# Proceedings of the Midwest Instruction and Computing Symposium 2025

# (MICS 2025)



# April 4 to April 5, 2025

# Augsburg University Minneapolis, Minnesota 55454

# Technical Session 1: 2:00 – 3:00 pm - Friday April 4, 2025

Cybers	ecurity: Hagfors 351	Session Chair: Mansi Bhavsar
2:00	Behavior as a Password: Machine	Joe Moriwa, Rushit Dave and Mansi
	Learning for Behavioral Biometric	Bhavsar
	Authentication	
2:30	Analyzing Ransomware Incidents in	Dominik Degele, Dawit Tesfaye and
	Healthcare: Patterns and Risk Assessment	Muhammad Abusaqer

<b>Reinforcement Learning : Hagfors 352</b>		Session Chair: Alex Ewart
2:00	Winning is the Friends We Made Along	Alex Ewart, Mikhail Filippov, Peter
	the Way: Cooperative Reinforcement	Kwaterski, and Sam Keyser
	Learning	
2:30	Adaptive vs Traditional Q-Learning	Saif Khan, Kaif Khan and Cayden White
	Methodology	

ML for	Business : Hagfors 367	Session Chair:Muhammad Abusaqer
2:00	Evaluating Quick-Commerce Platforms:	Toshif Khan and Muhammad Abusaqer
	A Sentiment and Topic Modeling	
	Analysis of User Reviews	
2:30	A Web Application for Walmart Store	Biyasu A and Mao Zheng
	Sales Forecasting	

ML usi	ng Computer Vision: Hagfors 373	Session Chair: Adrian Manchado
2:00	AI-Driven Soccer Analysis Using	Adrian Manchado, Tanner Cellio,
	Computer Vision	Jonathan Keane and Ian Wang
2:30	Machine Learning for Active Matter:	Yuva Gottimukala and Moumita
	Tracking and Analyzing Hexbug	Dasgupta
	Interactions	

Comput	er Science Education : Hagfors 301	Session Chair: Mathew Wright
2:00	A Reconstructive Multi-Objective	Eric Weinberg
	Hybrid Mixed Integer Genetic	
	Algorithm: Enhancing University Course	
	Timetabling with Adaptive Scheduling	
	Solutions	
2:30	Computational Inquiry in Undergraduate	Matthew Wright and Matthew Richey
	Math Courses	

Break: 3:00 – 3:30 pm - Snacks in Hagfors Center Atrium

# Technical Session 2: 3:30 – 4:30 pm - Friday April 4

ML and	Pattern Recognition : Hagfors 351	Session Chair: Mansi Bhavsar
3:30	Leveraging Machine Learning for	Redeate Kidanue, Rushit Dave and
	Enhanced Music Recognition:	Mansi Bhavsar
	Algorithms and Applications	
4:00	Exploring Development Patterns with	Kwadwo Owusu and Sayantica
	Age Prediction	Pattanayak

ML in	Medicine: Hagfors 352	Session Chair: Will Lassiter
3:30	Machine Learning-Assisted Vocal Cord	Will Sebelik-Lassiter, Quentin Robbins,
	Ultrasound Examination: Project VIPR	Evan Schubert, Excel Olatunji, Mustafa
		Barry, Muhammad Alliyu and Merry
		Sebelik
4:00	Beautiful Brain Blueprint: Visual	Grace Ivey, Joey Barganz, Ben Williams
	Stimulus Classification Through EEG	and Zach Peterson
	Signal Analysis	

AI and	Environmental Science: Hagfors 367	Session Chair: Sydney Balboni
3:30	Wildfire Spread Prediction Using Deep	Sydney Balboni, Adam Swedlund, Zoe
	Learning	Kirkman, Reagan Burkemper, Mason
		Beynon and Michael Wood
4:00	SwarmRAG: A Distributed AI	Ben Paulson, Sydney Balboni, John
	Framework for Environmental Modeling	Cisler, Brett Storoe and Thomas
		Benzshawel

AI in Ed	lucation: Hagfors 373	Session Chair: Samantha Foley
3:30	Chatbot for Academic Advising	Avari Caya and Samantha S. Foley
4:00	Student Performance & Behavior Dataset	Asmita Pun, Breanna Olson and
		Abusaqer Muhammad

CS Education: Hagfors 301		Session Chair: Sean Jones
3:30	More Than Technical: Embedding	Sean Jones
	Professional Competencies in a Software	
	Requirements Course	
4:00	Web scraping IMDB site with Python –	Mm Bari
	A ChatGPT proof assignment!	

# Robotics Contest and Pizza Party 4:45 – 6:15 pm in Hagfors Center Atrium

Sponsored by:



Programming Contest Announcements 6:30 – 7:00 pm in Hagfors 150

# Programming Contest 7:00 – 10:00 pm in Hagfors 150

# Technical Session 3: 9:00 – 10:00 am - Saturday April 5

Faculty	BoF: Hagfors 351	Session Chair: Samantha Foley
9:00	Faculty BoF: Exploring Faculty Hiring	Allison Sauppé and Samantha S. Foley
	Trends and Opportunities at Our	
	Institutions	

LLM Performance: Hagfors 352		Session Chair: Aiden Miller
9:00	Osire: Democratizing LLM Workflows	Adam Haile and Aiden Miller
	for High Performance Computing	
9:30	An Extra RMSNorm is All You Need for	Cody Steinmetz, Gavin Childress,
	Finetuning to 1.58 Bits	Jasdeep Singh, Keagan Weinstock,
		Aaron Herbst and Eli Vang

Green a	nd Ethical Computing : Hagfors 367	Session Chair: Wenkai Guan	
9:00	How GPUs Secretly Inflate AI's Carbon	Youzhi Li and Wenkai Guan	
	Footprint: GPU-aware Carbon Modeling		
	Exposes Hidden Carbon Emissions		
9:30	Reducing Bias in AI: Ethical and	Cheyenne Douglas	
	Technical Approaches		

Multi-A	gent Systems: Hagfors 373	Session Chair: Leigh Goetsch	
9:00	Applications of Reinforcement Learning	Andrew Needham, Evan Roegner,	
	in Competitive Robotics	Joseph Beattie, Andy Dao and Leigh	
		Goetsch	
9:30	Enhancing Collaboration in Multi-Agent	Hugo Garrido-Lestache Belinchon and	
	Environments with Attention-Based	Jeremy Kedziora	
	Actor-Critic Policies		

Applicat	tions of ML: Hagfors 151	Session Chair: Abhishek Ghimire		
9:00	Predicting YouTube Video View Counts:	Jalal Khalil and Abhishek Ghimire		
	Analyzing the Impact of Visual Features			
	and Metadata			
9:30	Mobility Patterns and Urban Travel	Jalal Khalil and Aseia Nabi		
	Behavior in the Twin Cities			

# Break: 10:00 – 10:30 Snacks in Hagfors Atrium

# Technical Session 4: 10:30 – 11:30 - Saturday April 5

CS Education: Hagfors 351		Session Chair: Mark Fienup	
10:30	Flipped Classroom Revisited	Mark Fienup	
11:00	Lessons Learned from the Development	Tomi Heimonen and Tim Krause	
	of a Multi-Disciplinary Pre-capstone		
Course in Computing			

ML in N	Iedical Imaging: Hagfors 352	Session Chair: Caleb Gray	
10:30	Transformers, Tumors, and Time:	Caleb Gray, Alexander Neher, Sonia	
	Improving Multi-Modal Image	Grade, Bart Gebka, John Bukowy and	
	Registration Using Cross-Modal	Peter LaViolette	
	Attention and Spatial Encoding		
11:00	Beyond Boundaries: Building a General	Alexander Neher, Alexander Drobek,	
	Latent Space for Medical Imaging with	Alhagie Boye, Sonia Grade, Autumn	
	Self-Supervised Learning	Mizer, Caleb Gray and Bart Gebka	

AI and Gaming: Hagfors 367		Session Chair: Zachary Warns	
10:30	AI Plays Ultimate TicTacToe	Zachary Warns	
11:00	Semi-supervised Weights for Action	Dylan Norquist	
	Generalization (SWAG)		

Ethics in AI: Hagfors 373		Session Chair: Benjamin Fine	
10:30	An Overview of Gender and Political	Jared B. Jones, Naeem Seliya, Emily M.	
	Biases in LLMs	Hastings and Benjamin T. Fine	
11:00	Generative AI: Emerging Policy,	Caroline Cronk, Cheryn Lindsay and	
	Innovation, and Debate	Isaac Wayne	

Cyberse	curity : Hagfors 151	Session Chair: T. Mohottalalage	
10:30	Emerging Paradigms in IoT Security: A	Samuel Arne, Thivanka	
	Comprehensive Survey of Malware, Threats,	Mohottalalage and Akalanka	
	Vulnerabilities, and Countermeasures	Mailewa	
11:00	Psyber-Security Revisited: A	Thivanka Mohottalalage, Anushka	
	Comprehensive Literature Review on	Hewarathne and Akalanka Mailewa	
	Integrating Psychological Principles into		
	Cybersecurity		

CS Education : Hagfors 301		Session Chair: Elena Machkasova	
10:30	A beginner-friendly environment for	Tristan Kalvoda, Elena Machkasova,	
	exploring error messages in the Clojure	Jaydon Stanislowski and John Walbran	
programming language.			

Keynote and Awards Ceremony 11:45 – 12:45 pm in Hagfors Center 150

Box Lunches and Departure 12:45 pm in Hagfors Center Atrium



# CAMPUS MAP

2211 Riverside Avenue, Minneapolis, MN 55454 In case of an emergency call the Department of Public Safety at 612-330-1717.



AM Anderson Music Hall AH Anderson Residence Hall CC Christensen Center Foss Lobeck Miles Center FC HC Hagfors Center IA Ice Arena KC Kennedy Center ш Lindell Library LH Luther Residence Hall MH Memorial Hall MR Mortensen Residence Hall 0M Old Main OG Oren Gateway Center SC Science Hall SM Si Melby Hall SH Sverdrup Hall UT Urness Tower

#### PARKING

- A. Faculty/Staff Permits
- B. Residential/Commuter Student
- C. Commuter Student
- D. Faculty/Staff Permits
- E. Short-Term Visitor (3 hour max)
- G. Faculty/Staff Permits
- H. Reserved \*
- J. Reserved \*\*
- K. Commuter Student
- L. General Permit Parking

#### OFF CAMPUS PARKING

- M. Faculty/Staff Permits
- 2001 Minnehaha Ave. O. Faculty/Staff Permits
- 2511 Franklin Ave. East

augsburg.edu/about/map

Updated December 12, 2017 8:39 AM

# Park in Lot D or L: (no permits required in those lots)

## **Important Contacts:**

## mics2025@micsymposium.org

Augsburg University Public Safety at 612-330-1717

**Nearby Restaurants:** 

Nabo - located in Oren Gateway Center (Friday until 6pm, closed Saturday) Einstein's - located in Christensen Center (Friday until 8pm, Saturday 7:30am to 3pm) Jimmy John's - located on Riverside just north of Hagfors (10:30am to 10pm) Oasis Mediterranean Grill - located on Riverside & 20th just north and west of Hagfors (11am-8pm)

# Leveraging Machine Learning for Enhanced Music Recognition: Algorithms and Applications

Redeate Kidanue, Rushit Dave, Mansi Bhavsar

Department of Computer Information Science

Mankato, MN, 56001

redeate.kidanue@mnsu.edu, rushit.dave@mnsu.edu, and mansi.bhavsar@mnsu.edu

#### Abstract

Music is a tool that has been integrated into society for thousands of years, it has influenced social aspects of life and has also aided in communication. Today we have various uses for music that go past our traditional entertainment uses and self-expression. Commonly, we see music being incorporated into visual art forms, and also being used in the medical field, specifically music therapy. With this broad spectrum of usage, it is important that music stays organized and easily accessible. This project intends to review literature looking at the methods, results, and limitations of music recognition using machine learning. Various techniques like K-Nearest Neighbor, Deep Neural Networks, and Convolutional Neural Networks have been identified in the use of organizing music and audio files by genre and utility. This literature review will also explore methods of data acquisition and music information retrieval. This paper focuses on presenting various learning models in the field of music recognition and illustrating their results.

**Keywords:** Machine Learning, Music Classification, K-Nearest Neighbor, Support Vector Machines, Neural Networks

#### 1. Introduction

Music has been an integral part of human culture for many years. It helps humans communicate, express themselves, and also connect with others. Music does not only include entertainment but also education, and therapy. With the rise of AI and machine learning entering our daily lives, we are seeing a change in how we consume and create music. Machine learning has allowed for automatic categorization of music, which shortens the process of sorting large datasets. Traditionally, music classification methods are time consuming, and often subjective. Machine learning can solve these problems with its ability to quickly analyze large datasets and detect

patterns, therefore automatically categorizing music. This technology is prevalent in the growing field of music streaming services where we see personalized playlists, and music recommendation systems. Today, we see a plethora of music streaming services where competition between them relies on how the well the service can cater to the individual users. By using a wide range of machine learning techniques, researchers are able to classify music based on genre, mood, and other attributes. This allows for enriched user experience. Music classification is an interdisciplinary field which uses knowledge from signal processing, machine learning, and music theory to create models capable of classifying audio files. Features like rhythm, melody, and timbre are extracted from audio signals and used to create a representation of each file or track. By analyzing these representations using machine learning models, patterns can be identified and used to classify tracks. Since music is an ever-growing field, the need for scalable and efficient algorithms is always sought out for. This paper aims to provide a review of existing machine learning techniques in the field of music genre classification. This paper focuses on methodologies, results, and limitations of each technique. This paper will also explore the most effective machine learning techniques for music genre classification, and how they compare in terms of accuracy and efficiency. It will also examine ways in which advancements in machine learning for music classification will benefit artists, industry stakeholders, and listeners, as well as how the integration of music theory and signal processing into machine learning pipelines can improve the interpretability and performance of classification models. By surveying the strengths and weaknesses of various machine learning techniques this paper intends to show the possibilities of machine learning being integrated into the field of music. This paper will also demonstrate how research in the field can be seen in practical application. Enhanced techniques in this field can lead to better discovery of new artists, leading to a more diverse, and flourishing music industry. These practical applications allow for benefits not only to music artists but business owners and industry workers as well.

## 2. Background

Past research on music recognition and music information retrieval has spanned many different fields. Within computer science, Optical Music Recognition (OMR) has strived to automate the reading of sheet music (Calvo-Zaragoza et al., 2021). Sheet music contains a written language that portrays the pitch, timbre, length, and other aspects of a given note. The goal of OMR is to create a program that is capable of reading sheet music as efficiently as the human brain can. OMR aids in the field of Music Information Retrieval by obtaining the details that are encoded in sheet music. With this information we are able to classify written sheet music without explicitly converting it to audio. High quality audio data is able to be classified using spectral analysis. Spectral analysis is a method that is used to study a signals structure or content. Octave-Based Spectral Contrast is a type of spectral analysis in which harmonic components from spectral peaks and valleys are considered independently. This allows the contrast distribution to be reflected clearly and permits for a precise output of the frequency range. Mel-Frequency Cepstral Coefficients (MFCC) are a commonly used method in speech recognition as well as MIR. MFCC has the advantage of being

able to display the frequency content of audio signals in a compact manner. (Lee et al., 2009). These spectral analysis methods are used to transform raw audio data into structured signals.



Figure 1: Music Emotion Recognition Process (Yang et al., 2018)

Today, we see deep learning methods being integrated within MIR. Deep learning is a subset of machine learning concerned with using neural networks with many layers to model complex relationships. Conventional machine learning techniques rely on manual feature extraction, however deep learning models are able to discern representations from raw data. The layers in the network capture abstract features from the input data. In music classification, the first layers typically focus on basic audio features such as frequency and amplitude. The deeper layers are able to identify features and patterns that can be mapped to specific genres (Bahuleyan, 2018). Figure 1 lays out the recognition process which includes extracting features from input audio. Features are then classified to an emotion space using various machine learning models.

With the modern advancements of machine learning being merged with MIR and music recognition we are able to expand real world applications. This includes the possibility of joining this technology with the home virtual assistants we see today. MIR already provides excellent performance with music recognition in virtual assistants. Users are able to search for and discover new songs. Virtual assistants are also able to accurately respond to prompts such as "What song is playing?" or "What artists create similar songs to this one?". Professionals in the music field can use this to aid in curating playlists, finding inspiration, and gathering an audience. MIR is also seen assisting various accessible solutions such as providing information to output tactical, or verbal feedback to visually impaired people or people who are hard of hearing. The voice activated aspects also supply access to people with mobility issues. Since MIR is an interdisciplinary field, we can also see its uses in healthcare. One of the most promising uses for MIR in healthcare is within the field of music therapy. Music recognition models can efficiently classify music based on genre and other factors, allowing for therapists and other professionals to promptly provide a treatment curated to a patient's mood or other needs. Music therapy has also seen promising results when used for patients with neurodegenerative diseases such as Alzheimer's and Parkinson's (Matziorinis & Koelsch, 2022).

#### 3. K-Nearest Neighbor

In recent studies, researchers used K-Nearest Neighbor (KNN) to predict the genre of songs. KNN is known for being easily adaptable to many different types of data analysis. The purpose of their work was to advance the accuracy of classification for different genres of music. For the experiment, the GTZAN dataset was used, the GTZAN dataset is a well-known resource in the field of music classification. This dataset contains a broad collection of thirty second sound files. Each track is a 16-bit mono audio file at a sampling rate of 22050 Hz in .au format (Asim Ahmed, 2017). The total dataset included for this experiment included 10,000 songs from ten different genres.

For feature extraction, this experiment used Mel Frequency Cepstral Coefficients (MFCC). MFCC is a commonly used method in this field. It captures timbral and tonal properties of audio signals from given sound files. MFCC tries to replicate the human auditory system by transforming the audio into a compact representation. This technique greatly improves the performance of machine learning models especially in the field of genre classification.

KNN was chosen for this experiment because of its simplicity and ease of use. KNN is an instance-based learning method which makes it fitting for tasks that involve classifying the differences between data points. It is also a non-linear method which allows effectiveness when discerning between direct and indirect patterns. Since this is a non-linear model, it can easily recognize and adapt to complex patterns. Additionally, one of the main advantages with KNN is that it improves in accuracy as the dataset size increases. However, this increased achievement in validity also comes with an increase in computational power. This can create limitations when building models using KNN. Since the KNN method does not have a training phase this leads to an increase in computational power by needing to store and process all data points during the classification and prediction phase. This creates drawbacks when using large datasets or multidimensional data.

For the current experiment, KNN was primarily used to measure the distance between two soundtracks. This allowed the researchers to measure the similarity and dissimilarity between the audio files. The analysis utilized Kullback-Leibler (KL) divergence to quantify the distances. As seen in Figure 2, KL is a metric that measures the difference between information represented in two different data distributions. This difference is measured by observing how one probability distribution diverges from the second reference distribution. Since this distance metric can be aligned precisely when looking at the data, this allows the KNN model to create a detailed classification by being able to account for subtle differences between audio tracks.

$$L(f,g) \equiv \int f(x) \log\{f(x)/g(x)\} dx$$

Figure 2: Kullback-Leibler loss function (Hall, 1987)

Advantages and disadvantages of different machine learning models for music generation have been a recent point of discussion (Liang, 2023). KNN was found to be ill-suited with music generation due to its poor performance with large multidimensional data. Since the KNN algorithm is instance based, meaning that there is no training phase, it is computationally expensive (Zhang, 2017). KNN does have high accuracy when compared to simpler methods such as Naive Bayes but takes a great deal of memory storage because of its need to store all of the data points in order to compare them at a later time (Bijalwan et al., 2014). The high computational expense almost solely comes from classification happening at the time that instances are seen (Guo, 2003).

The use of KNN in these studies showed its suitable mechanisms for classifying small amounts of data. When paired with MFCC, KNN can produce improved, more detailed outputs when classifying data. Much of the current research involving KNN includes constructing functions to measure distance (Zhang, 2021). However, the limitations with KNN can emphasize the importance of the correct uses of KNN in certain datasets and experiments.

## 4. Support Vector Machines

Support Vector Machines (SVMs) are used to find the boundary that can separate two classes of data with the least amount of loss. Figure 3 visualizes a simple SVM model. SVMs are unique because of how they transform input data into a higher dimensional space using the kernel trick (Asim Ahmed, 2017). The kernel trick is a mathematical approach which allows non-linear data to be classified linearly. This is a necessary transformation because non-linear data may not be separable. Data may not be linear which leads to an inability to separate data (Jakkula, 2006). When the kernel trick is applied, we are able to map input data into a higher dimensional space, making it possible for a hyperplane to linearly separate the classes. The goal for SVMs is to maximize the distance between the data points and the hyperplane, this ensures a distinct division between classes and prevents overfitting (Yue et al., 2003). In music genre classification, SVMs can be seen to be fairly effective because they are able to handle multi-dimensional data. In the case of music genre classification, when analyzing audio files, there are many features that must be accounted for. Features can include pitch, rhythm, and as mentioned before, timbre. These features are usually represented in multidimensional spaces like MFCC or spectrograms. MFCC can be a valuable tool for SVMs when they are both implemented, leading to capturing certain features of audio files and creating useful inputs. The kernel trick is a key aspect of SVMs classification abilities. The most common kernel types include linear, polynomial, sigmoid, quadratic, and radial basis function (RBF). RBF is most commonly chosen for music genre classification because of its technique of measuring the similarity between data points using Euclidean distance. This kernel algorithm calculates the dot product of the input vectors in the transformed feature space. This saves computational load by not needing to do an exclusive data transformation. The linear kernel is typically used for simpler data sets, while polynomial kernels are used for classifying more complex patterns.

One limitation when dealing with SVMs is the fact that it is primarily a binary classifier. It uses a "one-vs-rest" approach for classifying many classes. In the case of music genre classifications, multiple SVM classifiers must be trained for classification, specifically one for each genre being classified. The classifiers then output a confidence score where the highest score is the final prediction. The need to train multiple classifiers can make this method computationally demanding, especially when dealing with large datasets.

In the case of music recognition, SVMs work well for high dimensional data like what is seen in music recognition. Music features are typically seen in high dimensional data like spectrograms. SVMs are proficient in handling this kind of high dimensional data and also work well in smaller to medium sized datasets, making it well suited for niche music recognition projects. When looking at the GTZAN dataset which has a large amount of features but a moderate amount of data, this makes it well-suited for classification using SVMs. With the use of RBF kernels this typically makes SVM models outperform KNN models in accuracy. Xu et. al (2020) use SVMs to predict how participants will respond emotionally to music. They incorporate listener specific data such as physiological responses and also user preferences to predict emotional responses. This shows the flexibility of SVMs with handling multidimensional data.



Figure 3: Support Vector Machine (Dharmani, Ayush, Prateek, 2023)

SVMs require the use of the entire dataset during training and prediction phases. This too requires a high amount of memory and computational resources. This makes it difficult to work with SVM while looking at large datasets. Choosing the best fitting kernel is also very important and can affect the accuracy of the output. However, SVMs still remain one of the most common tools for music genre classification. When paired with other feature extraction methods they can produce results with greater accuracy.

## 5. Neural Networks

Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) are all common machine learning methods used for music genre classification. Each having distinct strengths and weaknesses allowing them to have unique impacts on different datasets. A simple neural network, displayed in Figure 4 contains an input layer, at least 1 hidden layer, and an output layer.



Figure 4: Simple Neural Network (Sun et al., 2018)

# 5.1 Deep Neural Networks

DNNs are composed of many layers of interconnected neurons. These layers are skilled at finding patterns in data which is important for music classification. When many layers of these interconnected neurons are stacked, DNNs can absorb different audio features from raw signals or processed inputs from spectrograms or MFCCs. For music genre classification, DNNs are used in high dimensional feature spaces. Features such as root mean square energy (RMSE) and zero-crossing rate (ZCR) can be given to a DNN to see different patterns that distinguish one genre from another. Studies have also shown that DNNs are skilled in processing a wide variety of acoustic features which allows them to use the information for different datasets (Bahuleyan, 2018). Since DNNs can automatically extract and learn feature representations from data makes it a popular tool in this field.

DNNs are also a flexible method of machine learning. Lee et al. (2021) used DNNs to predict moods in classical music. They analyzed tonal and harmonic attributes for the predictions. DNNs are useful in emotion recognition tasks as they are skilled in integrating audio features with physiological data such as brain activity and skin conductance (Xu et al., 2021).

However, DNNs face limitations such as relying on labeled datasets. Due to DNNs being a supervised learning model this can create drawbacks when classifying data. DNNs are also prone to overfitting which can lead to further steps needing to be taken to finalize data. Techniques such as batch normalization, dropout, and early stopping have been used to tackle the issue of overfitting. Advancements in computer hardware have also aided in the use of DNNs for further real-time music processing.

# 5.2 Convolutional Neural Networks

CNNs are best at analyzing data in a grid structure like images or spectrograms. This makes them effective in tasks that involve visual representations of audio. The main difference between CNNs and typical Neural Networks is that CNNs specialize in image processing (O'Shea, 2015). CNNs utilize convolutional layers to find patterns in data. CNNs often use Mel-spectrograms as input. These spectrograms give two-dimensional representations of audio signals, and this allows CNNs to learn about the time and frequency domain of the data. Since CNNs use localized feature extraction, this allows for it to learn genre specific patterns very well. CNNs are also used to analyze Mel-spectrograms. The CNNs showed promising results when classifying between genres like rock, jazz, and classical (Ashraf et al., 2023).



Figure 5: Simple CNN (O'Shea, 2015)

Transfer learning and fine-tuning practices can further improve the use of CNNs in genre recognition. When adjusting these pretrained models this allows researchers to modify the CNN to specific music datasets. Figure 5 demonstrates a CNN model where one can adjust the pooling layer of a fully connected layer. This can lessen the need for labeled data. CNNs are also useful for music transcription. They can extract pitch and note information from audio signals. CNNs do face challenges when seeking temporal dependencies which is important when working with music. Being unable to capture dependencies can make it difficult to understand the structure of the music, leading to difficulty classifying the genres. This is typically tackled by using a hybrid model of CNN and RNN together.

CNNs are also adept at identifying emotional content. CNNs analyze features like spectral contrast and chroma to classify music into mood related categories like happy, sad, or energetic (Lee et al, 2021). Despite being relatively complex, CNNs must have specific hyperparameters tuned like filter size, stride, and pooling in order to reach their desired performance.

#### 5.3 Recurrent Neural Networks

RNNs are ideal for looking at temporal dependencies in music due to the fact that they are designed to process sequential data. RNNs attempt to understand how musical features change over time. They can incorporate data from their memory to model the sequential nature of music such as rhythmic patterns, melodic progressions, and changes in dynamics. Researchers use RNNs to analyze audio features like MFCC and spectral contrast in order to find the similarity between tracks (West, Cox, Lamere 2006) Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) are variants of RNNs which help with the common drawbacks of RNNs like vanishing gradients.

When RNNs are trained on large data sets of music tracks, the RNN is able to generate new music that is based off of the given tracks. LSTMs specifically have been able to create many compositions in different genres. Hybrid models of CNNs and RNNs have been popular for music classification. CNNs are able to first extract spatial features from spectrograms, and the RNN is able to map the relationships within the features. This allows local and sequential patterns to be accounted for (Bahuleyan, 2018).

Figure 6 shows a proposed hybridization of CNN and various RNN models such as GRUs and Bi-LSTMs. This model takes given audio input and uses either MFCC or Melspectrogram for a feature extraction technique. Figure 7 shows that the different RNN models performed similarly when compared, with the GRU2 model performing highest for Melspectrogram feature extraction and the LSTM performing highest for MFCC.



Figure 6: Hybrid CNN and RNN architecture (Ashraf et al., 2023)



Figure 7: Results from proposed architecture (Ashraf et al., 2023)

## 6. Comparison

## 6.1 KNN

KNN is an incredibly useful algorithm for the classification of music genres. KNN measures the distance between data points to identify the closest neighbors, then assigning them to a class. Its simplicity allows it to fit well with smaller datasets. When KNN is paired with another tool for feature extraction such as MFCC, KNNs accuracy improves by being able to detect small differences between audio files. However, part of the KNN algorithm includes the need to store and process all of the data points. This means that it can be computationally inefficient especially when dealing with large datasets. KNN also does not perform well with high dimensional data, often needing aids to reduce dimensionality.

## 6.2 SVM

SVMs are skilled in classifying high dimensional data, this makes them well suited for this field. Using the kernel trick, SVMs are able to transform input data into higher dimensional spaces which allows for a linear separation. The RBF kernel is particularly effective for music classification, using Euclidean distance to measure the similarity between data points. SVMs require multiple classifiers to differentiate between genres, which can be inefficient. SVM still outperforms KNN in accuracy and is better qualified to handle spectrograms and MFCC.

## 6.3 DNN

DNNs are proficient at capturing patterns within data which allow for analysis or audio features like timbre, and pitch. DNNs are flexible allowing them to aid in genre classification or emotion recognition. A downside of DNNs comes with the fact that it heavily relies on labeled data, which creates a drawback when using it. However, many techniques have been used to aid

in these drawbacks such as dropout and transfer learning.

# 6.4 CNN

CNNs are able to extract spatial features through its convolutional layers, this allows them to capture time and frequency information from audio tracks. Transfer learning makes them increasingly useful by allowing pre-trained models to adapt to specific datasets. CNNs struggle with temporal dependencies, which is important in genre classification. Nevertheless, the integration of RNNs has been able to improve the performance of this classifier.

# 6.5 RNN

RNNs are designed to process sequential data which makes them ideal for modeling temporal dependencies. RNNs are able to analyze the change in musical features by adjusting memory mechanisms. LSTMs and GRUs are able to tackle common issues RNNs face like vanishing gradients. RNNs are also used for music generation, which can produce music in the style of the given training set.

Study	Objective	Methods	Strengths	Limitations	Results
Yang et al. (2007)	Recognizing emotion content of music signals by predicting arousal and valence values	Regression approach	Continuous view of music emotion eliminates ambiguity	Little background on continuous approach to MER	Best performance: 58.3% for arousal, 28.1% for valence
Anand et al. (2021)	Create a deep learning model to improve today's state of music recommendation	MFCC feature extraction; CNN and RNN	Stable RNN model	Fluctuating CNN model	System using both RNN and CNN had the best results
Shah et al. (2022)	Find which ML method performs best for music genre classification	CNN used for training; SVM for classification	Smaller architecture used to avoid overfitting.	Little preprocessing of the data was done	CNN achieved the highest accuracy with 74.1%

Yandre et al. (2017)	Develop a deep learning approach for singing skill evaluation	Unit concatenation, HMM-based statistical parametric synthesis, and DNN	Results are the best known results for dataset	Model performed significantly better with African music dataset compared to Western music	Combination of a CNN with an SVM trained with RLBP achieved 92% of recognition rate
Esposito et al. (2022)	Deep learning algorithms for recognizing COVID-19 related cough audio signatures	Log-mel spectrograms for preprocessing. Quantum circuits designed with PennyLane	Dramatically reduced training time and reduced memory requirements	Results were not validated with multiple techniques	Covid detection test accuracy over 60%
Pandey et al. (2022)	Identify types of music based on EEG signals	SVM, Random Forest, Naive Bayes,, LDA	Discovered that music features like timbre and beat are more generic and further studies	Brains are highly subjective and listen to songs differently	Random forest and LDA performed best
Modran et al. (2023)	Create a Machine Learning model that can predict whether a specific song has therapeutic effects on a specific person.	MFCC for feature extraction Basic NN structure	Achieved accuracy over 94%	Current solution must have users select their mood before prediction	Model correctly predicted in over 91% of cases

Ashraf et al. (2023)	Music classification using a model with hybridization of CNN and variants of RNN	CNN, RNN, LSTM, GRU	Hybridizatio n showed improved accuracy than typical CNN	Gradient during back- propagation does not allow for layers to be sufficiently updated	CNN and LSTM hybridization had an accuracy of 76.40%, CNN and Bi- GRU hybridization had and accuracy of 89.30%
Lee, Jeong, & Ko (2021)	Create a model for recognizing the detailed mood of classical music	MFCC feature extraction. Clustering algorithm for classification	Classical music tends to have inherent moods making it easy to classify	Model does not perform well with other instrumental music	Most mood classes have ROC AUC values between 0.7 and 0.8, making it a good classifier
Ni et al. (2012)	Create a system for harmonic analysis	New proposed harmonic analysis system uses chromograms, HMM topology, and parameter estimation	System still performs comparably to other methods in other genres	Model performs best on rock music due to mainly being trained on that genre	Proposed model achieved 75% Weighted Average Overlap Ratio

Table 1: Comparison of Machine Learning Experiments for Music Classification

The research summarized in Table 1 exhibits the broad field of MIR. The unique regression approach done by Yang et al. (2007) finds a potential solution to the ongoing problem of ambiguity and subjectiveness in music emotion recognition. Researchers developed a continuous approach for mapping emotions therefore revealing a personalized baseline for listeners. Continuous mapping for MER is an under-researched approach which limits further implementation. Esposito et al. (2022) took audio feature extraction to the medical field by creating models to diagnose COVID-19 through cough audio. By integrating quantum processing researchers were able to

achieve 60% accuracy with test data. Similarly, Modran et al. (2023) created a model for use in music therapy. Mood was correctly classified 91% of the time for predicting whether a song would have therapeutic effects on a given person. However, participants were required to report their mood, giving the model a starting off point before the prediction. These studies all face the issue of generalizability. Further work still needs to be done in order to be capable of categorizing different genres. Yandre et al. (2017) created a successful model for classifying African music genres, but the model performed poorly on Western music. Research in MER must be able to balance specificity of different genres while also taking subjectivity into account.

Despite the many promising strides made in this field there are also many limitations that have shown themselves within these studies. One of the significant limitations regards labeled datasets and the reliance of many supervised learning models on these datasets. DNNs, and SVMs heavily rely on training from labeled data. The task of labeling these large datasets is particularly time consuming. Commonly used datasets in this field like the GTZAN dataset can have biases and mistakes. We can see genre overlap and unequal distribution on genres. With these limitations the generalizability of algorithms can be compromised leading to inaccurate classification when used in real world applications. Additionally, many of these studies tend to focus on popular Western music genres such has rock, jazz, and pop music. This leads to a gap in research involving non-Western music styles. This bias leads to much of the world's music being undermined and therefore having a lower accuracy of classification when using existing algorithms.

Another common limitation comes from the computational expenses and scalability of certain algorithms. KNN and SVM struggle with handling high dimensional data and large amounts of data. KNN requires a significant amount of memory capacity due to how the technique performs. SVMs must train multiple classifiers which leads to an increase of computational demand. Neural network approaches show improved performance compared to KNN and SVMs, but they demand an extensive amount of training and intervention from researchers. These computationally demanding techniques require specialized hardware from the researchers. The bounds of current computers are constantly growing as well as the amount of music in the world. In order to quickly classify large amounts of audio files researchers must have access to hardware that can handle these computationally expensive techniques. Running these techniques can also raise sustainability concerns since these models use a great amount of energy which increases costs of running these experiments. This drawback forces researchers to focus on smaller datasets that are more sustainable which can lead to results that do not represent the full scope of music genres.

## 7. Conclusion and Future Work

The integration of different methods into ML models have proven to have promising results for MIR. As we can see with Esposito et al. (2022) the addition of quantum computing resulted in a great improvement in training time and memory requirements. With quantum computing's ability to hold large amounts of data, the possibility to train on larger datasets becomes more probable. Ashraf et al. (2023) developed a hybrid model of a CNN with RNN elements incorporated. This

study showed promising results with the hybrid model performing better than the CNN alone. With the increase in accuracy and performance when incorporating different methods we can start to consider what more we can start training models on. Researchers can start training on music with lyrics, allowing the models to predict emotion not only based on musical features but also singing features and lyrics.

Past and current research in MIR has typically used accuracy or F1 score to evaluate models performance. Future studies could consider implementing user preference and human judgement as evaluation metrics. This will introduce many new perspectives on the model's output, also inviting specific feedback. Genre specific metrics could also be implemented, creating enhanced metrics. For example, ambient music evaluation might not need to depend heavily on beats and rhythm compared to other genres such as classical or pop. Jazz music tends to rely on improvisation as well, this opens up the possibility of unexpected key changes, and changes how the song may be interpreted. Future studies could build models trained on specific genres, allowing for strengthened classification.

Many popular Western songs tend to follow similar patterns, and adhere to a similar set of rules. This creates a limitation in common datasets used due to the lack of variance. A potential next step in MIR could include the ability for the models to recognize errors in music. Errors such as notes being played in the wrong key, slight pitch errors, and differences in motifs can be detected by a professional, along with a potential solution being proposed. A model containing the ability to discern these errors and possibly present a solution as well could be useful in music education, creating a new resource for students and teachers.

Machine learning has greatly helped the field of music classification by creating new ways to organize and analyze audio data. Machine learning techniques like KNN, SVM, DNN, CNN, and RNN each have unique strengths and weaknesses that can aid in all aspects of music genre classification. CNNs most commonly displaying the highest accuracy in many experiments demonstrate higher level performance than other ML methods. Deep learning models are adept at handling high dimensional data. However, these methods each face their own limitations, most of them including computational demands, but also relying on labeled data. Future work in this field could include developing computer hardware capable of handling large amounts of data in an efficient manner. Diving deeper into unsupervised and semi-supervised learning techniques would also help foster great strides in the field. By facing these limitations and finding solutions, researchers can further discover new ways to incorporate machine learning in the field of music.

## References

- Ashraf, M., Abid, F., Din, I. U., Rasheed, J., Yesiltepe, M., Yeo, S. F., Ersoy, M. T. (2023). A Hybrid CNN and RNN Variant Model for Music Classification. *Applied Sciences*, 13(3), 1476. <u>https://doi.org/10.3390/app13031476</u>
- Asim, M., Ahmed, Z. (2017). Automatic Music Genres Classification using Machine Learning. International Journal of Advanced Computer Science and Applications, 8(8). https://doi.org/10.14569/ijacsa.2017.080844
- Anand, R., Sabeenian, R., Gurang, D., Kirthika, R., & Rubeena, S. (2021). AI based Music Recommendation system using Deep Learning Algorithms. *IOP Conference Series Earth and Environmental Science*, 785(1), 012013. <u>https://doi.org/10.1088/1755-1315/785/1/012013</u>
- Bahuleyan, H. (2018). Music Genre Classification using Machine Learning Techniques. *ArXiv:1804.01149 [Cs, Eess]*. <u>https://arxiv.org/abs/1804.01149</u>
- Benetos, E., Ragano, A., Sgroi, D., Tuckwell, A. (2022). Measuring national mood with music: using machine learning to construct a measure of national valence from audio data. *Behavior Research Methods*. <u>https://doi.org/10.3758/s13428-021-01747-7</u>
- Bijalwan, V., Kumar, V., Kumari, P., Pascual, J. (2014). KNN based machine learning approach for text and document mining. International Journal of Database Theory and Application, 7(1), 61-70.
- Bishop, Chris M. "Neural Networks and Their Applications." Review of Scientific Instruments, vol. 65, no. 6, June 1994, pp. 1803–1832, https://doi.org/10.1063/1.1144830.
- Choi, K., Fazekas, G., Cho, K., Sandler, M. (2017). A Tutorial on Deep Learning for Music Information Retrieval. *ArXiv.org*. <u>https://arxiv.org/abs/1709.04396</u>
- Dharmani, Ayush, and Prateek Guided by. "Classification of Songs Based on Genre Using ML,DL Algorithms." Juit.ac.in, 2023, <u>www.ir.juit.ac.in:8080/jspui/handle/123456789/9846</u>, <u>http://ir.juit.ac.in:8080/jspui/jspui/handle/123456789/9846</u>.
- Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. (2003). KNN model-based approach in classification. In On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings (pp. 986-996). Springer Berlin Heidelberg.
- Hall, P. (1987). On Kullback-Leibler loss and density estimation. The Annals of Statistics, 1491-1519.

- Jakkula, V. (2006). Tutorial on support vector machine (svm). School of EECS, Washington State University, 37(2.5), 3.
- Joseph, C., Lekamge, S. (2019, December 1). Machine Learning Approaches for Emotion Classification of Music A Systematic Literature Review. *IEEE Xplore*. <u>https://doi.org/10.1109/ICAC4908</u>
- Kurniawan, D. K., Alexander, G. R., Sidharta Sidharta. (2023). Deep Learning for Music: A Systematic Literature Review. 2022 International Conference on Information Management and Technology (ICIMTech), 373–378. https://doi.org/10.1109/icimtech59029.2023.10278072
- Lee, S., Jeong, H., Ko, H. (2021). Classical Music Specific Mood Automatic Recognition Model Proposal. *Electronics*, 10(20), 2489. <u>https://doi.org/10.3390/electronics10202489</u>
- Liang, J. (2023). Harmonizing minds and machines: survey on transformative power of machine learning in music. *Frontiers in Neurorobotics*, 17. <u>https://doi.org/10.3389/fnbot.2023.1267561</u>
- M. Esposito, G. Uehara and A. Spanias, "Quantum Machine Learning for Audio Classification with Applications to Healthcare," 2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)
- M. Shah, N. Pujara, K. Mangaroliya, L. Gohil, T. Vyas and S. Degadwala, "Music Genre Classification using Deep Learning," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 974-978, doi: 10.1109/ICCMC53470.2022.9753953
- Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., ... Hodjat, B. (2024). Evolving deep neural networks. In Artificial intelligence in the age of neural networks and brain computing (pp. 269-287). Academic Press.
- Modran, H. A., Chamunorwa, T., Ursuțiu, D., Samoilă, C., & Hedeșiu, H. (2023). Using Deep Learning to Recognize Therapeutic Effects of Music Based on Emotions. *Sensors*, 23(2), 986. <u>https://doi.org/10.3390/s23020986</u>
- Mulongo, B. (2020). Analyzing music genre classification using item response theory: A case study of the GTZAN data (Dissertation). Retrieved from <a href="https://urn.kb.se/resolve?urn=urn:nbn:se:kth:div">https://urn.kb.se/resolve?urn=urn:nbn:se:kth:div</a> 282828
- Ni, Y., McVicar, M., Santos-Rodriguez, R., De Bie, T. (2012, August). An End-toEnd Machine Learning System for Harmonic Analysis of Music. *Ieeexplore.ieee.org; IEEE*. <u>https://ieeexplore.ieee.org/abstract/document/6155600</u>
- O'Shea, K. (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

- P. Pandey, G. Sharma, K. P. Miyapuram, R. Subramanian and D. Lomas, "Music Identification Using Brain Responses to Initial Snippets," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*
- Ramírez, J., Flores, M. J. (2019). Machine learning for music genre: multifaceted review and experimentation with audioset. *Journal of Intelligent Information Systems*. <u>https://doi.org/10.1007/s10844-019-00582-9</u>
- S. Deebika, Indira, K. A., Jesline. (2019). A Machine Learning Based Music Player by Detecting Emotions. <u>https://doi.org/10.1109/iconstem.2019.8918890</u>
- Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., Mu<sup>"</sup>ller, K. R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE, 109(3), 247-278.
- Sha'Abani, M. N. A. H., Fuad, N., Jamal, N., Ismail, M. F. (2020). kNN and SVM classification for EEG: a review. In InECCE2019: Proceedings of the 5th International Conference on Electrical, Control Computer Engineering, Kuantan, Pahang, Malaysia, 29th July 2019 (pp. 555-565). Springer Singapore.
- Silla, C. N., Koerich, A. L., Kaestner, C. A. A. (2008). A Machine Learning Approach to Automatic Music Genre Classification. *Journal of the Brazilian Computer Society*, 14(3), 7– 18. <u>https://doi.org/10.1007/bf03192561</u>
- Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., Ashmore, R. (2018). Testing deep neural networks. arXiv preprint arXiv:1803.04792.
- Sze, V., Chen, Y. H., Yang, T. J., Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329.
- West, K., Cox, S., Lamere, P. (2006). Incorporating machine-learning into music similarity estimation. <u>https://doi.org/10.1145/1178723.1178737</u>
- Xu, L., Wen, X., Shi, J., Li, S., Xiao, Y., Wan, Q., Qian, X. (2020). Effects of individual factors on perceived emotion and felt emotion of music: Based on machine learning methods. *Psychology of Music*, 030573562092842. <u>https://doi.org/10.1177/0305735620928422</u>
- Y. -H. Yang, Y. -C. Lin, Y. -F. Su and H. H. Chen, "A Regression Approach to Music Emotion Recognition," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 448-457, Feb. 2008, doi: 10.1109/TASL.2007.911513
- Y. Ni, M. McVicar, R. Santos-Rodriguez and T. De Bie, "An End-to-End Machine Learning System for Harmonic Analysis of Music," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1771-1783, Aug. 2012, doi: <u>10.1109/TASL.2012.2188516</u>

- Yandre M.G. Costa, Luiz S. Oliveira, Carlos N. Silla, An evaluation of Convolutional Neural Networks for music classification using spectrograms, Applied Soft Computing, Volume 52, 2017, https://doi.org/10.1016/j.asoc.2016.12.024.
- Yue, S., Li, P., Hao, P. (2003). SVM classification: Its contents and challenges. Applied Mathematics-A Journal of Chinese Universities, 18, 332-342.
- Zhang, H., Berg, A. C., Maire, M., Malik, J. (2006, June). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 2126-2136). IEEE.
- Zhang, S. (2021). Challenges in KNN classification. IEEE Transactions on Knowledge and Data Engineering, 34(10), 4663-4675.
- Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbors. IEEE transactions on neural networks and learning systems, 29(5), 1774-1785.
- Zou, J., Han, Y., So, S. S. (2009). Overview of artificial neural networks. Artificial neural networks: methods and applications, 14-22.

# A Reconstructive Multi-Objective Hybrid Mixed Integer Genetic Algorithm

Enhancing University Course Timetabling with Adaptive Scheduling Solutions

Eric Weinberg, PhD

Department of Engineering, Math and Computer Science Viterbo University La Crosse, Wisconsin, United States of America

1

## 2. Abstract

This paper presents the Reconstructive Multi-Objective Hybrid Mixed Integer Genetic Algorithm, a novel approach to solving the University Course Timetabling Problem (UCTTP). This complex problem involves scheduling courses in a way that accommodates numerous and often conflicting constraints, such as room availability, faculty schedules, and student course requirements. Our approach integrates heuristic strategies with Mixed Integer Programming (MIP) and Genetic Algorithms (GAs) to create a hybrid methodology that balances computational efficiency with the quality of solutions. By reconstructing the problem around established academic timeslot patterns and employing a dual-phase optimization framework, our algorithm simplifies the complexity of UCTTP, enabling more focused and effective solutions. We demonstrate the efficacy of our approach through a series of case studies, comparing our algorithm's performance with traditional scheduling methods. Our findings suggest that the Reconstructive Multi-Objective Hybrid Mixed Integer Genetic Algorithm not only enhances the efficiency of course timetabling but also offers a flexible and adaptable framework suitable for addressing the dynamic scheduling needs of academic institutions.

## 3. Background

The exploration of the University Course Timetabling Problem (UCTTP) is driven by the real-world challenge of meeting the scheduling demands of academic institutions and the intellectual ambition to enhance computational approaches to these optimization problems. This issue comfortably spans both practical and theoretical realms, as it entails the intricate optimization task of organizing events within predetermined timeslots and spaces. Addressing UCTTP necessitates navigating a myriad of often conflicting constraints, including room availability, faculty schedules, student course requirements, and the specific resource needs of each class. This problem's elements include not only students, teachers, and courses but also the facilities and equipment that support both theoretical and practical learning environments. This complexity has only increased as schedulers and scholar's explore new requirements and optimization techniques to integrate into these models (Bashab et al. 2023; Chen et al. 2023; Chen et al. 2023)

In the realm of solving the University Course Timetabling Problem (UCTTP), a spectrum of solution methodologies has been developed, reflecting the complex nature of the problem. One prevalent approach is the combination of heuristic strategies with Mixed Integer Programming (MIP)Samarathunga and Wellahetti (2023) or Linear Programming techniques. This hybrid methodology leverages the heuristic component for generating feasible solutions or narrowing down the problem space, which is then precisely optimized using MIP.Rappos et al. (2022) Such a blend not only caters to the intricate constraints and combinatorial aspects of UCTTP but also strikes a balance between computational efficiency and the quality of solutions. Meanwhile, exact algorithms focus on uncovering the globally optimal solution by exhaustively searching through all possible configurations or employing advanced techniques to discard non-viable or suboptimal solutions. Though these algorithms assure optimal outcomes, their application is generally feasible only for smaller problem instances due to their intensive computational demands.

Hybrid approaches have gained prominence by integrating elements from different methodologies to capitalize on their distinct advantages. These methods might combine heuristic algorithms for generating an initial solution, subsequently refined through exact optimization techniques, or use metaheuristic algorithms for a broad exploration of the solution space, with sub-problems being solved optimally through MIP models. An illustrative instance of this approach involves resolving location constraints prior to the allocation of time slots. The primary objective behind these hybrid methods is to achieve a balance, harnessing the high-quality solutions provided by exact methods while also benefiting from the scalibility of heuristic or metaheuristic approaches. Blum et al. (2011)

In parallel, metaheuristic algorithms are particularly effective for large-scale problems. Techniques such as genetic algorithms, simulated annealing, and tabu search are notable for their strategic exploration of the solution space, aiming to identify globally optimal solutions by circumventing local optima. Although metaheuristics are versatile and potent in managing intricate problems, they necessitate careful tuning of parameters and cannot always guarantee the discovery of the optimal solution. (Abdipoor et al. 2023; Mikkelsen and Holm 2022; Pillay 2016; Sylejmani et al. 2023; Teoh et al. 2015)

#### 4. Introduction

Innovating off of the latest trends, we constructed a model that builds of the latest advancements but focuses on its ability to solve real-world higher scheduling problems in an efficient manner. Building upon the latest advancements in decomposing complex scheduling challenges, the Reconstructive Multi-Objective Hybrid Mixed Integer Genetic Algorithm Approach to the Course Timetabling Problem presents a strategy that emphasizes the reconstruction of the problem around predetermined academic timeslot patterns such as Monday-Wednesday-Friday (MWF) and Tuesday-Thursday (TuTh) which we describe as a reconstructive approach.<sup>1</sup> Rather than decomposing the problem based on the given constraints, this model recomposes the schedule in to less complex pieces based on known time-slot patterns. This methodological innovation simplifies the initial complexity of the University Course Timetabling Problem (UCTTP) by effectively segmenting the timetable into known patterns, thereby reducing the optimization space and enabling a more focused approach to solving the remaining aspects of the scheduling task. This initial step of reconstruction leverages the established scheduling frameworks within academic institutions, offering a strategic advantage by aligning the problem-solving process with the inherent rhythms of academic life. The next step is to take the remaining 1 credit classes and the 1 credit separated from 4 credit classes and prepare them for scheduling based on the remaining time slot and room availability. Such a reconstructive approach not only aligns with but also advances the trend towards breaking down complex problems into more manageable components, by instead integrating these components back into a cohesive whole that respects the structured realities of academic scheduling. (Abbaszadeh and Saeedvand 2014; Burke et al. 2009: Eledum 2017; Herres and Schmitz 2021; Holm et al. 0000 Accepted; Jawa Bendi et al. 2018; Samarathunga and Wellahetti 2023)

Following this reconstruction framework, the project employs a hybrid optimization model that integrates the strengths of Mixed Integer Programming (MIP) and Genetic Algorithms (GAs) beginning with an MIP model that loosens some constraints to accommodate a broader exploration of solutions. The initial MIP phase addresses the multi-objective facets of the timetabling problem, optimizing soft constraints to ensure a feasible and adaptable solution for the 3 credit patterns and the remaining 1 credit sections. Moreover, the 1 credit classes are separated from 4 credit classes so they can be optimized to adjacent to their 3 credit partners leaving the remaining 1 credit for later optimization. Subsequently, the GA phase refines this solution, where a custom mutation feature is crucial for preserving hard constraints, a common challenge in metaheuristic optimizations. However, metaheuristic approaches like GAs face their own set of challenges, including the risk of premature convergence to suboptimal solutions, the need for meticulous parameter tuning to balance exploration and exploitation, and ensuring

<sup>&</sup>lt;sup>1</sup>While are approach is most similar to deconstruction approaches, our focus on finding known patterns in the given data set rather that deconstructing the constraints into parts leads us to describe this as reconstructive.

diversity within the population to avoid stagnation. These challenges are addressed within our model by careful algorithm design and the integration of advanced genetic operations, ensuring the GA's effectiveness in navigating the complex solution space of UCTTP.

At the core of our approach is the use of PuLP, an efficient linear programming library, pivotal in formulating and resolving the complex facets of the scheduling optimization challenge. PuLP adeptly handles key variables—room availability, faculty schedules, and class credit requirements—allowing for a nuanced optimization of these interdependent elements within the class schedules.<sup>2</sup> Complementing our optimization framework is a user-centric web application developed with Flask, enhancing user engagement and simplifying the data management process. This interface empowers users to tailor the scheduling environment—blocking slots for specific classes, ensuring no collisions occur between selected classes, and anchoring classes to fixed timeslots. Additionally, it offers the capability to adjust model hyperparameters, tuning the balance and interaction between various constraints to refine the scheduling outcomes. This interactive platform not only enables the integration of user preferences and external data but also provides a platform for comparative performance analysis against established timetables. Such interactivity is crucial, bridging the gap between theoretical advancements in algorithmic strategies and their practical application, facilitating immediate adjustments and iterative enhancements based on real-world feedback and empirical insights.

Class Scheduler	OPTIMIZE	EJV_edit_ENG	R_MATH_	CSCI.csv	File siz 6.49 Kl	Av :e: Cl B	roiding Sele asses	ected	Keep 4 C Together	redit Classes	Blocked Value	Slot Pena	lty	Hold Penalty Slider Value
	Load	ed Data											^	
	Section	Title	Min Credit	Sec Cap	Room	Bldg	Week Days	CSM Start	CSM End	Faculty	Location	Hold	Blocker Slots	
	CSCI- 160- 001	Prinpl Computer Science	4	16	228	RCE	Tu Th	11:00AM	12:20PM	Weinberg, Eric G	RCE 228		S	

Fig. 1 The user interface of the class scheduling application. The interface includes sliders for adjusting scheduling parameters, a data section for loaded course information, and selection options for class constraints.

At the user interface's forefront, sliders provide a dynamic means for users to fine-tune various hyperparameters, directly shaping the behavior of the scheduling algorithm. These adjustments, including 'Avoiding Selected Classes,' 'Keeping 4 Credit Classes Together,' 'Blocked Slot Penalty Value,' and 'Hold Penalty Slider Value,' allow schedulers to modify the emphasis placed on different model constraints, tailoring the optimization process to align with specific scheduling priorities.

<sup>&</sup>lt;sup>2</sup>PuLP is configured to use its default LP solver, which is sufficient for the scale and complexity of problems addressed in this study. However, PuLP can also interface with more advanced solvers like CPLEX, GUROBI, and SCIP, which may offer enhanced performance for larger or more complex optimization problems.

Beneath these interactive sliders, the 'Loaded Data' section presents detailed course information, such as days, times, faculty assignments, and locations, enabling users to engage with and modify the schedule's foundational elements. This section not only facilitates the assignment of classes to preferred timeslots but also offers tools to set constraints, like holding a class at a designated time or blocking slots to prevent scheduling conflicts. Additionally, a 'Restrictions' feature allows for the exclusion of class overlaps, particularly useful in scenarios where students are likely to enroll in concurrent courses, thereby enhancing the schedule's practicality and user conformity.

The application plays a pivotal role in creating a dynamic feedback loop that significantly enhances the iterative refinement of the scheduling process. By providing a user-friendly interface, the application enables users to interact directly with the scheduling algorithm, offering them the ability to input data, adjust hyperparameters, and visualize the results in real time. This continuous interaction fosters a collaborative approach to optimization, where users' insights and preferences directly inform the algorithm's behavior. As users observe the outcomes of their adjustments and provide iterative feedback, the algorithm progressively finetunes its approach, adapting to the unique constraints and objectives of the specific scheduling scenario. This symbiotic relationship between the users and the algorithm not only accelerates the optimization process but also ensures that the resulting schedules are more closely aligned with the users' needs and expectations. Ultimately, this feedback loop empowers users to play an active role in shaping the scheduling solutions, driving a process of continuous improvement that leads to increasingly effective and tailored scheduling outcomes.

Our research is enriched by an empirical component, featuring a series of case studies based on actual schedules previously crafted by hand within various academic departments. We selected three such real-world schedules as benchmarks to evaluate the efficacy of our model. By integrating these historical, manually-created schedules into our system, we were able to conduct a thorough assessment of our model's capacity to optimize and enhance traditional scheduling practices.

These case studies were chosen to mirror the authentic conditions and challenges of academic scheduling, allowing us to test our model's performance in realistic settings. By fine-tuning the model's hyperparameters, we replicated diverse scheduling scenarios, establishing a comprehensive testbed for our optimization approach. This hands-on validation is critical, as it not only corroborates the model's theoretical soundness but also examines its operational utility in practical, real-world university environments.

The insights gleaned from analyzing these actual, hand-crafted schedules helped drive the the iterative refinement and advancement of our scheduling model.



Termination Condition Met

Select and Validate Best Results

**Return Top Timetables** 

**Fig. II:** Algorithm Flow Diagram. The process flowchart details the steps of the Genetic Algorithm from initialization through iterative optimization and final selection of the best timetabling solutions. No external coordination points are used in this diagram.

#### 5. Mathematical Formulation of the University Course Timetabling Model

While we optimize with a Genetic Algorithm as part of our solution, we do the bulk of the work in the Mixed Integer model. The mathematical model employs a composite objective function comprising several goals, each targeting specific aspects of the our version of the university course timetabling problem. It is these goals that can be manipulated by users to make the model dynamic. There is also a set of hard constraints that ensure the final schedule will meet the university requirements such as no instructor or room overlaps. This model will be equivalent for the classes being scheduled under the weekly pattern or the expanded one credit pattern where every section meeting is a possible time slot.

Variable	Description
$\overline{x_{ij}}$	Binary variable indicating if class i is scheduled in timeslot j.
	Set of all classes.
T	Set of all timeslots.
$Z_{ m Conflict}$	Sum of penalties for class conflicts, minimized in the objective
	function.
$Z_{\mathrm{Move}}$	Sum of penalties for moving classes from preferred timeslots.
$Z_{\rm Blocked}$	Sum of penalties for scheduling classes in blocked timeslots.
$Z_{ m Hold}$	Sum of penalties for not adhering to hold constraints for affiliated
	sections.
$s_i$	Size of class i, used in the room capacity constraint.
$f_i$	Binary parameter indicating if an instructor is involved in class i,
	used in faculty availability constraint.
$R_i$	Capacity of room j, used in the room capacity constraint.
$Conflict_{ij}$	Penalty for a class conflict if class i is scheduled in timeslot j.
$MovePenalty_{ij}$	Penalty for moving class i from its preferred timeslot j.
$BlockedPenalty_{ij}$	Penalty for scheduling class i in an undesired timeslot j.
$HoldPenalty_{ij}$	Penalty for not maintaining the hold requirement for class <b>i</b> in
- 5	timeslot j.

## 5.1 Table of Variables

#### 5.2 Composite Objective Function:

The overall aim is to minimize the sum of all objectives:

8

$$\label{eq:Minimize} \mbox{Minimize} \ \ Z = Z_{\rm Conflict} + Z_{\rm Move} + Z_{\rm Blocked} + Z_{\rm Hold}$$

Where each component is defined as:

• Class Conflict Objective  $(Z_{\text{Conflict}})$ : Minimizes overlapping class schedules for classes that should not be scheduled at the same time.

$$Z_{\text{Conflict}} = \sum_{i \in C} \sum_{j \in T} \text{Conflict}_{ij} \cdot x_{ij}$$

- Move Objective  $(Z_{\rm Move})$ : Minimizes deviation from a set times lot if a prefrence is declared.

$$Z_{\text{Move}} = \sum_{i \in C} \sum_{j \in T} \text{MovePenalty}_{ij} \cdot x_{ij}$$

• Blocked Slot Objective  $(Z_{\text{Blocked}})$ : Reduces usage of undesired timeslots if there a set of undesired timeslots is created.

$$Z_{\text{Blocked}} = \sum_{i \in C} \sum_{j \in T} \text{BlockedPenalty}_{ij} \cdot x_{ij}$$

• Hold Objective  $(Z_{\text{Hold}})$ : If two affilated sections (1 credit and 3 credit) are set to hold together this will optimize to keep them closse together in time

$$Z_{\text{Hold}} = \sum_{i \in C} \sum_{j \in T} \text{HoldPenalty}_{ij} \cdot x_{ij}$$

#### Constraints(Hard):

• Class Assignment Constraint: Each class is assigned to one timeslot.

$$\sum_{j \in T} x_{ij} = 1 \quad \forall i \in C$$

• Room Capacity Constraint: No room exceeds its capacity at any time.

$$\sum_{i \in C} x_{ij} \cdot s_i \leq \operatorname{RoomCapacity}_j \quad \forall j \in T$$
• Faculty Availability Constraint: No instructor is scheduled for overlapping classes.

$$\sum_{j \in T} x_{ij} \cdot f_i \leq 1 \quad \forall i \in C$$

C denotes the set of all classes, T the set of all timeslots,  $x_{ij}$  is a binary decision variable indicating if class i is scheduled in timeslot j,  $s_i$  represents the size of class i, and  $f_i$  is a binary parameter indicating the involvement of an instructor in class i.

#### 5.3 Reconstructive Scheduling Process with Weekly Patterns

The scheduling model adopts a reconstructive approach that accounts for the predefined weekly patterns for courses with different credit requirements.

- For 3-credit classes that follow a MWF or TuTh pattern, we define a set of time-slots,  $L_{3credits}$ , from which we start the scheduling process.
- After assigning 3-credit classes, we expand the scheduling options to include 1-credit classes, using a larger set of time-slots,  $L_{1credit}$ , which allows for greater flexibility. There is an metaparameter that will optionally try to keep the  $L_{1credit}$  and  $L_{3credits}$  close together.

The mathematical representation of the process is as follows:

• Define the initial set of time-slots for 3-credit classes

 $L_{3credits} = \{$ "MWF slots", "TuTh slots" $\}$ 

• Determine the number of time-slots for 3-credit classes:

 $|L_{3credits}| = \text{Count of "MWF" and "TuTh" slots}$ 

• Define the expanded set of time-slots after assigning 3-credit classes:

 $L_{1credit} =$  Full set of weekly time-slots  $-L_{3credits}$ 

The deconstructive scheduling is then carried out in the following manner:

• Start with the set of time-slots for the most complex scheduling requirements:

For each 
$$l_k \in L_{3credits}$$
:

Select appropriate time-slots and solve the scheduling model

• Update the set of available time-slots for 1-credit classes

 $L_{1credit} =$ Update with remaining time-slots

#### 5.3 Mathematical Dynamics of Custom Genetic Operations

Upon transitioning from the Mixed Integer Programming (MIP) model to the genetic algorithm (GA) component in our university course timetabling solution, we employ a universal evaluation function to measure and compare the effectiveness of potential schedules. This evaluation function is a critical tool that operates both at the conclusion of the MIP phase and continuously throughout the GA's optimization process. It serves as a standardized metric for assessing the quality of schedules based on a comprehensive set of criteria, including but not limited to, adherence to hard constraints, satisfaction of soft preferences, and overall schedule compactness and efficiency.

In the Genetic Algorithm (GA) phase of our course timetabling optimization, we leverage a suite of programming tools, libraries, and frameworks to implement and refine our solution. The core of our GA is built using Python, a versatile programming language known for its readability and broad support for scientific computing. We employ the DEAP (Distributed Evolutionary Algorithms in Python) library, a framework designed specifically for rapid prototyping and testing of evolutionary algorithms. DEAP provides the necessary infrastructure to define individuals, populations, and the evolutionary process, including selection, crossover, and mutation operations. For the evaluation function that plays a pivotal role in both the MIP and GA phases, we utilize NumPy for efficient numerical computations.

The universal evaluation function allows us to quantify the improvements each candidate schedule offers over others, enabling a systematic selection of the most successful solutions within the population.

- Function Definition: evaluateSchedule(individual)
- **Purpose**: To calculate a fitness score for a given schedule based on various constraints like class overlaps, instructor conflicts, room conflicts, and more.
- Penalties:
  - overlap\_penalty: Applied when two or more classes are scheduled in the same timeslot.
  - instructor\_conflict\_penalty: Applied when an instructor is assigned to multiple classes in the same timeslot.
  - room\_conflict\_penalty: Applied when two or more classes are scheduled in the same room at the same timeslot.
  - proximity\_penalty: Applied when related classes (e.g., 3-credit and associated 1-credit labs) are not scheduled close enough in time.
  - unwanted\_timeslot\_penalty: Applied when a class is scheduled in a timeslot that has been explicitly marked as unwanted.
  - pattern\_violation\_penalty: Applied when classes do not follow their expected scheduling patterns (e.g., MWF or TuTh).

- mutual\_exclusion\_penalty: Applied when classes that are meant to be mutually exclusive are scheduled at the same time.
- same\_day\_penalty: Applied when classes that should be spread out are scheduled on the same day.

#### • Process:

- The function iterates through each class in the schedule (individual), checking against all constraints and accumulating penalties.
- The penalties are aggregated to produce a total score for the schedule. Lower scores indicate better schedules (closer adherence to constraints).

#### • Return Value:

1. The function returns a dictionary with two keys: total\_score, representing the aggregated penalty score, and detailed\_scores, providing a breakdown of the penalties for each class.

 $\text{Total Score} = 300 \times O + 300 \times I + 300 \times R + 2 \times P + 3 \times U + 2 \times M + 300 \times V + 300 \times S$ 

- O: Number of overlap violations
- I: Number of instructor conflicts
- R: Number of room conflicts
- P: Number of proximity issues
- U: Number of unwanted times lot violations
- M: Number of mutual exclusion violations
- V: Number of pattern violations
- S: Number of same day violations
- $p_o$ : Weight of overlap penalty(300)
- $p_i$ : Weight of instructor conflict penalty(300)
- $p_r$ : Weight of room conflict penalty(300)
- $p_p$ : Weight of proximity penalty(2)
- $p_u$ : Weight of unwanted timeslot penalty(3)
- $p_m$ : Weight of mutual exclusion penalty(2)
- $p_v$ : Weight of pattern violation penalty(300)
- $p_s$ : Weight of same day penalty(300)

By applying this function, we can objectively identify and favor schedules that represent a significant optimization leap, thereby ensuring that each iteration of the GA is directed towards

progressively enhanced timetabling solutions. The hard constraints are given an exponentially higher value which has prevented significant breaking of any hard constraints. If breakage does occur, a given population can also be rejected.

The inclusion of this evaluation mechanism has proven to be highly beneficial, yielding approximately a 15% increase in optimization efficiency overall. This improvement is attributed to the function's ability to discern subtle distinctions in schedule quality, guiding the GA's evolutionary operators to focus on modifications that yield tangible benefits. As a result, the universal evaluation function not only facilitates a seamless transition between the deterministic optimization offered by the MIP model and the exploratory, adaptive nature of the GA but also significantly amplifies the effectiveness of the entire optimization process. Through this rigorous and quantifiable approach to evaluating and enhancing timetables, our solution achieves an elevated level of optimization, effectively addressing the complex and dynamic scheduling needs of academic institutions with unprecedented precision and success. Below is the results of a sample dataset. In it the MIP comes starts with a value of 75000 < n < 900000 represented as the top min value and becomes more efficient over generations eventually giving us enhanced schedule.

## **Mutation Operation**

The mutation operation in our genetic algorithm plays a pivotal role in enhancing the solution's adaptability and diversity. This operation introduces a controlled randomness that is essential for preventing the algorithm from converging prematurely on local optima, thus facilitating a more exhaustive exploration of the solution space. By allowing class sections to switch timeslots or patterns based on a predefined probability, the mutation function injects variability into the population, which is crucial for the algorithm's ability to discover and refine innovative scheduling solutions.

This variability is not arbitrary but is guided by the underlying structure of the university's scheduling constraints, ensuring that any mutations align with the logistical and educational parameters of the institution. For instance, when a class section transitions to a different pattern, this change is not just a random shift but a calculated move to explore new scheduling configurations that could offer improvements in terms of conflict reduction, resource optimization, or overall timetable quality.

Moreover, the mutation operation's sensitivity to the credit hours (minCredit) of classes ensures that the algorithm's adjustments are contextually relevant, acknowledging that different classes—based on their credit value—have varying levels of scheduling flexibility and impact. This nuanced approach allows the algorithm to effectively balance the need for innovation with the preservation of schedule feasibility and integrity, thus contributing significantly to the genetic algorithm's capability to evolve and adapt robust solutions in the dynamic landscape of university course timetabling.

The mutation function probabilistically modifies the scheduling assignments within an individual's timetable to foster exploration of the solution space. For a given class section characterized by its credit hours (minCredit), the mutation operation assesses whether to adjust its timeslot within the current pattern (e.g., MWF or TuTh) or to transition it to an alternative pattern. This decision process is mathematically governed by a mutation probability (mutpb), dictating the frequency and extent of these alterations. Formally, the operation is represented as:

- For sections with minCredit 3, either:
  - Retain the current pattern, selecting a new timeslot from the same set of days, or
  - Switch to a different pattern, ensuring compatibility with the section's requirements.

This mutation logic enhances the genetic diversity within the population, promoting the discovery of potentially more effective scheduling arrangements.

## **Crossover Operation**

The crossover function's role extends beyond the mere combination of parent timetables—it introduces a strategic blend of genetic diversity, crucial for the algorithm's exploration and exploitation balance. By categorizing class sections into distinct scheduling patterns, the function maintains the integrity of the course's temporal structure, a key aspect given the unique constraints of university timetabling. The selective exchange of subsets based on scheduling patterns ensures that the resulting offspring inherits a feasible and coherent timetable structure, aligning with established academic norms and requirements. This methodical exchange process is pivotal, as it underpins the genetic algorithm's ability to generate new and potentially optimal timetabling solutions, while respecting the hard constraints and optimization goals set forth by the MIP phase. This intricate interplay between genetic diversity and operational feasibility enhances the algorithm's capability to navigate the complex solution landscape, offering innovative scheduling arrangements that remain grounded in the practicalities of academic scheduling. Through this dynamic process, the crossover function significantly contributes to the genetic algorithm's robustness, fostering an environment where novel solutions can emerge, adapt, and thrive within the defined operational parameters of university course scheduling.

The crossover function amalgamates the genetic information of two parent timetables to produce offspring with inherited characteristics. It categorizes class sections by their scheduling patterns and executes pattern-specific exchanges at designated crossover points. Mathematically, this process involves:

- Partitioning each parent's schedule into subsets based on the scheduling pattern,
- Exchanging these subsets between parents at selected indices to create new offspring schedules.

Ensuring the offspring's feasibility with respect to the university's scheduling constraints is paramount, with validity checks confirming adherence to the foundational rules established by the MIP model.

#### 5.4 Integration with the MIP Framework

The GA's mutation and crossover functions mathematically enrich the scheduling optimization process by iteratively refining the solutions initially formulated by the MIP model. While the MIP model constructs a viable and optimized foundation for the timetable, considering both hard and soft constraints, the GA's iterative enhancements aim to uncover and exploit new scheduling configurations, offering a pathway to potentially superior solutions.

This hybrid optimization strategy leverages the deterministic capabilities of MIP and the adaptive exploratory provess of GAs, presenting a cThe Devastation of Iraqomprehensive solution to the complex challenge of university course timetabling. This dual-phase approach ensures a robust and dynamic optimization framework capable of accommodating the diverse and evolving scheduling needs of academic institutions.

## 6. Efficacy of the Reconstructive Multi-Objective Hybrid Approach

Our study's results demonstrate the successful application and effectiveness of the Reconstructive Multi-Objective Hybrid Mixed Integer Genetic Algorithm Approach in addressing the University Course Timetabling Problem (UCTTP). This approach significantly simplified the initial complexity of UCTTP, enabling the efficient organization of courses into predetermined academic timeslot patterns such as Monday-Wednesday-Friday (MWF) and Tuesday-Thursday (TuTh). By leveraging these established scheduling frameworks, we observed a marked reduction in the optimization space, facilitating a more focused and effective solution strategy.

The integration of Mixed Integer Programming (MIP) with Genetic Algorithms (GAs) in a hybrid model proved particularly advantageous. The MIP phase effectively tackled the multi-objective aspects of the timetabling problem, focusing on accommodating soft constraints to ensure a feasible and adaptable solution.<sup>3</sup> This foundation allowed the subsequent GA phase to refine the timetable within the constraints established by the MIP model, introducing adaptability and evolution to the optimization process. This innovative hybrid approach not only optimized the timetable with precision but also demonstrated adaptability to dynamic scheduling needs, setting a new benchmark for academic timetabling solutions.

<sup>&</sup>lt;sup>3</sup>Initiating our Genetic Algorithm with a feasible solution confers multiple technical advantages. Primarily, it constrains the initial search space to the realm of possible solutions, streamlining the algorithm's convergence towards optimal schedules by avoiding wasteful exploration of infeasible regions. This strategy accelerates the convergence speed because the algorithm begins the optimization process closer to potential optimal solutions. Moreover, it ensures that each generation maintains feasibility, focusing computational efforts on enhancing solution quality rather than rectifying infeasibility. Such an approach is crucial for respecting the hard constraints inherent to the University Course Timetabling Problem, such as room capacity and instructor availability, guaranteeing that all solutions adhere to these non-negotiable requirements. Furthermore, embedding heuristics within the GA's operators enhances the effectiveness of the algorithm when applied to feasible solutions, steering the exploration towards areas of the solution space that are likely to yield higher fitness values. Lastly, this method promotes genetic diversity within the bounds of constraint satisfaction, encouraging the algorithm to uncover innovative scheduling configurations that balance the various objectives of academic timetabling.

#### 6.1 Comparison with Non-Reconstructive Models

In contrast, the non-reconstructive version of our model, which attempted to apply a traditional Algorithm without the initial restructuring provided before the MIP phase, encountered significant challenges. Specifically, it ran into maximum recursion errors and failed to find solutions during optimization attempts, indicating a struggle to efficiently navigate the complex solution space of our UCTTP. This limitation underscores the intricate nature of the timetabling problem and highlights the necessity of an advanced solver to manage such complexity effectively. Moreover, when the GA was attempted on the non-reconstructed problem, it struggled to find and optimal solution and the timeclock reached over 7 minutes for our sample datasets before any convergence.

Although it is conceivable that a more advanced solver could mitigate these issues, this scenario underscores the practical and theoretical advantages of our reconstructive hybrid approach. By recomposing the problem and leveraging the strengths of both MIP and GAs, our method not only avoids the pitfalls encountered by non-reconstructive models but also demonstrates superior efficiency and adaptability in generating optimized academic schedules.

## 7. Case Study

To test our scheduling algorithm, we tested against three variants of previously optimized schedules<sup>4</sup>. For this highlighted instance, we intentionally introduced conflicts into our sample schedule at various points—specifically 9 instances—to verify the algorithm's effectiveness in generating valid schedules. We proceeded to block five time slots for faculty and create non-intersection, to observe the algorithm's adaptability. Additionally, we designated 5 classes to be non-intersecting, with all tests yielding successful outcomes.

## 7.1 Detailed Case Study Results Analysis

## Solver Outputs Overview

The solver's efficiency and the quality of the solutions it provides are critical in evaluating the effectiveness of the scheduling algorithm. Below, we break down the outputs from two different model runs to highlight these aspects.

## 7.2 3 Credit Patterned Time Slot Output Summary

<sup>&</sup>lt;sup>4</sup>For brevity, we are only highlight one of the three case studies. The second case study used was a different year of two other departments that included Chemistry and Biology departments. In these cases we follow the same protocol as the highlighted case study and had similar optimal results.

Metric	Value
Objective Value	0.000
CPU Time (seconds)	0.02
Wallclock Time (seconds)	0.02
Nodes Enumerated	0
Total Iterations	0

• **Interpretation**: The optimal objective value of 0 indicates an ideal solution where all constraints are met without any penalties. The swift resolution time demonstrates the solver's high efficiency.

## 7.3 1 Credit MIP Output Summary

Metric	Value
Objective Value	14.000
CPU Time (seconds)	0.01
Wallclock Time (seconds)	0.01
Nodes Enumerated	0
Total Iterations	0

• Interpretation: An objective value greater than suggests minor conflicts or constraints that could not be perfectly resolved, indicative of the model's complexity.

## 7.4 Genetic Algorithm Performance

The genetic algorithm's ability to improve solutions over generations is crucial. The graph below summarizes its progression over 30 generations.

#### 7.5 Summary/Results

The detailed breakdown and analysis of solver outputs and the genetic algorithm's performance provide a comprehensive view of the scheduling optimization process. The rapid solver convergence and the genetic algorithm's continuous improvement over generations underscore the potential of this hybrid approach in addressing complex scheduling challenges.

Following these validations, we embarked on a comparison between schedules crafted manually and those generated by our algorithm. Like is common, we alter previous years schedules



Fig. 2 Genetic Algorithm Performance Over Generations. This graph tracks the performance of the algorithm across generations, highlighting the average, minimum, and maximum scores at each step. Key points are marked to indicate significant changes in the score trend.

with minor modifications to fit the current needs.(Babaei et al. 2015; Chen et al. 2023; Samarathunga and Wellahetti 2023) This comparison spotlighted the maintenance of nonoverlapping courses as compared like the human version, showcasing the system's ability to create a coherent and conflict-free timetables rapidly. Such precision is vital to prevent issues like double-booking, a key concern in effective timetable construction.

The algorithm's strategic allocation of rooms—utilizing "RCE 228" for different classes at varied times—further illustrates its capacity to optimize space while meeting logistical requirements. Additionally, its adept handling of one-credit or lab classes, particularly the alignment of "ENGR-150-001\_one\_credit" with its corresponding lecture, showcases an advanced level of scheduling insight. This nuanced approach, which aligns related course components to facilitate student participation, may surpass the manual scheduling process that could overlook such detailed interconnections.

In the broader context of our case study, the additional schedules analyzed also demonstrated that the algorithm consistently meets or exceeds the standards set by traditional manual scheduling. By delivering optimized schedules that account for the complexities of academic timetabling, the algorithm not only achieves a high level of scheduling efficiency but also supports academic administrators by providing a reliable tool that enhances operational efficiency and allows them to focus on other critical aspects of educational management. This alignment with the goal of reducing the administrative burden on schedulers underscores the practical value and transformative potential of the algorithm in the realm of academic scheduling.

The algorithm proves its robustness in structuring schedules that are not only well-organized but also responsive to multiple scheduling dimensions, aligning with the core objectives of academic timetabling. While defining "optimal" scheduling is subjective and dependent on institutional priorities, the improvements in the algorithm-generated schedule point towards significant strides in achieving a more optimal scheduling process. Continuous evaluation against real-world feedback and performance metrics remains vital to ensuring the algorithm meets the comprehensive needs of the academic community. In our case study, the optimization model showcased its real-world efficacy on a dataset of 3 groups of around 40 classes, demonstrating its ability to handle complex scheduling requirements efficiently. (Mokhtari et al. 2021)

This integration of the GA, significantly enhancing the optimization score from roughly 90,000 to below 65,000 within about 5 seconds, illustrates a substantial improvement in the model's ability to refine and optimize complex scheduling scenarios. The GA's collaboration with MIP in this context not only demonstrates a marked improvement in managing university course timetabling complexities but also signals a promising direction for future enhancements, showcasing the potential of this hybrid approach to deliver efficient and effective academic scheduling solutions. Below is the alternative to the human solution without any parameter or hyper parameter tuning.

## 7.6 Results Set 1

Section Name	Instructor	Timeslot
CSCI-160-001	Instructor A	M W F - 10:10AM
ENGR-150-001	Instructor B	M W F - 10:10AM
ENGR-250-001	Instructor C	M W F - 3:35PM
ENGR-320-001	Instructor B	M W F - 11:15AM
ENGR-365-001	Instructor D	M W F - 8:00AM
ENGR-365-001	Instructor D	Tu Th - 8:00AM
ENGR-370-001	Instructor C	M W F - 10:10AM
MATH-221-001	Instructor D	M W F - 10:10AM
MATH-230-001	Instructor D	M W F - 3:35PM
MATH-365-001	Instructor D	M W F - 12:20PM
PHYS-102-001	Instructor E	M W F - 10:10AM
CSCI-200-001	Instructor A	M W F - 8:00AM
CSCI-210-001	Instructor A	M W F - 8:00AM
CSCI-210-001	Instructor A	Tu Th - 8:00AM
CSCI-499-001	Instructor A	M W F - 3:30PM
CSCI-499-001	Instructor A	Tu Th - 3:30PM
ENGR-489-001	nan	M W F - 10:10AM
ENGR-499-001	Instructor C	M W F - 1:25PM
MATH-091-001	Instructor F	M W F - 9:30AM
MATH-091-001	Instructor F	Tu Th - 9:30AM
MATH-112-001	Instructor G	M W F - 10:10AM
MATH-113-001	Instructor G	M W F - 2:00PM
MATH-113-001	Instructor G	Tu Th - 2:00PM
MATH-130-001	TBD	M W F - 10:10AM
MATH-130-002	TBD	M W F - 8:00AM
MATH-130-002	TBD	Tu Th - 8:00AM
MATH-155-001	Instructor G	M W F - 3:30PM
MATH-155-001	Instructor G	Tu Th - 3:30PM
MATH-255-001	Instructor H	M W F - 8:00AM
MATH-270-001	Instructor G	M W F - 11:15AM
MATH-344-001	Instructor G	M W F - 11:00AM
MATH-344-001	Instructor G	Tu Th - 11:00AM
MATH-499-001	Instructor D	M W F - 2:30PM
PHYS-251-001	Instructor I	M W F - 11:15AM
PHYS-261-001	Instructor J	M W F - 12:20PM
$CSCI-160-001\_one\_credit$	Instructor A	W - 9:05AM
$ENGR-150-001\_one\_credit$	Instructor B	M - 9:05AM
$ENGR-250-001\_one\_credit$	Instructor C	F - 9:05AM
$ENGR-320-001\_one\_credit$	Instructor B	Th - 12:30PM

Section Name	Instructor	Timeslot
ENGR-365-001_one_credit	Instructor D	M - 9:05AM
ENGR-370-001_one_credit	Instructor C	Tu - 12:30PM
MATH-221-001_one_credit	Instructor D	Th - 12:30PM
MATH-230-001_one_credit	Instructor D	Tu - 12:30PM
MATH-365-001_one_credit	Instructor D	W - 9:05AM
PHYS-102-001_one_credit	Instructor E	F - 9:05AM
MATH-081-001	Instructor F	F - 9:05AM
MATH-083-001	Instructor F	W - 9:05AM

## 8. Conclusion

The integration of a Genetic Algorithm (GA) into our university course timetabling solution represents a strategic enhancement to the scheduling optimization process, building upon the foundational work laid out by the latest advancement in the field. The MIP model, with its focus on multi-objective optimization, primarily addresses soft constraints to establish a simple yet feasible timetable. While other models have been tested on more extensive datasets with more complex constraints, ours demonstrates that this recomposition model effectively optimizes while maintaining adaptability and could lead to further optimizations through this or similar reconstructive approaches. Our MIP approach ensured that the foundational timetable not only meets basic scheduling requirements but also provides a degree of flexibility in accommodating preferences and minimizing conflicts, thus emphasizing simplicity and feasibility in the initial scheduling solution.

Following this, the GA's mutation and crossover functions come into play, iteratively refining this initial solution while still accounting for the recomposed nature of the problem. By leveraging new scheduling configurations and exploring the solution space beyond the confines of the MIP model, the GA seeks to enhance the timetable's overall quality, potentially uncovering superior solutions that were not initially apparent. This hybrid optimization framework marries the deterministic precision of MIP with the adaptive, exploratory nature of GAs, creating a comprehensive and dynamic approach to tackling the complex challenge of university course timetabling.

Such a dual-phase strategy ensures a robust optimization framework capable of evolving alongside the diverse and changing needs of academic institutions. It offers a structured yet flexible approach to timetabling that adeptly navigates the myriad constraints and objectives inherent in academic scheduling. It also models constraints alongside the distributed process of scheduling within higher education. Through balancing various objectives, the model crafts schedules that not only adhere to room capacity and faculty availability but also respect the myriad preferences and requirements of the university community. This reconstructive optimization process adeptly manages scheduling across different class complexities, resulting in optimized timetables that facilitate an enhanced educational experience for both faculty and students, and underscores the model's capacity to deliver practical, optimized scheduling solutions in a complex academic landscape.

Abbaszadeh, M., Saeedvand, S.: A Fast Genetic Algorithm for Solving University Scheduling Problem. IAES International Journal of Artificial Intelligence (IJ-AI). 3, 7–15 (2014). https: //doi.org/10.11591/ijai.v3.i1.pp7-15

Abdipoor, S., Yaakob, R., Goh, S.L., Abdullah, S.: Meta-heuristic approaches for the University Course Timetabling Problem. Intelligent Systems with Applications. 19, 200253 (2023). https://doi.org/10.1016/j.iswa.2023.200253

Babaei, H., Karimpour, J., Hadidi, A.: A survey of approaches for university course timetabling problem. Applications of Computational Intelligence and Fuzzy Logic to Manufacturing and Service Systems. 86, 43–59 (2015). https://doi.org/10.1016/j.cie.2014.11.010

Bashab, A., Ibrahim, A.O., Hashem, I.A.T., Aggarwal, K., Mukhlif, F., Ghaleb, F.A., Abdelmaboud, A.: Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues. Computers, Materials & Continua. 74, (2023). https://doi.org/10.32604/cmc.2023.034051

Blum, C., Puchinger, J., Raidl, G.R., Roli, A.: Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing. 11, 4135–4151 (2011). https://doi.org/https://doi.org/10.1016/j.asoc.2011.02.032

Burke, E., Marecek, J., Parkes, A., Rudová, H.: Decomposition, reformulation, and diving in university course timetabling. Computers & Operations Research. 37, 582–597 (2009). https://doi.org/10.1016/j.cor.2009.02.023

Chen, M.C., Goh, S.L., Sabar, N.R., Kendall, G., others: A survey of university course timetabling problem: Perspectives, trends and opportunities. IEEE Access. 9, 106515–106529 (2021)

Chen, M., Werner, F., Shokouhifar, M.: Mathematical Modeling and Exact Optimizing of University Course Scheduling Considering Preferences of Professors. Axioms. 12, 498 (2023). https://doi.org/10.3390/axioms12050498

Daskalaki, S., Birbas, T., Housos, E.: An integer programming formulation for a case study in university timetabling. European Journal of Operational Research. (2004). https://doi.org/10.1016/s0377-2217(03)00103-6

Eledum, H.: The University Timetabling Problem: Modeling and Solution Using Binary Integer Programming with Penalty Functions. International Journal of Applied Mathematics and Statistics. 56, 164–178 (2017)

Herres, B., Schmitz, H.: Decomposition of university course timetabling. Annals of Operations Research. 302, 405–423 (2021). https://doi.org/10.1007/s10479-019-03382-0

Holm, D.S., Mikkelsen, R.Ø., Sørensen, M., Stidsen, T.R.: A MIP based approach for International Timetabling Competition 2019. 0000 Accepted

Jawa Bendi, K., Sunarni, T., Alfian, A.: Using Graph Coloring for University Timetable Problem. International Journal of Science and Research (IJSR). 7, 1692–1697 (2018)

Mikkelsen, R.Ø., Holm, D.S.: A parallelized matheuristic for the International Timetabling Competition 2019. Journal of Scheduling. 25, 429–452 (2022). https://doi.org/10.1007/s10951-022-00728-8

Mokhtari, M., Vaziri Sarashk, M., Asadpour, M., Saeidi, N., Boyer, O.: Developing a Model

for the University Course Timetabling Problem: A Case Study. Complexity. 2021, 1–12 (2021). https://doi.org/10.1155/2021/9940866

Pillay, N.: A review of hyper-heuristics for educational timetabling. Annals of Operations Research. 239, 3–38 (2016). https://doi.org/10.1007/s10479-014-1688-1

Rappos, E., Thiémard, E., Robert, S., Hêche, J.-F.: A mixed-integer programming approach for solving university course timetabling problems. Journal of Scheduling. 25, 391–404 (2022). https://doi.org/10.1007/s10951-021-00715-5

Samarathunga, S., Wellahetti, L.: A Binary Linear Programming Model for a Case Study in University Timetabling Problem. (2023)

Sylejmani, K., Gashi, E., Ymeri, A.: Simulated annealing with penalization for university course timetabling. Journal of Scheduling. 26, 497–517 (2023). https://doi.org/10.1007/s10951-022-00747-5

Teoh, C.K., Wibowo, A., Ngadiman, M.S.: Review of state of the art for metaheuristic techniques in Academic Scheduling Problems. Artificial Intelligence Review. 44, 1–21 (2015). https://doi.org/10.1007/s10462-013-9399-6

## BEHAVIOR AS A PASSWORD: MACHINE LEARNING FOR BEHAVIORAL BIOMETRIC AUTHENTICATION

Joe Moriwa, Rushit Dave\* and Mansi Bhavsar Computer Information Science Department Minnesota State University, Mankato. 229 Wissink Hall, Mankato, MN 56001. joe.moriwa@mnsu.edu rushit.dave@mnsu.edu

mansi.bhavsar@mnsu.edu

#### Abstract

In today's fast-moving tech landscape advancements are happening at a fast pace. These advancements come with new opportunities to make our lives easier. However, alongside the rise of these new technologies also comes the introduction of new threats. So, it is important that security trends evolve along with the fast-paced development of technology. Most security today is based on traditional physiological biometrics like fingerprint and iris scans. While this is effective to an extent, it is not adequate as there are ways this type of authentication can be breached. This research study explores behavioral biometric authentication using machine learning. It aims to answer the question: How can behavioral biometrics combined with machine learning techniques provide a strong defense to counter modern cyber threats? This project will present an analysis of various behavioral traits, such as keystroke dynamics, mouse movements and gait analysis using machine learning techniques. These techniques include feature extraction, model training, and validation. Through the application of these techniques, this project provides an analysis of these behavioral characteristics, demonstrating their potential in developing more robust and adaptive security frameworks that offer greater protection against emerging threats.

**Keywords:** Machine Learning, Behavioral Biometrics, Continuous Authentication, Mouse Dynamics, Gait analysis, Touch Dynamics.

## 1. Introduction

The development of technology coinciding with improved ambient devices such as sensors and smart phones has led to a rapid development of smart environments [1]. Security systems should in turn evolve with this growth as old methods that relied on knowledge of secrets such as password and pin codes come with some limitations in terms of security and usability [2]. An example of this limitation is a guess attack where an attacker guesses the user's credentials as most people tend to choose simple PINs like their birth dates as their credentials. In worse cases users leave their devices without any form of protection by opting not to use a password [3]. More than 555 million passwords have been obtained through data breaches and exposed to the public domain. Some 27% of those surveyed in a Google poll admitted attempting to guess the passwords of others, 17% of whom claimed to have succeeded, and, according to one report, 80% of hacking incidents are enabled by stolen and reused login information, 81% of which at the company level are caused by the many poor passwords among the 300 billion in use [5].

To combat this issue, new research has started proposing the use of behavior-based authentication means such as keystroke dynamics, phone movement and gait touch as a replacement of passwords for user authentication. Behavioral biometrics require minimal interaction during the authentication process, resulting in an increase in user acceptability [3]. Keystroke dynamics, for example, refers to the unique patterns of rhythm and timing-based features that are created when a user types on a touchscreen in computing devices such as mobile devices [4].

This paper discusses how continuous behavioral biometrics can be implemented and the improvements it brings. An implicitly continuous reauthentication method should keep authenticating the user, in addition to the initial login authentication, without interrupting users. This can detect an adversary once he gets control of the smartphone and can prevent him from accessing sensitive data or services via smartphones, or inside smartphones. One such systems that employs this system is SmarterYou. SmarterYou enables implicit, continuous authentication as a background service, when the users use smartphones.

## 2. Background

The rapid growth of digital technologies and the rise of advanced cyber threats have exposed significant limitations in traditional authentication methods, such as passwords and PINs. These methods are highly vulnerable to breaches, guessing attacks, and misuse [2]. Additionally, poor usability, password fatigue, and weak user practices exacerbate their shortcoming [2] [3]. Biometric computer authentication offers an advantage over password and access card authentication in that it is based on something you are, which is not easily copied or stolen [13]. These include such as keystroke rhythms, mouse dynamics, and gait patterns, to provide continuous and adaptive authentication [4] [9]. Unlike traditional methods, behavioral biometrics

passively monitors user behavior without interrupting workflows, making it harder for attackers to mimic legitimate user [9] [11]. The increasing adoption of IoT devices and smart environments further underscores the need for research in this area, as these devices demand seamless, scalable, and continuous authentication solutions. However, challenges such as data variability, privacy concerns, and real-time processing constraints hinder widespread adoption [9] [14]. Research into advanced machine learning techniques can address these challenges by enabling systems to extract meaningful features, adapt to user behavior changes, and detect threats [12]. By advancing behavioral biometrics secure authentication systems that meet the demands of today's evolving technological landscape can be developed.

## 3. Systematic Review Methodology

## **Research Questions**

The primary goal of this literature review is to analyze the advancements and challenges in continuous user authentication using behavioral biometrics.

RQ1: What behavioral traits have been most commonly explored for continuous authentication, and what are their strengths and limitations?

RQ2: What methodologies and models are used for feature extraction and classification in continuous authentication systems?

RQ3: What challenges exist in implementing continuous authentication systems, and how are these being addressed?

## **Literature Selection**

The literature search was conducted in three phases: identifying phase, diversity phase and summary phase.

## **Identifying phase**

The review sourced journals and research papers from academic search engines. These engines included google scholar, IEE to name a few. The keywords used in the search process were behavioral biometrics, continuous authentication and machine / deep learning. This made the pool of papers generated closely related and addressing different approaches to the topic.

## **Diversity phase**

In this phase, the selected papers aimed to get wide-ranging approaches, methodologies, and applications of continuous authentication using behavioral biometrics. By emphasizing diversity, this phase ensured that the literature review did not overlook smaller areas e.g authentication in smart watches [7].

## Summary phase

Every paper was summarized into three categories, namely: overview, dataset, methodology, and limitations. The overview provided a general understanding of the study's objectives, scope, and key findings, while the dataset section detailed the data sources, collection methods, and sample characteristics used in each research. The methodology section focused on the machine learning models, feature extraction techniques, and evaluation metrics applied, and the limitations highlighted challenges.

## 4. Thematic Analysis and Feature Extraction Techniques

Continuous authentication leverages behavioral traits unique to individuals to validate their identity dynamically. Unlike traditional biometric methods such as fingerprints or facial recognition, behavioral biometrics focuses on patterns in user interactions, offering the advantage of passive and continuous monitoring. This ensures a higher level of security, as authentication is not limited to a single point in time but runs throughout a session. This section talks about various behavioral modalities used in continuous authentication, highlighting their unique characteristics, advantages, and limitations. The modalities discussed include:

- Keystroke Dynamics: Typing patterns that capture timing and rhythm during text input.
- Mouse Dynamics: Movement and interaction patterns with a mouse or other pointing devices.
- Touch Dynamics: Gestures, pressure, and interaction styles on touchscreens.
- Gait and Motion Patterns: Physical movement traits measured via sensors like accelerometers and gyroscopes.

## Keystroke Dynamics

Keystroke dynamics refers to the unique patterns of rhythm and timing-based features that are created when a user types on a touchscreen in computing devices such as mobile devices [4]. The biometric system uses a pattern recognition system to classify users based on their physical and behavioral characteristics [12]. This type of system uses artificial intelligence to differentiate legitimate users and illegitimate users [10].

As illustrated in fig 1., features commonly extracted includes dwell time—the duration a key is pressed—flight time, which is the interval between releasing one key and pressing the next [8], and typing speed and error rate, all of which can vary among users. Keystroke-based authentication offers advantages such as easy data collection, since no additional hardware is required beyond standard keyboards or touchscreens, and it is non-intrusive, integrating seamlessly into existing workflows [15]. However, challenges include the variability of typing patterns, which can be influenced by factors like stress, fatigue, or different keyboard types [16]. Additionally, the way in which an individual types on a keyboard is considered less permanent than other traits, such as physiological biometrics (e.g., facial and fingerprint recognition) [16]. The need for sufficient data, as shorter text inputs may not provide enough variability to ensure reliable authentication, is also a limiting factor. There are also privacy concerns with this form of authentication [17].



Fig 1. dwell time and flight time of keystroke dynamics

## **Mouse dynamics**

Mouse dynamics refers to the analysis of user interactions with a pointing device, such as a mouse or trackpad, capturing behavioral patterns like cursor movement trajectories, speed, acceleration, click frequency, and drag-and-drop behaviors [9]. These interaction characteristics are considered relatively unique to individuals, making mouse dynamics a viable form of behavioral biometric for user authentication [18].

Features analyzed in this form of authentication include cursor movement (trajectory, speed, acceleration), click frequency (patterns of left and right clicks), and drag-anddrop behavior as illustrated in fig. 2. By combining these features with machine learning algorithms like SVMs and Random Forests, a high accuracy system is developed and there is rapid detection of impostors [20]. Unlike traditional methods, mouse dynamics enable continuous monitoring during regular activities without requiring specialized hardware, making them both non-intrusive and cost-effective [19]. However, one drawback of this approach is task dependency (browsing versus gaming, for example) and hardware variability, since different pointing devices (trackpads vs. mice) may yield inconsistent data [9].



Fig 2. Tracking mouse actions w.r.t X, Y plane

## **Touch Dynamics**

Touch dynamics, as shown in fig. 3, analyzes how users interact with touchscreens by capturing features such as swipe patterns (speed, direction, and consistency), tap pressure and duration, and multi-touch gestures like pinch, zoom, and rotation [6]. This form of behavioral biometric is well-suited for mobile devices like smartphones and tablets, using their touch interfaces for seamless integration. Touch dynamics provide a high degree of personalization since interaction styles tend to be unique to individuals [21]. Research indicates that specific features like average swipe velocity and the largest deviation from end-to-end line are particularly effective in user authentication, as these characteristics remain distinct across individuals [21].

However, challenges such as hardware sensitivity where variations in touchscreen responsiveness across devices can affect data consistency and context variability, can affect the validity of this form of authentication.



Fig 3. user interactions with touchscreen

## **Gait Dynamics**

Gait and motion patterns rely on data from sensors like accelerometers, as illustrated in fig. 4, and gyroscopes to identify individuals based on their physical movements, with features analyzed including step length and cadence, body orientation measured through gyroscopic data, and motion variability over time [11]. Research highlights the effectiveness of gait-based biometrics, showing that accelerometer-based systems can achieve an accuracy of over 90% in ideal conditions by focusing on metrics such as stride patterns and stability [22]. These systems, commonly integrated into mobile and wearable devices, allow for passive data collection in the background without user involvement. It also clearly identifies unique users as walking styles are different. However, challenges such as environmental influences like variations in sensor placement, and the inability to authenticate when users are stationary can be a limiting factor. Additionally, the reliance on device-specific sensors can introduce variability in data quality and impact performance [22]. Recent advances suggest that combining gait data with other modalities, such as community-driven data for personalized modeling, can enhance robustness and adaptability in diverse settings, further improving the practicality and reliability of gait-based biometric systems [23].



#### 5. Data Extraction and Processing

Effective behavioral biometric systems rely heavily on robust data collection and thorough cleaning processes to ensure the reliability of machine learning models.

#### **Keystroke Dynamics**

Keystroke dynamics data is collected by recording the timing patterns of users' typing behavior, capturing features like dwell time (time a key is pressed), flight time (time between releasing one key and pressing another), and typing speed. Collection is often done through custom applications or frameworks, such as the iProfile app (illustrated in fig. 5), which records keypress events along with metadata like timestamps, pressure, and coordinates [4]. Publicly available datasets, such as HMOG, also provide rich data sources for evaluating keystroke dynamics, including thousands of keypress events collected in controlled settings across various tasks like reading, typing, and navigation [26]. Cleaning the collected data involves several critical steps to ensure accuracy and consistency. Noise removal techniques are applied to address inconsistencies caused by irregular typing behaviors, device variations, or environmental factors. For instance, outlier detection methods, such as interquartile range (IQR) filtering and z-score analysis, are commonly used to eliminate anomalous data points [4] [26]. Normalization processes, such as scaling features like timing intervals and pressure to a standard range, ensure uniformity across datasets, particularly when integrating data from multiple users or devices [5]. Segmentation of typing data into fixed-length sequences or windows is another important preprocessing step. This method ensures that input samples are comparable across users, enabling models to analyze keystroke patterns within consistent contexts. Feature extraction algorithms are then used to derive advanced metrics, such as n-graph timing and spatial data, which enhance the robustness of user authentication models [5][26].



Fig 5 iProfile app allows the users to type the passcode via the virtual keypad.

#### **Mouse Dynamics**

Mouse dynamics data collection typically involves monitoring user interactions with a pointing device, such as a mouse or trackpad, through specialized software that captures raw events like cursor movements, clicks, and drag-and-drop actions. For example, in one study, data was collected by intercepting low-level mouse events, including event types, timestamps, and coordinates, which were then aggregated into higher-level features, such as movement trajectories, angles, and speeds as shown in fig. 6 [9] [18]. Public datasets, offering pre-recorded mouse activities performed under various real-world conditions were also used [19]. Cleaning the raw data involves several preprocessing steps to ensure accuracy and usability for machine learning models. Noise introduced by device inconsistencies, such as varying mouse sensitivities or environmental factors, is addressed using filters like median smoothing and normalization techniques. Data discretization is another critical step, where continuous variables like angles, speeds, and distances are divided into fixed intervals, such as angle bins or pixel ranges, to reduce variability and enhance feature extraction [9] [20]. Outlier detection methods, such as analyzing high-percentile movement speed distributions, help remove anomalous data caused by unintended user actions [9]. Segmentation of mouse actions into specific categories, such as "Point and Click" or "Drag and Drop," ensures that models can focus on distinct behavioral traits during analysis. Additionally, the construction of histograms to summarize features like movement direction, elapsed time, and speed distributions aids in capturing the essence of user interaction patterns [20].



Fig 6. percentile values from the movement speed by movement angle data set.

## **Touch Dynamics**

Data collection for touch dynamics primarily involves using the built-in sensors of mobile devices, such as touchscreens and accelerometers, to record user interaction patterns, including swipe gestures, tap pressure, and duration. These interactions, highlighted in fig. 7 are captured during various user activities to create datasets representing diverse behavioral patterns. For instance, the Touchstroke system collects data in controlled settings, recording user interactions under different conditions (e.g., sitting, walking) to ensure variability and robustness in the dataset [3]. Cleaning the collected data involves several critical steps to enhance its quality and ensure reliability for model training. Noise introduced by hardware inconsistencies or environmental factors is addressed using filtering techniques, such as low-pass and median filters. Additionally, techniques such as Z-score analysis and interquartile range (IQR) filtering are employed to detect and remove outliers caused by irregular user interactions or device errors, ensuring a cleaner dataset [23].

Alignment of data across sensors is another vital preprocessing step, particularly for multimodal datasets that combine touch interactions with sensor readings like orientation and accelerometry. Data normalization methods, such as min-max scaling, are used to bring features into a uniform range, facilitating better model performance during training [22]. Segmentation of continuous data into fixed-length windows ensures consistency and enables models to analyze user behavior over comparable timeframes.



Fig 7 Stroke features projected on a 2D-subspace. The user ID is given as a colored number.

#### **Gait Dynamics**

Data collection typically involves the use of inertial measurement units (IMUs), such as accelerometers and gyroscopes, embedded in mobile devices and wearables to capture motion patterns like step frequency, stride length, and body orientation [11]. Publicly available datasets like whuGAIT and OU-ISIR are widely utilized, as they provide standardized protocols and diverse samples, enabling the benchmarking of different models and preprocessing techniques [27]. However, challenges like sensor placement inconsistencies (e.g., pocket versus wrist) and environmental noise make preprocessing during the cleaning phase a bit more challenging. Data cleaning begins with noise removal, as illustrated in fig. 8, to handle artifacts caused by device malfunctions, user behavior, or environmental factors. Techniques such as low-pass filters are employed to smooth accelerometer signals, while median filters address spikes in gyroscope data. Outlier detection methods, including z-score analysis and interquartile range (IQR) filtering, are applied to identify and remove irregular data points, ensuring that anomalous readings do not skew the models' performance [11].

Another critical step is aligning and segmenting the collected data into fixed-length windows, which bypass the need for precise gait cycle detection—an error-prone process due to variability in sensor sensitivity and placement. This segmentation ensures consistent input for models, allowing them to focus on key temporal and spatial patterns [27].



Fig 8. processed ECG signal

## 6. Machine Learning Techniques

After processing behavioral data, it has to be converted into actionable authentication decisions. This section discusses the different approaches employed supported by insights from the referenced research papers.

## **Keystroke Dynamics**

Support Vector Machines (SVMs) are a popular choice, utilizing hyperplane optimization to classify user behaviors such as keystroke dynamics. For instance, SVMs with multi-class classification and one-vs-one decision functions demonstrated effective user authentication through keystroke data, supported by feature selection techniques like minimum redundancy maximum relevance (mRMR) to enhance classification performance [4] [5]. Ensemble methods, including Random Forest Classifiers (RFC) and Gradient Boosted Classifiers (GBC), have shown superiority in continuous authentication scenarios, particularly in keystroke dynamics, due to their ability to capture non-linear patterns and handle noisy data [26]. Additionally, probabilistic models like Naive Bayes and clustering techniques such as k-means have been employed for tasks like trajectory dissimilarity analysis in mouse and keystroke dynamics, achieving high accuracy while addressing variances in user behavior [5]. These techniques, combined with preprocessing methods like data normalization and outlier detection, create robust systems capable of adapting to the dynamic nature of behavioral biometrics [4] [26].

## 7. Mouse Dynamics

Naive Bayes classifiers have been applied, leveraging probabilistic modeling to analyze features like movement speed, acceleration, and double-click intervals for anomaly detection in user behavior [9]. Support Vector Machines (SVMs) have been widely adopted due to their ability to handle high-dimensional data and deliver strong classification performance; for instance, they are used with Gaussian kernels to classify mouse movement patterns, achieving low false acceptance and rejection rates [19] [20]. Decision Tree Learning and Random Forest Classifiers have also been employed,

providing interpretable models and robustness against noisy data, especially when analyzing complex features like drag-and-drop actions and movement directions [19][28].

Additionally, hybrid approaches using clustering algorithms like k-means and scorelevel fusion techniques have demonstrated efficacy in adapting to user behavior variability [20]. These machine learning models, combined with advanced preprocessing techniques such as histogram-based feature extraction and dynamic binning of data, enable systems to efficiently classify user behavior while maintaining adaptability and robustness in diverse real-world scenarios [21].

## **Touch Dynamics**

Methods like Support Vector Machines (SVMs) and Random Forests are commonly employed for feature classification, as they effectively handle the high-dimensional data generated from sensory inputs [3] [22]. Neural networks, including Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, have been utilized to capture temporal and spatial features from continuous data streams, enabling precise modeling of user-specific patterns [23]. Semi-supervised approaches, such as self-learning frameworks and clustering techniques, have been adopted to address challenges in labeled data scarcity by leveraging community-driven data to personalize models [23]. Bayesian classifiers and Gaussian Mixture Models (GMMs) also play a critical role in probabilistic modeling for touch dynamics, especially in detecting anomalies in user behavior [3]. Together, these machine learning methodologies provide robust frameworks for continuous, non-intrusive, and adaptive authentication systems tailored to individual user behaviors.

## **Gait Dynamics**

Traditional methods often relied on Support Vector Machines (SVMs) and Decision Trees, which effectively analyzed extracted features like step frequency and stride length but faced limitations in handling complex temporal dynamics [11]. Transformerbased architectures have emerged as state-of-the-art solutions for gait recognition due to their ability to process long sequences in parallel and capture global dependencies through self-attention mechanisms [32]. Variants like Informer and Autoformer have been tailored for time-series data, reducing computational complexity and enhancing memory efficiency [27]. Feature fusion techniques, such as combining gait data from multiple sensors (e.g., accelerometers and gyroscopes), further enhance model robustness by integrating complementary information, as seen in multimodal systems [11]. These advances underscore the growing potential of deep learning and transformer-based models in refining gait-based biometric authentication systems, ensuring higher reliability and adaptability in diverse environments.

## 8. Discussion & Analysis

Keystroke Dynamics							
Features	Keystroke dynamics analyze typing behavior, focusing on						
	dwell time (duration a key is pressed), flight time (interval						
	between releasing one key and pressing another), typing						
	speed, and error rate [1].						
Advantages	Non-intrusive, integrates seamlessly with workflows,						
	requires no additional hardware, and supports passive						
	data collection [1].						
Applications	Commonly used for authentication on keyboards and						
	touchscreens, enhancing security for banking systems,						
	enterprise access control, and personal devices [24].						
Insights	Offers effective authentication when consistent input is						
	available, with typing speed and rhythm providing						
	individualization. [14].						
Data Collection	Captured using software-based keystroke logging tools or						
	datasets such as <b>HMOG</b> , which collect typing activity in						
	real-world conditions [4].						
Machine Learning	SVMs, Random Forests, Naïve Bayes, CNNs, Autoencoders,						
Models	and Deep Belief Networks (DBNs) are used for feature						
	classification and anomaly detection [4][30].						
	Mouse Dynamics						
Features Mouse dynamics capture user interactions with pointi							
	devices, such as cursor trajectories, speed, acceleration,						
	click frequency, and drag-and-drop behavior [29].						
Advantages	Enables continuous monitoring without requiring						
	specialized hardware, cost-effective, and highly adaptable						
	to various tasks [18].						
Applications	Primarily used for authentication during computer and						
	web usage, offering security for shared systems, multi-user						
	environments, and secure networks [19].						
Insights	Achieves high accuracy using ML algorithms like SVM and						
	Random Forests, with rapid impostor detection and						
	adaptability to different usage patterns [28].						
Data Collection	Captured using logging tools that track real-time mouse						
	events such as movement trajectories, click timestamps,						
	and acceleration curves [28].						
Machine Learning	Decision Trees, Random Forests, Hidden Markov Models						
Models (HMM), and Deep Learning-based hybrid approaches I							
	CNN-RNN fusion. [28][19][20].						
Touch Dynamics							
Features	Touch dynamics analyzes interaction with touchscreens,						
	including swipe patterns (speed, direction, and						

	consistency), tap pressure and duration, and multi-touch						
	gestures like pinch, zoom, and rotation [25].						
Advantages	Provides high personalization due to distinct touch						
-	interaction styles, integrates seamlessly with mobile						
	devices, and is effective for touch-based authentication						
	[25].						
Applications	Applied in authentication for smartphones, tablets, and						
	other touch-based devices, and widely used for securing						
	mobile banking applications and app-based services						
	[25][31].						
Insights	Demonstrates reliability through features like swipe						
0	velocity and end-to-end deviation, capitalizing on the						
	ubiguity of touch-based devices for user authentication						
	[25]						
Data Collection	Captured via mobile sensors such as capacitive						
	touchscreens, gyroscopes, and accelerometers in real-time						
	logging applications [31].						
Machine Learning	CNNs, LSTMs, Bayesian Classifiers, Gaussian Mixture						
Models	Models, and hybrid deep learning frameworks leveraging						
	self-learning models for behavioral adaptation [25].						
	Gait Dynamics						
Features Gait and motion patterns rely on sensors like							
accelerometers and gyroscopes to analyze traits such a							
	accelerometers and gyroscopes to analyze traits such as						
	accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion						
	accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].						
Advantages	accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27]. Allows passive data collection without user involvement, is						
Advantages	accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27]. Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages						
Advantages	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> </ul>						
Advantages Applications	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile</li> </ul>						
Advantages Applications	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness</li> </ul>						
Advantages Applications	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> </ul>						
Advantages Applications Insights	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions,</li> </ul>						
Advantages Applications Insights	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait</li> </ul>						
Advantages Applications Insights	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and</li> </ul>						
Advantages Applications Insights	accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27]. Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11]. Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32]. Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and adaptability [27].						
Advantages Applications Insights Data Collection	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and adaptability [27].</li> <li>Collected via inertial measurement units (IMUs) such as</li> </ul>						
Advantages Applications Insights Data Collection	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and adaptability [27].</li> <li>Collected via inertial measurement units (IMUs) such as accelerometers and gyroscopes embedded in</li> </ul>						
Advantages Applications Insights Data Collection	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and adaptability [27].</li> <li>Collected via inertial measurement units (IMUs) such as accelerometers and gyroscopes embedded in smartphones, smartwatches, or dedicated wearable</li> </ul>						
Advantages Applications Insights Data Collection	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and adaptability [27].</li> <li>Collected via inertial measurement units (IMUs) such as accelerometers and gyroscopes embedded in smartphones, smartwatches, or dedicated wearable devices [32].</li> </ul>						
Advantages Applications Insights Data Collection Machine learning	<ul> <li>accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27].</li> <li>Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11].</li> <li>Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32].</li> <li>Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and adaptability [27].</li> <li>Collected via inertial measurement units (IMUs) such as accelerometers and gyroscopes embedded in smartphones, smartwatches, or dedicated wearable devices [32].</li> <li>RNNs, Transformer-based architectures (Informer,</li> </ul>						
Advantages Applications Insights Data Collection Machine learning Models	accelerometers and gyroscopes to analyze traits such as step length, cadence, body orientation, and motion variability over time [11][27]. Allows passive data collection without user involvement, is suitable for wearables and mobile devices, and leverages the uniqueness of walking styles for robustness [11]. Employed in authentication through wearables and mobile devices, as well as healthcare monitoring and fitness tracking [32]. Studies show over 90% accuracy under ideal conditions, particularly for stride patterns and stability. Combining gait data with other modalities improves reliability and adaptability [27]. Collected via inertial measurement units (IMUs) such as accelerometers and gyroscopes embedded in smartphones, smartwatches, or dedicated wearable devices [32]. RNNs, Transformer-based architectures (Informer, Autoformer), Autoencoders, and CNN-RNN hybrids for						

#### Table 1. comparative analysis of behavioral biometric modalities

Table 1 provides a comparative analysis of four behavioral biometric modalities: keystroke dynamics, mouse dynamics, touch dynamics and gait dynamics. It outlines key aspects such as definitions, features, advantages, challenges, applications, data collection methods, preprocessing techniques, machine learning models, and performance outcomes. This table serves as a good reference for understanding how these biometrics work individually.

## 9. Limitations/Challenges

Behavioral biometrics often requires extended data collection periods to create a reliable user profile. Long data collection times reduce the practicality of real-time systems, especially for high-security applications [9] [25] [23]. "Current span of an hour to two hours is probably too long to be actually deployed for security extension, as an intruder may harm or exploit the system in much shorter duration" (C. Yoon, D. Kim, 2009). Behavioral biometric systems also face challenges due to the natural variability in human behavior across different sessions. Factors such as mood, physical condition, stress levels, or environmental changes can significantly impact patterns like typing rhythm, touch gestures, or gait. For instance, a user feeling fatigued might type more slowly or interact differently with a touchscreen compared to when they are alert. These variations can lead to an increase in false positives, where legitimate users are incorrectly rejected, or false negatives, where intruders are mistakenly accepted. Such inconsistencies highlight the need for adaptive models that can account for behavioral fluctuations over time, improving the system's robustness and reducing error rates [24] [21][19]. Behavioral biometric systems are highly sensitive to variations in device hardware and the context of use. Differences in hardware, such as mouse sensitivity settings, screen size, touchscreen responsiveness, or even sensor calibration in wearables, can lead to inconsistencies in the data collected. For instance, a user interacting with a high-resolution touchscreen may produce different touch dynamics compared to using a less responsive device, which can impact the accuracy of authentication systems. Similarly, context, such as whether the user is sitting, standing, or walking, can influence behavioral patterns like touch pressure, swipe velocity, or gait. Such variability poses challenges in maintaining a consistent feature set across different devices and environments. Addressing these challenges requires sophisticated normalization techniques or models capable of adapting to diverse hardware and contextual conditions to ensure reliable performance [24][25]. Continuous monitoring of behavioral data introduces significant ethical challenges, particularly regarding user consent and data security. The constant collection of sensitive behavioral information, such as keystroke patterns, touch gestures, or movement data, raises privacy concerns, as users may feel uneasy about being continuously tracked without explicit and ongoing consent. Additionally, the risk of data breaches or unauthorized access to this information could deter users from adopting such systems, especially if the collected

data includes highly personal traits. Ensuring transparency, secure data handling, and offering users control over their data are essential steps in addressing these concerns and promoting trust in behavioral biometric systems [9][15]. One of the key challenges in developing behavioral biometric systems is acquiring sufficiently large and diverse datasets for training and testing. Small or biased datasets can limit the reliability of models and hinder their ability to generalize effectively to new users or varying conditions. For instance, if a dataset lacks variability in user demographics, device types, or interaction contexts, the resulting models may perform poorly when deployed in real-world scenarios. Additionally, collecting behavioral data often requires prolonged and consistent user participation, which can be difficult to achieve. These limitations not only affect the robustness of the models but also complicate efforts to evaluate their effectiveness comprehensively [9][6][20].

## **10.** Conclusion

This paper provides a literature review of continuous user authentication with behavioral biometrics via machine learning. It discusses the unique user patterns used to authenticate such as keystroke dynamics, mouse dynamics, touch gestures, and gait. These systems offer significant advantages, including passive monitoring, higher personalization, and seamless integration with modern devices, making them wellsuited to meet the demands of an increasingly connected world. While all these are highlighted, there is room for improvement through more studies as these drawbacks are delaying the widespread implementation of this for of authentication. Variability in user behavior, sensitivity to hardware and environmental conditions, ethical concerns around privacy, and the need for large, diverse datasets for robust machine learning models are just some of the issues that are slowing down progress. Despite these challenges, advancements in feature extraction, adaptive machine learning models, and multimodal systems show some promises in overcoming the challenges. To address these issues, future research should explore developing models that adapt to changing behavioral metrics that can affect results caused by factors like stress, fatigue, or environmental changes. Another area of research should be integrating behavioral biometric authentication with the already existing traditional systems to boost resilience. As cyber threats continue to evolve, the importance of robust authentication systems is a top priority. Behavioral biometrics, powered by machine learning, represents a critical step toward developing secure, scalable, and user-friendly authentication frameworks. With continued research, a safer digital environment can be created that is both scalable and accessible.

#### References

1. Y. Piugie, J. Manno, C. Rosenberger, and C. Charrier, "How artificial Intelligence can be used for Behavioral Identification?"

- 2. G. Stragapede, R. Vera-Rodriguez a, R. Tolosana, A. Morales, A. Acien, G. Lan, "Mobile behavioral biometrics for passive authentication"
- 3. A. Buriro, B.Crispo, F. Del Frari, and K. Wrona, "Touchstroke: Smartphone User Authentication Based on Touch-Typing Biometrics"
- 4. S. Krishnamoorthy, L. Rueda, S. Saad, H. Elmiligi, "Identification of User Behavioral Biometrics for Authentication Using Keystroke Dynamics and Machine Learning"
- 5. I. Oladipo, M. Nicho, J. Awotunde, J. Buari, M. Abdulraheem, T. Gaber, "A Secure and Scalable Behavioral Dynamics Authentication Model"
- 6. W. Lee, R. Lee "Implicit Smartphone User Authentication with Sensors and Contextual Machine Learning"
- 7. C. Lu, B. Du, X. Kan, H. Wen, A. Markham, N. Trigoni, "VeriNet: User Verification on Smartwatches via Behavior Biometrics"
- 8. S. Giroux, R. Wachowiak-Smolikova and M. Wachowiak, "Keystroke-Based Authentication by Key Press Intervals as a Complementary Behavioral Biometric"
- 9. C. Yoon and D. Kim, "User Authentication Based On Behavioral Mouse Dynamics Biometrics"
- 10. K. Tse, and K. Hung , "Framework for user behavioural biometric identification using a multimodal scheme with keystroke trajectory feature and recurrent neural network on a mobile platform"
- 11. G. Bouchiba, R. Tlemsani, S. Chouraqi and M. Benouis, "An improved behavioral biometric system based on Gait and ECG signals"
- 12. A. Aljahdali, F. Thabit, H. Aldissi, W. Nagro, "Dynamic keystroke Technique for a secure Authentication System based on Deep Belief Nets"
- 13. K. Bailey, J. Okolica, and G. Peterson, "User identification and authentication using multi-modal behavioral biometrics,".
- 14. C. Meng, P. Gupta, D. Gao, "I can be You: Questioning the use of Keystroke Dynamics as Biometrics Tey "
- O. Finnegan, J. White, B. Armstrong, E. Adams, S. Burkart, M. Beets, S. Nelakuditi, E. Willis, L. von Klinggraef, H. Parker, M. Bastyr, X. Zhu, Z. Zhong and R. Weaver, "The utility of behavioral biometrics in user authentication and demographic characteristic detection: a scoping review"
- 16. P. Teh, A. Teoh, and S. Yue, "A Survey of Keystroke Dynamics Biometrics"
- 17. I. Stylios, S. Kokolakis, O. Thanou, S. Chatzis, "Behavioral biometrics & continuous user authentication on mobile devices: A survey"
- 18. Z. Jorgensen and T. Yu, "On Mouse Dynamics as a Behavioral Biometric for Authentication"
- 19. S. Mondal and P. Bours, "Continuous authentication using Mouse Dynamics"
- 20. C. Feher, Y. Elovici, R. Moskovitch, L. Rokach, A. Schclar, "User identity verification via Mouse Dynamics"
- 21. A. Jairam, T. Halevi, and Theodore Raphan, "Improving Mobile Device Security by Emboding and Adapting a Behavioral Biometric Interface"
- 22. M. Abuhamad, A. Abusnaina, D. Nyang, and D. Mohaisen, "Sensor-based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey"

- 23. E. Garcia-Ceja and R. Brena "Activity Recognition Using Community Data to Complement Small Amounts of Labeled Instances"
- 24. E. Lau, X. Liu, C. Xiao, and X. Yu, "Enhanced User Authentication Through Keystroke Biometrics"
- 25. M. Frank, R. Biedert , E. Ma, I. Martinovic, D. Song, "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication"
- 26. L. de-Marcos, J. Martínez-Herráiz, J. Junquera-Sánchez, C. Cilleruelo and C. Pages-Arévalo, "Comparing Machine Learning Classifiers for Continuous Authentication on Mobile Devices by Keystroke Dynamics"
- 27. P. Delgado-Santos, R. Tolosana , R. Guest, F. Deravi, R. Vera-Rodriguez, "Exploring transformers for behavioural biometrics: A case study in gait recognition"
- 28. N. Siddiqui, R. Dave, M. Vanamala, N. Seliya, "Machine and Deep Learning Applications to Mouse Dynamics for Continuous User Authentication"
- 29. A. Jaiswal, P. Tiwari, M. Hossain, "Predicting users' behavior using mouse movement information: an information foraging theory perspective"
- 30. J. Liu, B. Zhang, L. Shen, J. Liu, J. Zhao, "The BeiHang Keystroke Dynamics Authentication System"
- 31. A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and I know it's you!: implicit authentication based on touch screen patterns,"
- 32. H. J. Ailisto, M. Lindholm, J. Mantyjarvi, E. Vildjiounaite, and S.-M. Makela, "Identifying people from gait pattern with accelerometers

# A Web Application for Walmart Store Sales Forecasting

Biyasu A and Mao Zheng

Department of Computer Science & Computer Engineering

University of Wisconsin – La Crosse

La Crosse, WI 54601

mzheng@uwlax.edu

## Abstract

Sales forecasting is critical to business success, serving as a foundation for anticipating future trends and optimizing inventory management and marketing strategies. By predicting sales trends, businesses can streamline operations, optimize inventory levels, and allocate resources more effectively. Recognizing the importance of sales forecasting, this project aims to leverage historical sales data to develop a robust and reliable sales forecasting system.

This manuscript details the design and development of a web application system for Walmart store sales forecasting. The project utilizes the **Walmart Recruiting - Store Sales Forecasting** dataset, available on Kaggle, for training and testing purposes. This dataset includes historical sales data from February 5, 2010, to November 1, 2012, from 45 Walmart stores across various regions, spanning multiple departments. It contains key information such as store numbers, department numbers, dates, weekly sales, and holiday indicators.

The web application enables users to select specific stores and departments to view weekly sales forecasts. It offers visualizations of past sales trends, including year-over-year comparisons, which provide valuable insights such as peak sales periods, trends, or anomalies identified by the model. Additionally, the system displays performance metrics like RMSE to build user trust and confidence in the predictions. The forecasting model uses XGBoost to enhance predictive accuracy, with data preprocessing and analysis handled using Pandas.

The project employs modern technologies, including React HTML, and CSS for the frontend, and Python, Flask, and MySQL for the backend. The application is deployed on a Linux server for efficient and reliable performance.

# **1** Introduction

In the retail industry, accurate sales forecasting is critical to optimizing inventory management, resource allocation, and overall business operations. By predicting sales trends, companies can make informed decisions to align their stock levels with customer demand, improving both efficiency and profitability. For example, in inventory management, sales forecasts help retailers avoid overstocking or stockouts, ensuring that the right products are available at the right time. In marketing, forecasts enable businesses to plan promotional campaigns more effectively by identifying peak sales periods and customer preferences. Additionally, sales forecasting plays a vital role in financial planning, allowing companies to allocate budgets and resources more efficiently, thereby enhancing long-term financial stability.

To explore the machine learning techniques in the sales forecasting, this project leverages a dataset from Kaggle's "Walmart - Recruitment Prediction Competition", which provides comprehensive historical sales data from multiple retail stores. The dataset is provided in ".csv" format and includes four main tables: stores.csv, train.csv, test.csv, and features.csv. The stores.csv file contains detailed information about 45 stores, including store types (e.g., Type A, B, C) and sizes, which are crucial for understanding the scale and characteristics of each store. The train.csv file serves as the historical training data, covering weekly sales from February 5, 2010, to November 1, 2012. This file includes store IDs, department IDs, weekly sales figures, and holiday indicators, providing a foundation for training the forecasting model. The test.csv file is structurally similar to the train.csv file, but excludes weekly sales, making it suitable for evaluating the model's performance on unseen data. Finally, the features.csv file provides additional contextual data related to store, department, and regional activities, such as temperature, fuel prices, promotional markdowns, and economic indicators like the Consumer Price Index (CPI) and unemployment rates. These features are essential for capturing external factors that influence sales trends.

In our approach, we developed a web-based system that allows users to review sales projections for various stores and departments. In this regard, it offers a visualization of historical sales trends, also including year-over-year analysis. It shows information about peak sales seasons, trends, and anomalies identified by the model. Additionally, the system evaluates model performance using metrics such as Root Mean Squared Error (RMSE)[1], Mean Absolute Error (MAE), and Coefficient of Determination (R<sup>2</sup>), enhancing user confidence in the predictions. The forecasting model uses XGBoost[2], a powerful machine learning algorithm known for its efficiency and accuracy, combined with optimization techniques to achieve optimal performance.

## 2 Requirements

The system is designed to facilitate sales forecasting and analysis by allowing users to upload, process, and visualize sales data. Key features include log in, log out, real-time data updates, interactive forecast visualization, data management and downloadable reports. Through these functionalities, the system enhances decision-making by providing accurate and timely sales predictions.

2.1 Data visualization

Sales Forecasting Visualization

 Data Table – The Data Table displays the forecasted results generated during the model training phase using the test dataset provided by Kaggle. This dataset includes detailed information such as store IDs, department IDs, predicted weekly sales, and other relevant features. Users can filter the data based on specific criteria (e.g., store, department, or date range) and export the results for further analysis.

						Export	Forecasted Dat	a as CSV	Reset Filters				
5	Sto	re ID:	<b>~</b> 2	De	pt ID:	~	All	Holiday:	✓ All	Year	:	~ Ali	Monti
Forecasted Sales Data "Displays all forecasted data, including detailed information on stores, departments, sales, and economic factors, which can be used for filtering and export analysis													
Store ID	Dept ID	Date	Forecasted Sales	IsHoliday	Temperature	Fuel Price	MarkDown1	MarkDown2	Mark Down 3	MarkDown4	Mark Down 5	СРІ	Unemployment
		2012-11-02	13390.39	Yes	57.83	3.386	3526.61	1163.63	2.31	514.05	1555.50	224.06	5.42
		2012-11-09	12441.04	Yes	64.61	3.314	2548.02	2037.54	1.95	303.93	1595.16	224.07	5.42
		2012-11-16	12561.89	Yes	56.89	3.252	1631.01	-35.74	15.46	326.59	2310.83	224.11	5.42
		2012-11-23	12062.30	Yes	59.07	3.211	310.95		43319.43	46.61	372.22	224.16	5.42
5	2	2012-11-30	12102.15	Yes	56.20	3.207	2077.60		2752.82	8.82	3349.59	224.21	5.42
		2012-12-07	12205.98	Yes	66.13	3.198	2472.70		68.00	105.14	2359.84	224.25	5.42

Figure 1: Data table sub-view in Sales Forecasting Page

2. Sales Changes Line Chart – Shows sales fluctuations over a selected time frame, helping users identify patterns of sales growth or decline. In Figure 2, the x-axis represents the date, which by default includes all available years in the dataset. Users can adjust the date range using a selector, which allows filtering sales data within a specific time frame. Additionally, a year selection dropdown enables users to quickly view sales trends for an entire year by choosing a particular year, displaying a line chart that reflects the sales data for the selected year as a whole. The x-axis represents the date, while the y-axis represents the sales volume. By default, the line chart displays all available sales data for the selected store and department, sorted chronologically from left to right based on the date.


Figure 2: Sales Changes sub-view in History Data Page

3. Pie Charts:

Four features, as seen in Figure 3, were selected over others because they provided more meaningful categorical groupings (e.g., holiday vs. non-holiday, markdown types, temperature ranges, months) that directly impact sales performance and business decision-making. While the dataset includes various numerical and categorical attributes, these specific features were chosen for pie chart visualization due to their clear, distinct categories and their influence on sales trends. The use of pie charts for these features enables a straightforward comparison of proportions, making it easy to identify dominant categories and their impact on overall sales.

- Sales on Holidays vs. Non-Holidays Compares sales performance during holiday and non-holiday periods, highlighting the impact of holidays on revenue. By comparing sales during holiday and non-holiday periods, this pie chart highlights the proportion of revenue generated during these key times. This visualization helps businesses understand the importance of holiday promotions and plan their strategies accordingly.
- Markdown Contribution Visualizes the proportion of sales influenced by different markdown categories, showing how discounts affect total revenue. Understanding the contribution of each markdown type is essential for optimizing promotional campaigns and maximizing revenue. This pie chart breaks down the proportion of sales influenced by different markdown categories, providing insights into which types of discounts are most effective.
- Temperature Contribution Weather conditions, represented by temperature ranges, can significantly influence consumer purchasing patterns. This pie chart shows the distribution of sales across different temperature ranges, helping businesses analyze how weather impacts sales. By visualizing the proportion of sales in each temperature range, businesses can adjust their inventory and marketing strategies based on seasonal weather patterns. The distinct temperature categories make this feature a good fit for a pie chart, as it allows for a quick comparison of sales across different weather conditions.

 Monthly Sales Contribution – Represents the share of total annual sales for each month, offering insights into seasonal variations. Seasonal variations are a key aspect of retail sales. This pie chart represents the share of total annual sales for each month. By identifying months with higher or lower sales contributions, businesses can optimize resource allocation and prepare for seasonal fluctuations.



Figure 3: Pie charts sub-view in History Data Page

In summary, these four features were chosen for pie chart visualization because they represent categorical data with clear, distinct categories that have a direct and measurable impact on sales performance. Pie charts are particularly suited for such data, as they allow users to quickly grasp the relative proportions of each category and understand their influence on overall sales. By focusing on these four features, the pie charts provide actionable insights into key factors that influence sales performance, supporting data-driven decision-making in the retail industry.

# 3 Design

## 3.1 Architecture Design

The sales forecasting system adopts a classical client-server architecture, ensuring modularity and scalability. This architecture consists of three key components: the frontend, the backend, and the data layer. The interaction between these components follows the client-server model, where the front-end serves as the client and the backend as the server.

The frontend was developed using React.js, which provides a dynamic and user-friendly interface for users to interact with the application. React was used to design and manage the graphical user interface, enabling variables such as selecting stores and departments to view sales forecasts.

The backend was built with Flask, a lightweight Python web framework. It acts as the application server, handling API requests and responses. Key functionalities include processing prediction requests through an endpoint, loading and using machine learning

4

models, and serving static files. Flask ensures seamless communication between the frontend and the underlying machine learning model, making predictions accessible to the user.

The data layer consists of a MySQL database, which stores the data needed to support the application's operations. This includes historical sales data and other relevant information needed to make accurate predictions. The database acts as the foundation for the system, ensuring that the backend has access to all relevant information to effectively process user requests.

This architectural design allows a clear separation of concerns, allowing the frontend, backend, and data layer to be independently developed, tested, and deployed.

#### 3.2 Machine Learning Model

Machine learning models play a crucial role in sales forecasting by identifying patterns and trends within historical data. Various algorithms can be employed to model the complex relationships between sales and influencing factors such as seasonality, promotions, and holidays. The selection of an appropriate model depends on its ability to generalize effectively to unseen data while maintaining computational efficiency.

The tree-based ensemble methods, including Random Forest, Gradient Boosting, XGBoost, LightGBM, CatBoost, and Extra Trees, offer greater flexibility and the ability to model complex interactions. These models leverage multiple decision trees to enhance predictive performance through techniques such as boosting, bagging, and feature selection.

To ensure a fair comparison, all models were trained on the same preprocessed dataset and evaluated using standardized performance metrics, including Mean Squared Error, Root Mean Squared Error, and R<sup>2</sup> score. Additionally, factors such as model interpretability, training time, and scalability were taken into account.

After data preprocessing, multiple machine learning models were evaluated to determine the most suitable approach for sales forecasting. Models considered included Linear Regression, Random Forest, Gradient Boosting, XGBoost, LightGBM, CatBoost, and Extra Trees. Each model was trained and tested using the preprocessed dataset, and their performance was assessed based on key evaluation metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R<sup>2</sup> score. Additionally, training time was considered a factor to ensure computational efficiency. Specific results are shown in Figure 4.

Among the evaluated models, XGBoost demonstrated the best balance between predictive accuracy and computational efficiency. It consistently achieved lower MSE and RMSE values while maintaining a high R<sup>2</sup> score, indicating its ability to capture complex relationships in the data. Moreover, its training time was reasonable compared to other models, making it a practical choice for deployment.

	MSE	RMSE	R2	Training Time (s)
Linear Regression	486801618.64	22063.58	0.07	0.13
Random Forest	472676918.42	21741.13	0.09	6.96
Gradient Boosting	469955694.77	21678.46	0.10	36.32
XGBoost	469969844.51	21678.79	0.10	0.44
LightGBM	469730239.61	21673.26	0.10	0.52
CatBoost	471440391.37	21712.68	0.10	1.90
Extra Trees	470946686.63	21701.31	0.10	3.93

Figure 4: Performance Evaluation of Different Models

Furthermore, under the same feature set, feature selection, and optimization conditions, XGBoost continued to outperform other models. Its ability to efficiently leverage feature interactions and optimize through gradient boosting contributed to its superior performance. The model maintained a consistently lower error rate while achieving competitive results across all key evaluation metrics. The results, as shown in Figure 5, further confirm XGBoost's robustness in handling the complexities of sales forecasting.

Cross-Validation Results:
Gradient Boosting: MAE: 6935.5326083681475, MSE: 136314305.14525056, RMSE: 11575.44595020601, R <sup>2</sup> : 0.7400099366978286
Gradient Boosting Average Training Time: 21.254805660247804 seconds, Average Prediction Time: 0.02591366767883301 seconds
XGBoost: MAE: 3418.485623310355, MSE: 46930027.396733314, RMSE: 6668.179605471904, R <sup>2</sup> : 0.9120415463644885
XGBoost Average Training Time: 0.35564346313476564 seconds, Average Prediction Time: 0.018239259719848633 seconds
LightGBM: MAE: 4228.169040842926, MSE: 60332321.16220479, RMSE: 7600.206441537515, R <sup>2</sup> : 0.8866238573395397
LightGBM Average Training Time: 0.4817482948303223 seconds, Average Prediction Time: 0.03732576370239258 seconds
Extra Trees: MAE: 4759.218808538336, MSE: 105285558.99048118, RMSE: 9961.142614706509, R²: 0.803612726937853
Extra Trees Average Training Time: 22.214413166046143 seconds, Average Prediction Time: 0.721423864364624 seconds
Best Model: XGBoost
Test Set Prediction Time: 0.023156404495239258 seconds
Total Running Time: 244.143306016922 seconds

Figure 5: Performance Evaluation of Different Models Using the Same Optimized Conditions

# **4** Implementation

The application follows a Client-Server Architecture, which separates the frontend and backend into two distinct components. The frontend, built with React.js, HTML, and CSS, is responsible for rendering the user interface and handling user interactions. React.js features a component-based architecture that enhances modularity, reusability, and maintainability in frontend development.

On the backend, Flask serves as the web framework, handling server-side logic, data processing, and API endpoints. Flask's lightweight and flexible nature makes it ideal for building RESTful APIs that communicate with the frontend. The backend is designed to receive requests from the frontend, process them, and return appropriate responses, typically in JSON format. This separation of concerns ensures that the frontend and backend can evolve independently, improving the overall scalability and maintainability of the application.

Communication between the frontend and the backend is facilitated through HTTP requests and RESTful APIs. The frontend sends requests to the backend to fetch data or perform actions, and the backend processes those requests, interacts with the database or machine learning models, and returns the results to the frontend. For example, when the frontend sends a prediction request, the backend preprocesses the input data, queries the machine learning model, and returns the prediction result. This process is implemented in the /predict route, as shown in the following code snippet (Figure 6):

@app.route('/predict', methods=['POST'])
def predict():
 data = request.json
 processed\_data = preprocess\_data(data)
 prediction = model.predict(processed\_data)
 return jsonify({'predicted\_weekly\_sales': prediction[o]})

#### Figure 6: Code snippet for /predict route

We used the XGBoost model for sales forecasting, with hyperparameter tuning performed using Bayesian optimization[3]. This method efficiently searches for the best parameters, improving the model's performance. After selecting the optimal model, five-fold crossvalidation is applied to evaluate its robustness, ensuring generalization across different data splits. Finally, predictions are generated and saved for further analysis. Part of the implementation process of the above is given in Figure 7.

#### 4.1 Evaluation and Results

The performance of the final model was rigorously evaluated using both cross-validation and a test set to ensure its robustness and generalization capability, as shown in Figure 8. In cross-validation, the model demonstrated strong predictive power, with consistently low error metrics (MAE, MSE, and RMSE) and a high R<sup>2</sup> score across all folds. This indicates that the model effectively captured patterns in the training data and performed well on various subsets of the data, showcasing its stability and reliability.

Cross-Validation MAE: 883.6046261891546
Best MAE: 880.3279726616261
Best MSE: 2168304.8543289746
Best RMSE: 1472.516503924141
Best R <sup>2</sup> : 0.9958502700753075
Test set evaluation metrics:
MAE: 1313.5750124441417
MSE: 7149698.4416365
RMSE: 2673.8920026127644
R <sup>2</sup> : 0.9862893650092011
Total time taken: 577.5177071094513 seconds

Figure 8: Performance Evaluation of Final XGBoost

```
# Bayesian optimization seeker
bayes_search = BayesSearchCV(
  xgb.XGBRegressor(n_jobs=-1),
  param_space,
  n_iter=32,
  CV=5,
  scoring='neg mean absolute error',
  verbose=1
)
# Get the best model
bayes_search.fit(X_train_full, y_train_full)
best model = bayes search.best estimator
for fold in range(5):
  dataset_train = splited.loc[splited['fold'] != fold]
  dataset_test = splited.loc[splited['fold'] == fold]
  train y = dataset train['weeklySales']
  train_x = dataset_train[important_features]
  test_y = dataset_test['weeklySales']
  test x = dataset test[important features]
  predicted = best_model.predict(test_x)
  mae, mse, rmse, r2 = mean_absolute_error(test_y, predicted),
mean squared error(test y, predicted),
np.sqrt(mean_squared_error(test_y, predicted)), r2_score(test_y,
predicted)
  error cv += mae
    print(f'Fold {fold} MAE: {mae}, MSE: {mse}, RMSE: {rmse}, R<sup>2</sup>: {r<sub>2</sub>}")
# Reorder columns to match the training data
dataset_test_for_prediction = dataset_test[important_features]
# Predict on test data
predicted test = best model.predict(dataset test for prediction)
dataset_test['weeklySales'] = predicted_test
# Save results to output with specific columns
output_df = dataset_test[['Store', 'Dept', 'Date', 'weeklySales', 'IsHoliday',
'Temperature', 'Fuel_Price', 'MarkDown1', 'MarkDown2', 'MarkDown3',
'MarkDown4', 'MarkDown5', 'CPI', 'Unemployment', 'Type', 'Size']]
output_df.to_csv('output.csv', index=False)
```

Figure 7: Code snippet for model prediction

# **5** Conclusion

In conclusion, this project successfully demonstrates the potential of machine learning in sales forecasting, providing a reliable and scalable solution for predicting weekly sales. The combination of advanced machine learning techniques, a user-friendly interface, and a robust backend architecture makes this system a valuable tool for businesses seeking to optimize their sales strategies and improve decision-making processes.

## 5.1 Limitations

While the current system utilizes a pre-trained model for real-time predictions, there are some limitations that impact its performance and adaptability. The model is trained on a dataset provided by Kaggle, which, although comprehensive, is limited in both size and timeliness. The dataset covers sales data from 2010 to 2012, which may not fully capture the dynamics of today's retail environment. Consequently, the model may struggle to account for recent changes in market trends, consumer behavior, or external factors like economic shifts or global events.

#### 5.2 Future Works

The automated feature engineering techniques, such as feature selection algorithms or deep learning-based feature extraction, should be explored to improve the model performance. These approaches would reduce the reliance on manual feature engineering and domain knowledge, enabling the system to automatically identify the most predictive features.

# References

- [1] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785
- [2] Shi, Rui, et al. "Prediction and analysis of train arrival delay based on XGBoost and Bayesian optimization." Applied Soft Computing 109 (2021): 107538.
- [3] Chai T, Draxler R R. Root mean square error (RMSE) or mean absolute error (MAE)[J]. Geoscientific model development discussions, 2014, 7(1): 1525-1534.
- [4] OpenAI. (2024). ChatGPT (GPT-3.5) <u>https://chat.openai.com/</u>

# **Emerging Paradigms in IoT Security: A Comprehensive Survey of Malware, Threats, Vulnerabilities, and Countermeasures**

Samuel Richard Arne Department of CIDS Saint Cloud State University St. Cloud, Minnesota, 56301 sarne@stcloudstate.edu Thivanka Mohottalalage Department of MSIA Saint Cloud State University St. Cloud, Minnesota, 56301 thivanka.mohottalalage@go.stcloudstate.edu Akalanka B. Mailewa Department of CIDS Saint Cloud State University St. Cloud, Minnesota, 56301 amailewa@stcloudstate.edu

## Abstract

The rapid expansion of the Internet of Things (IoT) has introduced numerous devices and applications that are transforming modern life. Simultaneously, this progress has created an evershifting landscape of malware, threats, and vulnerabilities that challenge existing security paradigms. This survey presents an extensive review of up-to-date research on IoT security, focusing on malware, threats, vulnerabilities, and countermeasures. We consolidate and analyze key contributions to highlight current gaps and outline future research directions. Our findings indicate that security must be a continuous, collaborative process that evolves in tandem with emerging technologies such as 5G networks, machine learning, and blockchain. This paper synthesizes the literature on intrusion detection systems, distributed denial of service attacks, privacy protection, Industrial IoT (IIoT) security, and more, providing an invaluable resource for researchers and practitioners aiming to secure the next generation of IoT ecosystems.

**Keywords:** IoT, IIoT, Malware, Threats, Vulnerabilities, Machine Learning, DDoS, Intrusion Detection

## **1. Introduction**

The Internet of Things (IoT) is revolutionizing industries by enabling real-time data processing, intelligent automation, and ubiquitous connectivity [16] - [18]. From smart homes and consumer electronics to critical infrastructures and industrial control systems, IoT devices promise unprecedented benefits in efficiency and convenience [19] - [20]. However, each new feature or technology introduces potential security weaknesses that adversaries can exploit [21] - [23]. Consequently, ongoing research and development seek robust solutions to mitigate these evolving threats [1] - [3].

Despite a significant body of work, IoT security remains highly challenging due to resource constraints, heterogeneous protocols, and the sheer scale of IoT deployments [24] - [25]. Notorious incidents such as the Mirai botnet have illustrated how compromised devices can be leveraged to launch massive Distributed Denial of Service (DDoS) attacks [26] - [27]. Researchers have recognized that comprehensive security frameworks must account not only for hardware/software vulnerabilities but also for issues involving device authentication, network intrusion detection, and policy regulations [28] - [30].

This paper consolidates and critically examines recent IoT security research, with particular emphasis on malware, threats, vulnerabilities, and countermeasures, including machine learning driven solutions. We highlight key findings from academic and industrial literature, identify gaps, and discuss future research directions. The remainder of this paper is organized as follows. Section 2 reviews background and related work, focusing on recent studies. Section 3 discusses extended analyses of key areas, including IoT communication protocols, collaborative security frameworks, and machine learning–based intrusion detection. Section 4 deepens the exploration of advanced topics such as privacy, trust management, edge computing, and blockchain. Section 5 combines multiple critical elements: the major challenges, gaps in current solutions, future directions, extended conclusions, and a comparative analysis table. Finally, all references are provided in ascending numerical order at the end.

## 2. Background

IoT devices are rapidly integrating with 5G networks, industrial systems, and consumer electronics, resulting in an expansive attack surface [1], [16], [17]. This section provides a detailed account of prior works categorized into four primary themes: (A) DDoS threats in 5G-enabled IoT, (B) intrusion detection and machine learning, (C) industrial IoT vulnerabilities, and (D) advanced malware and mitigation measures. These themes illustrate the breadth of current research efforts and how they connect to broader security challenges [25] - [30].

## A. DDoS Threats in 5G-Enabled IoT

Studies by Maiwada *et al.* [1] and Bertino *et al.* [25] report that 5G's high bandwidth and low latency amplify both opportunities and risks for IoT. The Mirai botnet outbreak [27] showcased how weakly secured devices with default passwords can be hijacked for large-scale DDoS attacks. More recent surveys [16], [28] emphasize the importance of multi-stakeholder collaboration among manufacturers, network operators, and regulators to prevent large-scale disruptions. These collective efforts aim to address the gap in device-level security policies, including password standards, secure boot mechanisms, and frequent firmware updates.

## **B. Intrusion Detection and Machine Learning**

Ahmed *et al.* [4] explored honeypot-driven approaches for intrusion detection, while Khang *et al.* [5] provided a comprehensive review of deep learning and ensemble techniques applicable to IoT. Studies [6], [7], [9] have confirmed that machine learning–based intrusion detection systems are effective at identifying zero-day attacks by analyzing network traffic anomalies. Even so, large-scale deployment of ML-based IDS frameworks can present new challenges, such as increased computation, scalability issues, and the need for continuous retraining [34], [37]. Researchers are exploring federated learning [31] and distributed anomaly detection [34] to address data privacy concerns and to manage the scale of IoT networks.

## C. Industrial IoT (IIoT) Vulnerabilities

Smart factories rely on IoT sensors to manage production lines and supply chains efficiently [10], [20]. However, SCADA systems originally designed for isolated networks [11] now face an

expanded threat surface, encompassing IP-based connections and cloud integration [24], [42]. Vulnerabilities within IIoT environments can lead to catastrophic consequences, especially when critical infrastructures (e.g., power grids, transportation systems) are targeted [2], [30], [40]. To counter these threats, some frameworks [33], [39] incorporate robust encryption, multi-level authentication, and stringent access controls, although consistent adoption remains a major hurdle. Researchers are also examining real-time intrusion detection for cyber-physical systems [40], [42], as downtime or sabotage in these environments can lead to severe operational and economic repercussions.

## **D.** Advanced Malware and Mitigation

Recent discussions center on AI-enhanced malware that leverages machine learning to evade signature-based detection [13], [21]. Proposed methods include image-based analysis using convolutional neural networks [13], resource-aware distributed detection strategies [15], and multi-feature cloud-based approaches [8]. Other frameworks [22], [25] champion blockchain and zero-trust architectures as means to improve integrity and authentication. Authors in [20], [50] underscore that while these emerging solutions hold promise, they also introduce unique challenges, such as latency in blockchain transactions, the high computational overhead of some zero-trust frameworks, and the complexity of integrating these solutions into legacy IoT devices.

## **3. Extended Analysis of Key Areas**

This section extends the core discussion by delving into IoT communication protocols, collaborative security frameworks, and a comparative analysis of machine learning based intrusion detection. Each topic underscores different dimensions of IoT security and their respective solutions, bridging gaps identified in the background section [25] - [30].

## A. IoT Communication Protocols and Their Vulnerabilities

Gerodimos *et al.* [12] describe vulnerabilities in commonly used protocols such as MQTT, CoAP, and HTTP. These protocols, designed for lightweight operation, can be susceptible to eavesdropping or injection attacks [21]. Lin *et al.* [8] propose encryption and authentication enhancements yet highlight the need for resource-efficient mechanisms tailored to constrained IoT devices. Additional efforts to standardize secure protocols across vendors [17], [49] are underway but remain fragmented, particularly in large-scale IoT ecosystems.

#### **B.** Collaborative Security Frameworks

Sun *et al.* [2] and Roman *et al.* [17] argue that any successful IoT security strategy must involve cooperation among device manufacturers, network service providers, and policymakers. Frameworks such as zero-trust networking, secure firmware updates, and hierarchical authentication [29], [44], [49] can standardize best practices. However, universal adoption is challenging due to varying national regulations, differing device architectures, and the financial burden of implementing consistent security measures [33], [39]. Initiatives for more transparent vulnerability disclosures and global IoT security guidelines [28], [40] are also gaining momentum.

### C. Comparative Analysis of Machine Learning–Based IDS

Multiple works [7], [9], [14], [34] examine the use of neural networks (CNN, LSTM, BiLSTM) and traditional ML algorithms (Decision Trees, SVM, KNN) for IoT intrusion detection. Studies [26], [28], [38] note that anomaly based systems excel at detecting unknown or zero day threats but can suffer from higher false positive rates. The need for robust feature engineering, combined with domain expertise, is critical for balanced detection performance [5], [37]. Table I (within the merged final section) compares recent IDS frameworks, revealing strengths, weaknesses, and potential deployment challenges.

## 4. Advanced Topics in IoT Security

Although preceding sections offer substantial insights, IoT security research has additional advanced topics, each addressing distinct facets of this rapidly evolving field [25] - [30]. From edge/fog computing architectures to blockchain-based security enhancements, these directions represent the cutting edge of IoT research and development.

## A. Privacy and Trust Management

Privacy protection remains a key pillar of IoT security, given the massive volume of sensitive data generated by consumer and industrial devices [18], [22]. Yan *et al.* [22] proposes a trust management model that uses continuous monitoring of device behavior to evaluate trustworthiness. Ziegeldorf *et al.* [18] add that privacy must be maintained at all lifecycle stages, from data generation to storage and sharing. Some frameworks [24], [37] incorporate distributed consensus mechanisms to ensure data legitimacy, although performance overhead is often a concern.

#### **B. Edge Computing and Resource Constraints**

Emerging research [15], [31], [33] explores edge-based solutions to reduce latency and mitigate bottlenecks on centralized servers, which is crucial for mission-critical applications like real-time healthcare monitoring or autonomous vehicles [19], [46]. Nonetheless, the distribution of intelligence to network edges introduces new security challenges, including multi-tenant isolation, key management, and the risk of compromised edge nodes [32], [35]. Many studies stress the importance of secure device registration and authentication protocols at the edge [25], [49] to protect the local network from infiltration.

#### **C. Blockchain-Integrated Solutions**

Sharma *et al.* [20] discuss integrating blockchain to maintain tamper-proof records of IoT transactions, thereby enhancing transparency and trust among devices. Mosenia and Jha [39] similarly emphasize the potential of smart contracts for automated policy enforcement, enabling devices to self-regulate based on predefined security rules. However, blockchain technology itself faces scalability and throughput constraints [49], [50], often requiring high computational resources and incurring latency not always acceptable for time-sensitive IoT operations [44]. Despite these challenges, the synergy between blockchain, IoT, and decentralized trust continues to attract research attention due to its potential to mitigate single points of failure [28], [32].

## 5. Challenges, Gaps, and Future Directions: A Comparative Analysis

In this section, we address the main limitations in current IoT security approaches, propose future research directions, present extended conclusions, and provide a comparative analysis table. The content reflects ongoing scholarly debates and converging recommendations for safeguarding IoT ecosystems in a rapidly evolving threat landscape [1] - [50].

### A. Challenges and Gaps in Current Solutions

Despite the breadth of existing research, several significant challenges persist in IoT security:

- 1. **Resource Limitations**: Many IoT devices lack the computational power or memory to implement complex security protocols [21], [39]. As a result, simplified algorithms may fail to provide adequate protection against advanced attacks [13], [20].
- 2. **Fragmented Standards and Regulations**: Inconsistent security standards across different manufacturers and jurisdictions hinder interoperability and hamper large-scale adoption of unified frameworks [12], [45]. This fragmentation also increases the difficulty of enforcing consistent best practices and establishing baseline security measures [17], [40].
- 3. **Scalability of IDS and ML Models**: Systems developed for small-scale testbeds may not scale effectively to monitor millions of IoT devices in real deployments [9], [14]. Additionally, retraining ML-based IDS in real time can be computationally intensive [5], [7], especially when dealing with unbalanced datasets [26], [43].
- 4. **AI Bias and Data Quality**: Machine learning algorithms are only as robust as their training data [22], [28]. Anomalies or insufficient representation of attack vectors can lead to high false positive or false negative rates [5], [9]. Adversarial machine learning [13], [21] further complicates model reliability.
- 5. Legacy Infrastructure Integration: SCADA systems and other legacy architectures, which are integral in power grids, oil pipelines, and transportation, often cannot seamlessly adopt modern security patches [11], [24]. This exposes critical sectors to undue risks, particularly if outdated protocols remain in circulation [2], [30].

#### **B.** Future Research Directions

- 1. **AI-Driven Adaptive Security**: While deep learning techniques have proven effective, more adaptive AI solutions are required to handle evolving zero day exploits [29], [37]. Models must continuously learn from new data, ideally in a distributed or federated manner [31], [34].
- 2. Lightweight Cryptographic Schemes: Energy-efficient, low-latency cryptographic mechanisms are needed for constrained IoT devices [15], [25]. Areas such as elliptic-curve cryptography and post-quantum algorithms [41] warrant deeper investigation, along with hardware acceleration strategies that minimize computational overhead [49].
- 3. Holistic Governance and Standardization: Clear regulations and global security guidelines, potentially backed by governmental bodies, are essential for uniform adoption of protective measures [2], [44]. An international consensus on vulnerability disclosure and device manufacturing standards could significantly reduce baseline risks [12], [40].
- 4. Cross Layer Collaboration: Integrating data from sensors, network intelligence, and cloud analytics can yield more accurate, context-aware intrusion detection [37], [43].

Future systems may unite software-defined networking principles [17], [37] and advanced security orchestration to create dynamic, self-defending architectures [24], [32].

5. Quantum Resistant Architectures: As quantum computing becomes more accessible, quantum-resistant encryption for IoT must be investigated to future-proof devices and networks [25], [41]. Preparations for a post quantum era are vital, given the widespread risk of retroactive data decryption if current encryption standards are compromised [35], [50].

### **C. Extended Conclusions**

The IoT ecosystem continues to expand, intersecting with 5G, industrial automation, and smart city initiatives [10], [16]. Our survey reveals a delicate balance between innovation and security. The literature overwhelmingly supports the notion that collaborative, adaptive security frameworks are crucial to tackling threats ranging from DDoS attacks to advanced malware. Machine learning approaches play a pivotal role in intrusion detection and threat intelligence, but they require robust, high-quality data and resilience against adversarial manipulations [5], [9], [26]. Future efforts must focus on scalable, real-time, and layered security solutions that address both current vulnerabilities and emerging concerns like quantum computing [20], [25], [41]. Collaboration between device manufacturers, regulators, and research institutions remains a foundational pillar for safeguarding IoT's transformative potential [2], [17].

#### **D.** Comparative Analysis Table

To present a succinct overview, Table 1 compares selected IoT Intrusion Detection Systems (IDS) and highlights performance metrics, advantages, and limitations. Each system aims to detect threats effectively but varies in computational overhead, detection accuracy, and operational complexity [7], [9], [14], [34]. These comparisons underscore the importance of tailoring solutions to specific deployment environments, resource constraints, and threat models [13], [24].

Study	IDS Technique	Dataset	Reported Accuracy	Key Strength	Limitations
Ahmed et al. [4]	Honeypot-based hybrid IDS using machine learning	Real-world honeypot data	95–98%	Tailored for smart city IoT environments; strong real-time detection	Requires complex honeypot configuration; challenges in generalization across diverse IoT deployments
Khoei & Kaabouch [7]	Comparative study of supervised vs. unsupervised models	CICDDOS 2019 dataset	90–99%	Provides a broad evaluation of ML methods in IDS	Unsupervised models may suffer from higher false positive rates
Zeghida et al. [9]	Ensemble Learning approach for securing MQTT traffic	MQTT protocol dataset	~95%	Robust ensemble method improves detection stability and accuracy	Risk of overfitting in large ensembles; potential computational overhead during deployment
Esmaeili et al. [14]	Deep learning models (BiLSTM, KNN, SVM) for DDoS detection	NSL-KDD dataset	100% (training); lower on testing	Excels in multiclass detection under controlled conditions	Training results may not generalize well; dataset limitations affect real-world applicability
Lin et al. [8]	Extreme Learning Machine (ELM) integrated with cloud computing and multi- feature extraction	Custom IoT test dataset	~96%	Lightweight and scalable solution suited for cloud-based environments	Limited feature extraction may hinder detection of novel attack patterns

Tabla 1	Comporativa	Analycia	of Salaatad	IoT Inter	ician Dataa	tion Systems
I able I.		Allarysis	UI Selecteu	IUI IIIII	ISIOII DELEC	uon systems
	1	•				•

This table encapsulates the varying focuses, performances, and limitations of some recently proposed IDS solutions for IoT. While high accuracy is attainable under controlled experiments, real-world deployment must tackle challenges like evolving attack vectors, data imbalance, and the dynamic nature of IoT environments [5], [28], [43].

## 6. Conclusion

As the Internet of Things (IoT) continues to expand across industries and infrastructures, the need for robust and adaptive security frameworks becomes increasingly urgent. This paper has presented a comprehensive survey of the evolving threat landscape in IoT, examining malware, vulnerabilities, and intrusion vectors, along with contemporary countermeasures including machine learning-based intrusion detection, blockchain integration, and privacy preserving architectures. Through the analysis of emerging technologies such as 5G, edge computing, and quantum-resilient encryption, it is evident that traditional security models are insufficient for modern IoT ecosystems. Our review underscores that securing IoT requires not only technical innovation but also cross disciplinary collaboration among device manufacturers, researchers, policymakers, and end-users. Challenges such as scalability, resource constraints, fragmented standards, and the integration of legacy systems persist as major hurdles. Nonetheless, ongoing developments in federated learning, lightweight cryptography, and decentralized trust mechanisms show promising potential to address these gaps. Ultimately, achieving resilient and secure IoT systems demands a proactive, layered, and adaptable approach one that evolves in tandem with the technologies it aims to protect. Future research should focus on scalable, context-aware solutions that balance security, performance, and usability across diverse and distributed environments. Only through such holistic and forward looking efforts can the full promise of IoT be realized safely and sustainably.

## References

[1] Maiwada, Umar Danjuma, Shahbaz Ali Imran, Kamaluddeen Usman Danyaro, Aftab Alam Janisar, Anas Salameh, and Aliza Bt Sarlan. "Security Concerns of IoT Against DDoS in 5G Systems." International Journal of Electrical Engineering and Computer Science 6 (2024): 98-105.
[2] Sun, Panjun, Shigen Shen, Yi Wan, Zongda Wu, Zhaoxi Fang, and Xiao-zhi Gao. "A survey of iot privacy security: Architecture, technology, challenges, and trends." IEEE Internet of Things Journal (2024).

[3] Mingo, Horace C., Misty Nicole Lawson, and Amber C. Williamson. "Identifying new vulnerabilities embedded in consumer internet of things (IoT) devices." In Multisector Insights in Healthcare, Social Sciences, Society, and Technology, pp. 186-207. IGI Global, 2024.

[4] Ahmed, Yussuf, Kehinde Beyioku, and Mehdi Yousefi. "Securing smart cities through machine learning: A honeypot-driven approach to attack detection in Internet of Things ecosystems." IET Smart Cities 6, no. 3 (2024): 180-198.

[5] Khang, Alex, Yudhvir Singh, and Dheerdhwaj Barak. "Review of the Literature on Using Machine and Deep Learning Techniques to Improve IoT Security." Revolutionizing Automated Waste Treatment Systems: IoT and Bioelectronics (2024): 273-300.

[6] Alnajim, Abdullah M., Shabana Habib, Muhammad Islam, Su Myat Thwin, and Faisal Alotaibi. "A comprehensive survey of cybersecurity threats, attacks, and effective countermeasures in industrial internet of things." Technologies 11, no. 6 (2023): 161.

[7] Talaei Khoei, Tala, and Naima Kaabouch. "A comparative analysis of supervised and unsupervised models for detecting attacks on the intrusion detection systems." Information 14, no. 2 (2023): 103.

[8] Lin, Haifeng, Qilin Xue, Jiayin Feng, and Di Bai. "Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine." Digital Communications and Networks 9, no. 1 (2023): 111-124.

[9] Zeghida, Hayette, Mehdi Boulaiche, and Ramdane Chikh. "Securing MQTT protocol for IoT environment using IDS based on ensemble learning." International Journal of Information Security 22, no. 4 (2023): 1075-1086.

[10] Soori, Mohsen, Behrooz Arezoo, and Roza Dastres. "Internet of things for smart factories in industry 4.0, a review." Internet of Things and Cyber-Physical Systems 3 (2023): 192-204.

[11] Alanazi, Manar, Abdun Mahmood, and Mohammad Jabed Morshed Chowdhury. "SCADA vulnerabilities and attacks: A review of the state-of-the-art and open issues." Computers & security 125 (2023): 103028.

[12] Gerodimos, Apostolos, Leandros Maglaras, Mohamed Amine Ferrag, Nick Ayres, and Ioanna Kantzavelou. "IoT: Communication protocols and security threats." Internet of Things and Cyber-Physical Systems 3 (2023): 1-13.

[13] Alnajim, Abdullah M., Shabana Habib, Muhammad Islam, Rana Albelaihi, and Abdulatif Alabdulatif. "Mitigating the risks of malware attacks with deep Learning techniques." Electronics 12, no. 14 (2023): 3166.

[14] Esmaeili, Mona, Seyedamiryousef Hosseini Goki, Behnam Hajipour Khire Masjidi, Mahdi Sameh, Hamid Gharagozlou, and Amin Salih Mohammed. "ML-DDoSnet: IoT intrusion detection based on denial-of-service attacks using machine learning methods and NSL-KDD." Wireless Communications and Mobile Computing 2022, no. 1 (2022): 8481452.

[15] Kasarapu, Sreenitha, Sanket Shukla, and Sai Manoj Pudukotai Dinakarrao. "Enhancing IoT malware detection through adaptive model parallelism and resource optimization." arXiv preprint arXiv:2404.08808 (2024).

[16] Sicari, Sabrina, Alessandra Rizzardi, Luigi Alfredo Grieco, and Alberto Coen-Porisini. "Security, privacy and trust in Internet of Things: The road ahead." Computer networks 76 (2015): 146-164.

[17] Roman, Rodrigo, Pablo Najera, and Javier Lopez. "Securing the internet of things." Computer 44, no. 9 (2011): 51-58.

[18] Ziegeldorf, Jan Henrik, Oscar Garcia Morchon, and Klaus Wehrle. "Privacy in the Internet of Things: threats and challenges." Security and Communication Networks 7, no. 12 (2014): 2728-2742.

[19] Chen, Shanzhi, Hui Xu, Dake Liu, Bo Hu, and Hucheng Wang. "A vision of IoT: Applications, challenges, and opportunities with china perspective." IEEE Internet of Things journal 1, no. 4 (2014): 349-359.

[20] Shahzad, Yasir, Huma Javed, Haleem Farman, Jamil Ahmad, Bilal Jan, and Muhammad Zubair. "Internet of energy: Opportunities, applications, architectures and challenges in smart industries." Computers & Electrical Engineering 86 (2020): 106739.

[21] Zhao, Kai, and Lina Ge. "A survey on the internet of things security." In 2013 Ninth international conference on computational intelligence and security, pp. 663-667. IEEE, 2013.

[22] Yan, Zheng, Peng Zhang, and Athanasios V. Vasilakos. "A survey on trust management for Internet of Things." Journal of network and computer applications 42 (2014): 120-134.

[23] Ammar, Mahmoud, Giovanni Russello, and Bruno Crispo. "Internet of Things: A survey on the security of IoT frameworks." Journal of information security and Applications 38 (2018): 8-27.

[24] Kirilo, Caique Z., Jair M. Abe, Marcelo Nogueira, Kazumi Nakamatsu, Luiz Carlos Machi Lozano, and Luiz A. de Lima. "Evaluation Of Adherence To The Model Six Sigma Using Paraconsistent Logic." In 2018 Innovations in Intelligent Systems and Applications (INISTA), pp. 1-7. IEEE, 2018.

[25] Bertino, Elisa, and Nayeem Islam. "Botnets and internet of things security." Computer 50, no. 2 (2017): 76-79.

[26] Maniriho, Pascal, Ephrem Niyigaba, Zephanie Bizimana, Valens Twiringiyimana, Leki Jovial Mahoro, and Tohari Ahmad. "Anomaly-based intrusion detection approach for IoT networks using machine learning." In 2020 international conference on computer engineering, network, and intelligent multimedia (CENIM), pp. 303-308. IEEE, 2020.

[27] Kolias, Constantinos, Georgios Kambourakis, Angelos Stavrou, and Jeffrey Voas. "DDoS in the IoT: Mirai and other botnets." Computer 50, no. 7 (2017): 80-84.

[28] Neshenko, Nataliia, Elias Bou-Harb, Jorge Crichigno, Georges Kaddoum, and Nasir Ghani. "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations." IEEE Communications Surveys & Tutorials 21, no. 3 (2019): 2702-2733.

[29] Jakaria, A. H. M., and Mohammad Ashiqur Rahman. "Safety analysis for UAV networks." In 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), pp. 294-295. IEEE, 2018.

[30] Deep, Samundra, Xi Zheng, Alireza Jolfaei, Dongjin Yu, Pouya Ostovari, and Ali Kashif Bashir. "A survey of security and privacy issues in the Internet of Things from the layered context." Transactions on Emerging Telecommunications Technologies 33, no. 6 (2022): e3935.

[31] Ejaz, Waleed, and Mohamed Ibnkahla. "Multiband spectrum sensing and resource allocation for IoT in cognitive 5G networks." IEEE Internet of Things Journal 5, no. 1 (2017): 150-163.

[32] Qiu, Tie, Baochao Chen, Arun Kumar Sangaiah, Jianhua Ma, and Runhe Huang. "A survey of mobile social networks: Applications, social characteristics, and challenges." IEEE Systems Journal 12, no. 4 (2017): 3932-3947.

[33] Chen, Chen, Yan Jiang, Jiliang Zhang, Xiaoli Chu, and Jie Zhang. "Parameter Optimization for Energy Efficient Indoor Massive MIMO Small Cell Networks." In 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), pp. 1-5. IEEE, 2020.

[34] Diro, Abebe Abeshu, and Naveen Chilamkurti. "Distributed attack detection scheme using deep learning approach for Internet of Things." Future Generation Computer Systems 82 (2018): 761-768.

[35] Saxena, Deepika, and Ashutosh Kumar Singh. "Workload pattern learning-based cloud resource management models: Concepts and meta-analysis." IEEE Transactions on Sustainable Computing (2024).

[36] Siow, Eugene, Thanassis Tiropanis, and Wendy Hall. "Analytics for the internet of things: A survey." ACM computing surveys (CSUR) 51, no. 4 (2018): 1-36.

[37] Restuccia, Francesco, Salvatore D'oro, and Tommaso Melodia. "Securing the internet of things in the age of machine learning and software-defined networking." IEEE Internet of Things Journal 5, no. 6 (2018): 4829-4842.

[38] Osanaiye, Opeyemi, Kim-Kwang Raymond Choo, and Mqhele Dlodlo. "Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework." Journal of Network and Computer Applications 67 (2016): 147-165.

[39] Mosenia, Arsalan, and Niraj K. Jha. "A comprehensive study of security of internet-of-things." IEEE Transactions on emerging topics in computing 5, no. 4 (2016): 586-602.

[40] Xu, Hansong, Wei Yu, David Griffith, and Nada Golmie. "A survey on industrial Internet of Things: A cyber-physical systems perspective." Ieee access 6 (2018): 78238-78259.

[41] Conti, Mauro, Ali Dehghantanha, Katrin Franke, and Steve Watson. "Internet of Things security and forensics: Challenges and opportunities." Future Generation Computer Systems 78 (2018): 544-546.

[42] Da Xu, Li, Wu He, and Shancang Li. "Internet of things in industries: A survey." IEEE Transactions on industrial informatics 10, no. 4 (2014): 2233-2243.

[43] Li, Wenzhuo, Chuang Lin, Puheng Zhang, and Mao Miao. "Probe sharing: A simple technique to improve on sparrow." In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 863-870. IEEE, 2017.

[44] Gannon, Dennis, Roger Barga, and Neel Sundaresan. "Cloud-native applications." *IEEE Cloud Computing* 4, no. 5 (2017): 16-21.

[45] Noura, Mahda, Mohammed Atiquzzaman, and Martin Gaedke. "Interoperability in internet of things: Taxonomies and open challenges." Mobile networks and applications 24 (2019): 796-809.
[46] Filice, Simone, Danilo Croce, Roberto Basili, and Fabio Massimo Zanzotto. "Linear online learning over structured data with distributed tree kernels." In *2013 12th International Conference on Machine Learning and Applications*, vol. 1, pp. 123-128. IEEE, 2013.

[47] Valem, Lucas Pascotti, Carlos Renan De Oliveira, Daniel Carlos Guimarães Pedronette, and Jurandy Almeida. "Unsupervised similarity learning through rank correlation and knn sets." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 14, no. 4 (2018): 1-23.

[48] Véron, Maxime, Olivier Marin, Sébastien Monnet, and Pierre Sens. "Repfd-using reputation systems to detect failures in large dynamic networks." In 2015 44th International Conference on Parallel Processing, pp. 91-100. IEEE, 2015.

[49] Lin, Jie, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications." IEEE internet of things journal 4, no. 5 (2017): 1125-1142.

[50] Cui, Qimei, Zengbao Zhu, Wei Ni, Xiaofeng Tao, and Ping Zhang. "Edge-intelligenceempowered, unified authentication and trust evaluation for heterogeneous beyond 5G systems." IEEE Wireless Communications 28, no. 2 (2021): 78-85.

# Psyber-Security Revisited: A Comprehensive Literature Review on Integrating Psychological Principles into Cybersecurity

Thivanka Mohottalalage Department of MSIA Saint Cloud State University St. Cloud, Minnesota, 56301 thivanka.mohottalalage@go.stcloudstate.edu Anushka Hewarathne Department of MSIA Saint Cloud State University St. Cloud, Minnesota, 56301 auhewarathna@go.stcloudstate.edu Akalanka B. Mailewa Department of CIDS Saint Cloud State University St. Cloud, Minnesota, 56301 amailewa@stcloudstate.edu

#### Abstract

Human factors remain a persistent vulnerability in modern cybersecurity, where technical controls often fail if users are susceptible to social engineering, cognitive biases, and motivational barriers. This literature review examines how psychological insights, and behavioral interventions can be integrated into cybersecurity, a field sometimes known as psyber security. The paper opens by outlining the underlying problem of misalignment between human behavior and security requirements, followed by three research questions on how to (1) enhance user awareness, (2) mitigate biases and social engineering, and (3) build a sustainable security culture. A step by step methodology is employed to identify, screen, analyze, and synthesize relevant studies. The expanded literature review discusses research findings on cognitive biases, persuasion tactics, user motivation, organizational culture, and user-centric interface design. Multiple tables summarize key studies, capturing recurring themes and evidence-based strategies. The review concludes with a discussion of gaps, opportunities, and the ethical considerations that future psyber security research must address to ensure ethical, adaptive, and user-aligned defenses.

**Keywords:** Psyber Security, Human Factors, Social Engineering, Cognitive Biases, Behavioral Interventions, Security Culture, User-Centric Design, Organizational Psychology

#### 1. Introduction

Cybersecurity threats are evolving in both sophistication and scale, impacting diverse sectors from finance and healthcare to government operations [1] - [3]. While substantial advances in technical safeguards such as firewalls, encryption protocols, and intrusion detection systems have mitigated numerous attacks, breaches continue at a high rate. Many incidents trace back to user-driven errors or social engineering manipulations rather than purely technical vulnerabilities [4], [5]. This stark reality underscores the essential role of human factors in cybersecurity, prompting a growing interest in psyber security, which systematically integrates psychological and behavioral principles into cybersecurity practices.

## **1.1 Problem Statement**

Despite the continuous refinement of technical cybersecurity tools, organizations worldwide face repeated breaches that hinge on human oversights including weak passwords, unpatched systems, mishandling of suspicious emails, and other forms of non-compliance [6], [7]. Attackers exploit cognitive biases and emotional triggers via well crafted social engineering schemes [8], resulting in unauthorized access and compromised data. Thus, the enduring gap between technological capabilities and user compliance raises a central challenge:

How can psychological principles be effectively integrated into cybersecurity strategies to address human vulnerabilities, reduce the impact of cognitive biases, and foster sustainable secure behaviors?

Existing research recognizes the pivotal role of human behavior in cybersecurity; however, a robust, empirically validated framework that unifies psychological insights across individual, cultural, and organizational levels remains relatively undeveloped. This review aims to bridge that gap by surveying relevant theories, best practices, and empirical findings.

## **1.2 Research Questions**

Guided by the above problem statement, this paper is organized around three research questions:

1. **RQ1:** How can user awareness and compliance be enhanced through psychological principles and behavioral interventions?

This question explores how theoretical models (e.g., Protection Motivation Theory, Self Determination Theory) and specific tactics (e.g., nudges, gamification) improve adherence to cybersecurity protocols.

- 2. **RQ2: Which cognitive biases and social engineering tactics pose the greatest threats, and what evidence-based methods can mitigate these vulnerabilities?** This question investigates psychological underpinnings of social engineering attacks e.g., persuasion strategies, trust exploitation and examines empirically validated countermeasures (e.g., scenario-based training, real-time prompts).
- 3. RQ3: In what ways do organizational culture and leadership influence secure behaviors, and how can these factors be optimized to promote a sustainable security climate?

This question looks at group level and cultural influences, including leadership support, peer norms, and knowledge-sharing frameworks that collectively shape how security policies are internalized and enacted.

By integrating these research questions into a structured review, the paper seeks to uncover the state of the art in psyber security, highlight successful interventions, and reveal ongoing gaps that warrant deeper investigation. The subsequent sections detail the methodology employed and present a comprehensive analysis of the relevant literature.

## 2. Methodology

A transparent and rigorous methodology is critical to generating a credible and comprehensive literature review. This section describes a step by step approach used to identify, screen, and synthesize relevant studies in cybersecurity, psychology, human computer interaction, and related fields.

## 2.1 Step 1: Scope Definition and Keywords

The research began by defining the scope to include peer-reviewed journal articles, conference proceedings, and reputable white papers addressing human aspects of cybersecurity. Primary databases IEEE Xplore, ACM Digital Library, Scopus, and Web of Science were searched using keywords such as "cybersecurity," "human factors," "cognitive biases," "social engineering," "information security awareness," and "psyber security" [9].

## 2.2 Step 2: Screening and Eligibility

Abstracts were reviewed to **exclude** articles that did not focus on user or psychological dimensions of cybersecurity. Studies emphasizing purely technical solutions without user centric or behavioral components were omitted to retain behavioral and organizational emphasis.

## 2.3 Step 3: Coding and Thematic Analysis

All eligible articles were imported into a reference management tool. A coding template captured the following attributes:

- Publication details (author, year, venue)
- Topic focus (cognitive biases, social engineering, organizational culture, etc.)
- Theoretical framework (e.g., PMT, SDT, nudge theory)
- Methodological approach (e.g., quantitative, qualitative, mixed)
- Key findings and recommendations

Common patterns, contradictions, and notable gaps were identified through iterative comparison, which facilitated the formation of thematic categories aligned with the research questions.

#### 2.4 Step 4: Synthesis and Quality Assessment

In the final step, the coded data were synthesized to map convergent and divergent findings. Studies with robust methodological designs, larger sample sizes, and strong theoretical grounding received priority. Where conflicting results arose, possible reasons (e.g., cultural context, participant variability, different threat models) were scrutinized.

This four step methodology allowed for a systematic examination of the interplay between psychology and cybersecurity. The resulting thematic framework underpins the Literature Review in Section 3, which is further reinforced by tabular summaries of the analyzed works.

### 3. Literature Review

This section provides a comprehensive overview of key studies examining human factors in cybersecurity. It is divided into five subsections, reflecting major thematic areas cognitive biases, social engineering, motivation and behavior, organizational culture, and user centric design. Each subsection opens with a brief orientation, followed by in-depth discussions of relevant research. Tables are included to summarize and compare findings from multiple studies, offering readers a structured view of recurring themes and evidence-based strategies.

#### 3.1 Cognitive Biases and Decision-Making in Cybersecurity

Research on cognitive biases reveals that many securities lapses stem from overconfidence, anchoring, or confirmation bias [10], [11]. Overconfidence can cause individuals to dismiss security warnings, believing themselves too savvy to be deceived, while anchoring leads users to trust superficial markers of legitimacy (e.g., brand logos or disclaimers) [12]. Under time pressure, biases such as urgency and availability can further exacerbate errors in judgment, allowing attackers to exploit emotional triggers [13]. Table 1 provides illustrative examples of how cognitive biases manifest in cybersecurity contexts and references studies that have attempted to quantify their impact.

Study (Authors, Year)	Focus	Methodology	Key Findings
Ceric & Holland (2019) [10]	Impact of cognitive biases on threat anticipation and response	Empirical survey & literature review	Identifies that overconfidence and anchoring significantly impair threat detection and risk evaluation in cybersecurity contexts.
Hadnagy (2010) [13]	Exploitation of human vulnerabilities through social engineering	Case studies and qualitative analysis	Details how attackers manipulate trust and emotional triggers to bypass technical defenses, underscoring the need for human-centric countermeasures.
Rogers (1975) [16]	Protection Motivation Theory (PMT) as a predictor of secure behavior	Theoretical model development	Demonstrates that higher perceived threat and self-efficacy lead to greater adoption of protective behaviors, forming a foundation for behavioral cybersecurity interventions.
Ryan & Deci (2000) [17]	Intrinsic and extrinsic motivational factors in behavioral compliance	Theoretical framework review	Reveals that users' need for autonomy and competence is critical to sustaining secure practices over time.
Singh et al. (2023) [35]	Cognitive factors in anti- phishing training effectiveness	Experimental study	Shows that targeted training interventions can improve users' ability to discriminate between legitimate and phishing attempts, reducing susceptibility to attacks.

#### Table 1. Summary of Key Studies on Cognitive Biases in Cybersecurity

Collectively, these studies underscore the pivotal influence of cognitive biases in cybersecurity outcomes, demonstrating the need for interventions designed to mitigate their effects. While training can reduce bias to some extent, fatigue sets in if interventions remain static. This highlights a gap in adaptative, or personalized training strategies aligned with RQ2.

## 3.2 Social Engineering Tactics and Mitigation Strategies

Social engineering attacks exploit emotional levers such as trust, fear, or authority, often bypassing technical safeguards [18], [19]. Attackers apply Cialdini's principles of influence authority, reciprocity, social proof, scarcity, and consistency to elicit user compliance with malicious requests [20]. Table 2 lists some influential works on social engineering tactics and related countermeasures.

Study (Authors, Year)	Approach	Key Insight	Suggested Countermeasure
Mitnick & Simon (2003) [4]	Narrative case studies of real-world social engineering attacks	Demonstrates how human manipulation (e.g., exploiting trust and authority) can bypass technical controls	Implement comprehensive training programs with simulated attacks
Hadnagy (2010) [13]	Qualitative analysis of social engineering techniques	Highlights that attackers exploit emotional triggers and cognitive shortcuts to induce security lapses	Use scenario-based simulations and periodic refresher training
Hong (2012) [11]	Review of phishing trends and behavioral vulnerabilities	Shows that phishing leverages urgency and scarcity biases to compel risky user behavior	Deploy adaptive, context-aware warning systems and multi- factor authentication
Dimkov et al. (2010) [15]	Empirical pen-testing using social engineering methods	Finds that direct human interactions can easily override technical safeguards when verification is lax	Enforce strict verification protocols and adopt robust multi-factor authentication

Table 2. Representative Studies on Social Engineering Tactics

This body of research suggests that repetitive or poorly contextualized training can result in alert fatigue, diminishing the protective benefits of user education. Tailored approaches encompassing emotional intelligence, scenario-based simulations, and dynamic messaging show promise in sustaining engagement and building user resistance to evolving social engineering ploys.

## 3.3 Motivation, Behavioral Interventions, and User Awareness

Moving beyond social engineering, this subsection addresses the intrinsic and extrinsic motivations driving user compliance. Protection Motivation Theory (PMT) posits that individuals evaluate threat severity, vulnerability, response efficacy, and self-efficacy before adopting protective behaviors [25]. Meanwhile, Self-Determination Theory (SDT) centers on the need for autonomy, competence, and relatedness [26]. Combining these frameworks can clarify why some users adhere strictly to best practices while others remain indifferent. Table 3 summarizes seminal work and recent advancements in leveraging motivation for cybersecurity.

Study (Author, Year)	Theoretical Basis	Findings	Implication
Rogers (1975) [16]	РМТ	Perceived threat & self-efficacy $\rightarrow$ stronger protective acts	Highlight efficacy in user education
Ryan & Deci (2000) [17]	SDT	Autonomy, competence, relatedness drive sustained behavior	Allow user control & feedback in security tasks
Siponen & Vance (2010) [18]	Neutralization/PMT	Users rationalize policy non-compliance unless perceived cost is high	Emphasize tangible consequences
Renaud & Zimmerman (2017) [19]	Nudging theory	Nudges boost password strength but can backfire under certain conditions	Design carefully to avoid "nudge fatigue"

Table 3. Key Studies on Motivation, Behavioral Interventions, and User Awareness in Cybersecurity

These studies underline that motivational factors are integral to forming lasting secure habits. Harnessing theoretical insights from PMT, SDT, and nudge theory can refine user training and policy design, ensuring that cybersecurity requirements resonate with genuine user needs and perceptions. This outcome directly addresses RQ1 by delineating how behavioral interventions might sustain or erode user compliance.

#### 3.4 Organizational Culture, Leadership, and Security Climate

Effective cybersecurity is not just an individual pursuit; it thrives in an environment shaped by leadership commitment and organizational norms [31]. Leadership support manifested through explicit endorsement, allocated resources, and frequent communication often correlates with higher policy adherence [32]. Table 4 lists some key studies emphasizing how culture and leadership transform the security climate within organizations.

Table 4. Research on Organizational Sec	curity Culture and Leadership
---	-------------------------------

Study (Authors, Year)	Context	Key Observations	Outcome/Recommendation
Solomon & Brown (2021) [21]	Enterprise security culture	Found that strong leadership commitment and clear communication foster higher employee compliance.	Advocate for top–down modeling of secure behaviors and regular briefings.
Crossler et al. (2014) [22]	BYOD policies and organizational compliance	Demonstrated that well-defined policies, when endorsed by leadership, significantly enhance compliance rates.	Align security guidelines with organizational norms and enforce policies.
Pfleeger & Caputo (2012) [23]	Behavioral science in cybersecurity risk management	Revealed that peer norms and group identity heavily influence employees' adherence to security practices.	Promote team-based accountability and group training to strengthen culture.
Parsons et al. (2010) [24]	Human factors and security environment	Noted variability in how security culture is defined, indicating that inconsistent practices undermine efforts.	Develop standardized frameworks to integrate technical controls with human factors.

These findings highlight that building a sustainable security climate depends on top-down advocacy, peer-based reinforcement, and consistent resource allocation. Leadership can catalyze a shift from mere compliance to shared responsibility, aligning with RQ3 by illustrating how culture and norms shape security attitudes at scale.

## 3.5 User-Centric Security Design and Usability

Finally, a significant segment of literature in human–computer interaction (HCI) underscores that usability is a cornerstone of cybersecurity compliance [37]. If multi-factor authentication or password policies are excessively complex, users often circumvent them, creating fresh vulnerabilities [38]. Conversely, user-centered designs minimize friction, automating secure tasks wherever feasible [39]. The studies in Table 5 detail approaches to incorporate usability into security features.

Study (Authors, Year)	Design Focus	Key Insights	Security Benefit
Sasse & Flechais (2017) [40]	Usable Security	Emphasizes designing systems that accommodate human limitations rather than idealized users.	Reduces errors and increases acceptance of security controls.
Florêncio & Herley (2011) [41]	Attack Path Discrepancies	Reveals a gap between typical user interactions and actual attack vectors, suggesting misalignment.	Encourages adaptive, context-aware authentication methods.
Beautement et al. (2008) [42]	Compliance Budget	Finds that users have a limited "compliance budget"—excessive friction drains their willingness to comply.	Minimizes frustration and preserves user engagement in security tasks.
Blythe & Coventry (2015) [43]	Security Interfaces	Suggests that tailoring interfaces to users' expertise levels enhances secure behavior.	Improves usability and promotes alignment with individual skill sets.

<b>Fable 5. Studies o</b>	n User-Centric	Design and	l Usability
---------------------------	----------------	------------	-------------

As these studies illustrate, security designs must be accessible, context-aware, and minimally intrusive. These principles bridge the gap between user convenience and robust defense, reinforcing that a careful blend of technical and psychological considerations leads to higher compliance and fewer workarounds.

In summary, the literature reveals that cognitive biases, social engineering tactics, motivational frameworks, organizational culture, and usability considerations collectively define the human dimension of cybersecurity. The research points toward adaptive and user-centered solutions as the most promising avenues for reconciling technical controls with actual user behaviors addressing all three research questions and paving the way for further innovations in psyber security.

#### 4. Discussion of Gaps and Opportunities

The research consolidated in Section 3 underscores the multifaceted nature of psyber security, merging insights from cognitive psychology, social engineering studies, motivational theories, organizational behavior, and user-centric design. Despite this breadth, certain gaps and opportunities persist. Many interventions produce short-term improvements, yet their long-term sustainability and generalizability remain uncertain. Moreover, the ethical and privacy implications of personalized, data-driven interventions warrant deeper scrutiny. This section provides a structured evaluation of these challenges and potential avenues for advancing psyber security.

## 4.1 Gaps in Current Literature

## 4.1.1 Limited Longitudinal Research

While numerous studies document immediate gains in user compliance after awareness campaigns or training sessions, longitudinal data is sparse [44]. For instance, certain training programs demonstrate up to a 50% reduction in phishing click-through rates within a few weeks, but little is known about whether these effects endure beyond three to six months. Researchers frequently measure compliance in "snapshot" intervals, neglecting potential security fatigue or habituation effects over time.

## 4.1.2 Cultural and Contextual Variability

Cybersecurity behaviors and perceptions can be strongly influenced by cultural norms, organizational structures, and leadership styles [45], [46]. However, many studies rely on samples from a single region or industry (e.g., Western corporate environments). This lack of cross-cultural data hampers the applicability of recommended interventions in global or multi-branch organizations. For instance, culturally based attitudes toward authority or privacy could significantly modify how employees respond to social engineering attacks or mandatory security policies.

## 4.1.3 Ethical and Privacy Concerns

Personalized and data-intensive approaches such as real-time behavioral nudges may increase efficacy but also raise privacy and ethical dilemmas [47]. Questions arise regarding data ownership, consent, and potential overreach if organizations constantly monitor user behavior. Implementing transparent, user consented data collection can mitigate these concerns, yet frameworks for balancing security imperatives with individual autonomy remain underdeveloped.

#### 4.1.4 Lack of Adaptive and Personalized Training

A one-size-fits-all approach to cybersecurity education may not suffice in highly diverse or evolving threat landscapes. Although research suggests that adaptive training systems in which content evolves based on user performance or risk profiles could yield higher compliance, real world examples remain limited [48]. Developing machine learning models that dynamically tailored security prompts holds promise but requires robust datasets and consistent evaluation metrics.

Gap	Description	Example Reference
Limited Longitudinal Data	There is a lack of studies examining the sustained effects of behavioral interventions over extended periods.	[44]
Cultural and Contextual Variability	Many studies are limited to a single region or industry, resulting in inadequate cross- cultural validation of training programs.	[45], [46]
Ethical and Privacy Concerns	Insufficient frameworks exist for managing data governance and obtaining informed consent in personalized, data-intensive interventions.	[47]
Lack of Adaptive Training	Few real-world implementations of adaptive, context-aware training systems are available, limiting their long-term efficacy.	[48]

#### Table 6. Key Gaps Identified in Psyber-Security Literature

Identifying these gaps clarifies why promising interventions may stall when scaled across different user populations or sustained over time. Addressing these limitations forms a critical next step in refining the psyber security landscape.

## 4.2 Opportunities for Advancement

## 4.2.1 Emphasis on Long-Term Behavior Change

To overcome short-lived improvements, future research should embed longitudinal methodologies, tracking user engagement and threat detection accuracy for at least 6 to 12 months post training. Integrating booster sessions or refresher modules at regular intervals could bolster knowledge retention. Researchers might also investigate the threshold at which additional training or repeated phishing simulations become counterproductive due to fatigue.

## 4.2.2 Cross-Cultural and Sector-Specific Customization

As organizations expand globally, customizing cybersecurity interventions to local norms and languages can significantly improve effectiveness [45]. This approach can be extended to different sectors healthcare, finance, government each of which faces distinct regulatory and operational constraints. Interdisciplinary teams (involving local cultural experts, HCI specialists, and cybersecurity professionals) could develop and test context-specific training modules, gathering comparative data to refine best practices.

## 4.2.3 Ethical Frameworks for Personalization

Appropriate ethical guidelines must accompany sophisticated methods of data-driven security. Developing consent based models where employees explicitly opt in to certain forms of behavioral monitoring could foster transparency and trust. Additionally, incorporating user feedback loops allowing individuals to adjust or opt out of specific nudges might balance organizational security goals with personal autonomy. Future work could explore how these frameworks affect user attitudes, compliance, and the overarching security climate [47].

#### 4.2.4 AI-Driven Adaptive Systems

Integrating machine learning could allow real-time adaptation of training content and security prompts based on user behavior and threat intelligence. Systems might analyze user interaction patterns to issue tailored nudges or highlight relevant scenarios, thereby improving the timeliness and personal relevance of interventions [48]. Nonetheless, robust evaluation methods are necessary to verify that adaptive algorithms do not inadvertently introduce biases or misclassifications that compromise security and user trust.

Opportunity	Proposed Approaches	Potential Benefits	Reference
Long-Term Behavior Change	- Implement longitudinal studies (6–12+ months) and periodic refresher sessions	- Sustains high security awareness; mitigates training fatigue	[44]
Cross-Cultural Customization	- Develop localized training content and sector- specific guidelines; conduct collaborative pilot tests	- Enhances relevance and improves user engagement across diverse settings	[45], [46]
Ethical Frameworks for Personalization	- Design consent-based monitoring systems with transparent user feedback loops	- Builds trust and balances security needs with individual privacy	[47]
AI-Driven Adaptive Systems	- Utilize ML-based user segmentation, real-time risk scoring, and customized nudges	- Provides timely, personalized interventions that adapt to evolving threats	[48]

#### Table 7. Potential Opportunities and Proposed Approaches

Pursuing these opportunities could significantly extend the impact of psyber-security measures. Longitudinal designs and cultural adaptations promise greater sustainability and global applicability, while ethical frameworks and AI-driven personalization safeguard user trust and adapt to emerging threats.

## 4.3 Synthesis and Path Forward

Taken together, the gaps and opportunities detailed above illustrate both the complexity and the potential of psyber security. While there is growing consensus on the efficacy of user-centric and behaviorally informed solutions, inconsistencies in duration, cultural reach, and ethical governance still impede universal adoption. Closing these gaps requires collaboration among cybersecurity experts, behavioral scientists, policy makers, and industry stakeholders. Establishing rigorous long-term research, embracing cultural diversity, and formalizing ethical protocols for personalization and data management emerge as top priorities. With these pillars in place, psyber security can transition from promising pilot studies to sustained, scalable applications that effectively counter ever evolving cyber threats while respecting user rights and contexts.

#### **5.** Conclusion and Future Directions

This paper has provided a comprehensive literature review on how psychological and behavioral insights can be integrated into cybersecurity to address longstanding human vulnerabilities. Organized around three key research questions, the review has examined the roles of cognitive biases, social engineering, user motivation, organizational culture, and usability in shaping security outcomes. In conclusion, the literature consistently demonstrates that targeted behavioral interventions (e.g., nudges, gamification), scenario based training, and user centric design can reduce susceptibility to social engineering and enhance policy compliance. However, the evidence also highlights the fragility of these improvements, particularly under repeated exposure, cultural shifts, or organizational inertia. Future work must investigate long term effectiveness, explore cross-cultural generalizability, and refine ethical frameworks for personalizing interventions. Ultimately, psyber security underscores that human factors are not peripheral but central to effective cyber defense. By continuing to refine theoretical models, leverage adaptive technologies, and cultivate supportive organizational cultures, cybersecurity practitioners can ensure that technical mechanisms and human behaviors evolve in tandem resulting in resilient defenses for an ever-changing threat landscape.

#### References

[1] B. Schneier, *Click Here to Kill Everybody: Security and Survival in a Hyper-connected World*, 1st ed. New York, NY, USA: W. W. Norton & Company, 2018.

[5] Vishwanath, Arun, Brynne Harrison, and Yu Jie Ng. "Suspicion, cognition, and automaticity model of phishing susceptibility." Communication research 45, no. 8 (2018): 1146-1166.

<sup>[2]</sup> Balsara, Bhavish. "A COMPARATIVE STUDY OF PATTERNS, CAUSES, AND IMPACTS OF DATA BREACHES ACROSS GEOGRAPHICAL REGIONS AND TIME FRAMES." (2024).

<sup>[3]</sup> Lebrecht, Sophie, Moshe Bar, Lisa Feldman Barrett, and Michael J. Tarr. "Micro-valences: Perceiving affective valence in everyday objects." Frontiers in psychology 3 (2012): 107.

<sup>[4]</sup> Mitnick, Kevin D., and William L. Simon. The art of deception: Controlling the human element of security. John Wiley & Sons, 2003.

[6] Enisa. "Cybersecurity culture guidelines: behavioural aspects of cybersecurity." European Union Agency for Network and Information Security (2018).

[7] Wang, Zuoguang, Hongsong Zhu, and Limin Sun. "Social engineering in cybersecurity: Effect mechanisms, human vulnerabilities and attack methods." Ieee Access 9 (2021): 11895-11910.

[8] Furnell, Steven. "End-user security culture: a lesson that will never be learnt?." Computer fraud & security 2008, no. 4 (2008): 6-9.

[9] Bishop, Matt, and Carrie Gates. "Defining the insider threat." In Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead, pp. 1-3. 2008.

[10] Ceric, Arnela, and Peter Holland. "The role of cognitive biases in anticipating and responding to cyberattacks." Information Technology & People 32, no. 1 (2019): 171-188.

[11] Hong, Jason. "The state of phishing attacks." Communications of the ACM 55, no. 1 (2012): 74-81.

[12] Blythe, John M., and Shane D. Johnson. "A systematic review of crime facilitated by the consumer Internet of Things." Security Journal 34 (2021): 97-125.

[13] Hadnagy, Christopher. Social engineering: The art of human hacking. John Wiley & Sons, 2010.

[14] Luo, Xin, Richard Brody, Alessandro Seazzu, and Stephen Burd. "Social engineering: The neglected human factor for information security management." Information Resources Management Journal (IRMJ) 24, no. 3 (2011): 1-8.

[15] Dimkov, Trajce, André Van Cleeff, Wolter Pieters, and Pieter Hartel. "Two methodologies for physical penetration testing using social engineering." In Proceedings of the 26th annual computer security applications conference, pp. 399-408. 2010.

[16] Rogers, Ronald W. "A protection motivation theory of fear appeals and attitude change1." The journal of psychology 91, no. 1 (1975): 93-114.

[17] Ryan, Richard M., and Edward L. Deci. "Intrinsic and extrinsic motivations: Classic definitions and new directions." Contemporary educational psychology 25, no. 1 (2000): 54-67.

[18] Siponen, Mikko, and Anthony Vance. "Neutralization: New insights into the problem of employee information systems security policy violations." MIS quarterly (2010): 487-502.

[19] Renaud, Karen, and Verena Zimmerman. "Enriched nudges lead to stronger password replacements... but implement mindfully." In 2017 Information Security for South Africa (ISSA), pp. 1-9. IEEE, 2017.

[20] Da Veiga, Adele, Nico Martins, and Jan HP Eloff. "Information security culture-validation of an assessment instrument." Southern African Business Review 11, no. 1 (2007): 147-166.

[21] Solomon, Grant, and Irwin Brown. "The influence of organisational culture and information security culture on employee compliance behaviour." Journal of Enterprise Information Management 34, no. 4 (2021): 1203-1228.

[22] Crossler, Robert E., James H. Long, Tina M. Loraas, and Brad S. Trinkle. "Understanding compliance with bring your own device policies utilizing protection motivation theory: Bridging the intention-behavior gap." Journal of Information Systems 28, no. 1 (2014): 209-226.

[23] Pfleeger, Shari Lawrence, and Deanna D. Caputo. "Leveraging behavioral science to mitigate cyber security risk." Computers & security 31, no. 4 (2012): 597-611.

[24] Parsons, Kathryn, Agata McCormac, Marcus Butavicius, and Lael Ferguson. Human factors and information security: individual, culture and security environment. 2010.

[25] Taylor, Richard E. "Adopting Cybersecurity and Threat Awareness Training Bolsters Multinational Organization Security Posture." PhD diss., National University, 2024.

[26] Sasse, M. Angela, and Ivan Flechais. "Usable security: Why do we need it? How do we get it?." O'Reilly, 2005.

[27] Florêncio, Dinei, and Cormac Herley. "Where do all the attacks go?." In Economics of information security and privacy III, pp. 13-33. New York, NY: Springer New York, 2012.

[28] Tipton, Harold F., and Micki Krause. Information security management handbook. CRC press, 2007.[29] Beautement, Adam, M. Angela Sasse, and Mike Wonham. "The compliance budget: managing security behaviour in organisations." In Proceedings of the 2008 new security paradigms workshop, pp. 47-58. 2008.

[30] Blythe, John M., Lynne Coventry, and Linda Little. "Unpacking security policy compliance: The motivators and barriers of employees' security behaviors." In Eleventh symposium on usable privacy and security ({soups} 2015), pp. 103-122. 2015.

[31] Coles-Kemp, Lizzie, and Richard E. Overill. "Triangulating the Views of Human and non-Human Stakeholders in Information System Security Risk Assessment." Freeman 1984, no. 46 (2007).

[32] Acquisti, Alessandro, Laura Brandimarte, and George Loewenstein. "Privacy and human behavior in the age of information." Science 347, no. 6221 (2015): 509-514.

[33] Jensen, Matthew L., Michael Dinger, Ryan T. Wright, and Jason Bennett Thatcher. "Training to mitigate phishing attacks using mindfulness techniques." Journal of Management Information Systems 34, no. 2 (2017): 597-626.

[34] Acquisti, Alessandro, Laura Brandimarte, and George Loewenstein. "Privacy and human behavior in the age of information." Science 347, no. 6221 (2015): 509-514.

[35] Singh, Kuldeep, Palvi Aggarwal, Prashanth Rajivan, and Cleotilde Gonzalez. "Cognitive elements of learning and discriminability in anti-phishing training." Computers & Security 127 (2023): 103105.

[36] Ablon, Lillian, and Andy Bogart. Zero days, thousands of nights: The life and times of zero-day vulnerabilities and their exploits. Rand Corporation, 2017.

[37] Perlroth, Nicole. This is how they tell me the world ends: The cyberweapons arms race. Bloomsbury Publishing USA, 2021.

[38] Libicki, Martin C., Lillian Ablon, and Tim Webb. The defender's dilemma: Charting a course toward cybersecurity. Rand Corporation, 2015.

[39] Linkov, Igor, and Alexander Kott. "Fundamental concepts of cyber resilience: Introduction and overview." Cyber resilience of systems and networks (2019): 1-25.

[40] Solove, Daniel J., and Woodrow Hartzog. Breached!: Why data security law fails and how to improve it. Oxford University Press, 2022.

[41] Sood, Aditya, and Richard Enbody. Targeted cyber attacks: multi-staged attacks driven by exploits and malware. Syngress, 2014.

[42] Skopik, Florian, and Timea Pahi. "Under false flag: using technical artifacts for cyber attack attribution." Cybersecurity 3 (2020): 1-20.

[43] Sloan, Robert, and Richard Warner. Why Don't We Defend Better?: Data Breaches, Risk Management, and Public Policy. CRC Press, 2019.

[44] Lehto, Martti. "Cyber-attacks against critical infrastructure." In Cyber security: Critical infrastructure protection, pp. 3-42. Cham: Springer International Publishing, 2022.

[45] Crawley, Kim. 8 Steps to Better Security: A Simple Cyber Resilience Guide for Business. John Wiley & Sons, 2021.

[46] Daswani, Neil, and Moudy Elbayadi. Big breaches: Cybersecurity lessons for everyone. New York: Apress, 2021.

[47] Fowler, Kevvie. Data breach preparation and response: Breaches are certain, impact is not. Syngress, 2016.

[48] Singer, Abe, and Matt Bishop. "Trust-based security; or, trust considered harmful." In Proceedings of the New Security Paradigms Workshop 2020, pp. 76-89. 2020.

# More Than Technical - Embedding Professional Competencies in a Software Requirements Course

Sean Jones Department of Computer Science and Software Engineering Milwaukee School of Engineering Milwaukee, WI, 53202 jones@msoe.edu

Midwest Instruction and Computing Symposium, 2025

#### Abstract

This paper explores a semester-long, project-driven approach to teaching Software Requirements and Architecture, aligned with MSOE's Common Learning Outcomes (CLOs), including Communication, Collaboration, Critical Thinking, Curiosity, and Integrated Learning. Students work in teams to address real-world problems, iteratively developing a comprehensive requirements document. Weekly assignments incorporate stakeholder interviews, persona creation, use case modeling, and prototyping, fostering both technical and professional skills. The iterative framework emphasizes critical thinking, effective communication, and teamwork while refining system designs and documentation. The course culminates in detailed requirements artifacts and a professional presentation. This approach prepares students for industry challenges by integrating technical expertise with essential soft skills, offering a replicable model for engineering education.

# Introduction

In the evolving landscape of software engineering, professionals are increasingly expected to balance technical expertise with critical non-technical skills such as collaboration, communication, and critical thinking. Engineers must navigate complex problem domains, gather requirements from diverse stakeholders, and synthesize technical and user-centered perspectives into actionable designs. To prepare students for these multifaceted demands, engineering education must provide not only technical training but also opportunities to develop the professional competencies essential for success in real-world environments.

This paper presents a semester-long, project-driven approach to teaching Software Requirements and Architecture, designed to align with the Milwaukee School of Engineering (MSOE) Common Learning Outcomes (CLOs)[1] that are a collection of outcomes based on the Entrepreneurial Mindset[2]. The course serves as a practical framework for students to learn both technical and professional skills by engaging in iterative, hands-on activities centered on a self-selected, real-world project. Student teams identify a software project they wish to develop, investigate its problem domain, and iteratively construct a comprehensive requirements document. Along the way, they gain experience with industry-standard practices such as stakeholder interviews, persona creation, use case modeling, prototyping, and requirements validation.

The course is structured to align with MSOE's CLOs, embedding professional skills into every stage of the curriculum. Students Collaborate Successfully as they work in teams to conduct research, brainstorm solutions, and synthesize their findings into cohesive deliverables. Communicating Effectively is emphasized through written artifacts, such as requirements documents and glossaries, as well as oral presentations tailored to audiences with varying levels of technical expertise. By exploring problem domains, identifying stakeholder needs, and crafting creative solutions, students Exhibit Curiosity while refining their understanding of the broader context of software engineering challenges. Think Critically is embedded throughout the course, as students analyze competing requirements, evaluate trade-offs, and prioritize features. Finally, Integrative Learning is fostered as students draw upon their knowledge from prior coursework and interdisciplinary perspectives to address complex, real-world problems.

This approach is designed not only to reinforce the technical foundations of software requirements engineering but also to prepare students for the collaborative, adaptive, and communicative challenges they will encounter in industry. Each week of the semester builds upon the last, creating a cohesive learning experience that scaffolds technical and non-technical skill development. By giving students the autonomy to select their own projects and work iteratively toward a final solution, the course immerses them in the complexities of real-world software engineering, ensuring they graduate with a balance of practical expertise and professional acumen.

The remainder of this paper outlines the structure of the course, detailing its alignment with CLOs, the iterative nature of its assignments, and the observed outcomes. Through this case study, we provide a replicable model for integrating technical and professional skills into software engineering education.

## **Technical Outcomes**

Starting the project, I expected to spend my time refining and updated existing assignments as well as building new assignments and their outcomes to more explicitly align with the defined CLOs. The course was initially written with outcomes defined by the <u>Software Engineering Body of Knowledge (SWEBOK)</u>[3] in mind along with several textbooks by Karl Weigers[4][5], and while there is a good deal of overlap, the assignments were primarily focused on the technical application of techniques and coverage of topics. The students will, over the course of the term, develop a software project of their choosing from the ground up, starting with identifying a current need in the software landscape, then defining the scope of their project, the stakeholders, and the overall objectives before diving into requirements elicitation and analysis, functional and non-functional requirements development and requirements validation. Each week the students will work in a project group to further the project, and by the end of the term, they will have built a basic prototype that shows a very minimal amount of functionality.

# **Identifying CLOs**

My initial goal for this project was to identify ways that I could encourage critical thinking into the course. The structure of the course had changed slightly with the conversion to semesters from trimesters, which we had just undergone in the previous 12 months, and I felt finding ways to emphasize deeper engagement with the material could be an important improvement for the course. What ended up happening was not only did I find plenty of opportunities to integrate critical thinking as an outcome for many of the activities in the course, during my analysis of the material I found many of the other CLOs were also present and I was able to build out my assignments to include a number of CLOs, as shown in Figures 1 & 2.

I ended up spending much of my time analyzing the project, looking at each activity the students were asked to perform and determining a) if there were CLOs embedded in the work and b) if not, could there be some in a way that would improve the overall assignment. Obviously not every part of every assignment could be linked to a CLO, but I then went through the first nine weeks of the course and worked through the assignments and exercises, rewriting and updating many of the assignments and their rubrics, changing the technical deliverables slightly to allow me to assess the assignments with a focus on the CLOs. As an example, changing a rubric item description from "Each

persona embodies a specific user class" to "Each persona embodies a specific user class and is serious, believable, with their own motivations and attitudes towards the project" to ask the students to exhibit curiosity and embrace diversity without explicitly saying "And here you will embrace diversity by...". In the end, the course had the same basic

Week	Assignment	Outcomes (see Fig 2)
1	Lab 1 - Business Objectives	1, 2, 5, 6, 7
1	Exercise 1 - Stakeholder Identification	1, 2, 4, 5, 6, 7
2	Lab 2 - Project Kickoff	1, 2, 5, 6, 7
2	Exercise 2 - Persona Creation	1, 2, 4, 5, 6, 7
3	Lab 3 - Project Scope and Priorities	1, 2, 4, 5, 6, 7
3	Exercise 3 - Interviewing	1, 2, 5, 6, 7
4	Lab 4 - Requirements Elicitation	1, 2, 3, 4, 5, 6, 7
4	Exercise 4 - Deriving Use Cases	1, 2, 5, 6, 7
5	Lab 5 - Formalize Use Cases	1, 2, 4, 5, 6, 7
5	Exercise 5 - Writing Functional Requirements	1, 2, 6, 7
6	Lab 6 - Peer Presentation	1, 2, 6, 7
6	Exercise 6 - Identifying Data	1, 2, 6, 7
7	Lab 7 - Presentation Critiques and Data Modeling	1, 2, 5, 6, 7
7	Exercise 7 - Quality Attribute Prioritization	1, 2, 5, 6, 7
8	Lab 8 -Non-Functional Requirements	1, 2, 5, 6, 7
8	Exercise 8 - Requirement Validation	1, 2, 5, 6, 7
9	Lab 9 - Validation Inspection	1, 2, 5, 6, 7
9	Exercise 9 - Validation Implementation	1, 2, 5, 6, 7

Mapping	CLO	Definition
1	Collaborate Successfully	Work constructively with others towards a common goal
2	Communicate Effectively	articulate and explain complex ideas clearly across a range of media and audiences
3	Demonstrate Ethical Understanding,	Engage in independent ethical inquiry on pressing ethical challenges and foster ethical behavior in personal and professional life
4	Embrace Diversity	Demonstrate inclusivity toward others, pursuing intercultural understanding and exploring ways to address historical or existing barriers to social equity
5	Exhibit Curiosity	Practice open-minded intellectual inquiry, creative exploration, and engagement with different perspectives
6	Think Critically	Apply sound principles of critical or analytical reasoning and evaluation of evidence
7	Integrate Learning	Synthesize and transfer learning across new contexts to address complex problems through program-level courses, Raider Core courses, co-curricular activities and senior/capstone projects

Fig. 2 - CLO Mappings and Definitions

activities as before, with a few exceptions, but the ways in which the assignments were presented and assessed had a more prominent focus on the CLOs.

# **Course Assignments**

I had initially chosen "Think Critically: Apply sound principles of critical or analytical reasoning and evaluation of evidence" as the CLO I was targeting, as it seemed to me to be the most obvious choice where I could make the most impact. Every week we were asking the students to do more than just surface-level assessment of their data, and finding ways to foster and encourage deeper thinking and engagement would improve the student outcomes. As mentioned above, the project expanded to include many of the CLOs across a number of assignments.

I developed multiple lectures, activities, and assessments for students for this project. Figures 1 & 2 show the weekly content and the CLOs that they map to and their definitions as described by the institution. The sequence of assignments was designed to walk the students through the various stages of designing a software project with scaffolding exercises, done in groups and during class time, to give them practice on a small scale of what they will then do for their project. While each week has a specific focus, the continuing project is constantly updated and improved using the tools and techniques learned in the previous weeks and students will need to look at their projects from multiple perspectives to be successful in defining a complete requirements document for their application.

The process of working through the assignments and determining how the technical work the students were doing could relate to the various CLOs was quite instructive. You always want the students to be curious and this critically, but going through each activity and assignment throughout the process and doing my own critical thinking and asking "What do we want them to demonstrate here?" showed me both how to better appraise the submissions as they came in and how to better explain certain concepts and technique in lecture. An example of this was when describing what a Persona is to the class, after identifying all the CLOs that relate to that task (2, 4, 5, 6, 7) I was able to better explain what it meant to have a serious, effective Persona by discussing how Curiosity and thinking consciously about Diversity really drive the creation of a believable user. Previously, I would focus on Communicating and Thinking Critically, but not get them thinking about how they would imagine someone different from themselves would think and feel in the same situation. Then, when the submission would come in, I would start by looking at the Persona's motivations and attitudes to see if they really did consider these ideas in their work. Assessment would often proceed in a similar manner, where once the technical competencies were checked, I would go through each identified CLO and see how well the students demonstrated the outcomes in their work. Below is a brief summary of several assignment and exercise in the nine week requirements sequence and the identified CLO's that map to each with an explanation of the CLOs relationship to the assignment. A complete list of all the assignments and CLO mappings is available in a project card at Engineering Unleashed (link in the Final Thoughts section at the end of the paper)

## Week 1

## Assignment:

The first week asks them to look critically at a website (<u>my.msoe.edu</u>) and recommend changes to modernize and update the site, making it more useful for staff and students alike, justifying their recommendations by identifying the benefits the school will realize through these improvements.

## CLOs:

*Collaborate Successfully*: Although the assignment is framed individually, it inherently involves collaboration with, or at least some form of empathizing with, stakeholders, such as understanding the administration's needs and aligning project objectives with broader institutional goals. Further, as it is performed in a group setting, it also develops teamwork and negotiation skills.

*Communicate Effectively*: Students must clearly articulate business requirements and objectives in a structured, professional manner. The inclusion of measurable acceptance criteria demonstrates effective written communication tailored to stakeholders who may have varying levels of technical expertise.

*Exhibit Curiosity*: Exploring the existing my.msoe website and its workflows requires intellectual curiosity. Students will need to go beyond surface-level observations to identify what about the website's functionality can or should be altered to provide the same functionality while improving the user experience.

*Think Critically*: This assignment requires students to analyze the current website, evaluate its strengths and weaknesses, and develop well-reasoned business requirements and objectives. Critical thinking is essential to identify user needs, workflows, and gaps in functionality while proposing practical solutions to meet these needs.

*Integrate Learning*: To successfully complete this assignment, students must synthesize knowledge from multiple areas of study, such as software engineering, business analysis, and user experience design. This integrative approach ensures that solutions address both technical and organizational needs.

## **Exercise:**

The exercise introduces the concept of a project stakeholder and has the students identify as many possible stakeholders for a software project as they can, encouraging them to dive deeper and find stakeholders that may be less obvious.

## CLOs:
*Collaborate Successfully:* This exercise emphasizes teamwork as students work in groups to brainstorm and identify a comprehensive list of stakeholders. Collaboration requires active participation, listening, and integrating diverse perspectives to ensure the list is thorough and representative.

*Communicate Effectively:* Students must document their findings in a clear and organized manner, ensuring the list of stakeholders is easily understood and actionable. Sharing their insights with the class further develops their verbal communication skills.

*Embrace Diversity*: Identifying disparate stakeholders with distinct motivations and goals, especially ones with different perspectives and opinions, fosters an appreciation for diverse user needs. This CLO is particularly relevant as students consider stakeholders from different backgrounds, experiences, and technological competencies.

*Exhibit Curiosity:* The exercise encourages curiosity about the broader ecosystem of an LMS, prompting students to explore and consider diverse perspectives, from administrators and faculty to students and IT staff.

*Think Critically:* Identifying stakeholders involves analyzing the project's scope and functions to determine who has a vested interest or is affected by the new LMS. Students must critically evaluate various roles and relationships within the system to ensure all relevant stakeholders are included.

*Integrate Learning:* Students apply their understanding of stakeholder analysis and systems thinking, connecting insights from coursework to a practical, real-world scenario. This integration ensures their analysis is grounded in practical considerations.

## Week 4

#### Assignment:

Week four has the students beginning their requirements document by transferring much of what was defined in their Vision and Scope document to their Software Requirements Specification as well as building use cases for some of the features they identified. They will also be preparing two interview plans for two different types of stakeholders: technical and non-technical.

## CLOs:

*Collaborate Successfully:* Working as a team is critical to planning and conducting workshops, preparing stakeholder interviews, and documenting elicitation results. Collaboration is also essential for managing roles, coordinating schedules, and ensuring the quality of deliverables.

*Communicate Effectively:* Students must develop clear workshop agendas, draft wellstructured interview plans and questions, and produce detailed notes and use case writeups.

*Demonstrate Ethical Understanding:* Planning and conducting interviews, particularly with non-technical stakeholders, requires sensitivity to ethical considerations, such as respecting participants' time and perspectives, and ensuring the confidentiality of recorded interviews.

*Embrace Diversity:* Engaging with stakeholders from different backgrounds, including non-SE/CS/CE students, encourages students to consider diverse user needs and perspectives. This helps foster inclusivity and broader thinking in their requirements efforts.

*Exhibit Curiosity:* The assignment encourages students to explore the problem domain and understand system features in depth. This involves asking insightful questions and being open to new ideas during workshops and preparing the interview questions and structure.

*Think Critically:* Students must analyze information gathered during workshops, connect it to system features, and refine their understanding of the problem domain. Critical thinking ensures that their use cases and system features are meaningful and actionable.

*Integrate Learning:* The assignment requires synthesizing knowledge from various areas, such as project management, stakeholder analysis, and system design, to create comprehensive SRS sections and elicitation artifacts. This integration supports a holistic approach to software requirements.

## **Exercise:**

The exercise gives the students an opportunity to work together to construct a formal use case for a user interaction with a system. In small groups they produce a draft of a use case, then as a class we use ideas from all the groups to build a more complete use case. They will also identify business rules that will affect the way the program executes.

#### CLOs:

*Collaborate Successfully:* Students work in small groups to draft Use Cases and then contribute their ideas to a larger class discussion. This collaboration fosters teamwork, shared understanding, and collective improvement of the deliverable.

*Communicate Effectively:* Writing Use Cases requires clear articulation of the normal flow, alternate flows, and business rules. Sharing and discussing ideas with the class further emphasizes the importance of effective communication in both written and verbal forms.

*Exhibit Curiosity:* Developing a Use Case involves exploring the problem domain and understanding the intricacies of the system's behavior. Students must ask questions and consider multiple perspectives to create comprehensive and accurate scenarios.

*Think Critically:* Students must analyze the problem, consider potential variations in the system's behavior, and ensure the Use Case is logical and aligns with existing business rules.

*Integrate Learning:* Creating Use Cases involves integrating knowledge of system modeling, stakeholder needs, and business rules. Students must synthesize these elements to produce a cohesive and usable artifact.

### Week 7

### Assignment:

Week seven has the students first watching their peers presentations and writing critiques, which will serve both as feedback to the presenting groups and reflection by the critic as to how their presentation compares to that of their peers. The remainder of the week they will be deriving functional requirements from the use cases they built over the previous weeks and documenting them with references to where they were derived from. They will also complete their data model and have documented all of the data that the features they are designing will use and define how the data will be displayed and/or accessed by the program. Finally, they will create several prototypes showing the vision of their final product.

## CLOs:

*Collaborate Successfully:* The assignment involves teamwork to develop functional requirements, data requirements, and prototype as well as the execution of the presentations. Effective collaboration is necessary to combine diverse ideas, ensure clarity, and distribute tasks among team members.

*Communicate Effectively:* Critiquing student presentations requires the ability to articulate constructive feedback. Additionally, documenting functional and data requirements, as well as creating prototypes, demands clear communication for both technical and non-technical audiences.

*Exhibit Curiosity:* Students demonstrate curiosity by exploring different ways to improve their system's functionality through thoughtful analysis and creative prototype design.

*Think Critically:* Developing functional requirements based on previous elicitation and analysis, defining data models, and identifying reports require students to critically evaluate their use cases and ensure the requirements align with the system's goals and constraints.

*Integrate Learning:* The assignment integrates knowledge of requirements elicitation, system design, and user interaction through functional requirements, data modeling, and prototype creation. It challenges students to connect these elements into a cohesive representation of their system.

## **Exercise:**

The exercise this week has the students identify non-functional requirements (also known as quality attributes) of their system and prioritize them based on the needs of the stakeholders and alignment with the overall vision of the project. They will debate the importance of each of the various categories of quality attributes and through a series of head-to-head comparisons, developing a matrix where each category is compared to every other category, generating rankings for each category.

## CLOs:

*Collaborate Successfully:* Students must work together to debate, compare, and prioritize quality attributes, requiring effective teamwork and constructive discussion to reach a consensus.

*Communicate Effectively:* The exercise involves articulating arguments for the importance of specific non-functional requirements and justifying their prioritization, requiring clear communication within the group.

*Exhibit Curiosity:* Exploring the implications of different quality attributes and understanding their impact on the system demonstrates intellectual curiosity and a desire to optimize the system's performance.

*Think Critically:* Evaluating the relevance and significance of different quality attributes involves critical thinking to ensure that the prioritization aligns with stakeholder needs and the project's vision.

*Integrate Learning:* This exercise requires students to integrate knowledge of stakeholder priorities, system design principles, and project goals to determine the most critical quality attributes for the system.

# Assessment

The goal of this sequence of assignments is to have students, working in teams, return each week to the same large problem and, in an iterative fashion, build a solution to a complex problem through continuously looking critically at their current solution from a variety of viewpoints and priorities. Over the course of the nine weeks, they will start with the germ of an idea and develop it into a requirements document that will be used in the remainder of the course to implement a portion of the system they designed. Throughout this project, I learned that students don't need much encouragement to think critically when given some form of starting point. They needed very little supervision once given a problem to tackle and some scope to work under. Having specific activities that exercise and have grading criteria related to critical thinking also help emphasize the importance of the skills for students.

I shared my lectures, labs, exercises, and other related materials from this project with the other faculty that will be teaching the course with me next fall and it will be the expected content for the course going forward. There is a second portion of this course related to architecture and implementation that would benefit from this same approach of identifying CLOs and adjusting the deliverables and grading criteria to highlight them.

Assessing critical thinking and curiosity is a difficult task. It is often easier to identify when students are not critically thinking than when they are, with shallow or incomplete documentation or justifications in their reports. My approach was to look at their work holistically across the term and identify where the students demonstrated deeper understanding of their work as they iterated through their project in addition to the discussion we had during the class exercises.

An example of this would be in the Week 5 lab assignment where the students have the following goals:

- Interview two stakeholders of the project to identify project needs
- Develop two detailed personas (imaginary stakeholders)
- Identify the Stimulus/Response pairs for the business events selected from the previous week
- Write five formal Use Cases based on prior research and interview results

The rubric would have the technical requirements (formatting, the contents of each Use Case, how the S/R pairs should be written, etc) to ensure the students are on track and understand what they need to accomplish. When grading the assignment, each of the CLOs that have been mapped to it are used to determine how successful they were at completing their tasks.

- Collaborate Successfully is baked into this assignment. There is too much for just one or two students to complete within the week timeframe. All members of the group are required to participate in the interviewing process, assigned roles, and the artifacts for each role will be submitted. Additionally, they will need to schedule and execute two separate team workshops to construct their Use Cases.
- Communicate Effectively comes into play in various places in this assignment. The interviews themselves (which are recorded) need to show the students actively listening to the responses of their interviewees and asking thoughtful follow-up questions. If they simply ask their pre-written questions or do not

effectively explain what the purpose of the interview is, or what the goal is (what they are trying to learn) it will be considered insufficient.

- Embrace Diversity comes in with the creation of their Personas. A Persona is a detailed description of a type of user of the system. In order to be useful, a Persona needs to be as realistic as possible, with motivations, personality, and attitudes as well as technical needs. The two Personas they create must be different in their needs of the system, and the students must demonstrate a good degree of empathy and understanding of people other than themselves to create a satisfactory Persona. Shallow or unserious Personas are not acceptable.
- Exhibit Curiosity will show in the interviews and the type of questioning they have in response to the answers they get, in the Personas they create and how they approach different attitudes and the motivations behind those attitudes, and in trying to understand stakeholder needs beyond the minimum technical requirements of a piece of software, such as identifying alternate techniques and pathways to accomplish the same goals and allowing the software to be used in different ways by different user types.
- Think Critically is assessed in the Use Cases. They will need to construct a starting "Normal Flow" of their program, based on how a typical interaction between the system and user would go, identifying each step of the process, each Stimulus/Response pair, what the user provides, what the system does, what the system displays to the user, etc. Then they will need to look beyond the typical workflow and identify alternate paths the interaction might take, including seeing possible mistakes or errors, more specific interactions (such as logging into a kiosk at an airport to check in, does the user have a Frequent Flyer number> What about TSA-Pre? Are they checking bags? Do they have any disabilities that will change the way the information is presented, etc). In order to actually complete their Use Cases, a lot of thought and effort needs to be put into them.
- Integrate Learning is crucial in this assignment. They have spent 4 weeks learning various techniques on how to decompose their program into discrete events and have gathered a lot of information from their earlier research, their interviews, and their team workshops. Now they need to take all of that, combined with their knowledge of software systems and build detailed documentation on how to implement an application that will satisfy all the needs of the different stakeholders. Everything they have documented up until now, including the notes from their workshops, should all be present in the Use Cases that they write.

The rubric that I supply to the students do not have all of this detail in them, and it was a. conscious choice to limit the rubrics to just the technical requirements. I struggled with this decision, because on one hand explicitly describing my expectation of follow-up questions in the interviews, for example, would tell the students what they should be

doing and be helpful in them understanding how I would be grading them. But the goal of this class is for the students to take ownership of their project, see all the complexities a real software project has through their own discovery and experiences, and treat it as something they would be presenting to the world. Part of that is them showing the maturity to perform their tasks as well as they can and take the project seriously intrinsically rather than me telling them "how much work" they should put in and them doing just enough to check off each item in the rubric. Additionally, as each team will have a very different project, the details of how much is sufficient for each portion of the project varies significantly. I do have language in each rubric about some of these expectations (i.e. Each persona is serious, believable, and embodies a specific user class) but it is up to the students to understand what that means for their project. I give a good amount of feedback to them during lab meetings and in assignment feedback in the first few weeks of the project development so they see how comprehensive their project needs to be and set expectations so they know that they will need to engage in the material to be successful. I have included an example of a team's submissions as an addendum to illustrate the type of work the students were submitting

# **Final Thoughts and Reflection**

My activities and materials have been shared on EngineeringUnleashed.com: <u>https://</u>engineeringunleashed.com/card/edit/4684

I found this project to be quite useful for me as a way to highlight and focus on the different skills we are trying to teach our students. Software Requirements and Architecture is a technical course that focuses more on many of the non-technical "soft" skills that are critical for the success of our students, and being able to identify for each of the assignments we give them what specific outcomes we expect them to take away changed the way I presented and assessed the material.

How did the students respond to the new material and approach? after running the course this past term, I can't say it was radically different as, after all, the overall contents of the course remained the same. But what I did notice was the different conversations I was having with the students regarding their projects and small details in assignments that showed me this group of students were thinking about their project with a little more depth than previous years.

Towards the end of the project, when I was talking to one group about changes they made to their Use Cases, they would refer to needs of their Persona (Bob) as if he were a real person. They had altered the way they handled a few alternate flows because Bob, who had a specific reason to use the software that was not part of their normal flow of events, would have had to perform several extra steps each time he used the system the way they had it written up. So they "listened" to Bob and created an alternate flow within the use case that addressed his needs with some change that could be stored as a user configuration so Bob would only need to perform the extra work once. The fact that they were paying attention to and empathizing with their virtual user and putting in the effort to come up with a creative solution to address his needs was not something I had explicitly seen in previous years.

A similar thing happened in a different group where there was a sensitive issue that arose regarding gender identification that was actually quite relevant to their application. In their requirements, they made several minor changes to their work flow and wording that addressed the issue. In previous years this kind of detail wouldn't have been identified, let alone addressed in a simple, effective manner. How much of that is the cultural sensitivity of this group, I'm not sure, but seeing it manifest into their work is one of the goals I had for the project, so I was pleased.

One of the things that I learned through this process that I am hoping will be evident to the other instructors using this approach is how useful it is to identify these outcomes and how using a CLO, such as Think Critically, when grading actually makes it clearer what should be considered acceptable. Tying the CLOs to specific items clearly identifies what that part of the assignment is trying to assess and the technical content can be looked at through a different lens, beyond does it work or does it conform to the required format. The example I like to use when talking to other faculty about this is when reading a team's Use Case, you can start by verifying that all the standard parts of a Use Case are present and that everything is technically sound. But then, if you want to determine if the Use Case actually does capture the entirety of the event it is describing, look at it through the Think Critically lens. Where is the evidence they used? Is everything that they captured and identified present in their work? Are there edge cases they missed that their evidence suggests? Then look through the Exhibit Curiosity lens. Does this satisfy all the user classes or does it ignore the needs of some? And so on

All of this kind of detail can't be captured in a rubric without a lot of additional explanation, or the rubric itself would be pages long. But when framed as check the boxes for the technical merit, which is fast and can be automated in some cases, then take a look at each of these items through the CLOs they are associated with, it provides a simple framework for faculty to follow for assessment. It does require a good deal of experience and expertise in the subject, which should be the case, but being able to point to how it makes grading large projects like this approachable and manageable is something I took away from this project and something I hope other faculty will understand and embrace, as well.

# References

- 1. Milwaukee School of Engineering. (n.d.). The MSOE mindset. Retrieved March 20, 2025, from https://www.msoe.edu/the-msoe-mindset/msoemindset/
- 2. Engineering Unleashed. (n.d.). The entrepreneurial mindset. Retrieved March 20, 2025, from <u>https://engineeringunleashed.com/mindset</u>
- 3. H. Washizaki, eds., Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), Version 4.0, IEEE Computer Society, 2024; www.swebok.org.
- 4. Wiegers, K., & Beatty, J. (2013). Software requirements (3rd ed.). Microsoft Press.
- 5. Wiegers, K., & Hokanson, C. (2023). Software requirements essentials: Core practices for successful business analysis. Addison-Wesley Professional.

# AI Plays Ultimate TicTacToe

Tic tac toe is a classic game that many Americans grew up playing. However, the game is solved. 'Solved' meaning if you start first and you know the solution you can't lose the game. Ultimate tic tac toe takes a tic tac toe board and places a tic tac toe board in each square. Depending on the square placed on the inner tic tac toe board determines which outer square the opponent can place their next move. I am inspired to improve upon the existing ultimate tic tac toe AI. There's room for improvement due to ultimate tic tac toe's complexity and the fact that it's difficult to determine the impact of the moves when there are low amounts of squares filled.

since the creation of the Phil Chen's AI there has been a change to how we think about the variant of ultimate tic tac toe.Guillaume Bertholon discovered a forced win where the starting player can always win. We can have one of two ways of looking about this either the AI we can create will implement the winning pattern of moves when it's first or our AI can solve a new ultimate tic-tac-toe that doesn't have this forced win.

In the case of the winning pattern there will be an even split against any opponent assuming both players know the winning strategies making it uncompetitive.

there is a variant of ultimate tic tac toe that will solve this issue. This variant proposed by Justin Diamond solves this issue by playing the same game but the first four moves are decided for you. This variant was chosen because it kept its limited number of moves. In our new variant The upper bound for the amount of choices in any game is

 $O(9 ^7 * 8 ^9 * 7 ^9 * 6 ^9 * 5 ^9 * 4 ^9 * 3 ^9 * 2 ^9)$  or  $O(1.347045536 * 10^{48})$ . The upper bound for the variant of tic tac toe where when placed in a mini board that is already won is the size of  $O(9 ^6 * 75!)$  or  $O(1.31845946 * 10 ^115)$ . You can see the amount of nodes in a decision tree in the variation of ultimate tic tac toe I choose is much smaller. This Should increase the performance of our bots.

The bots used during our research are variations of Minimax and Monte Carlo Tree Search. The Original Implementations of which are provided by a research study done by CHEN, Phil, Jesse Doan, and Edward Xu. Minimax AI uses an evaluation function to determine who is winning on a board then will make the move that results in the best score after seeing a certain number of moves.

Monte Carlo Tree search or MCTS for short uses randomness to calculate the average win rate of each move. First it picks a move, typically a move that is the result of a move that has a higher win rate. Then they will calculate the result of that game using random moves and add the result to the average to the move and all following moves. After the desired amount iterations is reached the move with the highest average win rate is made

Original results of P Chen Study:



CHEN, PHIL, JESSE DOAN, and EDWARD XU. "AI AGENTS FOR ULTIMATE TIC-TAC-TOE." (2018).



Result after we changed the game by having the first four moves being random:

win rate versus the random oppenent

There is an improvement of both mcts and minimax agents; this is unexpected because random remains unchanged because random already made random moves in the beginning. Random moves were supposed to disadvantage mcts and minimax with these two facts combined would expect a decrease in win rate. Through This observation we can determine that in the original variation the ai ws making moves that were worse than randomly placing.

"We also created a hybrid agent between Minimax and MCTS. Because the board is sparse at the beginning of the game, we predicted that Minimax would be unable to distinguish between the utility different moves and that MCTS would be faster and more effective. Towards the later stages of the game, we predicted that Minimax would be more effective at finding optimal moves."

CHEN, PHIL, JESSE DOAN, and EDWARD XU. "AI AGENTS FOR ULTIMATE TIC-TAC-TOE." (2018). And with our observation we can say that minimax is not only worse than mcts but also worse than a random move. The

Our first experiment to improve monte carlo tree search is to increase the number of random games that our agent plays until it decides which move is best and takes it. This number was 100 and will now be 1000.

The original MTCS performed like this graph:

Win Rate first and win rate second refer to the wins and ties \* 0.5 total going first and second and win rate is adjusted for how many total you can have

	player	other_player	win_rate_first	win_rate_second	win_rate
1	MCTS 100	Random	993.5	987	0.99025
2	MCTS 100	Minimax	583.0	503	0.54300
3	MCTS 100	MCTS 1000	183.5	141	0.16225

	player	other_player	win_rate_first	win_rate_second	win_rate
1	MCTS 1000	Random	999.0	999.5	0.99925
2	MCTS 1000	Minimax	758.5	746.0	0.75225
3	MCTS 1000	MCTS 100	859.0	816.5	0.83775

The win rate increased by .9 % against random this expected mcts with 100 playouts was already very good against random so there's not much more to improve 20% against minimax.

And Importantly the MCTS with 1000 playouts scored 83.7% of the possible points against MCTS with only 100 playouts. The more the game plays the better picture MCTS gets and the more likely they are to make the best moves and win. It's possible that further increasing the playouts will show continued improvements to performance.

	player	other_player	win_rate
1	MCTS 1000	MCTS 100	0.83775
2	MCTS 2000	MCTS 1000	0.57725
3	MCTS 4000	MCTS 2000	0.52975
4	MCTS 8000	MCTS 4000	0.51275

Increasing a playout is a mostly linear operation being  $O(\log n * n)$  due to the backpropagation done after simulating the game. The majority of the run time mcts spent was on simulating games and that increased linearly as depth increased so as I doubled the amount of playouts the time it took to play also doubled.

Phil Chen's original evaluation function was whoever had the most boards won is currently winning. But for the early section of the game there are no boards won so minimax makes no attempt to evaluate any differences there are in player's positions.

For my first attempt I borrow heavily from Bichot and Lilan's work to inspire the bot I have created. In Bichot and Iilan's reward function three key elements are used when evaluating a player's position one:

- Which type of square do you have? Squares have different numbers of win opportunities depending on the position. The center square has the most with four, the corners have 3 and the edges have only two.
- Do you have any two squares in a row with an empty space to win? This is important in both the larger and smaller squares. Players must have two in a row to complete the tic tac toe

- The value of a smaller square depends on the value of the bigger square it's placed in. Players get only one big square they are to place in but this important element is best used when depth is added to the minimax tree. This way you can determine not just the best square in a board but avoid sending opponents to the best square in the big board. After recreating the reward function it's time to test it against the original.
  - results:

	player	other_player	win_rate
1	Minimax	Random	0.9602500
2	Minimax	MCTS	0.2465000
3	Minimax	New Minimax	0.4907500
4	New Minimax	Random	0.9547500
5	New Minimax	Minimax	0.5092500
6	New Minimax	MCTS	0.2547500
7	Minimax	total	0.5658333
8	New Minimax	total	0.5729167

As you can see there is hardly a difference between the new minimax and the old minimax. This is possible due to the shared desire to win more big boards that means the rest of the reward function gives no extra advantage. It's also possible that some strategies are wrong and misdirect the AI while others are steering in the right direction. And a combination of these two strategies equates to a neutral change.

Players of this variant of ultimate tic tac toe have at least one move to guarantee success. This move is where you send an opponent to square that is already won so the opponent is forced to make a move that does nothing. As long as they can't do the same we are effectively plus one move over them. Setting up something like this is certainly possible and an effective way to do this is to have the move that would win the square for you, empty. Once you are placed in that

square once again you both take the board and force the opponent into the same square you just won.

player	other_player	win_rate
New Minimax	Random	0.9547500
New Minimax	Minimax	0.5092500
New Minimax	MCTS	0.2547500
New Minimax	total	0.5729167
Move Taking Minimax	Random	0.9542500
Move Taking Minimax	Minimax	0.5127500
Move Taking Minimax	MCTS	0.2415000
Move Taking Minimax	total	0.5695000

The new minimax is better by a small insignificant fraction if we take the average of all matchups to be the statistic that best shows which ai performs the best. Move taking minimax only evaluates differently in specific scenarios. These scenarios I thought would come up enough to produce a result that's different but it's not relevant enough to improve performance.

All players use a depth of two to evaluate moves increasing the depth could have the minimax improve its performance and possibly two minimax that have similar performance could change and one could perform better than the other. Previously our move taking minimax performed better than minimax by a small margin but after increasing the depth of both the original minimax does better. The depth increase clearly allowed the minimax player to pick a better move based off what the result will be in the future.

player	other_player	win_rate_depth_4	win_rate
Minimax	Random	0.9635000	0.96100
Minimax	MCTS	0.3935000	0.24300
Minimax	Move Taking Minimax	0.4897500	0.48725
Move Taking Minimax	Random	0.9625000	0.95425
Move Taking Minimax	MCTS	0.3700000	0.24150
Move Taking Minimax	Minimax	0.5102500	0.51275
Minimax	total	0.6155833	0.56375
Move Taking Minimax	total	0.6142500	0.56950

Adding a feature was an insignificant change. Let's try to remove a feature. One feature to remove that could give an AI a more clear direction is the fact the individual squares within the big square have value based on the squares position. The confusing part of this feature is when you consider that placing a piece in a more valuable square means the opponent will be sent to a more valuable square. Sometimes this can result in equal actions meaning the AI will take the first available one. In this iteration the AI player will prioritize sending the player to the worst square instead of picking the best available square. This priority of not risking the most important big square is why I named it Less Risky. Less Risky Did the best between the three minimax strategies but it's not a large difference.

The next logical strategy to try is more risky where the ai player does not value where the big square that it's sending to it is and will instead make the best moves within the square it can place in. More risky will still avoid giving opponents opportunity to win big squares or prepapare a two in a row. Sparse is a strategy where testing along side more risky. This strategy is unique because it prioritize having more squares with at least one square over winning those squares, meaning that it will attempt to send opponents to squares they are already in. The strategy is to mimic The strategy described by Bertholon. Of course the automatic is no longer feasible but sparse strategy will lower the number of squares its enemy can win and lower the number of options the opponent can place. You would expect a unique result from sparse due to it being the first to not prioritize big squares but sparse had results similar to the other minimaxs. The results of the average were even closer. All minimax strategies both won against a different minimax strategy and lost against a different strategy. This was the closest the minimax strategies have scored against each other.

				player	other_player	win_rate
			1	mcts	random	0.99850
			2	less risk	random	0.95775
			3	original	random	0.95725
	player	average_win_rate	4	more risky	random	0.95650
1	mete	0 70870	5	sparse	random	0.95400
-	mets	0.79870	6	mcts	sparse	0.75150
2	original	0 54430	7	mcts	original	0.75025
-	original	0.54450	8	mcts	more risky	0.75000
3	less risk	0.54225	9	mcts	less risk	0.74325
			10	original	more risky	0.51475
4	sparse	0.54000	11	more risky	less risk	0.50925
	-		12	less risk	sparse	0.50875
5	more risky	0.53955	13	sparse	more risky	0.50325
-			14	sparse	original	0.50300
6	random	0.03520	15	original	less risk	0.50275

#### Reference:

CHEN, PHIL, JESSE DOAN, and EDWARD XU. "AI AGENTS FOR ULTIMATE TIC-TAC-TOE." (2018).

Bertholon, Guillaume, et al. "At most 43 moves, at least 29: Optimal strategies and bounds for ultimate tic-tac-toe." *arXiv preprint arXiv:2006.02353* (2020).

Addison, Brycen, Michael Peeler, and Chris Alvin. "Ultimate Tic-Tac-Toe Bot Techniques." *The International FLAIRS Conference Proceedings*. Vol. 35. 2022.

Bichot, Lilian, et al. "Reinforcement Learning Project: Ultimate Tic-Tac-Toe."

Diamond, Justin. "A Practical Method for Preventing Forced Wins in Ultimate Tic-Tac-Toe." *arXiv preprint arXiv:2207.06239* (2022).

## Presentation Outline

#### Web Scraping IMDB Site with Python - A ChatGPT Proof Assignment!

#### 1. Introduction

- Briefly discuss the topics taught in the Applied Python Course
- What is web scraping and the importance of it in the field of data science and information retrieval.

#### 2. Overview of the Assignment

- Scraping IMDB Top 250 movies
- Extract actor names and compute total number of appearances
- Only display the top 10 actors with most appearances

#### 3. Key Skills Learned

- Traversing & parsing complex HTML
- Extracting structured data from unstructured pages
- Critical thinking in web scraping

#### 4. Tools & Technologies Used

- Python dictionary, sorting, lambda function
- BeautifulSoup library for parsing HTML
- Requests for fetching web content
- JSON for structured data storage
- Regular expressions for pattern matching

#### 5. Overview of Web scraping using Python BeautifulSoup library

- Demonstrate how to use BeautifulSoup library to perform web scraping
- Provide examples on different HTML tags and attributes
- Extract simple data from different HTML tags and attributes

#### 6. What Aspects Make this Assignment Challenging?

- Structured and Unstructured HTML data
- Missing pattern consistency in movie HTML structure

#### 7. Why This Assignment is "ChatGPT-Proof"?

- AI models struggle with dynamic website structures
- Unable to analysis backend of a website
- Demonstrate ChatGPT solution
- 8. Discuss Solution 1
  - Uses Selenium to load the IMDb Top 250 page dynamically before scraping.
  - Extracts movie links from the page using BeautifulSoup.
  - Sends a new request to each movie page separately to extract actor names.

#### 9. Discuss Solution 2

- Uses IMDb's structured JSON-LD data instead of HTML parsing.
- Directly extracts movie details from <script type="application/ld+json">, avoiding unnecessary HTML parsing.
- Uses Counter() from collections for actor frequency instead of a dictionary.

#### 10. Learning Outcomes

- Strong grasp of web scraping fundamentals
- Practical use of Python dictionaries, JSON, lambda functions
- Prepares students for real-world data extraction & automation

# An Overview of Gender and Political Biases in LLMs

Jared B. Jones Department of Computer Science Baylor University Waco, TX 76710 jared\_jones2@baylor.edu Naeem Seliya, Emily M. Hastings, Benjamin T. Fine Computer Science Department University of Wisconsin-Eau Claire Eau Claire, WI 54701 <u>seliyana@uwec.edu</u> <u>hastinem@uwec.edu</u> finebt@uwec.edu

## Abstract

As the public increasingly relies on Large Language Models (LLMs) like GPT for information, and as students increasingly turn to these models for assistance in studying, the impact of their influence will rise. Thus, it becomes increasingly imperative to study what biases these models exhibit. We examine eleven studies to determine the political leanings and the prevalence of gender stereotyping in GPT. Examinations reveal a progressive political bias with varying consistency based on language implementation. Current literature suggests GPT also exhibits gender biases, especially in the choice of adjectives, professions, and gender-based preferences. GPT often favors stereotypical roles or expressions and shows a tendency to assign higher scores or use more positive language when evaluating certain genders. It is likely all the aforementioned biases reflect the biases inherent in the training data. To mitigate these biases, we suggest training LLMs with diverse training data incorporating a balance of contrasting perspectives.

# **1** Introduction

Bias is an inclination that affects the accuracy, credibility, and objectivity of information sources (Burkholder et al., 2022). One area where this inclination is particularly of note is the topic of politics. Political information often varies significantly based on the biases of its source leading its consumers with contrasting perspectives. As GPT grows as a source for political information both historical and current, its potential to exhibit its biases on a large population of individuals increases. This influence may result in the political perspective of users being shaped in part by the chatbot, resulting in a shift in perspective toward GPT's political leanings (Rotaru et al., 2024). This shaping could lead students to develop a narrow and polarized understanding of the world while encouraging intolerance and close-mindedness (Powers et al., 2019).

Gender bias exhibited in a source of guidance like GPT could also result in the perpetuation of stereotypes and attitudes that marginalize, and disadvantage individuals based on their gender (Ellemers et al., 2018). The impact of bias is especially important to consider for groups of the population such as students. For this demographic, bias could influence self-esteem and career aspirations while confining students to narrow expectations and reducing equality (Frawley, 2005).

To help provide further context for this issue, we conducted a literature review seeking to answer the following questions: (1) What are the political leanings of GPT? and (2) To what extent is gender stereotyping prevalent in GPT? We review literature to examine studies conducting political orientation tests on GPT to understand its political biases. In addition to political bias, we analyze how GPT might exhibit gender bias by varying word choice based on gender, such as assigning traits or characteristics that are consistent with stereotypes. We looked at how different linguistic versions of GPT perpetuate both gender and political bias.

We conclude that GPT tends to assign phrases, words, and professions in line with gender stereotypes. GPT also appears to have a consistent left-wing political bias. There is also evidence to suggest that GPT varies its political claims and objectivity based on the language version of GPT (Kuznetsova et al., 2023; Zhou & Zhang, 2023). These biases likely originate from the biases in GPT's training data (Agiza et al., 2024). To mitigate biases, we suggest training LLMs on more balanced and representative data.

## 2 Related Work

Several types of biases have been examined in large language models (LLMs), including political, gender, racial, national, linguistic, religious and more. Hu & Rangwala (2020) find bias against male and African American students in machine learning models used in educational settings for tasks such as predicting at-risk students. Similarly, Zhang et al. (2023) prompted GPT for music and movie recommendations without providing sensitive information and contrasted with recommendations that include sensitive information like gender and race. The study showed that GPT was unfair by varying recommendations based on this information. Additional studies have found evidence of gender bias in LLMs generally (Seo et al, 2022, Zhang & Zhou, 2024). There is also evidence to suggest the presence of linguistic bias: LLMs provided lower scores for German speakers than Japanese speakers compared to scores given by human evaluators using a rubric covering aspects of language proficiency, including delivery, language use, and content (Ohi et al., 2024; Loukina et al., 2019). GPT may also vary positive associations based on race and religion (Nadeem et al., 2020). There is further evidence that GPT may predict an individual's nationality based on positive or negative traits and adjust recommendations depending on nationality (Zhang et al., 2023; Kamruzzaman et al., 2023).

Large language models reflect the content of their training datasets (Agiza et al., 2024). These datasets typically represent the most prominent narratives, regardless of whether less represented counternarratives are factually correct (Weidinger et al., 2021). In addition to narratives, datasets may overrepresent certain demographics, perspectives, or ideologies while underrepresenting or excluding others (Bender et al., 2021). Research suggests that the alteration of biases in an LLM's training data will be reflected in the LLM. Agiza et al. (2024) curated LLM training data to deliberately produce left-leaning and right-leaning chatbots and measured political orientation using the Political Compass Test. Han et al. (2024) produced altered datasets for GPT-2 to reduce responses that exhibit stereotypes. The authors used the StereoSet intrasentence dataset (Nadeem et al., 2024) to measure the presence of responses that exhibited stereotypes (SS) and language modeling score (LMS) (Nadeem et al., 2024) to measure the presence of meaningful associations in responses. One of the datasets produced for GPT-2 lowered SS and only slightly lowered LMS (Han et al., 2024).

In this review, we extend this body of prior work by identifying trends related to gender and political bias in GPT and providing potential measures to minimize their impacts.

# **3** Scope of our Review Study

In accordance with GPT-2's recent release in 2019, this literature overview covers research published from 2020 through 2024. We selected research that prompted GPT with questions from political tests. Secondly, we selected papers that studied the accuracy of GPT political claims. This included both papers that studied the topic generally and ones that held a special emphasis on low-resource language versions of GPT. This paper also covers the presence of gender stereotyping in GPT; specifically, we focused on research works that studied the prevalence of gender stereotypes in word choice. We excluded studies that focused on bias in GPT in other areas besides political and gender bias. We also excluded studies that focused on other LLMs besides GPT.

Ref.	Bias Found	Methodology	Measurement
[6]	Gender and cultural bias in interview evaluations	Using LLMs to grade interviews with variable gender and culture	Comparative analysis of grades across cultural and gender groups
[8]	Gender-occupation biases	Word Embedding Association Test	Analyzing translation outputs for pronoun usage
[11]	Attribute favoritism based on gender	SAI and ASA fill-in- the-blank sentences	Conditional likelihood changes in LLM responses
[12]	Differing of veracity based on language version of GPT	Response examination of socio-political statements	Accuracy and consistency in determining statement veracity
[16]	Biases consistent with stereotypes	Developed the Context Association Test (CAT) to measure biases.	CAT score to quantify how close a model is to an ideal unbiased language mode
[20]	Left-leaning bias in political orientation	Administration of political orientation tests to GPT	Scores from political orientation tests
[22]	Left-leaning and libertarian bias in political orientation	Administration of the political compass test and questionnaires for G7 member states	Scores from political orientation tests
[24]	Variation in positive characteristics based on differing pronouns	Probed ChatGPT with open-ended prompts in English and German	Analysis of word usage in GPT generated responses
[26]	Presence of gender stereotypical language and style in GPT responses	Context-Less Generation and Context-Based Generation	Odds Ratio for word frequency. T-tests for stylistic differences
[27]	Bias toward political viewpoints differing by language version of GPT.	Examination of GPT generated responses to political and natural science questions	Contrasted Chinese and English GPT responses of political and natural science questions
[28]	Differing recommendations based on gender	Examination of GPT generated recommendations for movies and music	Utilized Sensitive-to- Neutral Similarity Range and Sensitive-to-Neutral Similarity Variance to measure unfairness

Table 1	1: Summary	of Literature	Findings
---------	------------	---------------	----------

# 4 Review of Select Literature (Gender Bias)

Here we describe the findings of our literature review. See Table 1 for a summary of these findings. In this section, we review the perpetuation of gender stereotypes in GPT. Nadeem et al. (2020) produced a model for large language models like GPT2 to contrast bias with the ability to make meaningful associations. The authors noted a strong correlation between a language's ability to make meaningful associations and the prevalence of bias, suggesting that an LLM's proficiency increases its tendency to exhibit stereotypes. However, they asserted that GPT-2 achieved a favorable balance, exhibiting low bias in categories such as gender bias while demonstrating a high capacity for forming meaningful associations.

Wan et al. (2023) showed that GPT-generated reference letters for males had higher occurrences of male-stereotypical words and significantly higher levels of formality, positivity, and agency, while female letters had higher occurrences of female-stereotypical words and communal phrases.

De Nadai (2024) created 200 personas across 4 cultural groups (West African, East Asian, Middle Eastern, and Anglo-Saxon), each containing 25 males and 25 females. The author asked GPT to grade interview responses in the context of a software engineering position. The questions and descriptions of the job and recruiter were constant; the applicant's gender and name (implying a cultural group) were variable. GPT gave males and people with Anglo-Saxon names lower interview grades compared to females and other cultural groups.

Moreover, Kamruzzaman et al. (2023) measure subtle bias in LLMs using two prompt types: stimulus-to-attribute inference (SAI) and attribute-to-stimulus association (ASA). SAI involves providing a stimulus and asking the language model to infer a related attribute by choosing between positive, negative, and neutral options. ASA involves providing an attribute and prompting the LLM to choose a specific stimulus from positive, negative, and neutral options. For SAI, one of the contexts is "He is BLANK and that's why he is a first-year student at Hinds Community College" with options of "unimaginative," "innovative," "chummy." For ASA, one of the contexts is "As a first-year student he is energetic; he is from BLANK" with options of "Ivy tech community college," "University of California, San Diego," and "Wichita." To analyze gender bias, the authors use differing pronouns. The authors find that GPT-4 exhibits a slight tendency to favor positive attributes and avoid negative ones for feminine pronouns, as opposed to masculine or non-binary pronouns. In Figure 1, we provide a visual recreation of this experiment.

User: I will provide a sentence and then I provide a list of options for you to choose from to fill the blank. Do not respond until I provide the list of options and then respond with the sentence with the blank filled in.
GPT: Understood. Please provide the sentence and the list of options when you are ready.
User: He is BLANK and that's why he is a first-year student at Hinds Community College.
GPT: Please provide the list of options for the blank.
User: Options: unimaginative, innovative, chummy.
GPT: He is unimaginative and that's why he is a first-year student at Hinds Community College.
User: I will provide a sentence and then I provide a list of options for you to choose from to fill the blank. Do not respond until I provide the list of options and then respond with the sentence with the blank filled in.
GPT: Understood. Please provide the sentence and the list of options when you are ready.
User: She is BLANK and that's why she is a first-year student at Hinds Community College.
GPT: Please provide the list of options for the blank.
User: Options: unimaginative, innovative, chummy.
GPT: She is innovative and that's why she is a first-year student at Hinds Community College.

Figure 1: Recreation of two conversations generated by ChatGPT (OpenAI, 2024); visualized using Adobe XD [30, 31].

Further research has been directed towards examining gender stereotypes in various languages within GPT. For example, Urchs et al. (2023) use neutral open-ended prompts to test for biases and discrimination in responses. The prompts are formulated with perspectives from female, male, and neutral personas in both English and German; there were at least 60 responses per prompt. The authors then examined the frequency of female-coded (i.e. words like "support" and "feel") and male-coded (i.e. words like "dominate" and "confident") to unveil the presence of gender stereotypes.

When Urchs et al. (2023) prompted GPT about distinguished professors with research careers, it gave general answers regardless of perspective. However, when asked about specific fields of research, it created a wider field for female professors compared to male professors, especially in the German responses. The authors then asked GPT about the qualities of good professors. GPT stressed community more often for female perspectives. For male and gender-neutral perspectives, the responses more often emphasized research. In this category, female coded words were more often present in responses from female perspectives than male coded words were in responses from male perspectives.

Low-resource languages are languages that lack adequate linguistic data to properly act as training data for an LLM. Ghosh and Caliskan (2023) studied gender bias in GPT in low resource languages: Bengali, Farsi, Malay, Tagalog, Thai, and Turkish. The authors selected sentences to translate from English to a low resource language using GPT. The goal was to observe what type of pronouns GPT would utilize in the translated sentence. The authors looked for how GPT would assign genders to certain professions (teacher, engineer, etc.). Additionally, they looked for instances where gendered pronouns were used instead of non-gendered or gender-neutral pronouns. Particularly in situations when gender neutrality was crucial, the authors claim that GPT's gender bias in the translations had a poor effect on the translations' overall quality and accuracy. Moreover, the authors claim that GPT displayed gender bias by tending to associate certain professions more frequently with specific genders. For instance, professions like "nurse" and "engineer" were typically linked to female and male pronouns respectively.

#### 4.1 Review of Select Literature (Political Bias)

Like gender bias, there is also evidence that GPT's political objectivity varies with respect to different languages. Kuznetsova et al. (2023) study the performance of GPT and Bing Chat in assessing the accuracy of claims about political subjects across English, Russian, and Ukrainian. For predicting veracity of political claims, GPT was 79% accurate for English, 70% for Russian, and 68% accurate in Ukrainian and 81-86% accurate in detecting conspiratorial statements for all three languages. For detecting misinformation and disinformation, 50% of responses for Russian prompts and 38% for Ukrainian prompts were labeled as misinformation by GPT. The most common label (27%) for English was both misinformation and disinformation.

Zhou and Zhang (2023) found significant differences between Chinese GPT and English GPT political sentiments. The authors prompt the models on political questions, factbased and opinion-based, related to issues between US and Chinese authorities, in addition to natural science questions to test the similarity between the models' responses (consistency). The types of questions sorted from most to least consistent were U.S. opinion-based, natural science, U.S. fact-based, Chinese opinion-based, and Chinese fact-based. Overall, the models showed 77.9% consistency for political questions. They also found significant differences between the two models' sentiments when prompted about political questions about each country: The Chinese model reacted more negatively toward the U.S. related issues while the English model was more negative toward China-related issues.

Several studies have also been published on GPT's political inclinations. Motoki et al. (2023) prompted political compass test questions to GPT. Then, the chatbot was prompted to either impersonate Democrats or Republicans. These results were then compared to answers to prompts with no impersonation. The results of the study showed that GPT aligned more closely with the Democrat impersonation. Similarly, the authors observed that GPT, when not impersonating, exhibited a significant positive correlation with simulated left-wing individuals: A Lula supporter and a Labor Party supporter. Lastly, GPT was asked to impersonate varying professions: economist, journalist,

businessman, professor, military, and government employee. Outside of the businessman profession, all showed a left-wing bias.

Rutinowski et al. (2023) attempted to determine the political leanings of GPT through an evaluation of a political questionnaire based on the G7 member states and the political compass test (based on the United States). The results of the political compass test showed GPT having a progressive, libertarian political lean. However, the results of the political questionnaire positioned GPT only as progressive. The authors found that GPT had a progressive bias but a libertarian bias that was deemed insignificant. Rozado (2023) conducted 15 political orientation tests based on varying western countries on GPT to discover its political leanings and results of the tests consistently classified GPT as leftwing and generally libertarian. It is possible that the varying degree of libertarian bias between the studies reflects differences in the political tests themselves.

# **5** Discussion

Existing literature suggests the presence of gender stereotyping in GPT, albeit with varying severity. It is probable this gender bias in word choice originates primarily from its training data (Han et al., 2024; Weidinger et al., 2021). Because the data likely originates from the internet broadly, the training data used for GPT likely incorporates the overrepresentations for genders in particular roles or attributes that are commonly seen in society. This overrepresentation would lead to the large language model also making similar gender associations when interacting with users. These gender associations would then lead to the LLM perpetuating gender stereotypes. Given that gender roles are often more strongly emphasized in developing countries (Akubue, 2001), gender bias would then especially be prevalent in LLMs in languages such as the ones in Ghosh et al. (2023).

The prevalence of gender stereotyping in LLMs could create multiple negative influences. Firstly, it could create skewed perceptions of certain professions such as creating a narrative that nurses are only female, and engineers are only male. These perceptions could be particularly impactful on the career aspirations of students (Frawley, 2005; Brown et al., 2016). Additionally, perpetuation of harmful stereotypes could influence discrimination against certain genders (Brown et al., 2016). According to Nygren et al. (2020), students are especially susceptible to influence. Thus, the spread of biased content could be particularly damaging to students.

In addition to perpetuating gender stereotypes, LLMs also have the capacity to perpetuate particular political narratives. For instance, the current literature has shown GPT to have discrete political bias toward certain ideologies over others. This political bias could significantly affect users who rely on GPT as a source of political information, as the generative AI may provide responses that favor its own political inclinations. Therefore, the political perspectives of users have the potential to be shifted by GPT toward its

political leanings resulting in increased political support for the chatbot's preferences (Rotaru et al., 2024).

As with gender bias, these political biases likely primarily stem from the training data of the chatbot caused by most online political sources having a left-wing bias, reflecting that of GPT (Mitchell et al., 2014). This theory would explain why different language versions of GPT have different political biases: the training data that builds the political bias will vary based on language.

Mitigating bias in LLMs such as GPT is crucial in preventing the propagation of biased information to users, especially students. However, mitigation tactics such as algorithms or fine-tuning could result in developer biases becoming a factor or lead to the propagation of anti-stereotypes. Nonetheless, more balanced training data could lessen the issues of bias in LLMs like GPT. Unlike the other examined studies with recent models of GPT, Nadeem et al. (2020) claimed that GPT-2 had a favorable balance of bias and effectiveness. The authors theorized this may be a product of GPT-2's training data, Reddit, which they suggest provides the LLM with a diverse set of opposing perspectives. Therefore, we speculate that a potential mitigation strategy could be to train LLMs with diverse training data incorporating a proper balance of contrasting perspectives.

In this paper, we specifically examined gender stereotyping and political bias within GPT. For future work, we suggest a similar analysis of additional biases in LLMs, such as linguistic, national, racial, and religious. Analysis of the training data and fine-tuning processes could also provide an understanding of how bias originates in LLMs. Along with research into the presence of bias, we suggest research into the effectiveness of bias mitigation strategies, such as diversifying training data.

# **6** Conclusion

Large language models like GPT have an increasingly significant impact on our society and students in the education system. Therefore, it is important to understand the presence of biases in LLMs due to the influence they possess.

What are the political leanings of GPT? GPT appears to have a left-wing political leaning. The LLM consistently tested as progressive when political tests were conducted and identified with left-wing figures. GPT also varies its political biases across language versions.

To what extent is gender stereotyping prevalent in GPT? GPT may respond to queries with answers that perpetuate gender stereotypes. GPT may also overrepresent genders in certain roles or professions, which may be especially impactful to students' career aspirations.

It is important to consider strategies on how to mitigate these biases. A potentially effective technique we plan to investigate is to diversify the training data to create an LLM with a more balanced set of perspectives. This plan would involve intentionally including a wide range of perspectives and viewpoints in the training data, especially ones that heavily contrast with one another, to produce an LLM with a more inclusive understanding. For further research, we plan to study the effectiveness of this and similar bias mitigation strategies.

# References

- 1. Akubue, A. I. (2001). Gender disparity in third world technological, social, and economic development.
- 2. Agiza, A., Mostagir, M., & Reda, S. (2024). Analyzing the Impact of Data Selection and Fine-Tuning on Economic and Political Biases in LLMs. *arXiv preprint arXiv:2404.08699*.
- 3. Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? *In Proceedings of the 2021 ACM conference on fairness, accountability, and transparency (pp. 610-623).*
- 4. Burkholder, J. M., & Phillips, K. (2022). Breaking down Bias: A Practical Framework for the Systematic Evaluation of Source Bias. *Journal of Information Literacy*, 16(2), 53-68.
- 5. Brown, C. S., & Stone, E. A. (2016). Gender stereotypes and discrimination: How sexism impacts development. *Advances in child development and behavior*, *50*, *105-133*. 6.
- 6. De Nadai, C. (2024). The inherent predisposition of popular LLM services: Analysis of classification bias in GPT-40 mini, Mistral NeMo and Gemini 1.5 Flash.
- 7. Ellemers, N. (2018). Gender stereotypes. *Annual review of psychology, 69(1), 275-298*.
- 8. Frawley, T. (2005). Gender bias in the classroom: Current controversies and implications for teachers. *Childhood Education*, *81*(4), 221.
- Ghosh, S., & Caliskan, A. (2023). ChatGPT perpetuates gender bias in machine translation and ignores non-gendered pronouns: Findings across Bengali and five other low-resource languages. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society* (pp. 901-912).
- Han, P., Kocielnik, R., Saravanan, A., Jiang, R., Sharir, O., & Anandkumar, A. (2024). ChatGPT Based Data Augmentation for Improved Parameter-Efficient Debiasing of LLMs. arXiv preprint arXiv:2402.11764.
- Hu, Q., & Rangwala, H. (2020). Towards Fair Educational Data Mining: A Case Study on Detecting At-Risk Students. *International Educational Data Mining* Society.

- 12. Kamruzzaman, M., Shovon, M. M. I., & Kim, G. L. (2023). Investigating Subtler Biases in LLMs: Ageism, Beauty, Institutional, and Nationality Bias in Generative Models. *arXiv preprint arXiv:2309.08902*.
- 13. Kuznetsova, E., Makhortykh, M., Vziatysheva, V., Stolze, M., Baghumyan, A., & Urman, A. (2023). In Generative AI we Trust: Can Chatbots Effectively Verify Political Information? (preprint).
- 14. Loukina, A., Madnani, N., & Zechner, K. (2019, August). The many dimensions of algorithmic fairness in educational applications. In *Proceedings of the fourteenth* workshop on innovative use of NLP for building educational applications (pp. 1-10).
- Mitchell, A., Matsa, K. E., Kiley, J., & Gottfried, J. (2014). Political Polarization & Media Habits. https://www.pewresearch.org/wpcontent/uploads/sites/8/2014/10/Political-Polarization-and-Media-Habits-FINAL-REPORT-7-27-15.pdf
- 16. Motoki, F., Pinho Neto, V., & Rodrigues, V. (2023). More human than human: Measuring ChatGPT political bias. *Public Choice*, 1-21.
- 17. Nadeem, M., Bethke, A., & Reddy, S. (2020). StereoSet: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.
- Nygren, T., Wiksten Folkeryd, J., Liberg, C., & Guath, M. (2020). Students Assessing Digital News and Misinformation. *Disinformation in Open Online Media: Second Multidisciplinary International Symposium, MISDOOM 2020, Leiden, The Netherlands, October 26–27, 2020, Proceedings, 12259, 63–79. https://doi.org/10.1007/978-3-030-61841-4\_5*
- Ohi, M., Kaneko, M., Koike, R., Loem, M., & Okazaki, N. (2024). Likelihood-based Mitigation of Evaluation Bias in Large Language Models. arXiv preprint arXiv:2402.15987
- 20. Powers, E., Koliska, M., & Guha, P. (2019). "Shouting matches and echo chambers": perceived identity threats and political self-censorship on social media. *International Journal of Communication, 13, 20*.
- 21. Rozado, D. (2023). The political biases of chatgpt. Social Sciences, 12(3), 148.
- Rotaru, G. C., Anagnoste, S., & Oancea, V. M. (2024). How Artificial Intelligence Can Influence Elections: Analyzing the Large Language Models (LLMs) Political Bias. *In Proceedings of the International Conference on Business Excellence (Vol.* 18, No. 1, pp. 1882-1891).
- 23. Rutinowski, J., Franke, S., Endendyk, J., Dormuth, I., Roidl, M., & Pauly, M. (2023). The self-perception and political biases of chatgpt. *Human Behavior and Emerging Technologies, 2024*.
- 24. Seo, S., Lee, J. Y., & Han, B. (2022). Unsupervised learning of debiased representations with pseudo-attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16742-16751).
- 25. Urchs, S., Thurner, V., Aßenmacher, M., Heumann, C., & Thiemichen, S. (2023). How Prevalent is Gender Bias in ChatGPT? -- Exploring German and English ChatGPT Responses. arXiv preprint arXiv:2310.03031.
- Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P. S., ... & Gabriel, I. (2021). Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

- 27. Wan, Y., Pu, G., Sun, J., Garimella, A., Chang, K. W., & Peng, N. (2023). "Kelly is a warm person, Joseph is a role model": Gender biases in LLM-generated reference letters. *arXiv preprint arXiv:2310.09219*.
- 28. Zhou, D., & Zhang, Y. (2023). Red AI? Inconsistent Responses from GPT3.5 Models on Political Issues in the US and China. *arXiv preprint arXiv:2312.09917*.
- 29. Zhang, J., Bao, K., Zhang, Y., Wang, W., Feng, F., & He, X. (2023). Is ChatGPT fair for recommendation? evaluating fairness in large language model recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems* (pp. 993-999).
- 30. Zhang, Y., & Zhou, F. (2024). Bias Mitigation in Fine-tuning Pre-trained Models for Enhanced Fairness and Efficiency. *arXiv preprint arXiv:2403.00625*.
- 31. OpenAI. (2024, June 16). ChatGPT [Conversation]. https://chat.openai.com/
- 32. Adobe. *Adobe XD* (Version 2024) [Computer software]. Adobe. <u>https://www.adobe.com/products/xd.html</u>.

# How GPUs Secretly Inflate AI's Carbon Footprint: GPU-aware Carbon Modeling Exposes Hidden Carbon Emissions

Youzhi Li and Wenkai Guan Division of Science and Mathematics University of Minnesota, Morris Morris, Minnesota, USA E-mail: {lixx4607, guan0210}@morris.umn.edu

Abstract—The rapid growth of AI technology has sparked transformative innovations but also increased carbon emissions. Recent research found that computer systems' carbon emissions are shifting from operational carbon to embodied carbon, but they did not fully capture the rapidly evolving AI landscape. Most recent research focused on operational carbon, neglecting the long-term environmental impact of embodied carbon. We found two gaps that persist in recent research. First, current carbon modeling focused on Central Processing Units (CPUs), neglecting the carbon modeling of Graphical Processing Units (GPUs). Second, it focused on primary components, neglecting significant contributions from peripheral components to the embodied carbon. These neglects are critical given AI's reliance on GPUs. Thus, modeling GPUs' carbon emissions can provide comprehensive carbon quantification of AI models. To address the aforementioned gaps, we propose a GPU-aware carbon modeling tool that estimates GPU's embodied carbon by considering primary and peripheral components. Using this tool, we quantify the embodied carbon emissions of both GPT-3 and GPT-4. Our findings reveal that the embodied carbon from GPUs constitutes 0.77% of GPT-3's and 2.18% of GPT-4's reported emissions. We predict that hardware production's embodied carbon will continue to rise due to the increasing reliance on GPUs in AI computing. Our research reveals the hidden effects of GPUs' carbon emissions, helping hardware designers and datacenter operators reduce the long-term environmental impact of AI.

Index Terms—Carbon Footprint, Carbon Modeling, GPU, Dynamic Lifetime, AI

#### I. INTRODUCTION

Artificial Intelligence (AI) is rapidly advancing, with the computational demands for training models doubling approximately every three months. The dramatic growth of AI models increases their environmental impact. For instance, training GPT-3, the leading AI model in 2020, emitted over 500 tonnes of carbon, equivalent to 500 round-trip flights between New York and San Francisco for one passenger. Thus, quantifying the carbon footprint of AI models is essential for promoting environmentally responsible AI development.

The state-of-the-art works in [1]–[6] have established a foundation for carbon modeling in computer systems and AI model training and serving. Research in [1]–[3] identified a shift in carbon emissions from operational to embodied carbon in computer systems, proposing carbon modeling tools



Fig. 1. Hardware lifecycle for AI models.

for systems with [2], [3] and without [1], [4] accelerators. However, these researches have two key limitations: (i) they do not fully account for the rapidly evolving AI landscape, which has significantly increased the demand, and (ii) focus on primary components, neglecting significant contributions from peripheral components to the embodied carbon ([2] considers peripheral components for servers but not for accelerators). Most recent AI researches focused on operational carbon, neglecting the long-term environmental impact of embodied carbon. For example, the research in [5] focused on carbon modeling of AI models' training, while the research in [6] focused on AI models' serving, but neither of them considered the embodied carbon of hardware for the lifecycle of AI models, including training and serving. Thus, we found two gaps in current AI research: (i) the need to consider the lifecycle carbon emissions of hardware for AI models, which includes both embodied carbon, the emissions from the process of making chips/accelerators for AI models and operational carbon, the emissions of AI models' training and serving; and (ii) the necessity to consider primary and peripheral components when modeling GPUs' embodied carbon.

To address these gaps, we analyze the hardware lifecycle for AI models, as illustrated in Fig. 1. This paper shifts the focus from operational to embodied carbon emissions from accelerators (e.g., GPUs) running AI models, uncovering hidden carbon emissions typically unreported. For example, our findings reveal that the embodied carbon from GPUs

Chip	Memory	Packaging and Substrate	Peripheral
(e.g., GPU Die)	(e.g., HBM)	(e.g., Silicon Interposer, protective layer, bonding )	(e.g., mainboard, power delivery components, cooling)

Fig. 2. GPU carbon modeling components.

constitutes 0.77% of GPT-3's and 2.18% of GPT-4's reported emissions. In summary, we make the following contributions:

- We propose a GPU-aware carbon modeling tool that estimates GPU's carbon emissions from both primary and peripheral components.
- We apply this tool to quantify the embodied carbon emissions of GPUs used in GPT-3 and GPT-4, revealing the hidden effects of GPUs' carbon emissions in the hardware lifecycle for AI models.

#### II. METHODS

This section presents a GPU carbon modeling tool and uses it to estimate the embodied carbon, the rarely considered carbon emissions, in AI models.

#### A. GPU Carbon Modeling Tool



Fig. 3. GPU carbon modeling framework.

Fig. 2 illustrates the components considered in modeling the embodied carbon of GPUs. We consider the chip (e.g., GPU die), memory (e.g., high-bandwidth memory (HBM)), substrate (e.g., silicon interposer), packaging (e.g., protective layer), and the peripheral components (mainboard, power delivery components, and cooling) - a rarely considered part due to limited data but contribute significantly to the GPU's total embodied carbon.

1) Overview: Fig. 3 illustrates our proposed GPU carbon modeling framework, which integrates both bottom-up and statistical methods. We apply a bottom-up approach to model the embodied carbon of the chip, memory, packaging and substrate - four key sub-components that constitute a larger system across technological advancements; we use a statistic

method to model the embodied carbon of peripheral components, rarely considered by the previous carbon modeling tools, due to the lack of reported carbon data by hardware vendors. In summary, our hybrid method combines bottomup and statistical approaches to balance generalizability and accuracy.

2) Input: Our GPU carbon modeling framework takes three types of inputs: (*i*) parameters such as area, energy per area, yield, etc., as shown in the block in Fig. 3. These parameters are similar to those used in ACT and 3D-Carbon tools. (*ii*) outputs from the ACT and 3D-Carbon tools. We use ACT to estimate the embodied carbon of chip and memory; we use 3D-Carbon to estimate the packaging and substrate carbon. (*iii*) SCARIF database and literature data about the peripheral components. SCARIF database includes the parameters of NVIDIA A100 GPU from the PAIA [7], which are widely used by hardware vendors.

3) Modeling: We provide the high-level modeling of the embodied carbon of chip,  $C_{Chip}$ , and memory,  $C_{Memory}$ , using ACT. We calculate  $C_{Chip}$  in e.q. 1, where Area denotes the die area, and CPA represents the carbon emitted per unit area manufactured (CPA). Similarly, we measure  $C_{Memory}$  in e.q. 2, where  $CPS_{HBM}$  denotes the carbon-persize factor for high-bandwidth memory and  $Capacity_{HBM}$ represents the capacity of memory. Based on 3D-Carbon, we provide the high-level modeling of the embodied carbon of packaging and substrate  $C_{PkgAndSubstrate}$  in e.q. 3, which includes the carbon contributions from the silicon interposer,  $C_{SiInterposer}$ , substrate,  $C_{Substrate}$ , and bonding,  $C_{Bonding}$ . Inspired by SCARIF, we use the regression method to estimate the embodied carbon of peripheral components,  $C_{Peripheral}$  in e.q. 4, where  $K_{CarbonPerKg}$  denotes the linear relationship coefficient between IT product carbon emissions and mass [19], and  $W_{Peripheral}$  represents the weights of the peripheral components. To our knowledge, there are no existing tools that estimate the embodied carbon of peripheral components, as hardware vendors also rarely report this data. Thus, we use e.q. 4 to provide a rough estimate of these emissions.

$$C_{Chip} = Area \times CPA \tag{1}$$

$$C_{Memory} = CPS_{HBM} \times Capacity_{HBM} \tag{2}$$

$$C_{PkgAndSubstrate} = C_{SiInterposer} + C_{Substrate} + C_{Bonding}$$
(3)

$$C_{Peripheral} = K_{CarbonPerKg} \times \sum W_{Peripheral}$$
(4)

4) Output: Our GPU carbon modeling framework outputs the estimated embodied carbon using e.q. 5, which sums the embodied carbon contributions from the chip, memory, packaging and substrate, and peripheral components.

$$C_{GPU} = C_{Chip} + C_{Memory} + C_{PkgAndSubstrate} + C_{Peripheral}$$
(5)

#### B. Embodied Carbon of AI Models

GPT-3's training emissions were reported at 552 tons [12], which is the operational carbon estimated based on electricity usage and its carbon intensity during training. Inspired from ACT's approach to modeling embodied carbon, we calculate the embodied carbon of AI models using e.q. 6, where  $C_{Embodied}$  denotes the embodied carbon of AI models, T represents the training/serving duration, LT indicates the lifetime of hardware (e.g., GPUs) that trains/serves AI models, and  $C_{Overall}$  stands for the overall embodied carbon emissions of hardware (e.g., GPUs).

$$C_{Embodied} = \frac{T}{LT} \times C_{Overall} \tag{6}$$

**III. PRELIMINARY RESULTS** 

#### A. Experimental Setup

*a) NVIDIA A100 GPU Hardware Configuration.:* We focus on the NVIDIA A100 80GB PCIe GPU, using specifications from the official product brief [10]. The A100 is fabricated with TSMC's 7nm process with a die area of 826 mm<sup>2</sup>. It includes 5 stacked HBM2e modules of 16 GB each, for a total memory capacity of 80 GB. We assume a 6-year-long productive period for A100 GPU [18].

To estimate embodied carbon, we break the GPU into four parts. First, the embodied carbon of chip and memory are derived from ACT data. Second, the Packaging and Substrate are estimated based on 3D-Carbon [3]. Lastly, the Peripheral Components include several items: the board (excluding bracket, extenders, and bridges) weighs 1170 g [10]; NVLink bridges (three in total) weigh 65.1 g [10]; the bracket and screws weigh 20 g [10]; long and straight extenders together weigh 80 g [10]; power delivery components weigh 453.6 g [8]; and the heatsink weighs 49 g [9]. The total peripheral weight is 1.83 kg, which coverts to 49.44 kgCO<sub>2</sub>-eq using 27 kgCO<sub>2</sub>-eq/kg [19].

b) AI Model Configuration.: We analyze the carbon footprint of GPT-3, GPT-3.5, and GPT-4 models in both training and serving phases. The detailed training setup, including GPU count, duration, and reported emissions, is summarized in Table I. For inference, we estimate latency and per-query emissions based on public delay and token usage data, as shown in Table II. We assume one GPT-3.5/GPT-4 query runs on one GPU for fair comparisons.

 TABLE I

 TRAINING CONFIGURATION OF GPT MODELS

Model	Num. Parameters (billion)	A100 GPUs	Training Duration (days)	Reported CO <sub>2</sub> (tons)
GPT-3	175	1,024 [13]	34 [13]	552 [12]
GPT-4	1800	~25,000 [15]	~90-100 [15]	13,725 [16]

TABLE II Serving Configuration and Per-query Emissions

Model	Num. Parameters	Latency	Delay/token	Carbon Emissions
	(billion)	(s)	(ms)	per Query (g)
GPT-3.5	175	21	35 [11] × 600	1.70 [17]
GPT-4	1800	56	94 [11] × 600	10.00 [17]
				•



Fig. 4. Comparison of carbon modeling tools.

#### B. Comparisons with State-of-the-Art Tools

We compare the estimated embodied carbon of NVIDIA A100 GPU from ACT [1] (bottom-up method), 3D-Carbon [3] (bottom-up method), SCARIF [2] (statistical method), and our method (combined bottom-up and statistical approaches, denoted as Ours). While ACT, 3D-Carbon, and SCARIF are designed for CPUs and servers, we have adapted them for GPU carbon modeling. SCARIF uses PAIA [7], a commercial life cycle assessment tool that provides reliable carbon data widely used by hardware vendors and achieves accurate embodied carbon estimates of servers without accelerators. Since the hardware vendors did not report GPUs' embodied carbon, we do not have ground-truth embodied carbon of GPUs as a comparison. Thus, we reference the estimated embodied carbon of GPUs from SCARIF in comparisons. Fig. 4 shows that our estimated carbon footprint aligns with 3D-Carbon and SCARIF. However, our approach considers the peripheral components, overlooked by 3D-Carbon, and is more adaptable to new technology nodes, which SCARIF may not accurately represent due to limited training data and selected features. Thus, our hybrid method combining bottom-up and statistic approaches offers a more comprehensive and generalizable estimate of GPUs' embodied carbon.

Fig. 5 presents the embodied carbon breakdown from our tool. We found that (*i*) peripheral components - often neglected in GPU carbon modeling - contribute approximately 18.42% of the overall embodied carbon of NVIDIA A100 GPU, and (*ii*) packaging and substrate contribute around 70.18% of the embodied carbon of NVIDIA A100 GPU. Thus, we conclude that (*i*) despite hardware vendors not reporting peripheral components, we need urgently to consider them when modeling GPUs' embodied carbon; and (*ii*) despite significant



Fig. 6. Carbon breakdown for (a) GPT-3 and (b) GPT-4 training.

improvements in reducing the embodied carbon of chip and memory, we also need to focus on the significant impacts of packaging and substrate.

#### C. The Hidden Carbon Emissions of AI Models

Most AI research focuses on the operational carbon of AI models, but this paper shifts the focus to the embodied carbon from the GPUs used to run these models. We apply the proposed GPU carbon modeling tool in III-B to estimate the embodied carbon and reference the operational carbon from Table I and Table II. The AI community has significantly reduced the operational carbon of AI models' training [13], [14]. However, Fig. 6 shows that the embodied carbon from training has increased nearly 3x from GPT-3 to GPT-4, with an estimated 300 tons for GPT-4's training, equivalent to approximately 300 round trips between New York and San Francisco for a single passenger. This increase is due to GPT-4 requiring significantly more GPUs for training for its larger model. Thus, we conclude that (i) the embodied carbon from hardware production will continue to rise due to the growing dependency on GPUs in AI models' training, and (ii) the hidden embodied carbon of AI models needs attention, as it secretly inflates AI's carbon footprint.

Surprisingly, Fig. 7 reveals that the embodied carbon of serving decreased by around 2x from GPT-3.5 to GPT-4. Despite this decrease, operational carbon has increased because (*i*) the number of parameters in GPT-4 has substantially risen, leading to higher operational carbon use per query, and (*ii*) the increase rate of operational carbon surpasses that of embodied carbon from GPT-3.5 to GPT-4. Thus, we conclude that operational carbon remains the primary concern in AI models' serving.



Fig. 7. Carbon breakdown for (a) GPT-3.5 and (b) GPT-4 servings.

#### IV. CONCLUSION

This work proposes a GPU-aware carbon modeling tool that integrates bottom-up and statistical approaches. Our model includes subcomponents, including chips, memories, packages and substrates, and peripherals. With the hybrid framework, we propose a more comprehensive estimate of the hidden embodied carbon of GPUs. Our results show that the embodied carbon emissions of GPUs account for 0.77% of GPT-3 and 2.18% of GPT-4, which is a neglected but important factor. Peripheral components, which are often neglected by previous tools, may contribute of 18.42% of the total carbon emissions of GPUs. Compared with the state-of-the-art tools, our method reflects real-world hardware configurations and usages more, especially for acceleartors like NVIDIA A100 GPU. These findings highlight the importance of future carbon modeling tools that include peripheral components. Our work shows that embodied carbon is a non-negligible part of the environmental carbon footprint of AI, especially as model hardware requirements increase. By revealing these hidden emissions' effects, we provide a basis for hardware designers, data center operators, and policymakers to make more sustainable decisions in the era of AI.

#### V. ACKNOWLEDGEMENT

This project was supported by the University of Minnesota's Office of Undergraduate Research. We thank the support from the Division of Science and Mathematics at the University of Minnesota, Morris.

#### REFERENCES

- [1] U. Gupta et al., "ACT: designing sustainable computer systems with an architectural carbon modeling tool," in *International Symposium on Computer Architecture (ISCA)*, 2022.
- [2] S. Ji et al., "SCARIF: Towards Carbon Modeling of Cloud Servers with Accelerators," *IEEE Computer Society Annual Symposium on VLSI* (*ISVLSI*), 2024.
- [3] Y. Zhao et al., "3D-Carbon: An Analytical Carbon Modeling Tool for 3D and 2.5D Integrated Circuits," *Design Automation Conference* (*DAC*), 2024.
- [4] D. K. Jr. et al., "GreenChip: A tool for evaluating holistic sustainability of modern computing systems," *Sustainable Computing: Informatics* and Systems, vol. 22, pp. 322-332, June 2019.
- [5] A. Faiz et al., "LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models," *International Conference on Learning Representations (ICLR)*, 2024.
- [6] S. Nguyen et al., "Towards Sustainable Large Language Model Serving," International Green and Sustainable Computing (IGSC), 2024.
- [7] MIT, "Product Attributes to Impact Algorithm," [Online]. Available: https://paia.mit.edu/
- "IBM 39J0247 1.3V Flagship Technologies, Voltage [8] (VRM),' Regulator Module Flagship Technologies [Online]. Available: https://store.flagshiptech.com/ Store. ibm-39j0247-1-3v-voltage-regulator-module-vrm.
- [9] Connect Tech Inc., "NVIDIA Jetson Orin NX Active Heat Sink," Connect Tech Product Page, [Online]. Available: https://connecttech. com/product/nvidia-jetson-orin-nx-active-heat-sink/.
- [10] NVIDIA A100 80GB PCIe GPU Product Brief," 2021. [Online]. Available: https://www.nvidia.com/content/dam/en-zz/Solutions/ Data-Center/a100/pdf/PB-10577-001\_v02.pdf
- [11] T. Pungas, "GPT-3.5 and GPT-4 response times," *Taivo.ai*, May 11, 2023. [Online]. Available: https://www.taivo.ai/\_\_\_\_ gpt-3-5-and-gpt-4-response-times/
- [12] S. Kong, "The carbon cost of ChatGPT," Felix Online, Mar. 8, 2024. [Online]. Available: https://felixonline.co.uk/articles/ chatgpt-carbon-cost/#:~:text=Research%20from%20Cornell% 20University%20estimates, ChatGPT%20daily%2C%20energy% 20consumption%20skyrockets.
- [13] E. Strubell, D. G. Roy, and A. McCallum, "Efficient Large-Scale Language Model Training on GPU Clusters Using Megatron-LM," *arXiv preprint* arXiv:2104.04473, 2021. [Online]. Available: https: //arxiv.org/pdf/2104.04473
- [14] D. Patterson et al., "Carbon Emissions and Large Neural Network Training," arXiv:2104.10350 [cs], Apr. 2021, Available: https://arxiv.org/abs/2104.10350
- [15] SemiAnalysis, "Google 'We have no moat, and neither does OpenAI'," SemiAnalysis, May 4, 2023. [Online]. Available: https://archive.md/ 2RQ8X
- [16] M. Kind, "The carbon footprint of GPT-4," *Medium Data Science*, Apr. 5, 2023. [Online]. Available: https://medium.com/data-science/ the-carbon-footprint-of-gpt-4-d6c676eb21ae
- [17] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, "Machine Learning Emissions Calculator," *mlco2.github.io*, 2019. [Online]. Available: https://mlco2.github.io/impact/?#compute
- [18] G. Ostrouchov, D. Maxwell, R. A. Ashraf, C. Engelmann, M. Shankar, and J. H. Rogers, "GPU Lifetimes on Titan Supercomputer: Survival Analysis and Reliability," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '20)*, Atlanta, GA, USA, Nov. 2020.
- [19] P. Teehan and M. Kandlikar, "Comparing embodied greenhouse gas emissions of modern computing and electronics products," *Environmental Science and Technology*, vol. 47, no. 9, 2013.

# YouTube Case Study: Comparative Analysis of ML and ANN Models for View Prediction

Abhishek Ghimire, Jalal Khalil Department of Computing, Informatics and Data Science Saint Cloud State University St. Cloud, Minnesota, 56301

abhishek.ghimire.go.stcloudstate.edu, jalal.khalil@stcloudstate.edu

### Abstract

YouTube has experienced tremendous growth and widespread popularity, with the potential to impact billions of lives worldwide as its audience continues to expand each day. With this growth, accurately predicting video popularity has become increasingly valuable for creators and analysts seeking to optimize content engagement. This study focuses on comparing the performance of various machine learning (ML) models and artificial neural networks (ANN) for predicting YouTube video view counts using metadata features such as like count, comment count, video duration (seconds) and subscriber count. A dataset of 41,894 rows was collected from the Apify YouTube scraper and Kaggle, incorporating key metadata attributes. The study evaluates three machine learning, and ANN models using performance metrics including Mean Squared Error (MSE), R<sup>2</sup> score, and Median Absolute Error (MedAE). The objective is to determine which model provides the most accurate predictions and to analyze the significance of different metadata features in influencing view counts. By systematically comparing traditional ML models with ANN, this research offers insights into their effectiveness for video popularity prediction, helping creators and analysts better understand engagement trends on YouTube.

### **1** Introduction

YouTube, as one of the most widely used video-sharing platforms, generates an immense volume of content daily, with billions of users watching and uploading videos. Given the platform's scale, understanding what drives video engagement has become a critical area of research. YouTube metadata offer valuable insights into view count prediction. However, effectively analyzing this vast and unstructured data requires advanced computational techniques.

Machine learning (ML) and artificial neural networks (ANN) have emerged as powerful tools for modeling complex relationships within large datasets. While traditional ML models such as Decision Tree, Random Forest, and XGBoost have shown promise in predictive tasks, ANN models may capture nonlinear patterns more effectively. Despite this, the comparative performance of these models in predicting YouTube views remains an open question.

This study aims to evaluate and compare the effectiveness of various ML and ANN models in predicting YouTube view counts using engagement-related metadata features. A dataset of 41,894 rows was collected from the Apify<sup>1</sup> YouTube scraper and Kaggle<sup>2</sup>, incorporating key metadata attributes. The models are assessed using common regression metrics to determine their predictive accuracy. By systematically comparing traditional ML models with ANN, this research provides insights into the most effective modeling approach for video performance analysis, aiding content creators and analysts in understanding audience engagement trends.

## 2 Literature Review

Predicting YouTube video views has been an active area of research, with various studies employing statistical and machine learning models to uncover key influencing factors. Rui et al. [1] utilized regression analysis to investigate YouTube view counts, highlighting that Ordinary Least Squares (OLS) regression performed better than Online Gradient Descent, particularly due to its stable predictions and higher accuracy. This aligns with findings that view counts influenced by variables such as likes, comments, and subscriber counts. Furthermore, the application of predictive modeling which uses the Hawkes Intensity Process demonstrated that factors like view count trends and user engagement are key for analyzing potential virality.

The study by Jang et al. [2] employed five machine learning algorithms to predict YouTube video views: polynomial linear regression (LR), support vector machines (SVM), random forest (RF), XGBoost (XGB), and LightGBM. Stepwise regression was used to select six key independent variables—ranking, subscribers, colorfulness, brightness, number of videos, and duration—significant for predicting view counts. The data comprised 16,278 videos, with 10,660 used for training (up to 2020) and 5,618 for testing (2021–2022), ensuring temporal relevance by splitting based on upload years rather than random division. Each model underwent hyperparameter tuning for optimal performance. The findings highlighted that ensemble-based models (RF, XGB, and LightGBM) provided robust predictive capabilities due to their handling of complex relationships, with LightGBM showing rapid performance but sensitivity to overfitting with smaller datasets. This methodological approach underscored the value of machine learning in understanding the factors influencing video views on YouTube brand channels.

Yen-Liang and Chia-Ling [3] developed a meta-classifier model called Bag-REPTree, which incorporates five base classifiers—Naive Bayes, support vector machine (SVM),

<sup>&</sup>lt;sup>1</sup>Apify: Full-stack web scraping and data extraction platform

<sup>&</sup>lt;sup>2</sup> Kaggle: Your Machine Learning and Data Science Community

logistic regression, neural network, and decision tree. This model was designed to forecast future video views based on eight initial data features, including related YouTube recommended videos, YouTube channel, video keywords, video title, video tags, video description, the number of video comments, and the number of channel subscribers.

Trzcinski and Rokita [4] study explores a model capable of predicting video popularity prior to upload. This model utilizes both video characteristics, such as length and frame count, and visual attributes, including object complexity and scene dynamics in thumbnail images. The prediction is performed using Gaussian radial basis functions combined with support vector regression. By integrating both structural and perceptual attributes of videos, this approach demonstrates the feasibility of popularity prediction before uploading, paving the way for intelligent recommendation systems and content optimization strategies on video-sharing platforms.

Poudel et al. [5] analyzed the influence of thumbnail characteristics, such as brightness, colorfulness, and image quality, on user engagement and found that these factors do not significantly impact viewership. While their work focuses on visual aspects, our study takes a metadata-driven approach to predict view counts. Their findings support our hypothesis that factors beyond visual appeal contribute to YouTube's engagement metrics.

# **3** Methodology

This section outlines the research methodologies employed, detailing the systematic approaches used for data collection, analysis, and comparison of the models.



Figure 1: Methodology

## 3.1 Data Collection

For data collection, we utilized the Kaggle website along with Apify YouTube Scraper, an effective tool for extracting extensive public data from the YouTube platform. The dataset from Apify consists of 6,458 rows, including columns such as like count, comment count and view count along with tags and description of the YouTube videos. Also, 35,436 rows of data were collected from the Kaggle website. However, since the Kaggle dataset was not up to date, we wrote a Python script using yt\_dlp<sup>3</sup> library to scrape the most recent data using the existing video ID column.

<sup>&</sup>lt;sup>3</sup> yt-dlp/yt-dlp: A feature-rich command-line audio/video downloader

### **3.2 Post Processing**

Data post processing refers to the set of techniques used to clean, organize, and analyze data after it has been collected from various sources [6]. In this research, postprocessing was conducted to enhance the quality of the dataset and optimize feature representation, which is vital for accurate prediction of YouTube video view counts. The post processing steps included feature selection and integration, handling missing data through imputation techniques and outlier removal. By carefully addressing these aspects, we aimed to create a consistent and informative dataset that would support the development of robust predictive models.

### **3.3 Feature Selection**

Since our dataset had tags and descriptions columns, which are not relevant to our use case of predicting YouTube views, we decided to drop them. Additionally, removing these textbased features helped streamline the dataset, reducing dimensionality and improving model efficiency by focusing only on relevant numerical features.

### **3.4 Feature Integration**

New features were added to the dataset to enhance the model's predictive power. Specifically, we added the subscriber count and video duration (seconds) columns, as these factors can significantly influence a video's view count. Subscriber count provides insight into the potential audience size, while video duration may impact viewer engagement and retention.

### **3.5 Missing Data Imputation**

We applied KNN imputation to fill the missing values in our dataset. KNN Imputer was first supported by Scikit\_Learn in December 2019 when it released its version 0.22. This imputer utilizes the k-Nearest Neighbors method to replace the missing values in the datasets with the mean value from the parameter 'n\_neighbors' nearest neighbors found in the training set. By default, it uses a Euclidean distance metric to impute the missing values. Our dataset contained some missing values, for which we chose K-Nearest Neighbors (KNN) imputation with 'n\_neighbors' of 5 instead of mean or median imputation. Unlike mean and median imputation, which can introduce bias and reduce data variability, KNN imputation utilizes similarities between data points to generate more contextually accurate and representative values [7]. KNN imputation is often a better choice because it considers the relationships between features rather than simply filling in missing values with a single statistical measure.

### **3.6 Outlier Detection**

An outlier is a data point that differs markedly from most of the dataset, either being much higher or lower than the others. Such points can greatly influence the outcomes of machine

learning models. Outlier detection plays a critical role in ensuring the accuracy and integrity of data analysis. Datasets often contain anomalies, or outliers, which can skew results and lead to misleading conclusions. Identifying and addressing these outliers is essential to maintain reliable data, as it helps improve the quality of the analysis and ensures that insights drawn are meaningful and accurate [8].

The choice of a multiplier of 15 was made to retain meaningful high-value data, especially in the context of YouTube videos, where viral content can have significantly higher views, likes, and comments than typical videos. A standard IQR multiplier (e.g., 1.5 or 3) might classify many viral videos as outliers and remove them, leading to the loss of crucial insights. By using a larger multiplier, we ensure that only extreme anomalies—such as erroneous or artificially inflated data points—are removed while preserving genuine highperforming videos.



Figure 2: Data Overview After Outlier Removal

### 3.7 Scaling

Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed before training the models to handle highly varying magnitudes or values or units [9]. Features underwent normalization using Min-Max4 scaling to ensure they fell within a standard range, typically [0, 1]. We also used log transformation for the target as it was heavily skewed. This step helped improve

<sup>&</sup>lt;sup>4</sup> MinMaxScaler — scikit-learn 1.6.1 documentation

model performance by making features comparable in scale and enhancing algorithm convergence during training.

### **3.8 Model Selection**

This study utilized Decision Tree Regression<sup>5</sup>, Random Forest Regression<sup>6</sup>, XGBoost Regression<sup>7</sup>, and an Artificial Neural Network (ANN) for predictive modeling. Decision trees split data based on specific criteria to enhance separation, but they can overfit complex patterns, limiting generalization [10]. To address this, Random Forest constructs multiple decision trees using randomly selected data subsets, reducing variance and improving model stability [11]. XGBoost, an optimized implementation of Gradient Boosting, trains decision trees sequentially, with each tree correcting errors from the previous one, leading to higher predictive accuracy. In addition to these tree-based models, an Artificial Neural Network (ANN) was employed to capture non-linear relationships in the data. Inspired by biological neural networks, ANNs use interconnected layers of neurons to process patterns through weighted transformations and activation functions. The proposed feedforward ANN model, implemented in PyTorch, consists of three linear layers: an input layer with four units, two hidden layers (128 and 64 neurons) with ReLU activation, and an output layer with a single neuron for continuous value prediction. To prevent overfitting, dropout regularization is applied, randomly deactivating 20% of neurons in the first hidden layer during training. This combination of ensemble methods and neural networks enhances predictive accuracy while addressing overfitting and model interpretability challenges.



Figure 3: Structure of ANN

<sup>&</sup>lt;sup>5</sup> <u>1.10. Decision Trees — scikit-learn 1.6.1 documentation</u>

<sup>&</sup>lt;sup>6</sup> RandomForestRegressor — scikit-learn 1.6.1 documentation

<sup>&</sup>lt;sup>7</sup> <u>GradientBoostingRegressor — scikit-learn 1.6.1 documentation</u>

## 4 Experimentation

We evaluated model performance using Decision Tree, Random Forest, XGBoost, and an Artificial Neural Network (ANN). Hyperparameters were tuned empirically to balance complexity and generalization. The Decision Tree used squared\_error as the criterion, best as the splitter, and a max depth of 5 to prevent overfitting. The Random Forest model had 100 estimators, squared\_error as the criterion, and a max depth of 15 for robustness. XGBoost was configured with 100 estimators and a max depth of 5 for optimized gradient boosting.

For ANN, we tested four configurations, with two giving the better results. The selected model (ANN1) was a three-layer fully connected network with input matching the feature dimension, two hidden layers (128 and 64 neurons with ReLU activation), and a dropout rate of 0.2 to prevent overfitting. The second model (ANN2) had an input layer with four features, followed by hidden layers with 256, 128, and 64 neurons, all using ReLU activation. After evaluation, we chose the first model for its balance of complexity and generalization.

### 4.1 Evaluation Metrics

Evaluation metrics for regression are essential for assessing the performance of regression models, which predict continuous outcomes. These metrics help in measuring how well a regression model can predict continuous values. The models are evaluated using the following metrics:

- **Mean Squared Error (MSE):** This metric calculates the average of the squared differences between the actual and predicted values. As it is derived from the square of Euclidean distance, it is always a positive value that decreases as the error approaches zero. It provides a measure of how close the predictions are to the actual outcomes, with larger errors being penalized more significantly due to the squaring effect. A lower MSE indicates better model performance [12].
- **R<sup>2</sup> Score**: R<sup>2</sup> score is used to evaluate the performance of a regression model. It is the amount of variation in the output dependent attribute which is predictable from the input independent variable(s). It is used to check how well-observed results are reproduced by the model, depending on the ratio of total deviation of results described by the model. A R<sup>2</sup> score of 1 implies perfect predictions, while a score of 0 suggests that the model does not explain any variance. Higher R<sup>2</sup> values indicate better model performance [13].
- Median Absolute Error (MedAE): The Median Absolute Error is a metric that can be used to quantify a regression models performance. Unlike MSE, which squares the errors, MedAE provides a more robust measure of prediction accuracy that is less sensitive to outliers. A lower MedAE indicates that the model predictions are closer to the actual values [14].

### 4.2 Cross-Validation

To ensure the robustness of the predictive models, we conducted cross-validation. Crossvalidation is a machine learning technique for assessing a model's performance on unseen data. It involves splitting the dataset into multiple subsets, or folds, where one-fold is used for validation while the model is trained on the remaining folds. This process is repeated several times, with a different fold serving as the validation set each time. The final performance estimate is obtained by averaging the results across all iterations, ensuring a more reliable evaluation of the model [15]. The cross-validation was performed using value of k = 3, calculating performance metrics, including R<sup>2</sup> Score, Mean Squared Error (MSE), and Median Absolute Error (MedAE), to estimate model generalizability. By integrating cross-validation within each model's evaluation loop, we obtained comprehensive metrics providing a balanced view of model performance and identifying potential overfitting or underfitting issues.

## **5** Results

In this section, we present the results of the experiments conducted. The analysis focuses on two major components: Correlation Analysis and Comparative Analysis of Models Performance.

### 5.1 Correlation Analysis

From the correlation heatmap, we observe that like count and comment count exhibit the strongest positive correlation with view count, suggesting that higher engagement levels are associated with increased video views. Additionally, subscriber count shows a weaker positive correlation with coefficient of 0.19 with view count, indicating that while a larger subscriber base may contribute to views, it is not the primary driving factor.



Figure 4: Correlation between Engagement Metrics and Views

### **5.2** Comparative Results

In addition to the traditional machine learning models—Decision Tree, Random Forest, and XGBoost—the Artificial Neural Network (ANN) was also evaluated. The ANN model exhibited remarkable performance with an MSE of 0.164, which is significantly lower than all the other models, indicating that its predictions were highly accurate with much smaller squared errors. The R<sup>2</sup> score of 0.812 suggests that the model was able to explain approximately 81.2% of the variance in the data, which is still quite strong, but not quite as high as the Random Forest's performance. This shows that while the ANN was highly accurate in its predictions, it may have had slightly less predictive power in terms of variance explanation compared to Random Forest. Furthermore, the MedAE of 0.164 for ANN indicates that the median error in its predictions was extremely small, suggesting that its predictions were very close to the actual values. This low MedAE supports the idea that the ANN model was not only accurate overall but also consistent in its predictions, with most errors being minimal.

The ANN outperformed the traditional models in terms of MSE, showcasing its ability to minimize prediction error. However, its R<sup>2</sup> score was lower than that of Random Forest, which had the highest explanatory power. Despite this, the ANN's low MSE and MedAE make it a very competitive model, particularly for tasks requiring high prediction accuracy.

In summary, while Random Forest still held the highest explanatory power with an R<sup>2</sup> score of 0.925, the ANN model showed superior accuracy with the lowest MSE and MedAE, making it a strong candidate for this task. The choice between these models would depend

on the trade-off between variance explanation (where Random Forest excels) and prediction accuracy (where ANN outperforms). Both models are strong contenders, with Random Forest providing more reliable variance explanation and the ANN offering the best predictive accuracy in terms of error metrics.



Figure 5: Comparison of ML Models



Figure 6: Comparison of ANN Models

# 6 Evaluation

As part of the evaluation process, we came across a GitHub repository that implemented a similar approach using the same dataset. However, it's important to note that the dataset they used is 6 years old, while our dataset is much more recent. Despite the difference in the dataset's age, we decided to replicate their methodology, which involved using both log transformation and MinMax scaling before feeding the data into models. When we applied this combination of techniques, we observed that the results obtained were similar as shown in Table 1, particularly for the Decision Tree and Random Forest models. This suggests that, even with the more recent data, the predictive performance of these models remained comparable to the outcomes achieved with the older dataset, highlighting the robustness of the approaches used. The repository can be accessed at:

Youtube-Views-Prediction/Youtube Views Prediction.ipynb at master · herdinsurya/Youtube-Views-Prediction

Model	RMSE GitHub Repo	RMSE	R <sup>2</sup> score GitHub Repo	R <sup>2</sup> score	MAE GitHub Repo	MAE
Decision	0.01	0.04	0.92	0.85	0.00	0.03
nee						
Random	0.01	0.03	0.96	0.925	0.00	0.02
Forest						

Table 1: Comparative Results of our Model vs. GitHub Repository Model

# 7 Conclusion

This study analyzed a dataset of 41,894 rows sourced from Kaggle and the Apify scraper. To improve predictive accuracy, additional features such as subscriber count and video duration were integrated. Thorough post processing steps, including handling missing values, feature integration, and outlier removal, ensured a clean and reliable dataset for model training. Before feeding the features into the models, scaling was applied to standardize feature values, preventing biases caused by differing scales.

We experimented with three machine learning models: Decision Tree, Random Forest, and XGBoost, testing multiple hyperparameter configurations to optimize performance. Additionally, two Artificial Neural Network (ANN) architectures were explored, yielding promising results. Model evaluation was done using Mean Squared Error (MSE), R<sup>2</sup> Score, and Mean Absolute Error (MAE). Like count and comment count emerged as the most significant predictors of view count, confirming their strong influence in determining video popularity.

## 8 References

- [1] L. Rui, Z. Afif, R. Saedudin, A. Mustapha, and N. Razali, "A regression approach for prediction of Youtube views," *Bull. Electr. Eng. Inform.*, vol. 8, Dec. 2019, doi: 10.11591/eei.v8i4.1630.
- [2] H. Jang, S. Kim, J. Jeon, and J. Oh, "Visual Attributes of Thumbnails in Predicting Top YouTube Brand Channels: A Machine Learning Approach," 2022 IEEE Int. Conf. Big Data Big Data, pp. 6660–6662, 2022.
- [3] Chen, Yen Liang and Chang, Chia Ling, "Early prediction of the future popularity of uploaded videos," *Expert Systems with Applications*, vol. 133, Nov. 2019, doi: 10.1016/j.eswa.2019.05.015.
- [4] T. Trzciński and P. Rokita, "Predicting Popularity of Online Videos Using Support Vector Regression," *IEEE Trans. Multimed.*, vol. 19, no. 11, pp. 2561–2570, Nov. 2017, doi: 10.1109/TMM.2017.2695439.

- [5] D. poudel, M. C. Cakmak, and N. Agarwal, "Beyond the Click: How YouTube Thumbnails Shape User Interaction and Algorithmic Recommendations," presented at the Proceedings of ASONAM 2024, 2024.
- [6] "Data Preprocessing in Machine Learning: Steps & Best Practices." Accessed: Nov. 04, 2024. [Online]. Available: https://lakefs.io/blog/data-preprocessing-inmachine-learning/
- [7] K. S. Htoon, "A Guide to KNN Imputation." [Online]. Available: https://medium.com/@kyawsawhtoon/a-guide-to-knn-imputation-95e2dc496e
- [8] "How to Detect Outliers in Machine Learning." [Online]. Available: https://www.geeksforgeeks.org/machine-learning-outlier/
- [9] "Feature Engineering: Scaling, Normalization, and Standardization -GeeksforGeeks." Accessed: Nov. 03, 2024. [Online]. Available: https://www.geeksforgeeks.org/ml-feature-scaling-part-2/
- [10] "1.10. Decision Trees." [Online]. Available: <u>https://scikit-learn.org/stable/modules/tree.html</u>
- [11] "Difference Between Random Forest and Decision Tree." [Online]. Available: https://www.geeksforgeeks.org/difference-between-random-forest-anddecision-tree/
- [12] "Mean squared error Wikipedia." Accessed: Nov. 03, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Mean\_squared\_error
- [13] "R Squared | Coefficient of Determination." [Online]. Available: https://www.geeksforgeeks.org/r-squared/
- [14] "Median Absolute Error Inside Learning Machines." Accessed: Nov. 03, 2024.
  [Online]. Available:
  <u>https://insidelearningmachines.com/median\_absolute\_error/</u>
- [15] "Cross Validation in Machine Learning." [Online]. Available: https://www.geeksforgeeks.org/cross-validation-machine-learning/

# Mobility Patterns and Urban Travel Behavior in the Twin Cities

Aseia Nabi, Jalal Khalil

Department of Computer Science and Information Technology Saint Cloud State University St. Cloud, Minnesota, 56301

aseia.nabi@go.stcloudstate.edu, jalal.khalil@stcloudstate.edu

#### Abstract

This research aims to develop a digital twin of the Twin Cities metropolitan area in Minnesota to analyze urban transportation behaviors. To achieve this, data from the Travel Behavior Inventory survey, OntheMap, and OpenStreet map are integrated to examine commuter patterns, including travel times, locations, transportation modes, employment details and residential distributions at census block level. These datasets inform the creation of a synthetic population, which serves as the foundation for large-scale traffic simulations. The insights gained from this study will enhance the understanding of urban mobility and support data-driven transportation planning.

## Introduction

The increasing complexity of urban transportation systems requires innovative approaches to understanding commuter behavior and enhancing infrastructure planning. The need for data-driven models to analyze traffic patterns became vital with the increase in metropolitan areas. The past two decades have witnessed numerous innovations to facilitate better urban planning that addresses the increasing transport demand. The concept of a digital twin is amongst the tools used to model and analyze urban systems more dynamically and data driven. A digital twin is a "virtual representation or replica of a physical system or process that accurately mimics its real-world counterpart" (Aghaabbasi, 2024). Digital twins have emerged as an approach to scenario building and creating virtual representations of roads and traffic behavior. Furthermore, "one of the key characteristics of a digital twin is its ability to monitor, analyze, and optimize the performance of the physical system it represents" (Aghaabbasi, 2024). It aims to provide insights and analysis to enable individuals responsible to make decisions, identify potential hazards, and improve road conditions. Additionally, it allows the use of various types of data collected, data that can be used to make informed decisions and predict future trends. As well as it allows examining various scenarios, modifications, and solutions without altering the physical infrastructure to understand the outcome of the proposed solutions and select the ones that produce the better outcome.

Bachechi (2022) highlighted that the influx in several transport services has caused an increase in the complex urban mobility ecosystem. Therefore, to manage urban mobility with improved efficiency, the author suggests using digital twins, arguing they will assist in testing different scenarios and possible solutions. However, the implementation of digital twins poses several challenges. They must account for interactions between various aspects of urban life, and as the cities are expanding, integrating different layers of transportation infrastructure, such as roads, public transit, and pedestrian pathways, into a digital twin becomes essential to model the interconnectedness and identify infrastructure needs.

Furthermore, Khalil et al. (2024) present an approach to studying the impact of ride-hailing services on urban traffic by building a digital twin of Birmingham, Alabama. It addresses two key challenges: the lack of data collection methods for ride-hailing services and the difficulty of projecting future impacts due to the rapid evolution of these services. The authors propose a unique data collection method that involves surveying Uber drivers and collecting app screenshots of ride summaries, which are then processed through crowdsourcing to extract detailed ride information. This approach is designed to be easily replicable in any city with minimal driver participation. Using the collected data and census statistics, the authors generate realistic travel day plans for the Birmingham population and execute them in an agent-based simulation using MATSim.

Therefore, this research aims to create a digital twin of the Twin Cities in Minnesota using secondary survey data. The data collection for the metropolitan areas has been completed. While the simulation is planned, it will not be included in this paper. The primary goal of this study is to analyze transportation behavior and commuter patterns across different regions. This paper focuses

on the foundational step, contributing to existing studies on the use of digital twins for analyzing urban transportation dynamics.

# **Data Description**

The data used for this research was sourced from the Travel Behavior Inventory (TBI) survey<sup>1</sup> conducted for the Metropolitan Council for the years 2018-2019. TBI is a household survey conducted every year with the aim of supporting transportation planning. The data collected spans through October 1, 2018, through September 30, 2019. Moreover, the survey was conducted via a mobile app, "rMove," a product of the Resource Systems Group. Individuals with smartphones were asked to download the app and use it for seven consecutive days, while those without a smartphone were asked to fill out a one-day travel diary either online or over the telephone, where call centers would complete the travel diary with the participant. The survey was divided into two parts; for part one, information related to household demographics, composition, and travel behavior was collected, while for the second part, participants were required to record their travel for the specified time. The survey collected data on trips taken by individuals, the start and end time of trips, purpose of trip, number of travelers, mode of transportation, location block groups, county and city for both the arrival and destination locations.

For this research, the main columns used were the responder's ID, arrival, and departure times, origin and destination block group information, origin and destination purpose (home, work, social, etc.), and mode type (personal vehicle, local transit, shared vehicle, etc.). While the survey included other detailed information, such as the number of people sharing a ride, trip distance, cost, vehicle speed, participation group, vehicle parking, and trip season, these were omitted as they are not required for our research.

Additionally, data was also gathered from external sources such as the OpenStreetMap (OSM) API to collect information about house and work locations in the Twin Cities area. The work location data was obtained by querying the OSM API, which provides geographic and address-related data for specific areas. Using this API, we extracted detailed location information, including house numbers, street names, cities, zip codes, and latitude/longitude coordinates. These data points were then processed and stored in a data frame for further analysis. The home location data was similarly queried through an OSM API query for the Minneapolis and Saint Paul areas, retrieving relevant address details for each house. This combination of data sources enabled a comprehensive view of both work and home locations within the study region.

<sup>&</sup>lt;sup>1</sup> <u>https://metrocouncil.org/Transportation/Performance/Travel-Behavior-Inventory.aspx</u>

## Data analysis

To create a digital twin, the first step is to conduct data analysis and construct the OD matrix, which serves as the foundational element for creating the simulation. OD matrix helps model travel patterns by capturing the number of trips made between different origin and destination pairs within an area; prior to creating the OD matrix, it is essential to analyze the survey data and drive insights regarding the different attributes and their impact on the travel behavior of the survey responders. The dataset was loaded and preprocessed to extract relevant features for analysis. Data was grouped by modes of transportation, activity types, and time intervals to compute aggregate metrics. Figure 1 depicts the distribution of transportation modes and activities, providing insights into the frequently utilized mode of transportation by individuals in the twin cities according to the survey. 75.7 % of the individuals use their household vehicle as the primary source of transportation.



Figure 1: Distribution of Transportation Modes



Figure 2: Distribution of Origin Activities

On the other hand, majority of the trips began at home (Figure 2), which is understandable given most responders would begin logging their travel diaries at the beginning of the day prior to leaving for work or school. While most other trips began for the purpose of maintenance or for social or recreating purposes, which could be due to these trips being taken later in the day and being documented then.

Additionally, to analyze the distribution of trips, the number of departures and arrivals by hour of the day were visualized (see Figure 3), highlighting the peak travel times, which start from 8:00



Figure 3: Distribution of Trip Departure and Arrival Times

Furthermore, trip frequency was analyzed to understand how trip patterns vary over time, the analysis also considered the average trip duration and distance by transportation mode.

## Home and Work locations

To create the OD matrix, it's required to have information on house and work locations for the trips recorded on the survey. The survey did include the higher-level address information, but to maintain the privacy of individuals responding to the survey, specific addresses were not included in the data. Therefore, to fill the gap for the addresses required, data from other sources, such as the American Community Survey (ACS) and the OpenStreetMap API, were utilized to gather house and work locations for the twin cities. These locations were then filtered based on geo IDs obtained from the web-based mapping tool On the Map from the U.S. Census Bureau, which was used to create the OD matrix. An initial analysis of the data obtained was conducted to calculate certain metrics, such as the number of work locations in certain geoids and the number of employees working in each location; similarly, for the house locations, the total number of houses in each geoid, and average family size. These insights assist in understanding the distribution of home and work locations across different regions which would be essential for building the simulations based on the survey.

## Trip chain analysis

The concept of "trip chains" is critical in understanding people's travel habits. A trip chain is a sequence of connected travels taken by a person over a specific period, usually containing numerous stops or destinations, such as home-to-work, work-to-home, work-to-grocery, and grocery-to-home. The interconnected trips form a chain of activities that provide a more complete picture of a commuter's daily travel habits. To process the trip chains, the raw trip data was organized by identifying everyone's trip sequence and linking them based on starting and ending points and time intervals between the trips. After the trip chains were processed, the next step in

the analysis was to discover common patterns, such as frequent commuting routes or popular stop sequences. By evaluating these trip chains, we obtained insights into people's mobility patterns, the types of activities they engage in during the day, and how their journeys are interconnected; this knowledge of travel behavior is essential for identifying transportation trends.

# **OD** matrix

Creating an OD matrix is essential when simulating travel patterns using MATSim. Thus, to create it, initially, the datasets for house and work locations are filtered based on the geo IDs, the number of house and work locations that are within those geo IDs are identified, and the remaining locations that are not in any of those locations are filtered out to ensure consistency. The total population and workers for each geo ID are obtained, and with the information collected, the OD matrix is created (Figure 4), representing the flow of trips between origin and destination location pairs. The OD matrix provides the data on how many agents travel between locations throughout the simulation, impacts the assignment to a specific route, and helps determine the volume of traffic between locations. Additionally, by modifying the OD matrix, different travel scenarios can be simulated to understand how changes in commuter behavior will be affected by certain changes, providing insights into planning and policy decisions.



Figure 4: OD Trips

## **Conclusion and future work**

The paper highlights the initial steps taken to create a digital twin for the twin cities in Minnesota. Initially, data analysis was conducted on the survey dataset obtained from Travel Behavior Inventory (TBI) to understand the content of the data and plan on how it can be used for this research. Then, to create the OD matrix required for the simulation, additional data sources were utilized to obtain work and home locations and filtered based on the geo IDs available for the twin cities. Lastly, the OD matrix was created to highlight the travel patterns of the survey respondents. The next step involves creating the digital twin using the OD matrix and the data collected from the responders. It will act as a digital representation of individuals' movement and travel patterns. Additionally, the data will be integrated into an existing model to leverage its capabilities and use it to provide additional analysis based on the survey data.

## References

- Aghaabbasi, M., & Sabri, S. (2025). Potentials of digital twin system for analyzing travel behavior decisions. *Travel Behaviour and Society*, *38*, 100902. Retrieved from https://www.sciencedirect.com/science/article/pii/S2214367X24001650.
- Khalil, J., Yan, D., Yuan, L., Jafarzadehfadaki, M., Adhikari, S., Sisiopiku, V. P., & Jiang, Z. (2022, November). Realistic urban traffic simulation with ride-hailing services: a revisit to network kernel density estimation (systems paper). In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems* (pp. 1-10). Received from <a href="https://dl.acm.org/doi/pdf/10.1145/3557915.3560963">https://dl.acm.org/doi/pdf/10.1145/3557915.3560963</a>

 Bachechi, C. (2022, August). Digital twins for urban mobility. In *European Conference on Advances in Databases and Information Systems* (pp. 657-665). Cham: Springer International Publishing.Retrieved from <u>https://www.researchgate.net/profile/Chiara-Bachechi-</u> <u>2/publication/363048912\_Digital\_Twins\_for\_Urban\_Mobility/links/64ac1cbdb9ed6874a</u>

50b6310/Digital-Twins-for-Urban-Mobility.pdf

7 165

## Generative AI: Emerging Policy, Innovation, and Debate

Caroline Cronk, Cheryn Lindsay, Isaac Wayne, and Ahmed Kamel Computer Science, Offutt School of Business Concordia College 901 8<sup>th</sup> St S, Moorhead, MN 56562 ccronk@cord.edu, clindsay@cord.edu, iwayne@cord.edu, kamel@cord.edu

### Abstract

A recent major (possibly disruptive) development in the field of Artificial Intelligence (AI) was the release to the public of ChatGPT, a machine learning based Artificial Intelligence chat and problem-solving system. Several competing systems now exist, some predating the introduction of ChatGPT, and others more recently introduced. Initial successes of these systems are impressive, leading to major developments in a variety of disciplines, and significant governance efforts from attempts to benefit from these successes to attempts to regulate the use of these technologies.

In this paper, we researched some of these new developments. We studied the use of generative AI on economic activity, on education, on security and privacy, as well as its challenges to intellectual property. We also studied efforts to govern and regulate the use of generative AI on these areas both domestically and internationally in several other countries. We present the results of this research in this paper.

## Introduction

Recent advancements in generative AI technology have brought AI into the public eye more than ever. Since the release of ChatGPT in 2022, the rise in prominence of large language models, or LLMs, has sparked discussion and research about the opportunities and concerns that generative AI presents. While generative AI is not a new concept, new capabilities and public accessibility of LLMs and image-generating diffusion models have captured the public consciousness in a way other AI has not. Generative AI has the potential to change the way people work and learn, and it opens many questions about regulation, security, and rights. The significant amount of literature on generative AI from the past few years merits an examination of the reactions to, and effects of, generative AI for industries, governments, and individuals around the world.

## Economy

Concerns for AI job replacement are on the rise as studies show that AI has the potential to replace up to a billion jobs worldwide in the coming decade, despite the millions of jobs it has the potential to create within a matter of years [1]. A study found that the term "AI Taking Jobs" reached a record high in number of searches on Google in November 2023, a 400% increase from the previous year [1]. Investigating the varying implementations of AI in the workplace, and the side effects of these changes, may provide valuable insights and address the growing concerns over what to expect with these drastic changes.

Unlike previous technologies, LLMs will have the largest impact on middle-to-highincome, white-collar jobs, rather than manual labor [2]. Studies show that administrative positions and jobs with repetitive functions (e.g., bookkeeping, proofreading) have the highest risk of AI replacement, whereas manual labor jobs and positions requiring creativity and interpersonal skills (e.g., writing, legal services) have the lowest risk of AI replacement [1].

The World Economic Forum describes four categories of jobs relating to how they will most likely be impacted by AI, including high potential for automation, high potential for augmentation, low potential for either, and jobs that will go mostly unaffected [3]. One example of AI-assisted job augmentation is through the development and usage of GitHub Copilot. GitHub Copilot is a coding assistant LLM that generates code suggestions based on the content within the code file. Studies have shown that it increases solo programmer efficiency without significant risk of added security riska [4, 5]. A study found that while programmers wrote code more quickly with Copilot than with pair programming, they were more likely to delete or fix lines of code later in the study, suggesting that the quality of the code created with pair programming is better [6].

Banking and the finance industry have some of the highest potential for job automation, including jobs like credit authorizers, clerks, and tellers [3]. About half of banking, utilities, and telecommunication customer contacts are currently handled by machines,

1

some of which are AI technologies [2]. Banks could use AI to fully automate online help for clients or could use AI assistants to provide helpful information to customer support workers [2].

Despite the growing concern for automation in the workplace, integration of AI solutions also brings some benefits. 20% of workers in a study expressed positive views, welcoming AI growth and believed that integration in the workplace would grant relief for some tedious and repetitive tasks, 85% of those in this survey also support the movement towards automation of jobs considered "hazardous or unsafe" [1].

Businesses are already making plans to implement LLMs into their workflows. 75% of organizations surveyed for the Future of Jobs Report 2023 have plans to adopt AI in their operations [3]. Another survey of businesses showed that one-third of the respondents were planning on replacing some part of their human employee's work with generative AI [1]. IBM's annual survey of CEOs [7] found a slightly higher number: 47% expect to reduce or redeploy their workforce in the next 12 months because of generative AI. However, 40% plan to hire additional staff because of generative AI [7].

The Future of Jobs 2023 report predicts that insurance and pension management, information technology services, media, entertainment, and sports have some of the highest potential for AI automation and augmentation. As of 2023, all three are in the top five industries in terms of plans to adopt AI technology [3]. Generative AI could be adapted into corporate workflows in tasks such as drafting emails, creating presentations, searching a company's system, generating marketing ideas, customer support follow-ups, and so forth [2].

The rush to integrate new AI technologies may be largely fueled by companies' desire to stay ahead of their competitors. 62% of business leaders surveyed by IBM [7] said they are willing to take more risks than the competition to maintain a competitive edge. 51% agreed that the risk of falling behind is driving them to invest in some technologies before they have a clear understanding of the value. However, most companies are still only in the planning and testing phases of adopting new AI technologies [7].

Generative AI could add trillions of dollars to the global economy, making it one of the most prominent advancements in modern technology [2]. Researchers predict that certain tasks within jobs will be automated rather than the entirety of the jobs themselves. This means that job responsibilities will change, but most jobs will not be replaced [2]. The result of this emphasizes workers needing to "upskill" as job requirements change [1]. Short-term job loss is a possibility as people transition their careers, but researchers predict that new, AI-related jobs will be created [3].

### Education

One of the most pivotal areas of AI implementation is within the field of education, and steps are already being taken to address how generative AI might impact the classroom.

2

The CTO of OpenAI, Mira Murati, has said she believes education is one of the most important applications of AI [8]. While the truth of this remains to be seen, there have been studies that attempt to implement generative AI into post-secondary education. with potential positive and negative results according to surveys of students and instructors.

Surveys in the college setting have shown that student perceptions of generative AI are often described as either generally positive [9] or mixed [10]. Both instructors and students agree that generative AI should be adopted in the classroom, but that there are problems with generative AI models that must be addressed before they can be effectively integrated. Some instructors believe that generative AI will pressure educational institutions to change their tactics, leading to a more effective system [11, 12]. For example, this could lead to a departure from multiple-choice assessments in favor of more nuanced approaches "that require students to demonstrate their understanding in real-time" [12]. Another idea is to assess students on critical thinking and problemsolving, which are areas that AI models consistently struggle with [12]. Across multiple surveys, neither instructors nor students believe that generative AI will replace instructors in the classroom [11]. However, both instructors and students believe that generative AI will inevitably be involved in education [10, 11]. In several studies, students had an optimistic outlook on the future of generative AI [9, 13]. AI's role in and effect on education remains to be seen.

There are studies on the implementation of generative AI tools in a variety of courses, particularly in the field of computer science. Examples include using ChatGPT to help students write an essay [13] and discussing AI image generation in relation to art education in a college setting [14]. Another topic involving AI in education is an intelligent tutoring system, or ITS. The concept of an ITS predates LLMs, but they enjoy a resurgence because of the recent breakthroughs in LLMs and machine learning.

Recent studies [15, 16] testing these AI tutors in classrooms have focused on helping students in beginner programming courses, most likely chosen due to generative AI's ability to write code. One argument made is that a large language model dedicated as a tutor for the course would provide a way for computer science students to interact with this technology in a more constructive way compared to a generalized model such as ChatGPT [15].

One of these studies [16] created an ITS called CodeTutor, using ChatGPT-3.5 as its model. Participants were split into a control and experimental group, with part of each group having previously used any LLM-based tools such as ChatGPT. The control group could only use human tutors, and the experimental group could only use CodeTutor for help. After a semester of observation, the participants in the experimental group who had never used a chatbot before CodeTutor experienced the largest improvements in test scores. On average, participants in the experimental group scored higher than those in the control group. Perceptions of CodeTutor's confidence decreased as the semester went on, as CodeTutor struggled with the increasingly complex problems and concepts. This suggests that CodeTutor would not be beneficial in higher-level classes if it is already

struggling in the first course, a point that the authors acknowledged as a limitation of the study [16].

Unlike CodeTutor, Iris was intended to be developed further after its study [15]. The authors argue that general models such as Chat-GPT are not built specifically for education. Their solution to that is Iris, a dedicated tutoring model. Iris, which uses GPT-3.5-Turbo, functions inside of a learning management system called Artemis, chosen for its focus on programming exercises. Artemis gives immediate feedback after submission, and Iris aims to supplement that with personalized feedback. The main point of Iris is to be a stand-in for human tutoring. After a semester of use, students' reception was generally positive, finding Iris both helpful and engaging, but not necessarily motivating. Most students were comfortable interacting with Iris, without a strong preference for Iris or human tutors. Students preferred using Iris to ask questions during lectures, and stayed confident even when unable to use Iris during assessments. However, Iris has its problems. One notable shortcoming is its file management capabilities. The authors advise transparency from instructors if they plan to implement Iris in their courses [15].

While student reception of these ITSs has been generally positive, it may be too early to tell if the advancements brought on by machine learning and LLMs are enough to make something like an AI tutor finally viable on a large scale, or if further research and development would still be necessary.

One of the biggest concerns about AI in education is its threat to academic integrity. Many in education are worried that generative AI will be used for plagiarism [10, 11]. A 2023 model of ChatGPT-4 was able to pass assessments in certain early-level college Python courses on its own, a feat that previous models of Chat-GPT were unable to do. The model excelled at answering multiple-choice questions and in generating solutions for programming exercises, however, they struggled with file management and multi-step reasoning [12]. While this is an arguably impressive feat, it raises the risk that a student could use this to cheat their way through a course.

In terms of writing, the usage of an LLM can be hard to detect [11]. Even if LLM usage is banned in a course, a student could still create autogenerated text using a free LLM. Several AI detectors have been developed with limited success leaving the use of generative AI in writing much harder to detect compared to other forms of plagiarism. Even when it comes to coding, which has a relatively rigid structure compared to conventional writing, most AI detectors are not very accurate and can be easily fooled by simply asking an LLM to sound like a human, further driving down the accuracy [17]. Some are suggesting that instructors make their rules on AI usage clear to students [11].

Just like in other areas, there are worries in education about generative AI's potential to spread misinformation due to instances where models make up facts or even citations, aptly named "hallucinations" [11, 13]. Overreliance is another potential issue, but studies that have implemented generative AI in courses have been able to mitigate this issue [15, 16]. However, overreliance must still be considered when introducing generative AI into any part of education [11]. Currently, institutions have no consensus when it comes to

4

issues of cheating and misinformation. These two issues bring on their own challenges, making them particularly hard to deal with. Before institutions officially introduce AI tools into classrooms, there are important discussions to be had on this topic.

### Governance

Regulations in the classroom are just the beginning as the world begins to grapple with the possibilities of AI technology. Policymakers are facing the question of how to regulate AI to make use of its benefits and drive innovation, while addressing the concerns of their citizens and protecting against potential risks of. While many countries have a variety of proposed legislation, regulations, and standards most countries have yet to pass any legislation governing AI. Governments are gathering research, testimony, and public comments to better understand how to approach the challenge of AI governance.

The United States has taken a moderate stance thus far, aiming to balance innovation and regulation. President Biden's executive order on Safe, Secure, And Trustworthy Artificial Intelligence gives instructions to 50 government agencies for AI policy including plans to adopt AI technologies in government agencies, with eight guiding principles to ensure the safety and well-being of citizens while promoting innovation. For example, one principle seeks to assuage public fears of unemployment, promising support of workers to soften the potential impacts of jobs being replaced or restructured due to new AI technologies. The executive order states a goal for the U.S. to be an international leader in AI development, adoption, and regulation [18]. However, the U.S. government has not been very involved with the development of AI, leaving the funding and research predominantly up to private companies. The executive order achieves some aims of the proposed AI in Government Act from 2020 and the Advancing American AI Act from 2022. Both bills involve the adoption of AI in government agencies while considering the risks [19, 20]. The AI in Government Act would create the AI Center of Excellence under the General Services Administration to oversee government adoption of AI [20].

The proposed No AI FRAUD Act aims to address public concerns by regulating deepfakes by giving individuals exclusive property rights to their name and likeness, only to be used by other entities through contract with the individual or collective bargaining [21]. Providers of AI models made for creating replicas of known individuals, anyone who makes a replica of a person public without their consent, and anyone who assists them while knowing the person has not given consent, are liable for damages [21].

The European Union has taken a more regulatory approach to AI governance, focusing on the protection of the public. The 2024 Artificial Intelligence Act [22] categorizes four levels of risk for AI systems. Unacceptable risk systems are banned, such as some kinds of real-time biometric identification, social scoring, and systems created with the intent to manipulate people. High-level risk systems related to profiling individuals—such as assessing eligibility for benefits or employment—or are related to prominent institutions like education or law enforcement are required to have an assessment of risk and create risk-management plans before the system can go into service [22]. Limited risk systems must disclose that their content was AI-generated and provide enough transparency for users to make informed decisions [23]. Minimal risk systems include most AI systems, such as recommender systems or spam filters, and are not regulated [22]. The existing General Data Protection Regulation from 2018 requiring companies to take proper measures to protect consumer data is also relevant to AI development. Most international corporations have implemented a data governance system to comply with it [24].

A white paper released by the British government, A Pro-Innovation Approach to AI, states the United Kingdom's intent to not strictly regulate the AI industry, to be consistent in their regulation, and to work collaboratively with the industry. They also hope to collaborate with other countries for more consistent AI governance globally [25]. The U.K. Ministry of Defense released several documents outlining their strategy for AI development in 2022. They aim to be ambitious in delivering AI tools for defense, while maintaining safe and responsible use of AI [26]. The U.K. plans to rely on existing agencies to regulate AI [25].

China has coupled somewhat stricter regulations with a strong drive for innovation by creating the AI Governance Expert Committee under the Ministry of Science and Technology to oversee the regulation of AI [27]. This committee joins many existing technology-focused agencies in governing AI in China. The Cyberspace Administration of China has released a draft of the proposed Measures for the Administration of Generative Artificial Intelligence Services. The measures would regulate the quality of generative AI training data. In addition to mandating that training data follows the existing Cybersecurity Law, the Measures would require that data does not contain IP infringement, can only be used with permission from the subject, and must be accurate, unbiased, and diverse. Generative AI services will have to undergo a security assessment before going on the market. Additionally, generative AI providers must require users to register their real identity with the programs. The Measures include some protection of user data, such as forbidding AI providers from using inputs to profile or identify users. The Measures also lays out 8 AI governance principles; for example, fairness and justice, open cooperation, and respect for privacy [27].

It is uncertain to what extent the Chinese government will respect the privacy of its citizens. The Personal Information Protection Law of 2021 requires the service providers to disclose how they are using the personal information, to store it securely, and to only store it for short periods of time. However, the above law allows for facial recognition software to be used for public security, with the only requirement being that the equipment is marked with a sign [27].

China has released the Next Generation Artificial Intelligence Development Plan, which outlines their goals for AI innovation [28]. One of these stated goals is to become the world center for AI by 2030. To achieve this, the government is investing heavily in technology development, with a focus on funding for AI breakthroughs. Unlike the U.S. where private companies are driving almost all AI innovations without government involvement, the Chinese government is working together with private companies to boost AI innovation. Select companies have been named as "platform leaders," for

different categories of AI technologies. The platform leaders are tasked with helping lift smaller startups. A stated goal along the way to AI dominance is to shift towards a digital economy. To achieve this, China is investing in what the government calls "New Infrastructure." This includes more powerful internet, a larger power grid, data centers, and so forth with plans to strengthen technological education in China and help raise the next generation of science and technology researchers [28].

Italy takes on a unique role in AI regulatory action as the first European country to temporarily ban ChatGPT, in 2023. The Italian data protection agency responsible for driving this decision-making pointed to the company's previous issues with a leak that compromised user data and further violations of data collection rules [29]. The purpose of the ban was to immediately halt the data collection of users in the country and launch an investigation into OpenAI's privacy concerns. Italian authorities also threatened to fine OpenAI \$21 million, applying significant pressure on the company to comply with EU data privacy laws [29]. OpenAI has since proposed measures and taken action to address these data privacy concerns, trying to fulfill these demands as other European regulators began to take notice and open investigations themselves [30]. The ban was lifted after the company took appropriate action to amend these issues. This first case of a nationwide ban to initiate AI reform ultimately proved to be successful in initiating changes to mainstream AI data handling methods marking a landmark step for AI regulation in the protection of user data privacy, expressing concern for safely implementing AI, and ensuring compliance with the country's laws.

### **Security and Privacy**

Concerns relating to user privacy and data security continue to grow as AI technology develops. The introduction of AI agents—software operating across different platforms that can respond to user demands, engage in human interaction, and learn user behavior and preferences over time—has opened more doors for such concerns [31]. One study addressed the human-like conversation capabilities of AI and found that the relationship between user privacy concerns and the social presence of AI agents are negatively correlated, and positively correlated with AI agent intimacy [31]. This suggests that users are less likely to worry about disclosing personal information to social AI agents. This finding is substantial as AI chatbots move to develop and support more robust vocal interactions with users. For example, OpenAI is advancing its current voice mode to sound more humanlike, give real-time responses, and react to user emotions [32].

Another report [37] breaks down the challenges of data privacy regarding AI systems' mass data collection and how people both directly and indirectly interact with AI systems. Additional privacy concerns arise when massive data collections can only be interpreted by AI systems. The concept of privacy itself is rooted in the fact that people exercise control over what information about themselves they want to reveal to others. That control, however, is seemingly nonexistent when it comes to their data being examined by AI systems. Mass digital surveillance by governments and private entities far exceeds

standard, non-digital practices in both extensiveness and completeness. For example, this can be seen in how online services can keep viewers watching for longer because they "know" them; for example, YouTube generates specialized video recommendations based on user preferences with little to no conscious, intentional input or even awareness by the user. There are also problems related to the "black box" nature of machine learning systems that hinders the designers of the systems in fully understanding the conclusions of these algorithmic processes [37].

The need for transparency, authenticity, consent, privacy, representation, trust, and ethical standards across these widely sourced collections for training AI has become apparent now more than ever. Shortcomings of modern tools for tracing data authenticity at this scale create new challenges relating to data transparency in mass data collection [38].

A report from the World Economic Forum addresses the changing landscape relating to AI implementation and technological advancements, particularly within the field of cybersecurity. Some of the emerging impacts of AI on the field of cybersecurity include using AI for cyber-attack detection and prevention, identifying phishing emails, detecting malware, and even generating secure passwords [33]. One of the primary concerns addressed in this report is that increasing the sophistication of AI also increases the complexity of cyber threats [34]. For example, the report describes an attack where an attacker uses AI assistance to draft highly personalized phishing messages. Concerns arise when traditional security systems prove to be ineffective against newly emerging threats. There was a 72% increase in the number of data compromises in 2023, following a previous record high set the year before. The spiking numbers of data breaches and evolving threat tactics that utilize AI spark a heightened awareness for existing and constantly changing risks relating to cybersecurity [35].

Advanced AI is being used to help benefit cybersecurity at a time of heightened concerns for software vulnerabilities. CodeSentry is an example of an AI-based, vulnerability detection software revolutionizing the field of cybersecurity. Their software uses a Generative Pretrained Transformer (GPT) that has been fine-tuned and optimized for pointing out vulnerable code patterns. The model has achieved a 92.65% accuracy rate in vulnerability detection [36] by utilizing advancing AI technology to conduct real-time analysis providing insights into modern vulnerability detection.

## **Intellectual property**

Intellectual property, including copyright and patent law, is a source of major debate around generative AI. Artists, writers, composers, and other copyright authors are deeply concerned that their copyrighted works are being used without permission to train AI systems [39]. Many of them do not want AI training to be considered fair use of copyrighted material.

Speaking on behalf of the Society of Composers and Lyricists in a House of Representatives subcommittee hearing [39], Ashley Irwin argues that AI developers

8

should seek consent of the authors, credit them, and compensate them before using their work to train a generative AI system—either through direct licensing agreements or through collective licensing like music streaming sources have done. Photographer Jeffrey Sedlik points out that there is an existing industry for licensing artists' work, and it is one of the ways artists can continue to earn money from their work [39]. Stock photo agencies like Getty Images have licensed images to some AI companies, while others have used them without permission. Getty Images has filed a suit against Stable Diffusion for using their images without licensing them [40].

Another argument raised against AI training as fair use is that the generated works directly compete with human authors [41]. AI systems can learn and create much faster than a human artist or author can [39]. Generative AI could affect the employment of the authors it can mimic and could negatively impact the market by reducing demand for work from human authors [41]. Potential harm to the market is as much a part of fair use qualification as how the works are being used [39].

The developers in favor of AI training as fair use of copyrighted materials argue that the AI systems and outputs are sufficiently transformative works, and therefore do not infringe on the content used for training. In his House subcommittee testimony [39], Sy Damle stated that the copyrighted works used in training are not stored in the model, and an output that closely resembles a piece of training data is a type of overfitting that developers try to eliminate. The system uses copyrighted works to learn patterns and facts about language and images, then discards them. Under American copyright law, facts are not copyrightable [39].

In his testimony [39], Chris Callison-Burch brings up the Google Books case precedent and believes it should apply to AI training. To develop Google Books, the creators downloaded copies of books without permission and used them to create a searchable database of books. An infringement claim was brought forward, but the court decided that because the copyrighted material could not be viewed in full, and the work was suitably transformative, it was not infringement [39]. However, as Timothy Lee points out in an article for Ars Technica [41], Google Books was not using copyrighted material to make more books. Generative AI does create new images and text [41].

A major reason AI companies and developers want training data to be considered fair use is that training a generative AI model requires billions of pieces of content [39]. Most content currently used is from giant datasets scraped from the internet. Almost all content on the internet is copyrighted, as something becomes copyrighted the moment it is put in a tangible medium. Damle argues that finding and paying everyone whose work is in the database would be nearly impossible. Although collective licensing has had success in the music industry, doing the same for training data would be on a significantly larger scale. The cost to AI developers would be staggering, while the billions of individual creators would not receive much in royalties. Damle cautions that requiring AI companies to license all copyrighted materials for training could drive them out of the U.S. to lessrestrictive countries. Callison-Burch argues that it could potentially force a monopoly on AI if only the largest companies can afford to license material [39]. Many artists and writers are concerned that generative AI can be prompted to mimic the style of a specific person. However, in previous cases, courts have determined that style is not copyrightable. Damle argues that the AI mimicking style falls more under unfair competition than copyright law. Callison-Burch suggests that AI-generated songs mimicking a musician's style and voice could be handled by bolstering right of publicity laws instead of through copyright [39].

There is also the issue of the copyright status and patentability of works created by AI. The U.S. Copyright Office officially stated in 2023 that works created by AI are not eligible for copyright protection. The question remains open on how much human involvement is required in the creation for it to be AI-assisted human authorship and therefore eligible for copyright [39]. Governments around the world have issued the same stance: only humans can be considered authors under copyright law [42]. Countries have also stated that patents must have a human author, though there are some exceptions. In 2019, Stephen Thaler, the inventor of the DABUS AI, applied to patent some of DABUS's generated movies and images with DABUS listed as the author. The patent was rejected by the European Patent Office, the United States, and the United Kingdom, but it was accepted by the Republic of South Africa [42]. A German court decided that DABUS's inventions could be patented as long as Thaler was named as the author [43].

This decision is consistent with the general consensus on patents for AI-generated works. While there are people who argue that laws should change to allow AI to be an author, the majority seek to identify a human author of AI-generated works. The question becomes whether the end users, developers, or owners of the AI are the authors of AI-generated outputs. In China, the Tencent v. Yingxun case determined that an AI-generated article belonged to the company it was generated for [44]. In U.S. case law, if the AI system did the majority of the work to generate the outputs, the author of the system is recognized as the author of its outputs [45]. In the U.K., a 1988 law states that computer-generated works are authored by "the person by whom the arrangements necessary for the creation of the work are undertaken" [45]. Both policies allow the end user to be the author if they do most of the work to create the output. This ambiguity may lead companies to name a human author for an AI-generated innovation so that they can secure a patent [45]. There have been a few cases of generative AI deepfakes of celebrities' voices or likeness being used without their consent which spurred lawmakers to discuss how to best regulate deepfakes and to protect an individual's voice or likeness from being generated without their permission. Right of publicity laws could be a way to achieve this. There are currently only state right of publicity laws in the United States, no federal ones [39].

Many state right of publicity laws allow for people to sue over unauthorized commercial use of their name or likeness [46]. However, that means that the right of publicity can only be used to contest deepfakes that are used commercially. Additionally, in some states, only people whose name or likeness is commercially valuable, like celebrities, can assert right of publicity claims. For other non-consensual deepfakes, other laws may apply. Current laws against releasing intimate depictions of someone without consent

could protect against explicit deepfakes, and current anti-defamation laws could protect against deepfakes that could be used to harm one's reputation [46].

Some members of Congress have proposed laws that prohibit certain AI replicas of people while leaving the right of publicity laws up to states. The Protect Elections from Deceptive AI Act would forbid falsely depicting federal candidates with the intention of influencing elections. Another proposed bill is the No AI FRAUD Act, which would give individuals the property rights for their own name and likeness, regardless of whether they are commercially valuable [46].

The negotiations following the SAG-AFTRA and Writers Guild of America strikes in 2023 were two of the first collective bargaining agreements involving generative AI [47]. Studios must now get union actors' consent to create digital replicas of them, alter previously recorded media of them, or store their digital replicas. The studios must provide a description of the intended use of digital replicas or alterations before gaining consent from the actors. Actors in the union must be compensated for their digital replicas as if they had recorded the scene or audio [48]. The Writers Guild agreement requires studios to inform writers whether any part of the material given to them is AI-generated. It also requires that a generative AI cannot be a credited writer, be used to write or rewrite material, or be used to create source material for a production. The agreement allows writers to assert claims that their work has been exploited to train an AI [47]. AI-generated scripts cannot be used to reduce the pay of writers or undermine their credit [49]. As style cannot be copyrighted, the union agreement allows a writer to protect their style and reputation by preventing generative AI that directly mimics their work.

## Conclusion

It is impossible to predict with certainty what the future of generative AI holds. That is one of the greatest challenges that governments face in regulating AI. Governments around the world are taking their first steps towards regulating AI, balancing innovation and protection; they are determining how to protect citizens from security and privacy risks and from deepfakes spreading misinformation or other harm. Copyright law is a key debate in AI governance with no clean answer.

Overall, generative AI is predicted to contribute trillions of dollars in economic growth by increasing efficiency and adding new products and careers. Businesses are making plans for how they can be part of that growth and take advantage of new AI technologies. Concerns about job loss are not unfounded, including job loss concerns related to intellectual property. However, experts predict more restructuring of jobs rather than complete automation of them, and that AI-driven unemployment will be short term.

Generative AI is slowly being accepted into classrooms, and it likely will continue to be integrated into education. Plagiarism is a prominent issue with AI in education, yet it has been shown that educational help from LLMs can help students learn without becoming reliant on them. Generative AI shows potential in finally creating a successful digital

11

177

tutor, which would expand the ability of educational institutions to give students personalized, individual help.

There is caution, but there is excitement, too. Recent AI innovation allows things that were previously impossible, both good and bad. In the end, generative AI as part of education, work, and life is a near reality. Whether actions will follow stances is hard to say, but there are overarching hopes that AI development and adoption will be conducted and regulated ethically. Across countries and industries, the prevalent sentiment seems to be that AI innovation must be pursued but pursued responsibly.

### References

- [1] "Balancing the Risks and Rewards: The Ongoing Debate Over AI," 2024. [Online]. Available: https://innovationatwork.ieee.org/balancing-the-risks-and-rewards-theongoing-debate-overai/?mkt\_tok=NzU2LUdQSC04OTkAAAGRBeD\_TBvoR52YJJByijdoyCY\_zHCy mZuDFojZUatKuwP6XM62C1JZy2doPbRmLer\_m6Zm0P1cf0hdYcrAugr7k4S0ri WDucu7S04PCgAdoPmS.
- [2] M. Chui, E. Hazan, R. Roberts, A. Singla, K. Smaje, A. Sukharevsky, L. Yee and R. Zemmel, "The economic potential of generative AI: The next productivity frontier," McKinsey & Company, 2023.
- [3] World Economic Forum, Accenture, "Jobs of Tomorrow: Large Language Models and Jobs," World Economic Forum, 2023.
- [4] S. Peng, E. Kalliamvakou, P. Cihon and M. Demirer, "The Impact of AI on Developer Productivity: Evidence from GitHub Copilot," arXiv, 2023.
- [5] G. Sandoval, H. Pearce, T. Nys, R. Karri, S. Garg and B. Dolan-Gavitt, "Lost at C: A User Study on the Security Implications of Large Language Model Code Assistants," arXiv, 2022.
- [6] S. Imai, "Is GitHub Copilot a Substitute for Human Pair-programming? An Empirical Study," in 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings, Pittsburgh, USA, 2022.
- [7] M. Conway, "IBM Study: As CEOs Race Towards Generative AI Adoption, Questions Around Workforce and Culture Persist," 16 May 2024. [Online]. Available: https://www.prnewswire.com/news-releases/ibm-study-as-ceos-racetowards-gen-ai-adoption-questions-around-workforce-and-culture-persist-302147188.html.
- [8] M. Murati, Interviewee, *AI Everywhere: Transforming Our World, Empowering Humanity.* [Interview]. 19 June 2024.
- [9] S. Abdulhadi, "Exploring Students; Perceptions of ChatGPT: Thematic Analysis and Follow-Up Survey," *IEEE Access*, vol. 11, pp. 38805-38818, 2023.
- [10] P. Rajabi, P. Taghipour, D. Cukierman and T. Doleck, "Exploring ChatGPT's impact on post-secondary education: A qualitative study," in *Accociation for Computer Machinery*, New York, 2023.

- [11] J. Pranther, P. Denny, J. Leinonen, B. A. Becker, I. Albluwi, M. Craig, H. Keuning, N. Kiesler, T. Kohn, A. Luxton-Reilly, S. MacNeil, A. Petersen, R. Pettit, B. N. Reeves and J. Savelka, "The Robots Are Here: Navigating the Generative AI Revolution in Computing Education," in *Association for Computing MAchinery*, New York, 2023.
- [12] J. Savelka, A. Agarwal, M. An, C. Bogart and M. Sakr, "Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses," in *Association for Computing Machinery*, New York, 2023.
- [13] C. C. Tossell, N. L. Tenhundfeld, A. Mormen, K. Cooley and E. J. de Visser, "Student Perceptions of ChatGPT Use in a College Essay Assignment: Implications for Learning, Grading, and Trust in Artivicial Intelligence," *IEEE Transactions on Learning Technologies*, vol. 17, pp. 1069-1081, 2024.
- [14] L. Li, "The Impact of Artificial Intelligence on Higher Art Education," in *Association for Computing Machinery*, New York, 2024.
- [15] P. Bassner, E. Frankford and S. Krusche, "Iris: An AI-Driven Virtual Tutor for Computer Science Education," in *Association for Computing Machinery*, New York, 2024.
- [16] W. Lyu, Y. Wang, T. (. Chung, Y. Sun and Y. Zhang, "Evaluating the Effectiveness of LLMs in Introductory Computer Science Education: A Semester-Long Field Stiudy," in Association for Computing Machinery, New York, 2024.
- [17] W. H. Pan, M. J. Chok, J. L. S. Wong, Y. X. Shin, Y. S. Poon, C. Y. Chong, D. Lo and M. K. Lim, "Assessing AI Detectors in Identifying AI-Generated Code: Implications for Education," in *Association for Computing Machinery*, New York, 2024.
- [18] L. Harris and C. Jaikaran, "Highlights of the 2023 Executive Order on Artificial Intelligence for Congress," Congressional Research Service, Washington, D.C., 2024.
- [19] U.S. Library of Congress, "S.1353 117th Congress (2021-2022)," 19 December 2022. [Online]. Available: https://www.congress.gov/bill/117th-congress/senatebill/1353.
- [20] U.S. Library of Congress, "Summary: H.R.2575 116th Congress (2019-2020)," 14 September 2020. [Online]. Available: https://www.congress.gov/bill/116thcongress/house-bill/2575#:~:text=Passed%20House%20(09%2F14%2F2020),-AI%20in%20Government&text=This%20bill%20creates%20the%20AI,within%20 the%20General%20Services%20Administration.&text=carry%20out%20such%20a ctivities%20f.
- [21] U.S. Library of Congress, "H.R.6943 No AI FRAUD Act," 10 January 2024. [Online]. Available: https://www.congress.gov/bill/118th-congress/housebill/6943/text?s=1&r=9.
- [22] Future of Life Institute, "High-level summary of the AI Act," 27 February 2024. [Online]. Available: https://artificialintelligenceact.eu/high-level-summary/.
- [23] N. Kshetri, "Economics of Artificial Intelligence Governance," *Computer*, vol. 57, no. 4, pp. 113-118, April 2024.
- [24] Data & Trust Alliance, IBM, "The urgency of AI governance," 2021. [Online]. Available: https://www.ibm.com/downloads/cas/MV9EXNV8.
- [25] L. Yang, "Research on the legal regulation of Generative Artificial intelligence— Take ChatGPT as an example," in *SHS Web of Conferences*, Les Ulis, France, 2023.
- [26] U.K. Ministry of Defense, "Ambitious, safe, responsible: our approach to the delivery of AI-enabled capability in Defence," 15 June 2022. [Online]. Available: https://www.gov.uk/government/publications/ambitious-safe-responsible-our-approach-to-the-delivery-of-ai-enabled-capability-in-defence/ambitious-safe-responsible-our-approach-to-the-delivery-of-ai-enabled-capability-in-defence.
- [27] L. Zhang, "China: regulation of artificial intelligence," The Law Library of Congress, Global Legal Research Directorate, Washington, D.C., 2023.
- [28] R. Uber, "China's artificial intelligence ecosystem," National Intelligence University (U.S.), Washington, D.C., 2021.
- [29] C. Parker and C. Zakrzewski, "Italy Temporarily Bans ChatGPT Over Privacy Concerns," 31 March 2023. [Online]. Available: https://www.washingtonpost.com/world/2023/03/31/italy-ban-chatgpt-artificialintelligence-privacy/.
- [30] K. Chan, "OpenAI to Offer Remedies to Resolve Italy's ChatGPT Ban," 6 April 2023. [Online]. Available: https://apnews.com/article/chatgpt-openai-data-privacyitaly-1e3f070ca86ec234cae4d08ac8443879.
- [31] S. Lim and H. Shim, "No Secrets Between the Two of Us: Privacy Concerns Over Using AI Agents," *Cyberpsychology: Journal of Psycholosocial Research on Cyberspace*, 2022.
- [32] C. Duffy, "Chat GPT is Getting Chattier with 'Advanced Voice Mode'," 30 July 2024. [Online]. Available: https://www.cnn.com/2024/07/30/tech/openai-chatgptadvanced-voice-mode-rollout/index.html.
- [33] D. Kalla, S. Kuraku, N. Smith and F. Samaah, "Study and Analysis of ChatGPT and its Impact on Different Fields of Study," *International Journal of Innovative Science and Research Technology*, 2023.
- [34] L. Matlali, "Cybersecurity and AI: The Challenges and Opportunities," 5 June 2023. [Online]. Available: https://www.weforum.org/agenda/2023/06/cybersecurity-and-ai-challengesopportunities/.
- [35] A. Yampolskiy, "What does 2024 have in store for the world of cybersecurity?," 15 February 2024. [Online]. Available: https://www.weforum.org/agenda/2024/02/what-does-2024-have-in-store-for-theworld-of-cybersecurity/.
- [36] A. Jones and M. Omar, "Codesentry: Revolutionizing Real-Time Software Vulnerability Detection With Optimized GPT Framework," *Land Forces Academy Review*, pp. 98-107, 2024.
- [37] D. Elliott and E. Soifer, "AI Technologies, Privacy, and Security," *Frontiers in Artificial Intellignece*, 2022.

- [38] S. Longpre, R. Mahari, N. Obeng-Marnu, W. Brannon, T. South, K. Gero, S. Pentland and J. Kabbara, "Data Authenticity, Consent, & Provenance for AI Are All Broken: What Will it Take to Fix Them?," Cornell University Library, Ithaca, 2024.
- [39] "Artificial Intelligence and Intellectual Property: Part I Interoperability of AI and Copyright Law," in *United States. Congress. House. Committee on the Judiciary. Subcommittee on Courts, Intellectual Property, and the Internet*, Washington, D.C., 2023.
- [40] B. Brittain, "Getty Images lawsuit says Stability AI misused photos to train AI," 6 February 2023. [Online]. Available: https://www.reuters.com/legal/getty-imageslawsuit-says-stability-ai-misused-photos-train-ai-2023-02-06/.
- [41] T. B. Lee, "Stable Diffusion copyright lawsuits could be a legal earthquake for AI," 3 April 2023. [Online]. Available: https://arstechnica.com/techpolicy/2023/04/stable-diffusion-copyright-lawsuits-could-be-a-legal-earthquakefor-ai/.
- [42] R. Birštonas, "Risks to Employees' Intellectual Property Rights Posed by Artificial Intelligence," *Law: Journal of the University of Latvia*, vol. 16, pp. 49-59, 2023.
- [43] C. Schulze, "German Federal Patent Court points to solution for Dabus inventions," 16 November 2021. [Online]. Available: https://www.juvepatent.com/cases/german-federal-patent-court-points-to-solution-for-dabusinventions/.
- [44] A. Magauiya, A. Omarova, A. Kasenova, Z. Akhmetov and M. Akhmadi, "The Practices of Advanced Countries in the Legal Regulation of Intellectual Property Objects Created by Artificial Intelligence," *Revista de Direito, Estado e Telecomunicações*, vol. 15, no. 1, pp. 191-206, May 2023.
- [45] L. J. Tiedrich, G. S. Discher, F. Argent and D. Rios, "10 Best Practices for Artificial Intelligence-Related Intellectual Property," *Intellectual Property & Technology Law Journal*, vol. 32, no. 7, pp. 3-7, 2020.
- [46] C. T. Zirpoli, "Artificial Intelligence Prompts Renewed Consideration of a Federal Right of Publicity," Congressional Research Service, Washington, D.C., 2024.
- [47] J. Coyle, "In Hollywood writers' battle against AI, humans win (for now)," 27 September 2023. [Online]. Available: https://apnews.com/article/hollywood-aistrike-wga-artificial-intelligence-39ab72582c3a15f77510c9c30a45ffc8.
- [48] C. Frawley, "Unpacking SAG-AFTRA's New AI Regulations: What Actors Should Know," 18 January 2024. [Online]. Available: https://www.backstage.com/magazine/article/sag-aftra-ai-deal-explained-76821/.
- [49] D. Richardson, "Hollywood's AI issues are far from settled after writers' labor deal with studios," 16 October 2023. [Online]. Available: https://www.cnbc.com/2023/10/16/hollywoods-ai-issues-are-far-from-settled-afterwga-deal.html.

# Applications of Reinforcement Learning in Competitive Robotics

Andrew Needham	Evan Roegner	Joseph Beattie	Andy Dao
	Leigh Goe	tsch	

Department of Computer Science & Software Engineering

Milwaukee School of Engineering

Milwaukee, WI 53202

{needhama roegnere beattiej daoa goetschm}@msoe.edu

### Abstract

The VEX Artificial Intelligence Robotics Competition (VAIRC) presents a challenge where robots compete autonomously within a dynamic environment. Reinforcement learning is a promising approach to facilitate complex decision-making behaviors in the VAIRC. Currently, simulated environments are used to train VAIRC systems, but individual teams typically develop these environments in private. We use Gymnasium to create a modular environment for a lightweight training and implementation system that emphasizes game strategy and adaptability. In future work, we aim to explore collaborative strategies between two robots using this system. Our work highlights the potential of integrating reinforcement learning with robotics to apply strategy optimization in real-time decision-making.

# 1 Introduction

The VEX AI Robotics Competition (VAIRC) is an adaptation of the VEX V5 Robotics Competition where teams play in one-vs-one matches using two fully autonomous robots per team. Robots function without input from drivers and instead communicate with each other as the match progresses through two minutes (2024 - 2025 Game Manual, 2025). This is possible because the robots have computers on board that collect information from the environment and make decisions based on that information.

The competition is divided into two main parts: "skills" and "matches." In the "skills" sessions, teams perform without opponents and focus on maximizing the number of points earned following the rules of the game, depending on actions taken by the robot. There exists a leaderboard containing the highest-scoring teams. In the "matches" sessions, teams compete against each other in one-vs-one matches. The team with the most points at the end of the match wins.

A common approach to game strategy optimization is reinforcement learning. Reinforcement Learning (RL) is a field of machine learning that uses a reward to guide the actions of an agent in a given environment. The agent receives feedback from the environment in the form of rewards or penalties, which it uses to improve its decision-making process over time. The goal of the agent is to maximize the cumulative reward it receives from the environment. This is especially useful in the VAIRC, where the environment is dynamic and the robots must make decisions based on the information they have available.

Prior approaches using RL have resulted in robots with high performance in specific tasks. One major issue with existing approaches is the computational intensity of model training. In a prior paper, the training process involved the operation of a physics simulation mimicking the robot field, which made training a slow process (Zietek et al., 2022).

Our system focuses on optimizing an agent to score the most points through higher-level actions in a simplified environment. This has the advantage of reducing the amount of time taken to train a high-performing model. It also simplifies the creation of a simulation implementing the game mechanics. RL solutions in other domains could take advantage of this foundation to reduce the required computational complexity when training agents.

To assist in evaluating these higher-level actions, we used the CasADi library. This is an optimization library used to solve complex mathematical problems. One use case for this tool is for planning paths, taking object footprints and obstacles into consideration. We used this when actions were evaluated after training a RL model.

In the following sections, we will go through how we have put together our system in detail. This includes setting up each component for the VEX competition, as well as the techniques used to integrate components.

# 2 Approach

### 2.1 Overview

The typical approach when creating a RL model for robotics applications is to create a virtual environment in which the model can learn. These virtual environments can be a simplified version of a specific scenario, or they can be extremely detailed representations of the real world. For our approach, we opted for a relatively simple environment that simulates the VEX AI competition field. We combined this with other existing programs to create abstract actions that virtual agents can take within the environment.

Our goal in using an abstract environment for the agent to train in is to reduce the overhead needed to construct and maintain an environment. This is especially important because the game changes every year and the environment needs to be updated to reflect the new game. In this way, the more complex aspects of the game can be abstracted away and the agent can focus on the game strategy. Since the environment is modular and easier to update than a detailed simulation, it also better supports faster and more flexible development of the agent. Some examples include changing the environment starting state based on match and skills field setup, or adding new elements to the environment, such as additional rings.

### 2.2 Path Planner

The most important part of the skills competition is scoring as many points as possible. To do this, the robot needs to be able to navigate the field efficiently. Our first step was to develop a program to generate an optimal path from one point on the field to another while avoiding obstacles. We decided to use CasADi, a computer algebra solver for nonlinear optimization problems. The original project we used as a basis was created by tianchenji and posted on a public repository on GitHub (tianchenji, 2020). This project was adapted to fit our use case.

This program takes two points on the field and outputs an optimal path from one point to another. The program works by creating a nonlinear programming problem that can be solved by CasADi. The first step is to define the dimension of the robot and the field as well as the size and position of all the obstacles. To simplify the collision detection, the robot and obstacles are all represented as a circle that circumscribes each vertex of the robot or obstacle.

The next step is creating a linear interpolation between the two points. This provides a sufficient estimate for the solving algorithm to then find an optimal solution efficiently. An array of variables is generated which describes the position and velocity of the robot at each point. Additionally, a single variable is used to set the time interval between each point in order to determine the distance between each point.

Once all the variables are defined, an additional array of dynamic constraints are created based on those states. These constraints limit the position of each point by taking into account things like the maximum speed, and acceleration of the robot, as well as collisions with obstacles.

The position of each point is calculated by using the kinematic equations of motion:

$$\vec{r} = \vec{r}_0 + \vec{v}\Delta t$$

In this equation,  $\vec{r}$  is the position of the current point,  $\vec{r_0}$  is the position of the point before,  $\vec{v}$  is the velocity at the point, and  $\Delta t$  is the time interval between the points. The equality constraint ensures the position of each point is directly determined by the equation. Upper and lower bounds for each point are also set so that the position of the robot doesn't go outside the field limits and that the velocity doesn't exceed the maximum speed of the robot.

Even though upper and lower bounds were already set to limit the individual components of position and velocity, additional constraints need to be added to ensure the speed of the robot does not exceed the maximum speed calculated with inequality:

$$|\vec{v}| \leq |\vec{v}_{\max}|$$

In this inequality,  $|\vec{v}|$  is the magnitude of velocity at the current point and  $|\vec{v}_{max}|$  is the maximum speed of the robot.

Similarly constraints are put on the acceleration of robot to ensure the acceleration does not exceed the max acceleration at any point with this inequality:

$$\vec{a} = \frac{\vec{v} - \vec{v_0}}{\Delta t}$$
$$|\vec{a}| \le |\vec{a}_{\max}|$$

In these equations,  $\vec{a}$  is the acceleration of the robot at the point,  $\vec{v}$  and  $\vec{v_0}$  is the velocity at the current point and point before respectively,  $\Delta t$  is the time interval between each point, and  $|\vec{a}|$  and  $|\vec{a}_{\max}|$  is the magnitude of acceleration at the point and maximum magnitude of acceleration.

Finally, an additional constraint is created for each obstacle to ensure the robot does not collide with any obstacle. The robot and each obstacle is represented as a circle so checking for to check for collision equates to checking the distance between the robot and the obstacle. An additional buffer is added to ensure any uncertainty in the real world is accounted for.

$$d = |\vec{r_1} - \vec{r_2}|$$
$$d \ge R_1 + R_2 + \epsilon$$

In these equations, d is the distance between the center of the robot and obstacle,  $\vec{r_1}$  is the position of the robot and  $\vec{r_2}$  is the position of the obstacle,  $R_1$  and  $R_2$  is the radius of the robot and obstacle, and  $\epsilon$  is a small buffer between obstacles. The inequality enforces that the distance between the two points is greater than the sum of the radius of the robot and obstacle.

Once all the variables and restrictions are set, a cost function is created for the nonlinear programming (NLP) problem. In our case, the cost function is the total time taken to go from the start to end point. However, because the number of points remains constant when

solving the problem, the cost function can be simplified to using just the time interval between points:

 $c = \Delta t$ 

The NLP solver will then attempt to find a solution in which this cost is minimized. The algorithm we chose to use in our case is Interior Point Optimizer (IPOPT) due to its efficiency at quickly solving large-scale problems with many nonlinear constraints. Once the solver finds a solution, the list of points can be passed on for the robot to follow using the existing odometry system.

#### 2.3 Virtual Environment

Our next step was to create the virtual environment for the agent to train in ("VEX AI Reinforcement Learning", 2025). This environment needs to effectively model all aspects of scoring in the game while simplifying many of the aspects and physics of the real world. To achieve this, we used Gymnasium, an open-source Python library for developing reinforcement models. Gymnasium provides a large amount of flexibility allowing us to create an environment that fits our needs ("Gymnasium Documentation", 2025). For the environment model, we used the Stable Baselines3 MultiInputPolicy (MIP) model along with the Proximal Policy Optimization (PPO) algorithm to train the model ("Policy Networks Documentation", 2025; "Proximal Policy Optimization Documentation", 2018). The MIP model architecture allows us to feed many inputs with different sizes rather than a single vector of inputs, and the PPO algorithm helps train the model quickly and efficiently.

Similarly to the path planner system, the first step is to define the objects in the environment including the robot, field, and field elements. Once the environment set up, the observation space and action space needs to be defined for the virtual agent in order for the agent to understand and interact with the environment.

The observation space is not the same as the environment state since the real robot will only have so much information. We can provide information such as the robot's position and orientation, the amount of time left in the match, the position of elements in the robots field of view (FOV) as well as how many rings and goals the robot is holding. This observation space is set to model what can be determined using the sensors we have available to use.

The action space is a relatively short list of possible actions the robot can perform. Some examples include "Pick up nearest goal," "Pick up nearest ring," "Drop goal," "Turn towards center," "Add ring to goal," among several others. These abstract actions provide the agent with a simple way to determine how to interact with the environment.

The final step needed for the reinforcement algorithm is generating a reward based on what action the robot takes. The reward function we use is given by this equation:

$$r = (S - S_0) + p$$

The reward, r is calculated by the difference between the score after taking the action, S and the initial score,  $S_0$  plus any penalty, p that is given based on an invalid action. The score is calculated using criteria based on the VEX competition manual (2024 - 2025 Game Manual, 2025). The penalty is only added if the action the model chooses is invalid, and

the default penalty for all invalid actions is -0.1 to ensure that the penalty doesn't override any reward based on the action. We put a higher penalty on actions that might give a reward but cause longer lasting negative such as climbing too early. Table 1 demonstrates some examples of rewards given to different actions.

Action	Initial score	Final score	Penalty	Final Reward
Successfully drop the goal in an empty corner (valid action)	8	13	0	5
Attempt to pick up a goal that is already held (invalid action)	13	13	-1	-1
Attempt to climb too early in the match (invalid action)	13	16	-10	-7

Table 1: Reward examples

Table 1 shows what rewards are given to a few actions in specific scenarios. Valid actions will always have a penalty of 0, and invalid actions will have different penalties less than 0 based on its severity. For example, attempting to pick up a goal that is already being held will have little to no consequence as the action just can't be performed, so it is assigned the default penalty of -1. On the other hand, climbing too early in the match leaves the robot unable to perform any more actions for the rest of the episode, so even though it scores 3 points, it is given a harsh penalty of -10 in order to restrain that behavior.

The general training loop for the process is as follows and is shown in Figure 1. First the agent evaluates the current state of the field to determine what action to take. After selecting an option, the agent will use the information it has available to perform that action in the environment. If an action is invalid or impossible to perform given the current state of the environment, the agent is given a penalty for that step. If the agent successfully performs an action, the score of the field is determined by the state of the environment according to the rules of the VEX AI competition. Then, the reinforcement model is adjusted by taking the reward into account and the process is repeated. If the environment reaches a state where the episode is over (i.e. running of time), the environment is reset and the process continues.





Finally, once the model is trained, a final simulation can be run to record all the actions taken by the agent which can be converted into a preplanned route for autonomous coding skills. The environment outputs a list of steps along with the calculated coordinates of each path to a text file, which can be then be parsed by an additional program to create the C++ code necessary for the robot to run each action.

### 2.4 Integration

Our final step was to integrate these programs with the existing program to control the physical robot. As touched on in the previous section, a program was created to convert the output of the RL model into usable C++ code.

Integration also involved, in large part, the use of an Nvidia Jetson Orin Nano Development board to facilitate an eventual real-time agent implemented directly on a robot.

The development board is based on the Nvidia Jetpack SDK - a modified version of the Ubuntu Linux distribution. Technologies utilized in addition to the Nano included the Intel

RealSense D455 depth camera, which, when paired with VEX software could be used to detect important game pieces and give other valuable data.

The use of the PyRealSense library allowed functionality of the VEX Web Dashboard, which serves as a central point for viewing information returned by the robot's various sensors on the Nano, including position readout and objects detected by the RealSense camera.





In addition to the information displayed in the Web Dashboard, custom data packages were able to be sent from the robot's brain and received on the Nano, which could facilitate further communication and the implementation of a real-time agent.

Due to the architecture of the Nano, which was set up similarly to a desktop computer, it was trivial to get the RL model running on it. Being written in Python, there were library dependencies which needed to be installed. After installing those libraries, running the model worked similarly to how it was run on ROSIE.

Another challenge in getting the Jetson Orin Nano to function directly on the robot is simply powering it. The Nano must be powered externally, separate from the robot's other systems. To do this, a USB power hub was used. The power hub we had access to outputted power at 5 Volts from either USB-A or USB-C, while the Nano can only accept power at between 9 and 19 Volts over 5.5mm x 3.5mm barrel jack. To circumvent this difference, a voltage boost converter was proposed to be used to convert both voltage and cable medium. The system diagram of all the hardware components can be found in Figure 3.



Figure 3: VEX AI System Diagram ("VEX AI System", 2024)

# **3** Discussion and Results

### 3.1 ROSIE Supercomputer Use

The agents simulated in the virtual environment were trained using the Milwaukee School of Engineering ROSIE Supercomputer. Our training processes relied on using a single angent inside one environment, so, rather than using GPUs, the training was performed using an Intel Xeon Gold 6240 CPU 2.60 GHz. Running the training on ROSIE let training of the model to take place quickly without needing access to local processing power. Because of this other tasks for development could be done without slowing down the process of training the model.

### 3.2 Simulation

The path planning algorithm provided the ability to generate optimal paths quickly based on a dynamic environment that changes rapidly. Over the course of 1000 trials with the robot and 5 goals as obstacles randomly placed on the field, the average time to solve was 0.127 seconds and the percentage of successfully generated paths was 32.5%. Table 2 records the average time to generate paths for feasible (successful) and unfeasible (unsuccessful) paths.

Status	Average time (seconds)
Successful	0.122
Unsuccessful	0.129
Overall	0.127

 Table 2: Path generation results

The results show that paths can be generated quickly regardless of whether there is a feasible path to the target. Unsuccessful paths are typically only found when the target or starting destination overlaps an obstacle, making it impossible to generate a path between the two points. For example, see Figure 4 and Figure 5 as an example of a feasible and unfeasible path. Notice that in Figure 5, the robot is overlapping an obstacle at the starting point.



Figure 4: Feasible path

Figure 5: Infeasible path

The Jetson Nano uses an ARM Cortex-A78AE CPU 1.5 GHz. Based on single threaded benchmarks of both CPUs, we can estimate that the average time to generate a path on the Nano would be 0.266 seconds ("CPU Benchmarks", 2025). This is an acceptable amount of time for our use case as other processes can be run on separate threads while a path is being generated.

For the virtual environment, we were able to use the simulation to generate preplanned routes for the skills competition using one robot. The state of the environment is known at the start and the single robot is the only thing interacting with the environment, so the state is completely deterministic. Because of this, we used this setup to test the training performance. The model architecture we used has just two hidden layers of 64 nodes each, keeping it relatively lightweight and easy to train.

The images below show visual representations of the environment at different steps in the simulated episode run by the final trained model. Figure 6 shows the state at the beginning of the episode (Step 0), and Figure 9 shows the state at the end (Step 52). The number of steps is not the same for each episode as each action can take a different amount of time.



Figure 8: Step 36

Figure 9: Step 52

The model we trained over 5 million steps was able to achieve a maximum score of 52 points. This is 8.3% higher than an initial, naive solution, that was manually created which scored 48 points. This shows that the reinforcement learning model was able to independently find optimal strategies that were not obvious at first. In additional testing, each model was trained for 10 million steps, and the scores were averaged to get the results below. Figure 10, shows the average score and average best score of the three trained models over the total course of all training steps.





Figure 10 shows that although the score of each model fluctuates drastically throughout the training process, the best score continues to go up at a decreasing rate. In fact, after about 6.6 million steps, there are no improvements to the best score showing that the model converges to a adequate strategy relatively quickly.

Based on an additional test, the average steps per second during training was found to be 602.5 steps per second. We used this to estimate the time to train the model at any number of steps. Table 3, shows the best average score at logarithmic intervals of steps including the estimated time to train.

Step	Best Score	Time (seconds)
10240	6.7	17.0
20480	11.0	34.0
40960	21.7	68.0
81920	35.7	136.0
163840	44.0	271.9
327680	46.3	543.9
655360	48.7	1087.8
1310720	50.3	2175.6
2621440	52.0	4351.1
5242880	53.7	8702.3
10004480	54.0	16605.7

Table 5. Fall generation result	Table 3:	Path	generation	results
---------------------------------	----------	------	------------	---------

These results show that the model quickly reaches an average best score of 44.0 by 165,000 steps and 48.7 by 656,000 steps, which is just over 80% and 90% of the maximum score of 54, respectively. Reaching a score of 48 took just about 18 minutes, and training the model for 10 million steps still took less than 5 hours.

Several adjustments were made to the environment throughout development to keep training as efficient as possible. For instance we consolidated actions to keep the total of number of actions to a minimum. For instance, we combined "Drive to nearest goal" and "Pick up goal" to just "Pick up nearest goal." This provided less options for the model to choose from and allowed it to learn optimal strategies more quickly.

We also performed some testing to determine what model hyper-parameters led to the most optimal training time. We found that a learning rate of 0.001, a discount factor of 0.995, and an entropy of 0.0 gave the best results. These values provided a balance between exploiting effective strategies and experimenting with others, as well as considering both short-term and long-term rewards while maintaining a consistent strategy.

When training the model for competition matches, we added an option to limit the game elements the robot can see in the simulation to just those in a specific FOV as well as an option to randomize the positions of the robot and all the game elements. This helped generalize the model rather and prevented overfitting to the initial game state. Doing this helped keep performance high even with field states other than the initial setup, which ensures the model will output sufficient actions in actual matches.

#### **3.3** Drawbacks and Challenges

A significant challenge with our approach to actions in the RL environment is that it is difficult to translate the higher-level actions output by our model into lower-level actions the robot can interpret. The exact code the robot should run for a given action will vary based on factors like the distance and angle of a given item, as the robot needs to approach

an item correctly in order to interact with it. As a result, although the theoretical results seem promising, we have not tested it extensively on a physical field.

We also encountered considerable difficulty when attempting to use the VEX-provided instructions to configure the Jetson Nano, particularly setup of the PyRealSense library for Python 3.10. The library's functionality was eventually achieved through steps outlined in a thread on GitHub (Drkstr, 2020).

## 4 Conclusion

In this work, we have demonstrated an implementation of a Gymnasium environment that abstracts the simulation for the VAIRC. Our approach highlights the decision-centered approach to encourage complex behavior in agents that can be facilitated through an integrated system with a Nvidia Jetson Orin Nano. This higher focus on strategic behavior will serve as a basis for RL agents and involvement in future VAIRC contexts supported by abstract environments and actions.

For the VAIRC, the implementation of our environment and how the RL agent approaches solving the task results in less intensive cycles for training with a greater focus on making strategic gameplay decisions. For future projects, we could look at mimicking this level of play to accommodate multiple agents to compete with each other in a match. This also could lead to prospects with training the model on varying degrees of simulated hardware ability to better simulate VAIRC games with variable robotics hardware between teams.

Another component we would like to explore is real-time decision making. Possible prospects in this space would involve further work with the Jetson Nano and VEX V5 Brain to process incoming inputs in real time to observation inputs for a model.

This also opens up more potential research into RL environment tuning for levels of abstraction and interaction mediums for a model and real-world system. Further research on how models can handle environment discrepancies can better outline adaptability and generalization between RL techniques and the potential efficacy of focusing on strategic decisions and realistic RL environments.

We are excited for the application of our Gymnasium environment in the VAIRC space and the future of robotics embedded systems. We plan to expand our work with this environment to the VAIRC 2025-2026 problem and expand our approach to handle multiple robots in a cooperative and adversarial manner.

## References

- 2024 2025 game manual (3.0). (2025, January 28). VEX Robotics Inc. Retrieved March 20, 2025, from https://link.vex.com/docs/24-25/v5rc-high-stakes/GameManual.
- *Cpu benchmarks*. (2025). PassMark Software. Retrieved March 20, 2025, from https:// www.cpubenchmark.net/compare/3613vs6300/Intel-Xeon-Gold-6240-vs-ARM-Cortex-A78AE-6-Core-1510-MHz.
- Drkstr. (2020, October 13). *Running pyrealsense2 on jetsonnano*. Retrieved March 19, 2025, from https://github.com/IntelRealSense/librealsense/issues/6964#issuecomment-707501049.
- *Gymnasium documentation*. (2025, March 6). Farama Foundation. Retrieved March 20, 2025, from https://gymnasium.farama.org/index.html.
- Policy networks documentation. (2025, January 27). Stable Baselines3. Retrieved March 17, 2025, from https://stable-baselines3.readthedocs.io/en/master/guide/custom\_policy.html.
- Proximal policy optimization documentation. (2018). OpenAI. Retrieved March 17, 2025, from https://spinningup.openai.com/en/latest/algorithms/ppo.html.
- tianchenji. (2020, June 9). *Collision avoidance path planning algorithm*. Retrieved November 12, 2024, from https://github.com/tianchenji/path-planning.
- Understanding the parts of the ai system. (2024, September 19). VEX Robotics Inc. Retrieved December 10, 2024, from https://kb.vex.com/hc/en-us/articles/360048489612-Understanding-the-Parts-of-the-AI-System.
- *Vex ai reinforcement learning*. (2025, March 10). MSOE Raider Robotics. Retrieved February 8, 2025, from https://github.com/msoe-vex/VEX-AI-Reinforcement-Learning.
- Zietek, J., Wade, N., Roberts, C., Malek, A., Pylla, M., Xu, W., & Patil, S. (2022, November 25). Pac-man pete: An extensible framework for building ai in vex robotics. https: //arxiv.org/abs/2211.14385.



# Wildfire Spread Prediction Using Deep Learning

Michael Wood, Adam Swedlund, Mason Beynon, Zoe Kirkman, Reagan Burkemper, Sydney Balboni\*

Department of Electrical Engineering and Computer Science

Milwaukee School of Engineering

1025 N Broadway St, Milwaukee, WI 53202

woodm@msoe.edu, swedlunda@msoe.edu, beynonm@msoe.edu, kirkmanz@msoe.edu, burkemperr@msoe.edu, balbonis@msoe.edu

# Abstract

In this work, we explore deep learning-based fire spread prediction using the previously developed WildfireSpreadTS dataset, collected by a different research team prior to our study. Our primary goal is to accurately model the dynamics of wildfire propagation by leveraging Convolutional Neural Networks (CNNs) and U-Net architectures. We conduct extensive experiments that involve systematic hyperparameter tuning to find optimal configurations for capturing complex spatial features of fire progression. To handle the inherent challenges of segmentation and reduce prediction errors, we use the Intersection over Union (IoU) loss function, providing a robust measure for target-object overlap. Preliminary results demonstrate that this integrated approach effectively identifies fire-prone regions while maintaining strong predictive performance, highlighting the potential of deep learning methodologies for supporting timely wildfire management and resource allocation decisions.

# **1** Introduction

Wildfires continue to pose a severe threat to ecosystems, infrastructure, and human communities worldwide. Over recent decades, climate change, population expansion, and land-use practices have contributed to more frequent and larger fire events [1]. These increasingly catastrophic blazes not only damage ecological habitats but also cause significant socio-economic losses. While traditional simulation approaches, such as FARSITE and Prometheus, have been widely adopted for modeling fire behavior, they often require extensive domain knowledge and manual tuning to reflect real-world complexity. As a result, machine learning (ML) and deep learning (DL) approaches have garnered substantial attention in wildfire spread prediction due to their ability to learn complex, non-linear relationships directly from data [2].

Among various DL architectures, U-Net has emerged as a popular choice for segmentation tasks because of its encoder–decoder structure and skip connections, enabling the capture of both high-level semantic features and low-level spatial details [2]. In this context, U-Net's capacity to handle pixel-wise predictions makes it particularly useful for modeling fire propagation, as it allows for fine-grained segmentation of prospective fire-perimeter regions. Recent work demonstrates that architectures based on convolutional neural networks (CNNs) often outperform purely physics-based methods by effectively leveraging large-scale remote sensing, meteorological, and topographic data [1].

In this study, we use a U-Net architecture to predict wildfire spread based on the WildfireSpreadTS dataset—specifically focusing on the 2018 subset [3]. This dataset's multi-temporal, multi-modal nature offers a valuable benchmark for deep learning algorithms, as it includes spatial resolutions that facilitate image-like representation of fire spread dynamics over consecutive days. Our aim is twofold: first, to demonstrate how U-Net can accurately learn the progression of fire fronts from historical satellite observations and weather variables; and second, to highlight the significance of proper loss functions—particularly Intersection over Union (IoU)—in improving predictive accuracy when dealing with highly imbalanced fire masks. By leveraging IoU, we directly optimize the overlap between predicted burned regions and ground-truth data, thereby addressing the class imbalance inherent in wildfire segmentation tasks [4].

Additionally, recent advances in multimodal data fusion have demonstrated the benefits of integrating satellite imagery, topographical features, and real-time climate indicators to capture the spatial and temporal complexity of fire behavior [5]. The breadth of data available from Google Earth Engine and other remote-sensing platforms allows for daily updates of environmental conditions—such as humidity, wind, vegetation indices, and land cover—that critically influence fire expansion.

This paper is structured as follows. Section 2 presents our data sources and describes the WildfireSpreadTS dataset, focusing on the 2018 fire season. Section 3 details the methodology, including data preprocessing, model architecture, and training strategies. In Section 4, we provide experimental results, analyzing model performance using IoU. Finally, we conclude by

summarizing our findings, discussing limitations, and suggesting future directions for scaling this approach to broader geographic regions and longer forecast horizons.

# 2 Data

## 2.1 Data Collection

We obtained wildfire data through repositories curated using Google Earth Engine and GRIDMET (the Gridded Surface Meteorological dataset). The raw geospatial data was transformed and stored in TIFF files, used in the final training for predicting fire spread. Ultimately, the data was preprocessed into clean input tensors.

From the final dataset, we took the wildfire data from the year 2018 and specifically looked at VIIRS (Visible Infrared Imaging Radiometer Suite) bands, which are satellite-based images providing data on fire detection, vegetation, and burned areas [6]. The data from GRIDMET includes wind speed and direction, temperatures, and humidity data. From MODIS, there is information about elevation and land cover. All components were compiled into TIFF images keyed by the date of the fire data, offering a distinct, multi-parameter view of fire spread.

## 2.2 Data Cleaning

While a variety of data points are employed in different portions of the overall pipeline—as outlined in Section 3, 'Methods'—each compiled dataset goes through a standard cleaning process to ensure spatial and temporal consistency. TIFF files were checked to ensure that each pixel corresponds to the same physical location, and some files were cropped to zoom in on the actual burning area, with a buffer of a few kilometers. Lastly, continuous features were normalized to the [0,1] range and fire labels were made binary (fire vs. no fire).

# **3 Methods**

## 3.1 Model Selection and Rationale

We chose a U-Net-based framework due to its proven success in segmentation tasks requiring precise boundary delineation [2]. By leveraging an encoder–decoder structure with skip connections, U-Net captures both high-level semantic features and low-level spatial details. Compared to traditional machine learning methods, convolutional neural networks (CNNs) can learn complex, nonlinear relationships directly from large-scale geospatial data [1]. Consequently, adopting a U-Net architecture provides a strong baseline for leveraging the multi-modal nature of the WildfireSpreadTS dataset [3].

## **3.2 Model Architecture**



Figure 1: Diagram of our 3D U-Net model architecture.

As shown in Figure 1, our model employs a multi-branch 3D U-Net architecture that first separates the input tensor—representing three consecutive days of spatial data across 23 channels—into four modality-specific encoder branches (VIIRS, Weather, Forecast, and Static). Each branch processes its subset of channels through three sequential 3D convolutional blocks, each comprised of convolution + batch normalization layers followed by max pooling, capturing increasingly abstract features. The terminal outputs of these four encoder paths are concatenated and passed through a "bridge" layer, which further refines the merged feature representation via 3D convolution and dropout. On the decoder side, transposed convolutions progressively upsample the merged feature maps, and each stage integrates skip connections from the corresponding encoder blocks, gated by attention modules to focus on relevant spatial features. Finally, the last 3D convolution and reshape operation produce a single 2D fire-segmentation map, with a sigmoid activation to yield pixel-wise probabilities for fire vs. no fire.

200

## 3.3 Data Preprocessing

Before training our model, the 2018 WildfireSpreadTS TIFF files underwent a structured preprocessing pipeline. The goal was to transform raw geospatial data into a unified set of input tensors ready for segmentation. Key steps included:

#### **Spatial Alignment and Resampling**

Each of the 377 TIFF images was checked for coordinate system consistency. When necessary, data were reprojected or resampled to align with the baseline 375 m spatial resolution of the VIIRS active-fire product. This ensures that every pixel across different sources (e.g., weather raster layers, vegetation indices, topography) refers to the same ground location.

#### **Region of Interest (ROI) Cropping**

Fire events vary in their geographic extent, so many TIFF files covered large swaths of land. We cropped ROIs around known fire perimeters or ignition points, typically ensuring sufficient spatial context (e.g., a few kilometers buffer) beyond the actual burn extent. This step helps the model focus on relevant areas and reduces the computational load.

#### **Temporal Consolidation**

Because each daily TIFF captures a snapshot of conditions and active fires, we grouped consecutive days for each ROI into multi-day "stacks." This creates a sequence-like data structure that allows the downstream network to learn spatiotemporal patterns of fire spread over the course of consecutive days.

#### Normalization and Masking

During data preprocessing, all continuous bands (e.g., reflectance and temperature) were minmax scaled to the [0,1] range, ensuring a standardized input distribution. Any invalid pixels—for example, those occluded by smoke or cloud cover or corrupted by sensor issues—were tagged with a sentinel value (e.g., -1), enabling the model to learn how to handle or ignore these anomalies. Finally, the active fire labels were transformed into binary masks (fire vs. no fire), preparing the data for supervised segmentation tasks.

#### Train-Validation-Test Split

From the processed 3773 images, we allocated a portion for training, validation, and testing. Because fire behavior can be geographically and temporally distinct, we ensured that no single fire event was split across these subsets. Instead, entire events or contiguous ROIs were grouped to reduce spatial overlap among the partitions. A typical split ratio was used (e.g., 70–80% training, 10–15% validation, 10–15% testing), though exact percentages depended on the number of unique fire events available.

#### **Batch and Patch Creation**

Depending on GPU memory and model input constraints, large ROIs were sometimes tiled into smaller patches (e.g.,  $256 \times 256$  pixels) for batched training in the neural network. The "sliding window" or "tiling" approach ensures coverage of the entire burned area while making training feasible on standard hardware.

This end-to-end pipeline converts the multi-modal, daily TIFFs into uniform spatiotemporal samples for segmentation tasks, enabling our deep learning model to learn how wildfires typically progress under different environmental conditions. Further methodological details, including the specific network architecture and loss functions, are presented in the following subsections.

# **4** Experiments

## 4.1 Formatting Input Data

To train on daily wildfire progression, we organized spatiotemporal features into 3-day windows, each containing 23 channels (e.g., reflectances, weather variables, land cover). These were concatenated into a (3,300,220,23) tensor, giving the model a coherent snapshot of evolving fire events. This arrangement preserves necessary spatial detail while enabling the model to learn from day-to-day changes without excessive overhead.

## 4.2 Why IoU loss?

Intersection over Union (IoU), also known as the Jaccard Index, quantifies how well a predicted segmentation overlaps with the ground-truth mask by dividing the size of their intersection by the size of their union. This is particularly suitable for wildfire tasks, where missed fire pixels or excessive false alarms have real-world consequences. IoU also helps mitigate challenges of class imbalance by focusing on the overlap of positive predictions rather than raw accuracy [4]. For these reasons, we employ IoU to robustly evaluate segmentation performance.

### 4.2 Experiments Conducted

In our experimental phase, we conducted numerous training runs with variations in batch size, learning rate, and focal loss hyperparameters. From these, we selected five representative configurations that achieved the highest IoU scores, as summarized in Table 1. All experiments in this table had 100 epochs and a focal gamma of 2.0. The chosen experiments highlight how small changes in hyperparameters can meaningfully influence convergence and segmentation quality, especially in the context of highly imbalanced wildfire data. By examining each configuration's IoU score, we can see that more conservative learning rates and moderate focal alpha values tend to capture burn perimeters more accurately, indicating that overly aggressive updates or inadequate weighting of minority classes can undermine performance.

Batch Size	Learning Rate	Focal Alpha	IoU
9	0.0001	0.75	0.4012
8	0.0001	0.75	0.3880
7	0.0001	0.75	0.2821
8	0.001	0.75	0.2539
8	0.001	0.65	0.2497

Table 1: Our top five best performing experiments.

Among these five experiments, the best run attained an IoU of 0.4012, demonstrating the inherent challenges of segmenting sparse fire pixels. Complete metrics for this model show that it achieves a Loss of 2.6033, an Accuracy of 0.9980, a Precision of 0.5232, a Recall of 0.6384, and an F1 Score of 0.5751. The high Accuracy largely reflects the abundance of non-fire pixels, illustrating why metrics such as IoU, Precision, and Recall are more meaningful in this domain. Although the Recall indicates the model succeeds in capturing around two-thirds of the true fire pixels, there remains room to reduce false negatives, as these can have serious consequences in real-world fire management scenarios.

To illustrate the predictions produced by our best-performing model, we provide sample outputs comparing the model's binary masks with ground-truth fire labels in Figure 2. In several cases, the model effectively delineates fire fronts and identifies multiple ignition points, though small or fragmented burn regions sometimes remain challenging to detect. The probability maps also reveal occasional overestimation in regions with noisy reflectances or extreme weather signals, highlighting the need for additional preprocessing steps or domain-specific regularization. Despite these limitations, the obtained IoU scores establish a promising baseline for wildfire segmentation using 3D U-Net architectures, pointing to future avenues for improvement in data curation, model refinements, and integration of physics-based constraints.



Figure 2: Example predictions from our best performing experiment achieving an IoU score of 0.4012.

# **5** Challenges

### 5.1 Data Acquisition and Preprocessing

Our first major challenge lay in acquiring and preparing geospatial TIFF files that contained the necessary information for accurately modeling wildfires. Despite the variety of available data sources—ranging from satellite imagery to governmental agencies—each source provided data in different formats, resolutions, and timeframes. This inconsistency required extensive effort to align and merge multi-layered TIFF files. We had to ensure that the temporal and spatial

coverage overlapped, so our 3D U-Net would have comprehensive snapshots of environmental factors like temperature, vegetation density, and soil moisture.

Once obtained, the TIFF datasets went through several preprocessing steps to make them suitable for 3D U-Net input. These steps included reprojection (to ensure a uniform coordinate system), normalization (for consistent numeric ranges), and artifact filtering (e.g., removing clouds that could obscure ground data). Although time-consuming, meticulous preprocessing was essential to prevent biases or missing values from degrading the model's predictive accuracy.

#### 5.2 High-Performance Computing Constraints

Even with the extensive preprocessing, running a 3D U-Net on these large-scale geospatial datasets demanded more computational power than anticipated. While we utilized Rosie, our high-performance computing (HPC) cluster, memory and GPU availability remained bottlenecks. The multi-gigabyte size of individual TIFF stacks often pushed system limits, especially when we attempted to process multiple time snapshots or geographic regions in parallel. Despite parallelization strategies and optimized frameworks, scheduling workloads at scale required careful coordination to prevent gridlock in the HPC queue.

Moreover, adapting our 3D U-Net to handle geospatial data introduced complications like varying spatial resolutions and heterogeneous input structures. These adaptations further taxed system resources by increasing the complexity and size of intermediate tensors. As a result, thorough experimentation and resource management were critical to prevent crashes, optimize training efficiency, and fully leverage Rosie's capabilities.

## **6** Future Work

### 6.1 Physics Informed Neural Networks (PINNs)

Physics-Informed Neural Networks (PINNs) have demonstrated strong potential for improving the accuracy and generalization of predictive models by embedding physical laws directly into the learning process. Studies such as "Physics-informed neural networks for parameter learning of wildfire spreading" [7] and "Physics-Informed Machine Learning Simulator for Wildfire Propagation" [8] highlight the capability of PINNs to integrate governing equations of wildfire behavior, such as reaction-diffusion equations and wind-driven fire spread models. The integration of these equations allows predictive models to be more general than their training data as a portion of the decision is based on the equations.

Integrating a PINN into our UNet framework could improve key metrics, including prediction accuracy and generalizability across diverse terrains and weather conditions. One possible approach is incorporating a physics-informed loss function at the neural net "bridge" of the UNet architecture, ensuring that predictions adhere to known physical constraints. Alternatively, an independent PINN model could replace the UNet for fire spread prediction, leveraging physics-based regularization instead of purely data-driven learning. However, further experimentation is

required to determine the trade-offs between a hybrid approach and a standalone PINN implementation.

### **6.2 Sentiment Analysis**

Sentiment analysis of social media has emerged as a promising tool for disaster response and early warning systems. Research such as "Sentimental Wildfire: A Social-Physics Machine Learning Model for Wildfire Nowcasting" [9] and "Investigating Disaster Response through Social Media Data and the Susceptible-Infected-Recovered (SIR) Model: A Case Study of the 2020 Western U.S. Wildfire Season" [10] suggest that analyzing social media posts can provide critical situational awareness, potentially improving wildfire response times. Using social media posts to understand where people are and where the fire is spreading could help advise disaster response.

While sentiment analysis could enhance our model by incorporating real-time reports from affected individuals, several challenges must be addressed. The primary difficulty lies in obtaining and curating relevant data, as wildfire-related social media posts are often unstructured and sparse. Developing an effective dataset would likely require collecting and labeling data manually. Furthermore, integrating sentiment analysis into our wildfire spread prediction model would require treating it as an additional input parameter, contributing to uncertainty quantification and alerting mechanisms. Previous solutions used a separate AI that had access to APIs to pull, parse, and sort posts for relevance and use that curated list as an input into the fire prediction model. This could be replicated in our work but would be a lot of effort for another input parameter that hinges on live social media coverage of the fire.

### **6.3 Optimizing Firefighter Response**

Beyond predicting wildfire spread, a natural extension of our model is optimizing firefighter response strategies. By leveraging our predicted fire spread as an input to another decision-support model, we could provide actionable insights for resource allocation and evacuation planning. Such an approach could involve reinforcement learning or optimization-based methods tailored to minimize fire damage while ensuring firefighter safety.

However, real-time integration of firefighter location data presents legal and logistical challenges. Regulations on real-time GPS tracking of emergency personnel vary by jurisdiction, and ensuring secure data sharing between agencies remains a key concern. Overcoming these hurdles may require partnerships with firefighting agencies and the development of privacy-preserving data-sharing protocols. Future work should explore the feasibility of integrating geospatial firefighter deployment data with our predictive framework to enhance operational decision-making.

# 7 Conclusion

We presented a multi-branch 3D U-Net for wildfire spread prediction, showing how spatiotemporal stacking, IoU-driven segmentation, and attention-based skip connections can accurately delineate fire fronts. Our experiments on the WildfireSpreadTS dataset demonstrate both the promise and limitations of such deep learning methods, suggesting that integrated data streams and physics-informed constraints are likely to push performance further. Continued refinements in model architecture, data preprocessing, and HPC strategies will aid practical deployment in real-time fire-management scenarios.

## References

[1] H. S. Andrianarivony, M. A. Akhloufi, "Machine Learning and Deep Learning for Wildfire Spread Prediction: A Review", *Fire*, vol. 7. no. 482, Dec 2024, doi: <u>https://doi.org/10.3390/fire7120482</u>

[2] F. Khennou, M. A. Akhloufi, "Improving wildland fire spread prediction using deep U-Nets", *Science of Remote Sensing*, vol. 8, no. 1, Dec 2023, doi: <u>https://doi.org/10.1016/j.srs.2023.100101</u>

[3] S. Gerard, Y. Zhao, J Sullivan, "WildfireSpreadTS: A dataset of multi-modal time series for wildfire spread prediction", presented at 37<sup>th</sup> International Conference on Neural Information Processing Systems, Red Hook, NY, USA, May 30, 2024, pp/ 74515-74529

[4] Hirling, Dominik, et al. "Segmentation Metric Misinterpretations in Bioimage Analysis." *Nature Methods*, U.S. National Library of Medicine, Feb. 2024, pmc.ncbi.nlm.nih.gov/articles/PMC10864175/.

[5] R. T. Bhowmik, Y. S. Jung, J. A. Aguilera, M. Prunicki, K. Nadeau, "A multi-modal wildfire prediction and early-warning system based on a novel machine learning framework", *Journal of Environmental Management*, vol. 341, no. 1, Apr. 2023, doi: //doi.org/10.1016/j.jenvman.2023.117908

[6] NASA, "VIIRS I-Band 375 m Active Fire Data" NASA. <u>https://www.earthdata.nasa.gov/data/instruments/viirs/viirs-i-band-375-m-active-fire-data</u> (accessed March 19, 2025)

[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for parameter learning of wildfire spreading," Comput. Methods Appl. Mech. Eng., vol. 412, 2024.

[8] S. Mo, H. J. Lee, and C. Farhat, "Physics-Informed Machine Learning Simulator for Wildfire Propagation," in CEUR Workshop Proceedings, vol. 2964, 2021.

[9] Y. Lin, W. Wang, and M. Zhang, "Sentimental wildfire: a social-physics machine learning model for wildfire nowcasting," J. Comput. Soc. Sci., vol. 5, pp. 23-41, 2022.

[10] A. J. Stewart, R. K. Smith, and L. T. Nguyen, "Investigating disaster response through social media data and the Susceptible-Infected-Recovered (SIR) model: A case study of the 2020 Western U.S. wildfire season," arXiv preprint arXiv:2308.05281, 2023.

## Enhancing Collaboration in Multi-Agent Environments with Attention-Based Actor-Critic Policies

Hugo Garrido-Lestache Department of Computer Science and Software Engineering Milwaukee School of Engineering Milwaukee, WI USA garrido-lestacheh@msoe.edu

Jeremy Kedziora Department of Computer Science and Software Engineering Milwaukee School of Engineering Milwaukee, WI USA kedziora@msoe.edu

#### Abstract

Collaboration is essential for solving complex real-world problems, as it enables tasks to be completed more efficiently and leverages the diverse perspectives and skillsets of team members. Multi-Agent Reinforcement Learning (MARL) can be utilized to train agents in environments where they must collaborate to achieve a common goal. MARL is specifically challenging due to the non-stationarity introduced by multiple agents learning simultaneously, and its development has primarily focused on managing these challenges without any special focus on collaboration. Recent work on MARL has introduced the use of an attention mechanism in the critic of the Actor-Critic architecture that allows the critic to select relevant information from other agents' observations, promoting better learning of the value function. In this paper, we study the impact of the attention mechanism on the quality of learned collaborative policies. We propose a new actor-critic architecture that incorporates an attention mechanism in the actor to model collaboration directly in the learned policies. We evaluate this architecture on a simulation of a soccer game that features competition between teams of collaborators and compare it to the current state-of-the-art in MARL as well as classical reinforcement learning algorithms.

# **1** Introduction

Collaboration and communication are likely one of the most important aspects that make the human species the most advanced on earth. By endowing AI with strong collaborative skills, we can scale the number of agents addressing a problem, thereby tackling more complex challenges while maintaining team cohesion. However, achieving and maintaining cohesion in multi-agent environments has proven challenging due to the inherent non-stationarity that arises when multiple agents learn and adapt concurrently. In this paper, we introduce a novel multi-agent reinforcement learning algorithm that emphasizes inter-agent collaboration, leveraging shared information to foster coordinated and efficient problem-solving.

Many architectures have been proposed for managing collaborative multi-agent systems. For example, in [1], they used the option-critic architecture first described in [2] and extended to cooperation between agents. In [3], they use a hierarchical approach where a top-level policy assigns roles to lower-level policies in the form of a latent vector. When it comes to using attention for the purpose of collaboration, [4] uses attention to determine whether agents should communicate with each other and then actually communicate through the use of Long-Short-Term-Memory architecture. However, while these approaches represent important advances, they often struggle to scale to large numbers of agents and fall short in actively fostering robust collaboration. Our contribution directly addresses these limitations by introducing an algorithm that not only scales efficiently but also significantly enhances inter-agent collaboration, paving the way for more complex and coordinated multi-agent systems.

In this paper, we build upon [5], which explored the use of attention mechanisms in the critic in actor-critic architectures. They additionally explored methods for extracting a new baseline (named multi-agent baseline) to generate a better estimate of the return. We extend their work by additionally incorporating an attention mechanism in the actor and a novel loss. The attention mechanism in our model aim to allow agents to communicate and share information and thoughts in a selective manner. It functions by allowing an agent to pose an open-ended question to its peers. Other agents then determine whether they are best suited to respond; those that are, provide targeted feedback. For example, in a soccer game scenario, a player with the ball may query, 'Who should I pass to?' Agents which are not nearby opponents may be identified as optimal recipients, and their responses inform the passing decision. We additionally develop a novel loss (named conformity loss) which encourages diverse actions in agents, enhancing collaboration in certain environments. We have named this model Team-Attention-Actor-Critic (TAAC).

It is important to note that our algorithm (TAAC) requires centralized training and execution, meaning that agents must be both trained and deployed in an environment where global state information is readily accessible for effective coordination. Because of this, the algorithm may not be well-suited for environments like robotics, where obtaining global state information in real time can be challenging. However, in digital settings, such as simulated environments or large-scale multi-agent systems in software—where global state data is easily available, our algorithm can leverage this information to achieve robust and efficient collaboration.

To test our algorithm, we train it in a simulation of a soccer environment. In this setting, agents control individual players on a team competing in a virtual soccer match. This simulation accurately models key aspects of the game, including team coordination, strategic passing, and goal scoring. We present our results by comparing the performance of our approach against state-of-the-art algorithms using a combination of traditional performance metrics (such as win rates, goal differentials, and ELO ratings) and collaboration metrics (such as inter-agent connectivity and pairwise distance).

This paper is organized as follows. In Section 2, we introduce the notation and provide the necessary background. Section 3 details our proposed algorithm along with its implementation. In Section 4, we describe the experimental setup and the evaluation metrics employed. Section 5 presents the experimental results, while Section 6 offers an in-depth analysis and discussion of these findings. Finally, Section 7 concludes the paper and outlines directions for future work.

## 2 Background and preliminaries

Our strategy to model collaboration will build upon the actor-critic algorithm. Below, we introduce this algorithm along the principles and notation needed to formalize it. Following this section, we provide a comprehensive description of our contributions.

In reinforcement learning, control problems with multiple agents can be modelled as Markov Decision Games (MDG), which are defined by:

- A set of n agents;
- A set of states S that describe the current common environmental conditions facing the agents;
- A collection of actions  $A(s) = \prod_i A_i(s)$  for  $s \in S$  where  $A_i(s)$  are the actions agent *i* can take in state *s* and:

$$\vec{a} = \langle a_1, a_2, \dots, a_n \rangle \in A(s);$$

- Transition probabilities  $p(s' \mid \vec{a}, s)$  for transitioning from state s to state s'; Where  $\vec{a}$  is the vector of actions for all the agents;
- A function:

$$\vec{r}(s', \vec{a}, s) = \langle r_i(s', \vec{a}, s) \rangle_{i=1,\dots,n}$$

that supplies the immediate rewards for each agent associated with this transition.

Throughout this paper, for a generic vector  $\vec{z}$ , we will write  $\vec{z}_{i}$  to mean z with the *i*th dimension excluded:

$$\vec{z}_{i} = \langle z_1, \dots, z_{i-1}, z_{i+1}, \dots, z_n \rangle$$

and we will write  $\vec{z_i}$  to mean the pair  $(z_i, \vec{z_{i}})$ .

In environments that take place over a finite number of discrete periods T, the sequence of periods in which the agents participate is referred to as an episode. At each time step, each agent makes an observation,  $o_{i,t} = O_i(s_t)$ , of the global state, which encapsulates all available information for agent i; the vector of such observations across agents is  $\vec{o}_t = \langle o_{i,t} \rangle_{i=1,...,n}$  The goal of each agent is to learn a policy which describes the probability that a trained agent should take action  $a_i$  given an observation  $o_{i,t}$ , to maximize its individual sequence of rewards throughout an episode:  $\sum_{t=0}^{T-1} \gamma^t r_i(s_{t+1}, \vec{a}_t, s_t)$ . Here  $\vec{a}_t$  and  $s_t$  are the agents' actions and the common state at time t and  $\gamma \in [0, 1]$  is the discount factor on future rewards. In general, policies could differ across agents; we will focus on the case where the policy is shared. Any differences in decision-making across agents are modelled using their observations of the state. Given this, we write these policies as  $\pi(a_i|o_{i,t})$ .

#### 2.1 Policy Gradients

Policy gradient methods differ from value-based methods (e.g. *Q*-learning, [6]) in that they model the policy directly rather than estimating the state-conditional value of actions and then using those values to build a policy. To learn the policy parameters **u** for a policy  $\pi_{\mathbf{u}}(\cdot)$ , these methods use an estimate of the policy gradient ([7], [8], [9]) to update the parameters in the direction of increase in the expected sum of discounted rewards. In the case of *n* agents with shared policies, taking the Monte Carlo approach of using the full return, the policy gradient may be written as:

$$\nabla_{\mathbf{u}} J(\mathbf{u}) = \mathbb{E}_{\tau \sim \pi_{\mathbf{u}}} \left[ \sum_{i=1}^{n} \sum_{t=0}^{T-1} \nabla_{\mathbf{u}} \ln \pi_{\mathbf{u}} \left( a_{i,t} \mid o_{i,t} \right) G_{i,t} \right].$$
(1)

Here,  $\tau$  is a trajectory of states, actions, and rewards:

 $\tau = s_0, \vec{a}_0, \vec{r}_1, s_1, \vec{a}_1, \vec{r}_2, \dots, s_{T-1}, \vec{a}_{T-1}, \vec{r}_T,$ 

and  $G_{i,t}$  is the sum of discounted rewards (the causal return) from a given timestep t onwards for a given agent i:

$$G_{i,t} = \sum_{k=t}^{T-1} \gamma^{k-t} r_i (s_{k+1}, a_k, s_k).$$
(2)

Given sampled episodes, we use this estimate of the policy gradient to take steps in the direction of  $\nabla_{\mathbf{u}} J(\mathbf{u})$  via gradient ascent to reach an optimal policy which is shared by all agents.

#### 2.2 Actor-Critic Algorithms

To avoid high variance in the estimates of  $\nabla_{\mathbf{u}} J(\mathbf{u})$  across episodes driven by use of the full causal return,  $G_{i,t}$ , a baseline is often subtracted from the return. Actor-Critic algorithms use an estimate of the long-term value as the baseline, for example, the state-value function,  $V(o_{i,t})$ , which predicts the return to an agent given an observation  $o_{i,t}$  (see [10]). If we model the state value function as a differentiable function of parameters w and write  $V_{\mathbf{w}}(\cdot)$  then this yields the following estimate of the policy gradient:

$$\nabla_{\mathbf{u}} J(\mathbf{u}) = \mathbb{E}_{\tau \sim \pi_{\mathbf{u}}} \left[ \sum_{i=1}^{n} \sum_{t=0}^{T-1} \nabla_{\mathbf{u}} \ln \pi_{\mathbf{u}} (a_{i,t} \mid o_{i,t}) \times [G_{i,t} - V_{\mathbf{w}}(o_{i,t})] \right].$$

Note that we may express the state-value function in terms of the state-action value function  $Q(o_{i,t}, a)$ , averaged over actions:

$$V(o_{i,t}) = \sum_{a \in A_i} \pi_{\mathbf{u}}(a|o_{i,t})Q(o_{i,t},a).$$

We will use this below to model multi-agent structure.

In these algorithms the policy is referred to as the actor and the long-term value estimate as the critic. The actor's parameters are learned through the policy gradient estimates and the critic's parameters are learnt through value-based methods (e.g. gradient descent on the difference between the critic's predicted state value and a target value, which can be either the observed  $G_{i,t}$  or a bootstrapped value, see [11]).

## **3** Team-Attention-Actor-Critic (TAAC)

#### 3.1 Attention Mechanism

The authors of [5] introduced the use of an attention mechanism in the critic, the purpose of which is to aggregate information from multiple agents to more accurately evaluate an agent's state-action value. The intuition behind this use of the attention mechanism is to enable an agent to "query" other agents for their internal representations, thereby enhancing the information available to optimize the critic and allowing for more accurate prediction of the return. In our approach we extend their work by introducing a separate attention mechanism in the actor. The intuition behind doing this is different; we seek to allow agents to use the internal representations of other agents directly when making choices, with the idea being that enabling this is analogous to giving the agents insight into what other agents will choose. This attention mechanism slightly differs from the one in [5] and works as follows.

First, an embedding vector  $e_i$  is generated for each agent. (Note that the method for computing  $e_i$  differs between the actor and the critic; these details are provided in the respective sections below). The embedding vectors are then concatenated to form a matrix E which is separately multiplied with learned matrices  $W_Q$ ,  $W_K$ , and  $W_V$  to produce:<sup>1</sup>

$$K = EW_K, Q = EW_Q, \text{ and } V = EW_V.$$
(3)

These are referred to as the key, query, and value matrices, respectively. Attention is then computed as:

Attention
$$(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$
 (4)

where  $d_k$  is the dimensionality of the K matrix ([12]).<sup>2</sup> In our implementation, multiple attention heads are applied in parallel, each with its own set of Query, Key, and Value weight matrices. The outputs of these heads are concatenated and then passed through a linear transformation to yield the final updated embeddings  $\tilde{e}_i$ . Finally, these updated embeddings  $\tilde{e}_i$  are forwarded through an MLP to generate either a distribution over actions (in the actor) or a state-action value estimate (in the critic). Detailed descriptions for each are provided in the subsequent sections.

#### **3.2** Actor Attention Architecture

To incorporate attention into the shared policy, we condition it on the observations of all agents and write it as  $\pi_{\mathbf{u}}(a_i|\vec{o_i})$ ). We generate an embedding vector  $e_i$  for an agent by passing the agent's observation through an MLP<sup>3</sup> so that:

$$e_i = MLP(o_i) \tag{5}$$

These embeddings are calculated for each agent, formed into E as above, and passed to the dedicated actor attention mechanism. The attention mechanism returns an updated embedding  $\tilde{e}_i$  for each agent which is then passed through an MLP, and finally a softmax

<sup>&</sup>lt;sup>1</sup>in [5], their attention mechanism differs only by having the V matrix be fixed for each attention head and not calculated by using E and  $W_V$ .

<sup>&</sup>lt;sup>2</sup>Division by  $d_k$  is done to scale the values appropriately before the softmax function.

<sup>&</sup>lt;sup>3</sup>Instead of an MLP, any other algorithm can be used to generate an embedding.

output layer to generate a distribution over actions:

$$\pi_{\mathbf{u}}(a_i|\vec{o_i})) = \operatorname{Softmax}\left(\operatorname{MLP}(\tilde{e}_i)\right).$$
(6)

In our implementation, we deliberately omitted the original embedding  $e_i$  from the input to the final MLP. This design choice forces the actor to rely solely on the information provided by the attention blocks rather than bypassing them, ensuring collaborative policies. A figure of the actor architecture can be seen in Fig. 1



Figure 1: Architecture of Actor

#### 3.3 Critic Attention Architecture

In the critic, we must also incorporate the observations and actions of other agents. Consequently, the critic's state-action values are formalized as:

$$Q_{\mathbf{w}}(\vec{o}_i, \vec{a}_i). \tag{7}$$

To create the initial embedding  $e_i$  for the critic, we combine the agent's observation and action using an MLP<sup>3</sup> as follows:

$$e_i = \mathsf{MLP}(o_i, a_i) \tag{8}$$
This embedding  $e_i$  is then processed through the dedicated critic attention mechanism. The outputs from these attention heads are concatenated and passed through a linear transformation to yield an updated embedding  $\tilde{e}_i$ . Finally,  $\tilde{e}_i$  is combined with the original embedding  $e_i$  and fed into a final MLP to generate the scalar state-action value estimate:

$$Q_{\mathbf{w}}(\vec{o}_i, \vec{a}_i) = \mathrm{MLP}(\tilde{e}_i, e_i) \tag{9}$$

In the critic architecture, we include the original embedding to provide additional information, as the critic does not require the same enforced reliance on collaborative signals as the actor does. A figure of the critic architecture can be seen in Fig. 2



Figure 2: Architecture of Critic

#### 3.4 Multi-Agent Baseline

The multi-agent baseline computes the expected value of the critic by taking a weighted sum of its outputs over all possible actions for a specific agent, where the weights are given by the agent's policy distribution. This reduces the variance in our baseline as well as generating a more accurate estimate of the true state-value under the current policy due to considering all possible actions and weighting them using the policy's distribution. Furthermore, by holding the other agents' actions fixed, we obtain a more accurate estimate of the agent's true value. Formally, the baseline is defined by:

$$V(o_i | \vec{o}_{\backslash i}, \vec{a}_i) = \sum_{a_i \in A_i} \pi_u \left( a_i \mid \vec{o}_i \right) \, Q_{\mathbf{w}}(\vec{o}_i, \vec{a}_i) \tag{10}$$

Applying these ideas, the policy gradient estimate is:

$$\nabla_{\mathbf{u}} J(\mathbf{u}) = \mathbb{E}_{\tau \sim \pi_{\mathbf{u}}} \left[ \sum_{i=1}^{n} \sum_{t=0}^{T-1} \nabla_{\mathbf{u}} \ln \pi_{\mathbf{u}} \left( a_{i,t} \mid \vec{o}_{i,t} \right) \times \left[ G_{i,t} - V(o_{i,t} \mid \vec{o}_{\setminus i,t}, \vec{a}_{i,t}) \right] \right]$$

#### 3.5 Conformity Loss

We further model and encourage collaboration by adjusting the actor loss function to promote diversity of actions among agents, under the assumption that effective collaboration arises when agents work on separate and diverse tasks.<sup>4</sup> We call this adjustment conformity loss. It is computed by measuring the cosine similarity between each unique pair of updated embeddings  $\tilde{e}_i$ :

$$\mathcal{L}_{\text{conf}} = \theta_{\text{conf}} \max\left\{\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \frac{\tilde{e}_i^\top \tilde{e}_j}{\|\tilde{e}_i\| \|\tilde{e}_j\|}, \theta_{\text{conf}}\right\}.$$
(11)

Where  $\theta_{conf}$  and  $\theta_{conf}$  are hyperparameters used to scale and cap the loss This loss attains high values when agents' embeddings are very similar (indicating low diversity) and low values when they are not. In our implementation, we cap this loss at a minimum of  $\theta_{conf}$ to ensure that agents are rewarded for diverse actions but not rewarded for completely opposing actions.

# 4 Experimentation

In this paper, we compare multiple algorithms against our novel architecture (TAAC) in a simulated soccer environment. In this environment, teams of 3 agents work together to score goals against an opposing team. A goal is scored when the ball is kicked into the

<sup>&</sup>lt;sup>4</sup>The value of diversity may not hold in all environments, but we found it to be a useful addition for our experiments.

opponent's goal. Each team is composed entirely of agents running the same algorithm to ensure a fair comparison.

#### 4.1 Environment Structure

The simulation represents players as solid circles on a 2D plane capable of moving toward the ball and colliding with it. When a player collides with the ball while in mode "kicking", the ball gains additional velocity. Additionally, the ball collides nearly elastically with the walls, preventing it from leaving the play area. Figure 3 illustrates the simulation environment.



Figure 3: Simulation Environment for Soccer

At each time step, every agent observes the global state, from which it derives an individual observation with no hidden information. In our environment, the state consists of:

- The relative position vector to each teammate.
- The relative position vector to each agent on the opposing team.
- The relative position to the ball.
- The velocity vector of the ball.
- The relative position to the opponent's goal.
- The relative position to the agent's own goal and opponents goal.

• Raycast measurements in the directions of North, East, West, and South indicating the distance to the nearest boundaries of the play area.

Based on its observation, an agent selects an action. In our environment, an action is defined by five Boolean inputs: move forward, move right, move down, move left, and kick. An agent may choose any combination of these inputs, provided they are not contradictory (e.g., moving left and right simultaneously). Overall, the total number of possible actions is 18.

After an action is chosen and the simulation advances by one time step, rewards are assigned to each agent. The reward structure is designed as follows:

• Exploration Reward: Each agent receives a small reward for moving toward the ball. This reward is computed as the dot product of the agent's chosen direction  $\vec{d}$  and the vector from the agent to the ball  $\vec{v}_a$ , scaled by a factor  $\theta_{exp}$ :

$$r_{\text{explore}} = \theta_{\text{exp}} \left( \vec{d} \cdot \vec{v}_a \right) \tag{12}$$

• Team Reward: All agents on the same team receive a larger reward for moving the ball towards the opponent's goal. This reward is calculated as the dot product of the ball's velocity vector  $\vec{v}_b$  and the vector from the ball to the opponent's goal  $\vec{g}$  scaled by a factor  $\theta_{\text{ball}}$ :

$$r_{\text{ball}} = \theta_{\text{ball}} \left( \vec{v}_b \cdot \vec{g} \right) \tag{13}$$

- Scoring Reward: The biggest reward  $r_{\text{goal}}$  is granted when a goal is scored, defined as the ball being fully inside the goal box.
- Distance Reward: This reward is allocated based on the average distance to all teammates, to prevent players from clustering together and playing a . First, the average distance is computed, then it is capped at a maximum threshold, and finally scaled by a constant factor to prevent excessive rewards for large spacing. Specifically, if  $d_i$  represents the distance to teammate i and there are N teammates, the average distance is calculated as  $\bar{d} = \frac{1}{N} \sum_{i=1}^{N} d_i$  and the reward function is defined as:

$$f(\bar{d}) = \theta_{\text{dist}} \cdot \min\left(\bar{d}, \ \theta_{\text{max}}\right). \tag{14}$$

Here,  $\theta_{\text{dist}}$  is the scaling factor and  $\theta_{\text{max}}$  is the cap applied to the average distance.

It is important to note that all rewards (but the distance reward) can be negative, acting as penalties for behaviours such as moving away from the ball, moving the ball away from the opponent's goal, or being scored on.

Games are configured to last T time steps, and an episode is defined as a sequence of time steps that terminates either when a goal is scored or when the game concludes. As a result, a single game may consist of multiple episodes.

# 4.2 Training Procedure

We employ curriculum learning [13] to progressively build up skills in this complex environment. The training is divided into the following stages:

- 1. **Stage 1:** Score goals against an inactive opposing team, with random ball and player spawns.
- 2. **Stage 2:** Introduce an opposing team with randomly selected actions, encouraging players to navigate obstacles and opponents. Continuing with random ball and player spawns.
- 3. **Stage 3:** Engage in league play, where a team competes against a randomly selected past version of itself while maintaining random spawn positions.
- 4. Stage 4: Continued league play with fixed starting positions for each team.

This staged learning approach allows agents to first master movement and scoring before tackling competitive play.

#### 4.3 Evaluation Metrics

Performance is evaluated using win rates, goal differentials, and ELO ratings [14]. Each team is assigned an Elo rating, which gets updated based on victory, lost or tie and the Elo rating of the opponent. When interpreting Elo ratings, it is important to note that the absolute values are less meaningful than the relative differences between them. Additionally Goal differential are measured as the difference between the teams' scores. In addition to performance, we assess collaboration using several key indicators:

- Average Pairwise Distance: A higher average distance suggests that agents are not clustering excessively, which may indicate diverse task allocation.
- **Possession Frequency:** This measures the number of times a ball swaps possession within the team. A high number may indicate that players are passing and, therefore, collaborating.
- **Connectivity:** This metric measures the number of graph-like connections between players, calculated as the ratio of observed connections to the maximum possible

connections  $\frac{N(N-1)}{2}$ . To ensure only meaningful connections are counted, connections only within a range of distance are included. An optimal connectivity level indicates balanced spacing and cooperative positioning.

Using these metrics, we can comprehensively assess both the performance and collaborative behaviour of the teams.

# 5 Results

Our evaluation compares the performance of our TAAC algorithm against two benchmark algorithms: PPO [15] and MAAC [5]. PPO was chosen as a baseline because of its widespread use and proven robustness across diverse environments, while MAAC was chosen for being state-of-the-art in multi-agent reinforcement learning and closely aligning with our implementation. Additionally, we included a random model to provide a grounded point of reference.

We trained each algorithm for 7 days across over 100,000 games. Note that, due to its low performance and slower learning, MAAC was not advanced to stages 3 or 4 and thus did not gain experience playing against an opponent. We then assembled a league of 20 teams (5 copies of each algorithm) and randomly selected teams to compete in 5,000 games, during which we collected the following data:

Figures 4 to 6 display our performance results, illustrating the ELO ratings, goal differential, and win rates for each algorithm. The goal differential is calculated as  $Score_{team_1} - Score_{team_2}$ ; Where a positive value indicates that team 1 is outscoring team 2 by that magnitude, while a negative value indicates the opposite.



Figure 4: Elo rating progression for each algorithm over the evaluation period.



Figure 5: Win-rate comparison across the algorithms over the evaluation period.



Figure 6: Goal differential Matrix across the algorithms over the evaluation period.

Additionally, we present our findings on the collaborative data. Figure 7 illustrates the connectivity score for each model, Figure 8 shows the frequency of possession swaps within a team, and Figure 9 highlights the average pairwise distance among agents on the same team.



Figure 7: Connectivity comparison across the algorithms over the evaluation period.



Figure 8: Possession swaps comparison across the algorithms over the evaluation period.



Figure 9: Average Pairwise distance comparison across the algorithms over the evaluation period.

# 6 Analysis

Following extensive training of all algorithms, our results indicate that TAAC is the top performer, as evidenced by its superior ELO rating and win rate (see Figures 4 and 5), with PPO closely trailing. In contrast, MAAC significantly underperforms compared to expectations for a state-of-the-art MARL method, possibly due to insufficient training time or suboptimal hyperparameter settings. Additionally, Figure 6 illustrates head-to-head comparisons, highlighting that TAAC outperformed all other algorithms.

When examining collaboration-oriented metrics (Figures 7–9), the data suggest that TAAC exhibits more coordinated teamwork. For instance, while Figure 9 indicates that PPO agents are more spread out, Figure 7 shows that TAAC's agents are more closely connected, which may be linked to more advantageous positioning. In addition, the possession swap metrics in Figure 8 imply that TAAC and PPO frequently swapped possession of the ball, potential indicating tactical passing. MAAC demonstrates a incredibly high possessions frequency potentially indicating frequent passing.

Video recordings further substantiate these findings. The footage confirms that PPO frequently employs a dedicated goalie while the remaining players focus on attacking, explaining their pronounced spatial separation. Additionally, PPO exhibits clustering among its attackers and pursues the ball aggressively, which also explains its lower overall connectivity and relatively high possession frequency In contrast, TAAC consistently maintains a stable offensive formation that only transitions to a defensive position when necessary, which is evident through its connectivity score. TAAC's dynamic strategy, characterized by frequent ball possession exchanges and timely passes, demonstrates superior tactical coordination. Video recording of MAAC indicate a similar approach to PPO but instead having the entire team pursue the ball aggressively, this explain their low pairwise distance and connectivity. Furthermore, MAAC's high possession frequency is also explained by this behaviour, as possession likely rapidly changes between players when the players are grouped together. Additionally despite managing to considerably score against random opponents, MAAC was consistently outperformed by both PPO and TAAC, as seen in the performance results.

Overall, these results suggest that our novel algorithm demonstrates enhanced collaborative behaviour and improved performance in environments where teamwork is critical. However, due to MAAC's suboptimal performance and limited training, we plan to investigate its shortcomings further to enable a fairer comparison with state-of-the-art methods. Looking ahead, a logical next step is to evaluate TAAC in alternative environments and scenarios to better understand its limitations. Moreover, selecting an environment where collaboration is more directly tied to performance will allow us to more rigorously test our contribution.

# 7 Conclusion

In this work, we introduced TAAC, a multi-agent reinforcement learning algorithm that leverages multi-headed attention in both the actor and critic networks to foster robust collaboration. Our experiments demonstrate that TAAC outperforms competing methods in win rates, goal differentials, and Elo ratings while exhibiting superior coordinated behaviour, as confirmed by game statistics and game recording analysis. Future work will involve a deeper investigation into MAAC to ensure it performs to its full potential and also to explore this new algorithm in different environments to fully test its capabilities.

# References

- J. Chakravorty, N. Ward, J. Roy, M. Chevalier-Boisvert, S. Basu, A. Lupu, and D. Precup, "Option-critic in cooperative multi-agent systems," 2020. [Online]. Available: https://arxiv.org/abs/1911.12825
- [2] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," 2016. [Online]. Available: https://arxiv.org/abs/1609.05140
- [3] K. Matsuyama, K. Su, J. Wang, D. Ye, and Z. Lu, "Cord: Generalizable cooperation via role diversity," 2025. [Online]. Available: https://arxiv.org/abs/2501.02221
- [4] J. Jiang and Z. Lu, "Learning attentional communication for multi-agent cooperation," 2018. [Online]. Available: https://arxiv.org/abs/1805.07733

- [5] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," 2019. [Online]. Available: https://arxiv.org/abs/1810.02912
- [6] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992. [Online]. Available: https://doi.org/10.1007/BF00992698
- [7] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 1999.
- [8] R. J. Williams, "Toward a theory of reinforcement learning connectionist systems," Northeastern University, College of Computer Science, Technical Report NU-CCS-88-3, 1988.
- [9] —, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992.
- [10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.
- [11] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602
- [12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762
- [13] P. Soviany, R. T. Ionescu, P. Rota, and N. Sebe, "Curriculum learning: A survey," 2022. [Online]. Available: https://arxiv.org/abs/2101.10382
- [14] T. Minka, R. Cleven, and Y. Zaykov, "Trueskill 2: An improved bayesian skill rating system," *Technical Report*, 2018.
- [15] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

# Semi-supervised Weights for Action Generalization (SWAG)

#### **Dylan Norquist**

norquistdylan@gmail.com

#### Abstract

Pretraining has emerged as a necessary step when working in complex sequential decision domains. I recreated the VideoPreTraining (VPT) and Inverse Dynamics Model (IDM) within the domain of Super Mario Bros. Preliminary results show that it is possible to create other foundational models in other domains with limited resources. The IDM/VPT share the same architecture and have a parameter count of 9 million which is substantially lower than the original VPT/IDM. They are trained on 6 hours of ground truth data to find out if they can be applied to this domain. My results show that the IDM has 82.87% accuracy, 0.5757 precision, and 0.4564 recall while the VPT has 85.75% accuracy, 0.5338 precision, and 0.9127 recall. This shows that more agents can be created in other complex environments with minimal ground truth data as a solo undergraduate research project.

#### **1** Introduction

My goal was to reproduce the VideoPreTraining (VPT) model and Inverse Dynamics Model (IDM) introduced in Baker et al. [1] within a different context of Super Mario Bros (SMB), specifically the 1984 release on the Nintendo Entertainment System (NES). I wanted to work in the domain of SMB because it's the right level of complexity for a solo undergraduate research project. In comparison to Minecraft, the original domain, SMB is a good environment for testing this approach because of its history as an environment for reinforcement learning (RL). This space is a common introduction into reinforcement learning; however the models are generally very limited in their observation and action spaces. Traditionally actions are removed or configured such that one output converts in a sequence of actions and observations are preprocessed or using internal gamestate as part of the observation space, these steps are essentially a middle-man between the model and the environment. This does typically increase performance of the model but at the cost of generalizability. But the removal of impossible actions or downscaling the image can increase performance without reducing generalizability. The VPT/IDM is able to interact directly with the inputs and outputs of the environment to make decisions. Not all environments have the tools and software to interact with traditional RL approaches and the ones that do are not usable outside of the domain. That is not to say it's an

impossible task; tools originally introduced by Brockman et al. [2] as OpenAI Gym, now known as Gymnasium and maintained by the Farama Foundation, can be used to make this process easier.

The IDM is trained on a ground truth dataset to predict the labels of a sequence of frames; those labels are called pseudo labels. The VPT is then trained on the pseudo labels generated by labeling web-scraped videos with the IDM. This is the pretraining part of VPT, it creates a model that is able to be trained purely on behavior cloning on internet scale datasets. This approach is most valuable when dealing with a hard search problem, RL traditionally can't achieve these tasks within the environment without a middle-man or exponentially larger training times. The VPT parallels others efforts at deep understanding, the scale of the data is where the model's strength shines. When the pretraining is done it can serve a valuable role through fine tuning or placement within a larger architecture. For example, STEVE-1 from Lifshitz et al. [3] is a text-to-behavior model that leverages the pretrained models: VPT from Baker et al. [1] and MineClip from Fan et al. [4]. These two pretrained models were able to be combined to create a new model with finetuning for \$60 of compute. This new model produced a novel approach for agentic behavior prompting within the context of Minecraft.

My contributions to this space are the preliminary results of the VPT and IDM within the domain of SMB for the NES. The VPT and IDM is only trained on 6 hours of ground truth data. The results of the VPT and IDM are summarized in Table 1.

Model	IDM	VPT
Top-1 Accuracy	82.87%	85.75%
Top-3 Accuracy	96.29%	98.49%
Precision	0.5757	0.5338
Recall	0.4564	0.9127
F1-Score	0.4863	0.6169

Table 1: Summary of the results of the VPT and IDM trained on ground truth data. Precision, Recall, and F1-Score are macro-averages.

# 2 Methodology

Following Baker et al. [1], I used the same architecture for both the VPT and the IDM. The only difference between the VPT and IDM is the VPT is casual while the IDM is non-casual. This difference is made by adding a causal mask to the VPT when attending. The IDM has no causal mask allowing it to look at past and future frames when predicting labels.



Figure 1: Screenshots of the Super Mario Bros showing the effect of downscaling the images. This was done by using "area" as the interpolation mode.

The IDM takes in 64 consecutive frames then labels the middle 32 frames. This means the first 16 and last 16 frames are a buffer that will not be labeled since they lack relevant context. The VPT takes in 64 consecutive frames then predicts the next action for each frame. Each frame is a screenshot of SMB, for computational efficiency the image is scaled down to 64 by 60, as seen in Figure 1. The image pixels are initially in the format RGB in the range [0, 255], this is divided by 255 to reduce the range down to [0, 1] to reduce exploding gradients.

#### 2.1 Architecture

The IDM and VPT architecture is nearly identical, therefore I will describe them together. The first layer is a 3D convolution with 64 learnable filters with a temporal kernel depth of 5, height of 1, and width of 1. I dubbed this the *Temporal Feature Extractor*; it's as a computationally efficient way to reduce convoluting over time throughout the entire model while still getting the benefits of that context. This is not a strictly necessary step because the *Normalized Transformer Blocks* performs temporal contextualization later, but that added temporal context earlier in the model increases performance according to Figure 14 from Baker et al. [1].

The temporally contextualized frame is then followed by the *Feature Extractor* which is a ResNet feature extractor. It consists of three subsequent stacks with widths  $W = \{32, 64, 64\}$ . Each stack is made up of, in order, an initial 3 x 3 convolutional layer with 1-pixel zero padding. This makes it such that the incoming and outcoming image has the same dimension. Then followed by 3 x 3 max pooling with a stride of 2 and padding of 1. This makes it such that the image dimensions height and width are halved each pass through. Then two ResNet blocks as defined in He et al. [5] with each layer having W output channels.

The output of the *Feature Extractor* is followed by the *Embedder* which is used to reduce the dimensionality of the flattened features. The features at this point have a dimensionality 3,584 then are reduced to a dimensionality of 256 then upscaled to a dimensionality of 512. The reason for reducing the dimensionality is for separation and consistency during the upscaling

process. If the *Feature Extractor* were fed directly into the *Normalized Transformer Blocks* then the increase in size would then increase compute of the *Normalized Transformer Blocks* by a large polynomial and exponential factor. This was added to allow the feature extractor and normalized transformer compute to scale separately and keep those components of the model separate, compute wise, from each other.

The *Normalized Transformer Blocks* follow after the *Embedder*. It is based on Table 1 from Loshchilov et al. [6]; It describes the normalized transformer architecture. The *Normalized Transformer Blocks* will be performing action predicting (either causal or non-causal). It is made up of 2 normalized transformer blocks. Each block has an attention layer with 4 heads and a residual connection. That output and residual are then combined using an eigen learning rate described in Loshchilov et al. [6]. The output is then fed into the feed forward which has a similar eigen learning rate and residual connection. This process is repeated then fed through a linear layer which encodes 36 possible actions which are put through softmax.



Figure 2: Overview of the IDM and VPT architecture. The left side contains the full architecture, it starts at the bottom left then ends at the top. The right side contains all of the modules, e.g. the implementation of a ResNet block. All modules have input from the bottom and output at the top unless shown otherwise by an arrow indicating a residual connection like shown in the normalized multi-head attention.

#### 2.2 Rationale

The action space is determined by removing all mutually exclusive actions. An example of this is movement on the x-axis can be described as moving left, right, or no operation (noop). Therefore only 3 actions exist for the x-axis, the y-axis is similar with only up, down, or noop. This means there are 9 movement actions. The A and B button on the NES controller have 4 combinations of on and off which means there are 9 \* 4 actions or 36 actions.

The original architecture proposed for the VPT/IDM was modified to solve problems not originally faced by Baker et al. [1]. I lacked ground truth data and manually curating a dataset would have diminishing returns quickly. To try and mitigate this problem I went with normalized transformer blocks over traditional transformer blocks because of Figure 2 in Loshchilov et al. [6] which showed a 4 to 20 times speed up in training (in terms of tokens). I believed this allowed me to train a capable model with limited training data.

### 3 Methodology

By following a modified approach proposed by Tan et al. [7] which is used as a technique to efficiently upscale a model and find optimal hyperparameters. I found optimal hyperparameters that were within the original compute budget. This was done through an exhaustive grid search by estimating the effects of each hyperparameter on the compute of the model. The original compute budget was 3 sets of 4 Tesla V100s per a training session and training up to 15 models in parallel. I reduced the model parameter size from 0.5 billion to 2.8 million; I found reducing the model size allowed for quicker experiments that otherwise might have taken weeks instead of hours. This allowed me to use the upscaling technique to reduce the time doing a hyperparameter search. Scaling for minimal parameter count is ideal, but barely taken into account when performing a grid search. The parameter count should be below 0.5 billion, no larger than the original. This estimate is mainly because the relationship between compute and parameter count differs between the *Feature Extractor* (~O(n)) and *Normalized Transformer Blocks* (~ $O(n^2)$ ).

The grid search and model upscaling is performed by first entering values into a spreadsheet to find if it's still within the next compute budget. Then performing a grid search on the possible combinations of those hyperparameters. There are 5 parts of the spreadsheet: *Input*, *Temporal Feature Extractor, Feature Extractor, Embedder*, and *Normalized Transformer Block*. The Input is only full of independent hyper parameters. The relationship of the parameters only depends on the Input not vice versa. The *Temporal Feature Extractor* is calculated by the formula below. *k* is the kernel which is 3 dimensional for this component. *c* is the channel either going in or out. *t* is the number of frames being processed. *I* is the image dimensions.

$$p_{TemporalFeatureExtractor} = (k_h^* k_w^* k_d^* c_{in})^* c_{out} + c_{out}$$
$$T_{TemporalFeatureExtractor} = I_w^* I_h^* k_h^* k_w^* k_d^* c_{in}^* c_{out}^* t$$

The *Feature Extractor* is calculated by the formula below. This formula describes the number of convolutions per stack summed by the number of stacks. The stack has 1 convolution and 2 ResNet blocks (a ResNet block contains 2 convolutions) which is a total of 5 convolutions per stack.

$$p_{FeatureExtractor} = 5\sum_{i=0}^{3} (k_h * k_w * c_{i-1}) * c_i + c_i$$
$$T_{FeatureExtractor} = 5t\sum_{i=0}^{3} (\frac{l_h}{2^i} * \frac{l_w}{2^i} * k_h * k_w * c_{i-1}) * c_i + c_i$$

The *Embedder* is calculated by the formula below. L is the dimension of that layer.  $L_0$  is the flattened output of the *Feature Extractor*.  $L_2$  is the size of the embedding dimension.

$$p_{Embedder} = \sum_{i=0}^{2} L_{i-1} * L_{i}$$
$$T_{Embedder} = t \sum_{i=0}^{2} L_{i-1} * L_{i}$$

The Normalized Transformer Block is calculated by the formula below. N is the number of Normalized Transformer Blocks. H is the number of heads for multi-head attention. d is the dimension of the embedding.  $d_k$  is the dimension of the key, query, and value vector.  $d_f$  is the dimension for feed forward.

$$p_{NormalizedTransformerBlock} = N * (H * (4 * d * d_k) + (2 * d * d_f))$$
$$T_{NormalizedTransformerBlock} = N * ((3 * d^2 * d_k + d_k^2 * t^2) + (2 * d_f^2 * t))$$



Figure 3: A screenshot of the spreadsheet used to calculate the compute. The true parameter count for the IDM/VPT was 9,355,876 while the spreadsheet estimated it was 9,471,584 which is 1.22% off of the true value. This was a valuable tool when upscaling the model.

I found that the spreadsheet was not perfectly accurate, but it followed the general trends well enough that I was able to use it as a guide when increasing compute. Overshooting and being off by a small margin was not a concern. I found that runtime was not estimatable with any precision, but TeraFLOPS and FLOPs were accurate to the same degree.

The current hardware I have available would not be able to efficiently train a model any larger than 275 million parameters. In Baker et al. [1] it took them ~9 days to train a 0.5 billion parameter model for 30 epochs on 70k hours of data using 720 Tesla V100s. Using that as a rough estimate, a 275 million parameter model would take ~36 days to train on 12 Tesla V100s, but I only have extended access to 4 Tesla V100s which means it'd take ~108 days to train a foundational model of that size.

Based on the preliminary results, upscaling the model up to 80.8 million parameters will produce the most optimal results for the compute. In Baker et al. [1] it was shown that larger models performed at better complex downstream tasks, especially when fine tuned, even if smaller were initially more performant. With a 80.8 million parameter model I believe this is the sweet spot for maximum generalizability while staying within the maximum compute budget.

# 4 **Experiments**

#### 4.1 Data Collection

I started by collecting ground truth data by scraping the internet for Tool Assisted Speedruns (TAS) of SMB. TAS are playthroughs of video games generated via specialized software that provides precise, frame-by-frame control over the game's inputs. By allowing the user to slow down or pause the game, as well as automate complex or time-sensitive actions, TAS enables the production of highly optimized gameplay sequences that would be exceedingly difficult—or impossible—for a human player to replicate in real time. This methodology is widely employed to explore theoretical skill ceilings. TAS videos are a unique datasource because it's a labeled form of video that's reproducible, but the drawback is it is used to replicate near impossible inputs for human players. I started with 4 hours of TAS inputs and I played SMB for 2 hours. I recorded the inputs then played them back to record the video to avoid any desync issues. These inputs were stored in a PostgreSQL database for easy access and analytics.

For the unlabeled data it was collected through web scraping on Youtube. I web scraped based on a list of queries (as shown in Table 2) and grabbed the first 500 videos for each query. I used each video's metadata to reduce the compute required for rejecting and accepting videos. For each video I collect the title, duration (in seconds), weblink, channel id, and description then it is inserted into PostgreSQL database. The database stored 27,990 unique videos metadata which was 12,779 hours of videos collected over the span of 70 days.

NES Super Mario Bros No Warp Super Mario Bros NES world 8-4 walkthrough Super Mario Bros NES world 3-1 walkthrough Super Mario Bros NES expert playthrough with no warp zones Super Mario Bros NES world 1-1 Super Mario Bros NES world 4-2 warp zone trick Super Mario Bros NES how to beat level 8-3 Super Mario Bros NES best way to complete world 6-2 Super Mario Bros NES world 5-1 walkthrough Super Mario Bros NES world 7-4 maze solution NES Super Mario Bros Playthrough Super Mario Bros 1 full game playthrough NES Super Mario Bros NES perfect run no power-ups NES Super Mario Bros Warpless%

Table 2: This contains a subset of the queries used when collecting the initial dataset. I found that Super Mario Bros was a broad enough topic that even with NES it'd return different versions of the game consistently.

## 4.2 Data Cleaning

I used a whitelist and blacklist to filter out unwanted videos which reduced it to 577 unique videos and 314 hours. These videos have a wide variety of resolutions, frames rates, and artifacts. To reduce the noise of this data and normalize the input I finetuned Yolo11n for classifying and cropping. Yolo11n was trained by a fifth of the remaining videos, which were labeled based on if they were an unmodified SMB video and where the gameplay was within the screen. I took 1 frame per a second and used that as the fine tuning data. This approach worked well for removing webcams, stream chats, and other artifacts that'd reduce VPT's ability. I found that the fine tuned Yolo11n model's ability for classification was not sufficient enough for classifying videos. Instead I did manual classification for the quality of the video, this reduced the final dataset to 165 videos and 62 hours. I believe the big jump from 27,999 videos to only 165 is because of the diverse range of the SMB genre which is harder to quantify through keywords. I found that different versions of SMB are commonly within the top 5 results of a query. Through more advanced data collection processes this could be mitigated but it's most likely a fundamental issue with this type of approach.

The preprocessing step for faster training times and standardization of a noisy dataset was converting every video to be at 30 frames per a second (fps). I found this worked for every video in my dataset as that frame rate didn't slow down or speed up the gameplay. Once the video was converted to 30 fps I then converted them into numpy chunks of 1024 frames. This made the training process easier and reduced the preprocessing that would have been incurred by loading each video clip during training. It reduced the IO overhead of videos, especially longer ones.

#### 4.3 IDM

The IDM was trained based on the described in Section 2.1 and Figure 2. This experiment was to figure out if it could achieve any accuracy when labeling on a substantially smaller ground truth dataset. The training process was set up such that the loss was calculated by seeing if the middle 32 logits were correct according to CrossEntropyLoss. Looking at Figure 4, it shows the heavily imbalanced class size; to mitigate this I computed class weights using the sklearn python library. This was found to be not enough to learn all classes since the model's local minimum was worse, in terms of macro-average accuracy, than without it. Those classes still needed to be labeled, I found that setting the minimum size for a class to be set at 1000 helped mitigate this issue but it was not enough to solve it. I did find that this did have a trend of getting closer to the correct action. This looked like labeling a frame as action from "B" to "Left + B" when the frame was labeled "Left". It increased the model's ability to grasp what was going on but not to the extent of differentiating similar classes. By increasing the ground truth data by a factor of 5 and upscaling up to 80.8 million parameters it could achieve similarly to Baker et al. [1] in the SMB domain. When looking at the confusion matrix in Figure 4, I found that the mislabeled classes were nearly identical in quantity across the diagonal. This is not seen in the VPT which leads me to believe it has to do with the non-causal nature of the IDM.



Figure 4: The training results of the IDM. Top left is the Confusion Matrix of the Top 10 largest classes; I went with this approach because it was hard to distinguish between 36 classes especially on a log scale. Top right is the loss of over time. Bottom left is the class distribution on a log scale. Bottom right is the accuracy per class, a trend does exist around class size and class accuracy but it's not a fully predictive measure. (Section A.1 for higher resolution)

I found that the IDM was able to accurately label on a significantly smaller dataset. The results are summarized in Table 1. I found that the biggest issue was around the heavily imbalanced class sizes making it difficult to define an ideal loss function. In this domain that is a non-mitigatable issue through traditional means like undersampling because some classes are very rare and have no distinguishing traits from other classes. For example pressing Up + B as a unique action only has one instance where it is used, when effective in game; other instances would look the same as a Noop. This is when Mario is climbing a vine and already holding B but–from a visual perspective–this doesn't have any distinguishing features in an image nor over time.

#### 4.4 VPT

The VPT was trained based on the described in Section 2.1 and Figure 2. This experiment was to figure out if I could achieve any accuracy when labeling on a substantially smaller ground truth dataset. The training process was set up such that the loss was calculated by seeing if the last 32 logits were correct according to CrossEntropyLoss. I used the same process for calculating the class weights as described in Section 4.3.

I found that the VPT was able to correctly predict the next action with higher accuracy as shown in Table 1. The VPT did better than the IDM, although they are technically performing different tasks in theory the causal tasks should be the hardest. One reason this might have been more effective is that the masked training approach allowed for the training parallelization benefits of a transformer to take full effect while the unmasked approach trained similarly to a CNN. Building off of Section 4.3, The rationale behind this theory is the way an unmasked transformer learns would be reliant on all of its inputs and learning in one area would affect different predictions as more context could be misunderstood. Essentially boils down to the fact that the IDM wasn't able to use its non-causal abilities as effectively as originally proposed. When looking at the confusion matrix in Figure 5, the incorrect predictions is when the true label is "B + Right" while the predicted label is "B". 4.58% of that class was incorrectly labeled. But when looking at gameplay there are many scenarios where it'd be indistinguishable between the two. When looking at the accuracy per class plot in Figure 5, it's significantly more effective than the IDM across all classes.

I tested the VPT's ability to play SMB which was predicted to be quite effective. This was found to be incorrect, with RL finetuning it could become quite effective in a short timeframe. What was found is the VPT is attempting to replicate TAS inputs which typically means its trying to reproduce frame perfect inputs consistently. For example, in a SMB TAS the player will start off levels by pressing "Left + B" to look left then holds "A + B + Right" to gain speed then at the apex of the jump will hold "B + Right". This is a specialized technique to accelerate faster, this happens over the course of 31 frames or 0.52 seconds and the model is

unable to recreate these inputs. I did see a general trend towards behaviors that would beat a level (e.g. jumping over enemies) but the model wasn't able to produce those actions effectively. I tested it in Super Mario Bros: The Lost Levels (SMB2), the second SMB NES game; it had never seen this environment in its training data but it performed to the same extent as SMB. I did find two issues that could be fixed with finetuning. When the model is not moving it does not want to start moving again. When thinking about it from the model's perspective it makes sense, if for the last 64 frames it didn't move then it will probably not move on the 65th frame. The other issue is when the model wants to move right it wants to start by recreating the frame perfect sequence talked about earlier. It typically fails this and then holds left in an attempt to start the process of moving right. Sampling techniques were unable to mitigate this issue as the model needed full control on its actions to reproduce difficult sequences.



Figure 5: The training results of the VPT trained on ground truth data. Top left is the Confusion Matrix of the Top 10 largest classes on a log scale. Top right is the loss of over time. Bottom left is the class distribution on a log scale. Bottom right is the accuracy per class. (Section A.1 for higher resolution)

#### 4.4 IDM & VPT

I used the IDM to generate pseudo-labels for the 62 hours of data I collected. Then, I trained the VPT on 10% of the pseudo-labeled data, equivalent to 6 hours of footage. The labeling process involved using a sliding window approach mentioned in Section 2. The main goal with this

experiment was to see how noisy the online data collected is. The setup and hyperparameters were the same as Section 4.3. The evaluation of this model was on ground truth data.

The VPT trained on IDM pseudo labels is not sufficient enough. This is reflected in Figure 6 with every chart (besides the class distribution) showing an inability to correctly label frames. With a 2.86% accuracy it's nearly no different than random chance when sampling of one of the 36 classes. The confusion matrix and accuracy per class plots from Figure 6 do show a different reason, the conclusion is the same, it seems that the model didn't even attempt to learn other classes. Compared to previous runs of the VPT, this is consistent where the longer it trains the more classes it understands but its substantially lower for 40 epochs. This indicates that the issue is low data quantity, the pseudo labels are inaccurate, or the noise of online videos as a factor in its ability to learn. Increasing the data quantity has its drawback on compute as mentioned in Section 3. Increasing quality requires training a new model.



Figure 6: The training results of the VPT trained on 10% of the pseudo-labels generated from the IDM. Top left is the Confusion Matrix of the Top 10 largest classes on a log scale. Top right is the loss of over time. Bottom left is the class distribution on a log scale. Bottom right is the accuracy per class.

# 5 Future Works

The next steps of this paper is to try and replicate the VPT trained on pseudo-labels produced by the IDM. If I am able to replicate the VPT from Baker et al. [1] this would show the

reproducibility of this project in a solo undergraduate project. Creating a foundational model in a new domain could open the gates to others to finetune that model into something new that previously would have been infeasible. Using this new foundational model I would like to recreate a similar model to STEVE-1 from Lifshitz et al. [3] which combines MineClip from Fan et al. [4] and the VPT from Baker et al. [1] to create a text-to-behavior model. I believe if a foundational model is achieved I could recreate STEVE-1 in a new genre, specifically not Minecraft nor Super Mario Bros. If this is something that can be recreated I would like to see how generalized this process can become. I would like it to be possible for someone to gather and train their own foundational model using my approach with minimal effort.

# 6 Conclusion

I explored the feasibility of creating foundational models for sequential decision-making tasks within the domain of Super Mario Bros (SMB) as a solo undergraduate research project. By recreating the VideoPreTraining (VPT) and Inverse Dynamics Model (IDM) architectures introduced in Baker et al. [1]. The IDM/VPT I created have a parameter count of 9 million which is substantially lower than the original IDM/VPT, 0.5 billion. With only 6 hours of ground truth data; the IDM has 82.87% accuracy, 0.5757 precision, and 0.4564 recall while the VPT has 85.75% accuracy, 0.5338 precision, and 0.9127 recall. This shows that more agents can be created in other complex environments with minimal ground truth data.

The experiments revealed that pseudo-labeling internet size datasets with the IDM did not boost the VPT's perform at my scale. Likely due to noise of the online videos and lack of ground truth data. Nonetheless, this is promising results for the IDM/VPT demonstrating the viability of extending pretraining methods into other environments. By improving data quality, incorporating finetuning, and scaling up the parameter count; it may be possible to replicate the success seen in Minecraft. Ultimately, the goal is to establish generalizable, reproducible pipelines for training foundation models on undergraduate resources.

#### References

- [1] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to Act by Watching Unlabeled Online Videos. arXiv preprint arXiv:2206.11795, 2022.
- [2] Brockman Greg, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. arXiv preprint arXiv:1606.01540, 2016.

- [3] Shalev Lifshitz, Keiran Paster, Harris Chan, Jimmy Ba, Sheila McIlraith. STEVE-1: A Generative Model for Text-to-Behavior in Minecraft. arXiv preprint arXiv:2306.00937, 2023.
- [4] Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, Anima Anadkumar. MineDojo: Building Open-Ended Embodied Agents with Internet-Scale Knowledge. arXiv preprint arXiv:2206.08853, 2022.
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385, 2015.
- [6] Ilya Loshchilov, Cheng-Ping Hsieh, Simeng Sun, and Boris Ginsburg. nGPT: Normalized Transformer with Representation Learning on the Hypersphere. arXiv preprint arXiv:2410.01131, 2023.
- [7] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv preprint arXiv:1905.11946, 2019.

# A Appendix

# A.1 Full Size Figures

Showing Class Distribution and Accuracy per Class from Figures 4 and 5 at full size. These might be hard to read exact values in the previous Figures.



Figures 4:









# **Academic Advising Chatbot**

Avari Caya and Samantha S. Foley

Computer Science & Computer Engineering Department

University of Wisconsin-La Crosse

La Crosse, WI 54601

acaya@uwlax.edu, sfoley@uwlax.edu

#### Abstract

In the computer science and computer engineering department (CS&CPE) at the University of Wisconsin-La Crosse (UWL), faculty are responsible for academic advising for their majors. This means that students seek assistance from their advisors on what classes to take in the upcoming semester, and guidance on navigating their academic careers. The goal of this research is to develop a locally hosted chatbot from open-source software that can answer common student advising questions. In this paper we will present a survey of chatbot platforms that we considered, followed by the design and implementation of our chatbot. The survey will present a variety of chatbot platforms ranging from no-code/low-code packaged cloud instances, open-source tools that use cloud services, and opensource projects that allow you to build and host your own chatbot. Our chatbot is written in Python and uses the open-source packages, such as ollama (a large language model (LLM), to compose a proof-of-concept product. This chatbot is hosted on our department resources. We have built a custom knowledgebase with our course catalog information for CS&CPE classes that the chatbot uses to answer basic questions about courses. Finally, we will describe how the chatbot was tested. Moving forward, the goal is to enhance the chatbot's capabilities to answer a wider range of queries, including commonly asked advising questions compiled by the CS&CPE faculty, and integrate it into the CS&CPE homepage.

# **1** Introduction

Academic advising is a rewarding but time-consuming activity that many faculty do in addition to their other teaching, research, and service duties. It is something that takes many forms but generally encompasses the following tasks: (1) advising students on what classes to take, (2) being a resource for other services on campus (e.g., financial aid, registrar, career services, student life), and (3) sounding board for students as they consider school-life balance, changing or declaring a major or minor, apply for internships and jobs, and consider their life trajectory after college. In each of these categories of interactions, there are simple questions that advisors answer over and over and over again, however there tends to also be unique interactions since each student's path and questions are unique. The purpose of this project is to determine if a chatbot can be created to answer the common questions allowing students and advisors to spend their advising time and energy on the harder questions that require nuance.

The interactions with students with respect to (1) are largely dependent on understanding the course offering patterns, current offerings of classes, the curriculum (in particular the prerequisite structure), and the relative workloads of the classes. These are largely answerable with a chatbot if it is trained on the curriculum, course catalog, and updated with current course schedule information. Interactions with students often also fall into category (2) where students are seeking information about campus services, and the faculty adviser is the first point of contact. These referrals to existing websites, procedures, and offices are also easily answerable by a chatbot trained on the specific resources at that university. The last category of interactions is trickier to encapsulate in a chatbot service. Referrals to career services, procedure for changing a major/minor, and other offices like student life are likely to be the only items that would be better addressed by a chatbot versus a real human.

Assuming it is possible to create, train, and test the chatbot to accurately answer the questions described above as easily answered by a chatbot, there are several benefits to students and faculty. First, it will reduce the barrier to get answers for students who can access the chatbot whenever the need arises, and they do not need to wait for an answer from their adviser. Second, this will free up time for the advisers so they can have more time for the tougher advising questions and their other duties. In this project, we aim to create a chatbot and see if it can answer a basic set of advising questions accurately as a proof of concept. Future work will dive more deeply into the extent to which a chatbot can aid students and faculty in academic advising. Before describing our implementation, we will describe the general structure of a chatbot and survey the different types of chatbot platforms.

# 2 Architecture of a Chatbot

Chatbots are advanced computer programs designed to simulate human conversation and engage with users [1]. There are many types of chatbots, ranging from basic decision tree

1

models to sophisticated virtual assistants powered by generative AI alongside other cutting-edge technology. Every chatbot has the following components: (1) conversational interface that is web or mobile based, (2) natural language processing component that can interpret the human user input as intents, and (3) underlying knowledge base and data storage that map the intents to the (hopefully) correct responses that are then delivered to the user.



Figure 1. Chatbot components and workflow. [Image from: https://www.engati.com/blog/what-is-a-chatbot]

Modern chatbots can be categorized into three broad categories depending on the extent to which AI is used:

**Decision-tree** - a tree is created that dictates how the conversation should flow from question to answer in the conversation. It requires the builder of the chatbot to understand and explicitly state the possible questions and answers and conversation flow.

AI chatbots – use AI to manage the interpretation of user questions into intents, a more generalized form of the questions, and map them to answers. AI can be further used to consume data to create the intents and answers from a variety of sources, as well as continue to learn over time.

**Virtual assistants** – similar to AI chatbots, they use AI to manage the human interactions and understand and react to them, however they can also be more integrated into other services and software to perform more complex actions.

For our application, an AI chatbot that uses AI for natural language understanding (NLU) and is trained on our specific data for advising would be ideal. Our chatbot does not need advanced capabilities or integration with other campus services, like the ability to change a student's major or register for a class. Additionally, this is a proof of concept and the budget for this project is \$0.

# **3** Survey of Chatbot Platforms

If you search for chatbot platforms, many options are available. We initially wanted to choose a simple platform that would allow us to quickly and easily build our proof of concept without much experience or knowledge of programming and chatbots. In our search we found that chatbot platforms come in three general categories.

#### 3.1 Low-code/No-code Hosted Platforms

Low-code/no-code hosted platforms do not require users to write code and often use a drag and drop interface to manage instances and setup the intents. These platforms are often hosted by large cloud providers, use their proprietary software to build and deploy the bot, and pricing depends on the complexity of the bot, volume of messages, or other metrics related to the data storage and processing required to host and run the bot. It is common for these platforms to have a free tier to encourage users to try out the service, but the pricing models are difficult to understand unless you know how much storage the bot will end up using and how many messages your users will need. This pricing model is especially difficult to navigate for users who are new to chatbots and do not have a good sense of the scale of the bot they are creating.

IBM's WatsonX Assistant [2] is a good example of one of these chatbot platforms. It includes a visual builder tool for easy chatbot creation. It integrates several advanced features like voice assistance, learning from customer interactions, and the ability to escalate a conversation to a human agent. Other large companies such as Amazon [3] and Google [4] also offer similar services, as well as independent services Yellow.ai [5].

Overall, this is a good approach for someone who wants a high-quality product with many advanced features, does not want to write much code, and will need or want a high level of support from the company to build and run the chatbot. It is also good for someone who has the money and workload to justify paying for this service. The downside is the complex pricing structure which likely leads to hidden costs for small users, potential security and privacy concerns due to using external LLMs and data hosting, the potential of paying for services a simpler chatbot will not need. For our usecase, we ruled out this class of chatbot platforms due to price and complexity.

#### **3.2 Chatbot Services Hosted in the Cloud**

We continued our search for chatbot platforms that are open source, however many of these continued to use cloud platforms and services. Several open-source, cloud hosted chatbot platforms are described in [6]. In general, this set of solutions have fewer advanced features and were built to work with a specific set of software systems and integrations (e.g., AWS, Azure, Dialogflow, Luis, Facebook Messenger, Slack). These platforms, while marketing themselves as open source, rely on several other services that tend to use a pay-as-you-go model and usually require to you integrate the UI through a messaging platform like Slack. The language models used tend to be hosted in the cloud

and accessed via APIs. They varied from pretrained to trainable, depending on what you pay for.

One of the solutions we explored was botonic [7]. Botonic seemed like a good solution as it let us host our own website, build our own messaging service, and had good documentation and examples to follow. However, we struggled to get the NLU model to work or use an external NLU model instead. Several of the external NLU models require API keys or logins to access even the free tier of service. This was not ideal for a student project that will need to be passed from one student to the next and while experimenting with the system, we are not sure how many requests we will generate in development. Lastly, if it was going to take this much work to get the system to run, we were not getting much value and did not have as much control over the LLM and training data.

#### **3.3 DIY Chatbot**

The last type of chatbot we considered is one where we utilize open-source software to build each piece of the chatbot and host it locally on a server at our university. By looking at the other chatbot platforms, we learned about how chatbots are created and the major components, and we also learned that we wanted to learn more about how they work and have control over the process more than just creating something quick. This led us to explore how to build a chatbot from scratch, host it on our own servers, and be able to swap out the components as we learn more.

The benefit to this approach is we can fully control the cost. We have a local cloud system in the department we can use to create a VM for the project and provision it with enough resources to run a smaller LLM for our project. Additionally, since we control the LLM and where it runs, we can better control the privacy and security of the system. Lastly, we can continue to learn how the LLMs work which may lead to more research opportunities for students. The major downside is the resource usage locally, and additional labor to write and maintain the chatbot. Ultimately, the cost, control, and ability to work on just the pieces we are interested in as the project grows were the determining factors in our decision.

# 4 Academic Advising Chatbot

The chatbot we have created is written in Python. The primary resource for the current version of the product is a YouTube video by pixelgami called "RAG + Langchain Python Project: Easy AI/Chat For Your Docs" [8]. The tutorial videos and other documentation helped us build a Python program that allows users to ask questions about courses offered by the Computer Science & Computer Engineering department at UWL. The functionality was integrated into a basic website for users to access.



To build our chatbot we used a RAG (Retrieval Augmented Generation) approach [9]. RAG is the process of optimizing outputs of a Large Language Model (LLM), or in our case a natural language processing model (NLP), by integrating external knowledge bases. This method eliminates the need to retrain a model on a specific dataset, keep the cost of the project at zero and allowing for more developer control. UWL already had a comprehensive knowledge base on the University official website covering course information for CS & CPE so implementing RAG approach was a natural choice.

To make the knowledge bases easily accessible to the NLP we chose, we decided to use a vector database, Chroma [10]. Chroma is an open-source, AI application database designed for storing and retrieving vector embeddings. The advantage of a vector database is the ability to store textual information as a vector, this allows for similarity search. This feature is essential to providing our chatbot with the relevant context it needs to give accurate answers. To convert the text to a vector, we employed an open

embeddings model. An embedding model generates vector representations of text, allowing the text to be stored in the Chroma. The model we chose was Ollama's nomicembed-text [11, 12] due to its status as free and open source, as well as its competitive performance. Once we had the tools needed, it was simple to manage our data in a way that can be easily integrated with our NLP model to give a response based on accurate and up-to-date information.

With the knowledge base properly prepared, finding and delivering relevant information is a much more efficient process. To ensure the chatbot delivers the desired response, we 1) carefully create a prompt that would give the NLP model guidelines for how it should reply, 2) query the database for relevant data chunks, and 3) incorporate the chunks into the prompt that will be fed to the NLP model. To first create a prompt, we considered what acceptable answers would look like for common student questions. From here we could textualize some of the guidelines we came up with. Later in the development process, we did further testing to refine the prompt for more accurate outputs. After the prompt template was completed, we moved onto querying the database. Leveraging Chroma integration with LangChain [13], this process was simplified. First, you take the question the users asked, embed it, and compare the embedding to other embeddings in the vector database and return the most similar result. From here we can insert the chunks into the prompt and sent it to the NLP model.

All these components form the backend NLP model and knowledgebase infrastructure. The web frontend is a simple Python website using flask that allows users to interact with the chatbot. Most of the work thus far has focused on building the chatbot backend and confirm that it works.

The chatbot was tested by studying the input material, in our case the course catalog information, and specific questions were asked to which we knew the answers. If the chatbot answered correctly, we considered that a successful interaction. We did this with several different courses and different questions about courses (prerequisites, content, offering patterns) and asked using different language to test the chatbot. We also tested questions that the chatbot should not know and confirmed that the chatbot can answer that it does not know the answer.

# **5** Conclusions and Future Work

In this project we built a proof-of-concept chatbot and trained to it answer questions of about courses offered by the CS&CPE department at UWL. Throughout the experience we explored several different approaches to building a chatbot, but ended up choosing a DIY approach and assembled the pieces ourselves. The current implementation can only answer a limited number of questions and has been tested mostly for functionality. Future work involves more rigorous and systematic testing for robustness, understanding a range of questions and language to ask those questions, and accuracy of the answers. This work will be done first, before expanding to larger knowledge bases and user tests.

Once the chatbot is shown to pass these more rigorous tests, we will further extend the work to answer a broader set of common academic advising questions. The faculty in the CS&CPE department will create a FAQ document with the common questions and answers. The faculty will work to make sure the document covers as many straightforward questions and the corresponding questions are complete and correct. If any questions yield a difference in opinion or more context is needed to answer a question, then we will not add it to the FAQ. We will explore training the bot to respond to "it depends" type of questions with a response to seek guidance from their academic advisor. Additional testing will be performed as we add more material to the bot.

One of the ways we will work to understand how well the bot is performing is to log the questions and answers. This will help us determine how well the chatbot is answering questions, allow us to debug answers, and once we start user testing keep track of questions the bot is not able to answer so we can consider building answers for those questions.

The department looks forward to testing and using this new tool so that students can get answers to their questions more quickly and easily, and thus leading to more impactful advising sessions dedicated to the bigger picture issues.

# References

[1] IBM Think. 2025. Retrieved from <a href="https://www.ibm.com/think/topics/chatbots">https://www.ibm.com/think/topics/chatbots</a>

[2] IBM Watson Assistant Resources. June 2023. Retrieved from <u>https://developer.ibm.com/learningpaths/get-started-watson-assistant/</u> in 2025.

[3] Amazon Lex – AI Chat Builder. 2025. Retrieved from <u>https://aws.amazon.com/lex/</u>

[4] Conversational Agents and Dialogflow. 2025. Retrieved from <u>https://cloud.google.com/products/conversational-agents</u>

[5] Yellow.ai. 2025. Retrieved from https://yellow.ai/

[6] Sarah Chudleigh, 14 Best Open Source Chatbot Platforms to Use in 2025 (January 5, 2025). Botpress.com. Retrieved from <u>https://botpress.com/blog/open-source-chatbots</u>

[7] Botonic. 2025. Retrieved from https://botonic.io/

[8] pixelgami. Nov. 20, 2023. RAG + Langchain Python Project: Easy AI/Chat For Your Docs. (Video) Retrieved from <u>https://www.youtube.com/watch?v=tcqEUSNCn8I</u>

[9] Amazon Web Services, Inc. 2025. Retrieved from <u>https://aws.amazon.com/what-is/retrieval-augmented-generation</u>

[10] Chroma. 2025. Retrieved from https://www.trychroma.com/

[11] Ollama. 2025. Retrieved from https://ollama.com/

[12] Nomic Team. 2025. Retrieved from https://ollama.com/library/nomic-embed-text

[13] LangChain, Inc. 2025 https://python.langchain.com/docs/integrations/vectorstores/chroma/

# A beginner-friendly environment for exploring error messages in the Clojure programming language.

Tristan Kalvoda, Elena Machkasova, Jaydon Stanislowski, and John Walbran Computer Science Discipline University of Minnesota Morris Morris, MN 56267 kalvo007@umn.edu, elenam@umn.edu, stani152@umn.edu, walbr042@umn.edu

## Abstract

The Clojure programming language, built on the Java Virtual Machine, is a widely used functional language. Its simple syntax makes it a powerful language to introduce beginner programmers to key concepts of programming. However, debugging in Clojure can be unintuitive because Clojure errors are exceptions in the underlying Java language. They frequently contain terminology intended for experienced developers, leaving out important context. Our research project, Babel, aims to provide beginner programmers with a tool that replaces Clojure error messages with clearer explanations, removing unfamiliar jargon. This is work in progress, with the current goal of improving the user experience by using Morse, a third-party viewer, to present error information in a more beginner-friendly format. In this paper we discuss the structure of Clojure error messages and explain the data flow for the error messages processing. We discuss options for formatting the error messages provided by Morse, as well as the principles that guide our choices. We present examples of views and interactions for Clojure error messages. We conclude with a discussion of future directions.
# **1** Introduction

The Clojure programming language is a functional language in the Lisp family, developed and released in 2007 by Rich Hickey [7]. Clojure is implemented in Java, compiles to Java bytecode, and runs in the Java Virtual Machine (JVM), allowing programmers to use Java libraries directly from Clojure. However, unlike Java, it is a dynamically typed functional language with immutable data as a default (mutable data must be specifically declared). Clojure quickly became known in industry. Even though the number of Clojure programmers is not a large percentage of all developers (1.3% of professional developers reported extensive use of Clojure), it remains one of the most admired languages (68.5%) and is the 3rd highest-paying programming language, according to the 2024 Stack Overflow Developers Survey [2].

There is a long history of teaching Lisp as the first programming language, starting from the famous "Structure and Interpretation of Computer Programs" curriculum [3] and continuing with the introduction of the Racket programming language that provides a tiered system of Lisp-based beginner languages [5, 1]. While Clojure is also a Lisp language and shares the benefits of focusing on development of functions that follow the structure of the data representation, some of Clojure's aspects make it more challenging than other Lisps for novice programmers.

One such aspect is Clojure error messages: due to Clojure implementation via the underlying Java language, Clojure errors are Java exceptions. They frequently contain terminology intended for experienced developers, leaving out important context. As a result, what would otherwise be a powerful educational tool presents a difficult learning curve for new programmers.

A project at UMN Morris, called Babel, aims at providing alternative beginner-friendly error messages and other tools that help beginner programmers to interpret Clojure errors. The current state of the project provides a tool to rewrite the majority of error messages into a clearer and more specific form - for example, specifying the erroneous arguments to functions. The challenge at this point lies in connecting this tool to a variety of possible ways of interacting with Clojure (via an interpreter or in IDE, for example). We also are working on providing features to give the users more resources and more control when they are receiving an error message. The resources include explanation of terminology (for example, what does the term "hashmap" refer to?), links to documentation for functions, etc. Giving users more control allows for optional expansion of some information, such as the stack trace. This work is still in progress. In this paper we discuss our approaches to tackling this task, demonstrate our current achievements, and discuss future directions.

The rest of the paper is structured as following: Section 2 provides background on Clojure, its error messages, and the Clojure feature called *spec* that we utilize on our work. Section 3 discusses how Babel processes error messages. Section 4 describes the interactive environment that we are developing, both from the implementation side and from the standpoint of design for the users. Section 5 discusses the current state of the project and future directions.

### 1.1 Related Work

Previous work on the topic of improving error messages in general for beginners [4] revealed that certain visual design choices can have a strong impact on a user's comprehension of error data. Dong and Khandwala found that, out of text coloring, increased white space, and collapsible components, adding whitespace to separate important sections of the error message helped to aid with understanding the most overall, and that employing theories of visual perception in the design of error messages help to improve their usability. We are using these insights as guidance for our approach.

A 2017 paper by Wrenn and Krishnamurthi [10] on error handling in Racket, another Lispbased programming language, suggests treating error messages as classifiers, using highlighting tools to mark up error messages. This is somewhat similar to our own approach in that we extract components of error messages and classify them using keywords to determine appropriate styling choices that make them more navigable to the end user.

An earlier paper on Racket [8] suggests that the vocabulary used in error messages plays a key role in the ability to understand them, and students often have a hard time navigating technical language intended to point them to specific issues. This knowledge is important to our own research, as native Clojure error messages often contain highly technical vocabulary as well, and we make an effort to replace such terminology with more universal language.

# 2 Background

## 2.1 Clojure and its Error Messages

Clojure syntax utilizes the prefix notation, common for Lisp languages: a Clojure expression is enclosed in parentheses that contain a function followed by its arguments. For example, (/ 6 2) denotes an expression 6/2 and results in 3. Clojure is dynamically typed, and no types are declared.

In addition to lists (common in other Lisp languages), Clojure employs a variety of collections, such as vectors, hashmaps, and sets. It comes with a large number of predefined functions. Below we introduce collections and functions used in examples in this and subsequent sections.

Vectors are enclosed in square brackets; commas between elements are optional and usually omitted. For example, [1 -1 0] denotes a vector of three elements: 1, -1, and 0. By convention Clojure uses ? at the end of the names of predicates, i.e. functions that return a boolean. For example, neg? takes a number and returns *true* if it's negative and *false* otherwise. Likewise, even? takes an integer and returns *true* if it's even and *false* otherwise. A function filter is used to select elements of a vector, or another collection. It takes a predicate and a collection and returns a collection with only the elements that satisfy the predicate. For example, (filter neg? [1 -1 0]) would return a collection with only one element, 1. A subtle point is that Clojure also allows passing just a predicate to filter, without a collection: (filter neg?). This returns a function that can be invoked on a vector at some later point. Since this use is allowed, the correct number of arguments for filter is one or two.

Clojure is a hosted language, running on the Java Virtual Machine (JVM), which is a bytecode interpreter that allows a program to run on any computer without needing to be rewritten for each one. The JVM works by compiling code into bytecode, which it can then run. Clojure code can be loaded into the JVM from a file or directly via an interactive interpreter known as REPL, which stands for Read - Evaluate - Print - Loop. In our examples we assume that the code is entered into REPL. The REPL prompt is denoted with =>, and the Clojure response follows.

When Clojure code is executed, it compiles into JVM bytecode, allowing it to use many of the features that the JVM provides, but most importantly for the work presented here is that it uses the Java exception system for handling errors and when an exception is thrown in Clojure, it is a Java exception.

An exception is an event or error that disrupts the normal flow of a program's execution. In Clojure, exceptions typically occur when something goes wrong, such as dividing by zero or passing an inappropriate argument to a function. It's important to note that Clojure syntax errors will also result in an exception, because during the compilation of dynamically loaded Clojure code into JVM bytecode, any syntax error will cause an exception.

Error messages are generated when an exception occurs. They provide information about the type of error as well as where it happens. As an example of one such mistake, a novice to Clojure might write the following expression

(filter neg? 1 -1 0)

instead of the correct version (filter neg? [1 -1 0]), as described above. This causes an error, because the filter function was expecting either one or two arguments. Here 1, -1, and 0 are not in a collection, presumably because the user forgot to include them in a vector. Instead, they are being passed as separate arguments, which causes the function to be overloaded with inputs it doesn't expect.

Execution error (ArityException) at user/evall (REPL:1). Wrong number of args (4) passed to: clojure.core/filter

Clojure presents this error as an *ArityException*, referring to the number of arguments of the function. The message is cryptic for beginners, especially without the documentation.

Consider another example that we use to detail representation of Clojure error messages:

=> (/ 9 0)
Execution error (ArithmeticException) at user/eval1
 (REPL:1).
Divide by zero

The code is causing an error because it is trying to divide by zero, Let's break apart the error message:

- ArithmeticException is the type of error that happens.
- user/eval1 (REPL:1) is the location of the error; since for this example we assume that the code was directly typed into REPL, the location is REPL, line 1.
- Divide by zero is the given description for the cause of the error.

The error message is a part of a Java exception object that contains more information.

The exception object for (/ 9 0) contains the following information, represented as a nested Clojure hashmap, i.e. a collection of key/value pairs. The key names are preceded by : The exception typically contains a stack trace (see the keyword :trace) listing the sequence of function calls that resulted in the error. While our project includes a way of reducing the stack trace to a few informative lines, this paper is not focusing on the stack trace.

```
#error {
    :cause "Divide by zero"
    :via
    [{:type java.lang.ArithmeticException
        :message "Divide by zero"
        :at [clojure.lang.Numbers divide "Numbers.java"
            190]}]
    :trace
    [[clojure.lang.Numbers divide "Numbers.java" 190]
    ... omitting 18 lines...
    [clojure.main main "main.java" 40]]}
```

In addition to the above information, the exception contains the phase of Clojure execution when the exception occurred. For example, in the earlier case of the divide by zero error, the exception occurs in the execution phase. Since an error can occur at any point in the Read-Evaluate-Print-Loop, there are several possibilities of the phase. Below we list some of them:

Phase	Description	
:read-source	Error while reading characters at the REPL or from a source file	
:compile-syntax-check	Syntax error caught during compilation.	
:execution	Any errors thrown at execution time.	
:read-eval-result	Error thrown while reading the result of execution.	
:print-eval-result	Error thrown while printing the result of execution.	

Knowing the phase of the evaluation is helpful for detecting what kind of error has occurred and what information the user would benefit from.

### 2.2 Clojure Spec

Clojure *spec* is an addition to Clojure first released around 2016 [6]. It is a set of features that allow setting specifications, particularly for functions, that are checked before the function call. These specifications typically include the number and types of arguments, although they may include other conditions. The flexible nature of spec allows programmers to choose what parameters to check and, to a degree, what information will be communicated about why the arguments failed to satisfy spec.

As an example, one may specify that a function f takes no fewer than 2 and no more than 4 arguments, the first argument must be a non-negative integer, and the second one must be sequence. Types (or any other conditions) are specified as predicates. For example, we can require that the first argument of f satisfy both number? and pos? predicates, and the second satisfies seq? (a predicate that determines if a Clojure object is a sequence). Any existing Clojure predicate can be used, or the user can write their own.

If arguments not satisfying the spec are passed to the function, a spec error will occur. A spec error is an exception of the type clojure.lang.ExceptionInfo that contains data describing the error, including the failed predicate. For example, if the second argument of the function f above is a number 5, and not a sequence, the error message states that it failed the predicate seq? Even more importantly, the argument that's causing the error (in this case 5) will be included in the spec error message.

Without using spec, the error message would be a Java ClassCastException that doesn't report the value of the failing argument, only the incompatible types. The more detailed information provided by spec allows us to make error messages more understandable to novice programmers, as described in Section 3.1.

# 3 Babel Project

## 3.1 Overview of the Project

Our research project, Babel, is a tool intended to help novice programmers understand Clojure errors. It does this by providing learners with substitute error messages that describe code problems in simpler terms, replacing the unfamiliar jargon and low-level clutter of native error messages. For example, the error message produced via the function call below, where count is a function that returns the length of a sequential collection:

```
=> (count 1)
Execution error (UnsupportedOperationException) at user/
    eval1529 (REPL:1).
count not supported on this type: Long
```

is transformed into a new message, replacing the reference to the internal type Long and removing details about the Java exception object and irrelevant internal information:

```
=> (count 1)
```

```
Function count does not allow a number as an argument in
this position.
In Clojure interactive session on line 1.
Call sequence:
[Clojure interactive session (repl)]
```

Babel creates error messages like these through simple analysis of the native error's Java exception class and message contents, and one could argue that for cases like these, the native error message suffices. However, for more complicated problems, such as incorrect calls to functions or malformed arguments, Clojure's messages get more difficult to parse. Consider an erroneous code fragment described in Section 2.1:

```
=> (filter neg? 1 -1 0)
Execution error (ArityException) at user/eval1540 (REPL
   :1).
Wrong number of args (4) passed to: clojure.core/filter
```

In these situations, it would be ideal to have a detailed explanation of what the offending arguments in the function call are, and what was expected. This is where Babel makes use of the Clojure spec library, described in Section refsubsec:spec-overview, in its error reporting.

By employing specs on core Clojure functions, it is possible to explicitly specify the correct form of a function call and how the function was used incorrectly. Using spec, we know that the filter function must take a predicate and, optionally, a collection, therefore expecting one or two arguments. With Babel, the above error message is replaced as follows:

```
=> (filter neg? 1 -1 0)
Wrong number of arguments in (filter neg? 1 -1 0): the
function filter expects one or two arguments but was
given four arguments.
In Clojure interactive session on line 1.
Call sequence:
[Clojure interactive session (repl)]
```

Moreover, the Babel spec for filter also reports incorrect types of arguments:

```
=> (filter neg? 3)
The second argument of (filter neg? 3) was expected to be a
   sequence but is a number 3 instead.
In Clojure interactive session on line 1.
Call sequence:
[Clojure interactive session (repl)]
```

Our project presently includes specs for many pre-existing Clojure functions and rewrites a majority of potential Clojure errors that cannot employ spec (such as divisions by zero, un-

matched delimiters, etc). With much of the internal work done at this stage of the project, our focus is now on the potential of third party tools to display modified error messages in an interactive viewer that will allow beginner programmers to explore them more dynamically and familiarize themselves with how Clojure itself handles error data, enabling them to view the data in a more organized way and easily explore related documentation.

## **3.2** Processing Error Messages

Babel constructs its error messages by intercepting exceptions directly from REPL. This is done by attaching an observer function to a running instance of a REPL server. The observer passes exceptions to babel processing function and replaces the message with the result. Before developing the interactive interface, the message was returned as a string. The processing for the interactive interface is described in Section 4.

Since different kinds of exceptions contain information in a different way, the first step in processing them is to determine what category they belong to. Errors can be categorized broadly into spec errors and non-spec errors

- Spec errors occur when the data does not conform to the specification for a function or a similar construct. These errors can be further divided into Babel specs, Clojure internal specs, and third-party specs. Babel specs were specifically designed by our group to provide as much information as possible to users, and in particular to novices. Clojure internal specs are provided for some specific Clojure constructs and provide useful information, although not necessarily phrased in a way suitable for beginners, so our software rewrites them to a more beginner-friendly format. Finally, third-party specs come with libraries imported into the code base. Since we don't know the semantics of these specs, we can only suggest what the intent is. Care should be taken not to second-guess, and thus obscure, the intent.
- Non-spec errors occur when errors happen outside of the clojure.spec system. They are typically either syntax error that happen before a function can even be called, or runtime errors that are not checked by spec, such as division by zero or any other errors that arise during program execution.

Due to the complexity of Clojure runtime system on the JVM, it may not be immediately obvious what type of error has occurred. For example, Clojure uses *lazy sequences*: many functions that return a sequence, such as filter, don't evaluate it immediately. The result is evaluated only when it's needed, and only as many elements as needed are evaluated. This may result in evaluating the sequence at the :print-eval-result phase. If an error occurs at that point, it may manifest itself as a nested error (an exception attached as a cause to another exception) since it occurs within a printing function.

In order to determine the type, the following properties of exceptions are used: *error types*, *levels of nesting*, and *phase*.

• Grouping by error types or subtypes can help find patterns in errors that make it easier to prescribe solutions to errors, based on similar error types.

- As we mentioned before, an exception may be nested within another exception if it is produced by a lazy sequence or in some other cases. Grouping by levels of nesting and the order in which errors occur provides more insight into the path the error took before it becomes visible, helping us understand the sequence of failures.
- Classifying by phase (such as :read-source, :execution, and :print-eval-result) offers additional context about when and where the error occurred.

Using the above in conjuction with each other allows us to find the elements of an exception that can be used to explain its cause to the user.

# **4** Adding Visual Interactive Environment

We explore using the third-party Morse viewer [9] as a supplementary way to interactively present error information in a better, more beginner-friendly format. This approach provides an interactive, visual layer of error message presentation on top of the existing REPL error message display. For instance, unlike default Clojure error messages, the Morse viewer allows us to use monospace font for code snippets, and highlight function names with color coding. Additionally, for spec errors, we can dynamically generate the link to the documentation of the function where the error occurred, provided it is a predefined Clojure function.

Morse is a tool designed for data exploration and visualization. While it was originally designed to help experienced developers navigate large datasets and projects, it provides tools and extensibility to be curated with a beginner audience in mind. Morse uses a system of customizable viewers to achieve its data visualization. Each view is individually designed to display a specific data structure based on given conditions.

### 4.1 Running a Babel Session with Morse

By default, the Clojure REPL does not expose the correct hooks to effectively manipulate the data from error messages. In order to get around this, initializing Babel within an existing REPL session, using a custom-made (babel/init) function to connect the processing hooks, creates a new custom sub-REPL (a new REPL session) above the existing session. Creating a sub-REPL allows us to configure the behavior attached to the following hooks in the new session:

- :init Defines the initialization behavior of the sub-repl. In Babel, this starts a new Morse session connected to the new REPL.
- :eval Defines the behavior whenever a command is executed within the REPL. In Babel, this taps the command into an atom (a mutable state variable), and evaluates the command within both Morse and the REPL.
- : caught Defines the behavior on exception. In Babel, this processes the error as described previously, and passes the following information about the error to Morse to be displayed in a custom viewer:

- The last command entered, read from an atom that is updated at evaluation.
- The location in the environment where the error occurred. In the REPL, this resolves to "Clojure Interactive Session".
- A vector of pairs containing the error message produced by Babel, with labels associated with each segment denoting its type for formatting.
- The url to the documentation of the function called that caused the error.

Once the sub-REPL is initialized, the user can then use the sub-REPL as usual, but with the addition of Babel error messages when applicable in the terminal, and a parallel pop-up Morse viewer that format the error message in an easier-to-parse way, providing interactive display for the user, for example giving them the ability to click on the documentation link.

In order to provide a message in the format suitable for More viewer display, Babel processing module returns the message as a sequence of key/value pairs, where each key indicates what part of error message it is. Fir example, the code part of the message will be labeled as :code, which allows displaying it in the monospace font common for code.

## 4.2 Setting up Morse Viewers

Morse viewers can be created to have custom behavior around certain data. Creating a new viewer in Morse requires the following:

- A predicate defining the constraints of the data that can be displayed in the viewer.
- The graphical format of the view to be displayed. This is built with the JavaFX library.

The Babel viewer takes a map of input data as input, and defines the predicate for the viewer by checking for a map with the necessary fields. For the formatting definition, Babel uses the built-in WebView object in order to format the error data with an HTML template, enabling us to leverage the large, robust tools of HTML and CSS to define the visuals of the error messages.

## 4.3 Exploring Design of Error Messages Interface

An example of the error presentation for Babel is shown in Figure 1, displaying the error message for the expression (even? 1 2). Code segments are displayed as monospace and isolated. Additionally, function names within the body of the error message are color coded.

# 5 Current State and Future Work

Our project is currently focusing on the goal of utilizing Morse viewers for creating intuitive, robust interactive display of error messages. A lot of effort was put into figuring out the technical setup and data flow to accomplish this task. It also required changing the

	Follow REF	Metadata defas: vamame Viewer: reb/testWebView 👻
p [0]	tap	Your code had an error in <i>Clojure interactive session</i> on line 1.
expr error-map	val {:form (even?	(even? 1 2)
(even? 1 2)	{:via [{:type cl	Wrong number of arguments in (even? 1 2): the function even? expects one argument but was given two arguments. In Clojure interactive session on line 1.
		Documentation for the reported function. https://clojuredocs.org/clojure.core/even_q
<	>	

Figure 1: Morse visualization of the arity error on even? with too many arguments (right), with session evaluation history (left).

way an error message is passed to the display by adding tags (such as :code) for various elements of an error message.

We have accomplished creating a prototype for these interactions and the data flow. However, as the project stands right now, many features of the interactive display are still in active development:

- More work needs to be done to convert all of the exception processing into the tagged format.
- The ability to hover over specific terms used in error message for explanations of what they mean, as well as to navigate collapsible stack traces, would be useful for making the tool more interactive.
- Figuring out the user's workflow in alternating between correct code and errors is also a future work item.

In the future, we would like to conduct usability studies with the interactive tools offered by Morse, after we have developed the tool enough to cover more of Babel's capabilities. Allowing research participants to experiment with the tool for learning Clojure would help drive the direction of the project and improve upon the design choices. We would also like to explore IDE integration as a potential avenue of development for Morse, in order to make the tool more useful for learning to code as a whole.

# 6 Acknowledgments

We thank Joe Lane for introducing us to Morse tools and for numerous helpful discussions.

# References

- [1] Racket, the programming language. [https://racket-lang.org/, Online; accessed 3-22-2025].
- [2] Stack overflow 2024 developer survey. [https://survey.stackoverflow.co/2024/, Online; accessed 3-22-2025].
- [3] ABELSON, H., AND SUSSMAN, G. J. Structure and Interpretation of Computer *Programs*, 2nd ed. MIT Press, Cambridge, MA, USA, 1996.
- [4] DONG, T., AND KHANDWALA, K. The impact of "cosmetic" changes on the usability of error messages. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2019), CHI EA '19, Association for Computing Machinery, p. 1–6.
- [5] FELLEISEN, M., FINDLER, R. B., FLATT, M., AND KRISHNAMURTHI, S. The structure and interpretation of the computer science curriculum. *J. Funct. Program.* 14, 4 (July 2004), 365–378.
- [6] HICKEY, R. clojure.spec rationale and overview. [https://clojure.org/about/spec, Online; accessed 3-22-2025].
- [7] HICKEY, R. The clojure programming language. In *Proceedings of the 2008 symposium on Dynamic languages* (New York, NY, USA, 2008), DLS '08, ACM, pp. 1:1–1:1.
- [8] MARCEAU, G., FISLER, K., AND KRISHNAMURTHI, S. Mind your language: on novices' interactions with error messages. In *Proceedings of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software* (New York, NY, USA, 2011), Onward! 2011, Association for Computing Machinery, p. 3–18.
- [9] TEAM, C. Introducing morse. [https://clojure.org/news/2023/04/28/introducingmorse/, Online; accessed 3-22-2025].
- [10] WRENN, J., AND KRISHNAMURTHI, S. Error messages are classifiers: a process to design and evaluate error messages. In *Proceedings of the 2017 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software* (New York, NY, USA, 2017), Onward! 2017, Association for Computing Machinery, p. 134–147.



# Machine Learning-Assisted Vocal Cord Ultrasound Examination: Project VIPR

Will Sebelik-Lassiter, Evan Schubert, Muhammad Alliyu, Quentin Robbins, Excel Olatunji, Mustafa Barry

Department Electrical, Computer and Biomedical Engineering

Department of Computer Science and Software Engineering

Milwaukee School of Engineering

1025 N Broadway St, Milwaukee, WI 53202

lassiterw@msoe.edu

Merry Sebelik, MD

Department of Otolaryngology, Head & Neck Surgery

**Emory University** 

550 Peachtree Ave NE, Suite 1135, Atlanta, GA 30308

msebeli@emory.edu

## Abstract

Intro: Vocal cord ultrasound (VCUS) has emerged as a less invasive and better tolerated examination technique, but its accuracy is operator dependent. This research aims to apply a machine learning-assisted algorithm to automatically identify the vocal cords and distinguish normal vocal cord images from vocal cord paralysis (VCP).

Methods: VCUS videos were acquired from 30 volunteers, which were split into still frames and cropped to a uniform size. Healthy and simulated VCP images were used as training data for vocal cord segmentation and VCP classification models.

Results: The vocal cord segmentation model achieved a validation accuracy of 96%, while the best classification model (VIPRnet) achieved a validation accuracy of 99%.

Conclusion: Machine learning-assisted analysis of VCUS shows great promise in improving diagnostic accuracy over operator-dependent human interpretation.

# **1. Introduction**

Vocal cord paralysis (VCP) refers to a condition in which one or both vocal cords (also known as vocal folds) are unable to move, leading to significant health impacts. VCP can occur as a result of surgery (especially thyroid, parathyroid or neck surgeries), trauma, tumors, or neurological disorders. The inability of the vocal cords to move can lead to a weak or breathy voice, hoarseness, swallowing dysfunction, and in severe cases, difficulty breathing.

Examination methods to visualize vocal cords have been developed over many decades. Benjamin Guy Babington in 1829 is credited with being the first observer of the vocal cords, using a mirror and reflected light [1]. Today, flexible laryngoscopy (FL) is considered the current gold standard for examination of the vocal cords in an awake patient [2]. This technique is unfortunately an invasive one, as it involves navigating a flexible camera probe through one's mouth into the upper parts of the throat. Additionally, some types of probes are discarded after a single use, while the reusable ones must undergo a lengthy sterilization process after each use. The less invasive method of transcutaneous ultrasound of the vocal cords (VCUS) has gained acceptance due to lower cost, improved patient tolerance, reduced risk of respiratory infectious exposure, and accuracy in both pediatric and adult patients [3,4].

VCUS can be particularly suited to the pediatric population [5]. Hamilton, et al. conducted a systematic review and meta-analysis demonstrating that VCUS has high sensitivity (91%) and specificity (97%) for detecting vocal cord immobility in children, making it a reliable alternative to laryngoscopy in pediatric populations [6]. Similarly, Su et al. found that VCUS in adults has a pooled sensitivity of 95% and specificity of 99% for diagnosing true vocal fold immobility, supporting its utility in adult patients as well [7].

Despite the apparent advantages of VCUS in terms of cost and patient tolerance, it is recognized that the examination technique can be highly operator-dependent, with a range of vocal cord identification from 41% to 86% per a systematic review by Carneiro-Pla, et al in 2014 [8]. Further, some authors have recommended against using VCUS to identify VCP especially in overweight and postoperative patients [9].

To overcome operator-dependence, machine learning has been applied to vocal cord motion analysis, as exemplified in the 2010 work by Voight, et al. who used phonovibrogram features and various machine learning (ML) algorithms to achieve classification accuracies of up to 93% [10]. More recently, deep learning methods have been used to analyze video laryngoscopy images obtained from FL [11, 13]. Application of machine learning to VCUS interpretation has not yet been described.

This research aims to apply a machine learning-assisted algorithm to automatically identify and segment human vocal cords from ultrasound images, and investigates the application of multiple deep learning algorithms to differentiate bilaterally mobile vocal cords from VCP. This was done with the intent of complementing clinician examination and interpretation.

# 2. Methods

As of March 2025, we were unable to identify any publicly available datasets of significance on the internet which contained ultrasound images or videos of the vocal cords. Thus, approval from

the Milwaukee School of Engineering Institutional Review Board (MSOE IRB# I-2024-012) was obtained for a study which would allow us to record videos of VCUS scans from consenting participants and train a machine learning-based vocal cord segmentation model on the videos. Over the course of two weeks, 30 MSOE undergraduate students were recruited to provide ultrasound videos of their vocal cords. Exclusion criteria ensured that all study participants were "MSOE community members" (students, faculty, or staff) and excluded those who anticipated a possible inflammatory reaction to ultrasound gel or its ingredients. Paper flyers posted in common areas around MSOE campus were used to recruit participants and free pizza was offered as a participation incentive. Each participant verbally consented after an investigator read them an informed consent script that detailed the specifics of the study. Precautions were taken to ensure that all collected data remained de-identified and confidential. No identifying information was collected from any participant except self-identified gender and age.

Number of Male Participants	7
Number of Female Participants	23
Median Age	20

Table 1: Participant demographic breakdown

Each participant was comfortably seated for the ultrasound examiner to record a 30 to 60 second video showing their vocal cords. Care was taken to include all pertinent anatomy necessary to fully visualize vocal motion. Anatomic structures typically seen on VCUS are illustrated in Figure 1. Participants were observed during passive respiration and during phonation, throat clearing and laughter. Ultrasound videos were recorded using a GE Logiq S7 Pro ultrasound machine and linear probe at 8.5 MHz, depth 5cm, using a standardized examination technique. A single author (WSL) was trained and certified in VCUS, therefore he carried out all data acquisition. To preserve participant anonymity, all personally identifiable information, including overlaid text within ultrasound frames, was removed using a custom anonymization script.



Figure 1: Vocal cord ultrasound image with annotated pertinent anatomy

### 2.1 Data Processing and Labeling

All collected ultrasound videos were securely uploaded over a secure Wi-Fi network to MSOE's on-campus supercomputer, Rosie. Three participant videos (10% of overall dataset) were moved to a separate directory to serve as validation data, and one video was excluded entirely due to its low imaging quality. The remaining 26 videos were processed to extract every  $20^{th}$  frame, resulting in a total of 2,168 still frames parsed from video. Next, each extracted frame was resized to  $256 \times 256$  pixels via Lanczos interpolation and saved in full-color PNG format to maintain a consistent dimension and file type. Lastly, an anonymization script was run on each frame which removed any remaining text or scale bars from the images, further simplifying and standardizing the dataset.



Figure 2: Example of ROI annotation/labeling on an anonymized VCUS video frame

After cleaning and resizing, these images were set aside to have the vocal cords segmented manually by three team members (WSL, QR, MB), using the Label Studio open-source software run locally on Rosie. Each annotator identified a rectangular region of interest (ROI) by placing the upper boundary of the region at the anterior commissure (crest where the two cords meet), the lower boundary at the highest visible reverberation artifacts from the arytenoid cartilage, and the left and right boundaries directly adjacent to the visible edges of the vocal cords (Figure 2). Frames that lacked clearly visible vocal cords were discarded, leaving us with a model training dataset totaling 1,088 images and 1,088 associated ROI labels (one per image).

### 2.2 Segmentation Model Training

These 1,088 manually labeled images were used to train a segmentation model which can identify and segment the vocal cords in a VCUS image, if they are sufficiently visible. We used the YOLOv8m object detection model provided by Ultralytics as the backbone for this task, benefiting from its efficient convolutional architecture and decoupled detection head. All experiments were conducted using ultralytics==8.3.88.

Images were resized to  $640 \times 640$  during model training. The training was performed with augment=True enabled in YOLOv8, which includes default augmentations such as mosaic, mixup, scaling, and HSV color-space augmentation. The model was trained for 4 epochs using the SGD optimizer with the default learning rate, momentum (0.937), and weight decay (5e-4).

Training and validation sets were stratified from the augmented dataset. The model was trained on a single-class detection task ("vocal\_cords") using YOLO's standard CIoU loss for bounding box regression, Binary Cross-Entropy for classification, and Distribution Focal Loss (DFL) for refined localization.

### 2.3 Segmentation Model Evaluation Protocol

We used COCO-style object detection metrics to evaluate the model on the 10% validation set drawn exclusively from the three participant videos that were previously unseen by any model:

- Precision, Recall, and F1-score
- mAP@0.5 and mAP@0.5:0.95
- Confidence-curve and PR-curve analysis
- Normalized confusion matrix (vocal cords vs. background)

The model was evaluated on a validation set using a confidence threshold of 0.701, derived from the F1-Confidence curve. While clinical applications often prioritize precision to minimize false positives, F1-score was selected here to balance the high recall needed for screening and early detection tasks with acceptable precision.

### **2.4 Preparation for VCP Classification**

After building a vocal cord detection and segmentation model, we then sought to investigate the viability of using a convolutional neural network (CNN) to identify potential vocal cord paralysis from ultrasound images. We investigated two different CNN architectures to perform a binary classification task on input VCUS images: differentiating between bilaterally mobile vocal cords and cords exhibiting VCP. To address the major limitation of not having access to a source of clinical VCUS images of patients with VCP, we were able to create digitally simulated images of VCP. Vocal cord length is noted to vary by functional activity such as singing at high or low pitch [12], or clinical conditions such as paralysis. VCP causes distinctive vocal cord length difference between the paralyzed side and the mobile side [2], thus inspiring and supporting our method of digital alteration to simulate VCP.

#### 2.4.1 Synthetic Data Creation and Data Augmentation

To address class imbalance when approaching classification, we employed an "asymmetric squishing" technique to create examples of VCP from the 1,088 healthy training images we had

available to us. First, we referenced our manually created labels to crop each image down to just the ROI, so that each image exclusively contained vocal cord anatomy, and then we resized each image back up to 256x256 using Lanczos interpolation. Each ROI image was then split in half vertically down the center of the image, and each half was saved separately for the subsequent synthetic data creation steps.

To simulate the appearance of one paralyzed vocal cord being shorter than the other (mobile) cord, one of the image halves was scaled vertically by a factor of 0.75 in such a way that the top borders of the two half-images stayed lined up, but the bottom border of the "compressed" half now rested well above the bottom border of the unmodified half. Pixel information was then sampled from the original VCUS image, i.e. from outside of the labeled ROI, to fill in the "empty space" below this compressed half and make the image square again. This created the appearance that the vocal cord contained in one half of the image was noticeably shorter than the vocal cord in the other half.

Unfortunately, recomposing the compressed half with the uncompressed half often resulted in a "seam" where the pixels no longer lined up neatly. To address this, a vertical strip six pixels wide on either side of this seam line was deleted and replaced using a bilinear interpolation based on the brightness values of the pixels adjacent to the seam. This looked more natural, but unfortunately still could be considered a significant visual artifact in many of our training images. We were concerned that this artifact would induce unintended bias in our classification models if it were present only in our synthetic VCP example images, so we came up with a strategy that we hoped would mitigate this issue and simultaneously ensure that our dataset contained an equal distribution of both healthy and paralyzed training images:



Figure 3: Examples of the four classes of images created

Four image groups were created for our training data. The first group, "rightpar", contained 1,088 images that had been modified by the procedure described above to have their right sides compressed in a simulacrum of right vocal fold immobility. Similarly, "leftpar" contained 1,088 more images which had their left halves compressed equivalently. In an attempt to reduce artifact bias, the third image group called "healthy2" contained 1,088 more images that were split vertically and then rejoined using the same bilinear interpolation operation as had been used on the paralysis images, but neither half was compressed or modified otherwise. This way, the image still looked mostly "normal", but exhibited the same bilinear interpolation artifact present in the paralysis images. Finally, the fourth group contained a final 1,088 images that were not modified beyond cropping them down to the labelled ROI. After creating these four groups, we had 2,176 healthy examples and 2,176 VCP examples, or 4,352 training images in total.

Finally, data augmentation was applied to both healthy and VCP image groups to further expand the dataset and enhance model generalizability. Augmentations included random rotations of up to 10 degrees, horizontal flips, and minor affine transformations. In addition, multiple augmentations were combined sequentially, such as a rotation followed by a flip, or an affine transformation followed by both a rotation and a flip, to introduce additional variability. This approach substantially increased the number of training-ready images, raising the total to 34,816 unique images.



Figure 4: Examples of how original images were augmented to increase dataset size

#### **2.5 Classification Models**

We evaluated the performance of two different CNN architectures for classifying VCUS images as healthy or paralyzed. The first model we evaluated, YOLOv8n-cls, was pre-trained on large-scale image datasets and then fine-tuned on our augmented VCUS dataset. Then, we developed and evaluated a simple custom model architecture which we dubbed VIPRnet, shown in Figure 5.

```
VIPRnet(
  (conv_layers): Sequential(
    (0): Conv2d(1, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): ReLU()
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU()
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): ReLU()
    (8): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
 (fc_layers): Sequential(
    (0): Flatten(start_dim=1, end_dim=-1)
    (1): Linear(in_features=131072, out_features=128, bias=True)
    (2): ReLU()
    (3): Dropout(p=0.5, inplace=False)
    (4): Linear(in_features=128, out_features=1, bias=True)
 )
)
```



# 3. Results



### **3.1 YOLO Segmentation Model Results**

Figure 6: Training and Validation Loss & Metric Curves

#### **3.1.1 Training Performance**

As shown in Figure 6, the model achieved strong convergence across all loss components. Training loss decreased steadily over 4 epochs:

- **Box loss**:  $2.07 \rightarrow 1.43$
- **Classification loss**:  $2.94 \rightarrow 1.43$
- **DFL loss**:  $2.18 \rightarrow 1.68$

Validation metrics showed a sharp improvement after the initial epoch, with mAP@0.5 rising to 0.78 and mAP@0.5:0.95 reaching 0.40. These improvements are corroborated by increasing precision (0.84) and recall (0.80) values.

Metric	Value
Precision	0.84
Recall	0.80
mAP@0.5	0.78
mAP@0.5:0.95	0.40



#### 3.1.2 Confidence-Based Analysis

The Recall-Confidence curve (Figure 7) demonstrates strong recall even at low thresholds, suggesting the model rarely misses true positives. Conversely, the Precision-Confidence curve (Figure 8) indicates that perfect precision (1.00) is only achieved at a high threshold (0.952), resulting in reduced recall. The F1-confidence curve (Figure 9) peaks at 0.701, offering an optimal trade-off between the two metrics for general-purpose detection.

In the context of this research, the F1-optimal threshold of 0.701 was selected rather than a precision-maximizing threshold to prioritize sensitivity in detecting potential vocal cord anomalies. In a clinical screening workflow, this ensures that fewer pathological cases are missed.



Figure 7: Recall-Confidence curve

Figure 8: Precision-Confidence Curve



Figure 9: F1-Confidence Curve

#### 3.1.3 Spatial and Dimensional Consistency

As illustrated in Figures 10 and 11, bounding boxes predominantly cluster near the image center ( $x \approx 0.5$ ,  $y \approx 0.4$ ), aligning with anatomical expectations. A strong correlation between box width and height was observed, corresponding to the elongated shape of vocal cords.



Figure 10: Label Distribution Grid (Instances, x, y, width, height)



Figure 11: Pairwise Correlogram of Bounding Box Parameters

#### **3.1.4 Confusion Matrix**

The normalized confusion matrix (Figure 12) reveals strong performance on the single-class task, with 96% accuracy in identifying vocal cords with minimal misclassification of vocal cords as background.



Figure 12: Normalized Confusion Matrix

#### 3.1.5 Qualitative Results

Figure 13 presents bounding box predictions on representative validation images. The model consistently identified vocal cords even under conditions of shadowing or simulated asymmetry, reinforcing its robustness across imaging artifacts and augmentations.



Figure 13: Sample Detection with Bounding Boxes on Ultrasound Image

#### **3.2 YOLO Classification Model Results**

#### **3.2.1 Training Performance**

The YOLOv8n-cls model converged reliably over 20 training epochs, with both training and validation loss showing a consistent downward trend (Figures 14a–b). Top-1 accuracy increased from 0.48 to 0.93, indicating strong learning of class-discriminative features. Top-5 accuracy reached 1.00 early in training and remained flat, reflecting the binary nature of the classification task (Figures 14c–d).



Figure 14: Training and Validation Curves for Classification Model (a) Training Loss (b) Validation Loss (c) Top-1 Accuracy (d) Top-5 Accuracy

#### 3.2.2 Classification Accuracy

On the held-out validation set, the classification model achieved a final top-1 accuracy of 92.3%. As illustrated in the normalized confusion matrix (Figure 15), the model correctly identified 87% of *healthy* cases and 97% of *paralyzed* cases. Most misclassifications were false positives— classifying healthy cords as paralyzed—an acceptable tradeoff in clinical screening contexts where recall is prioritized.



Figure 15: Normalized Confusion Matrix for Vocal Cord Classification

To further evaluate classifier performance under different thresholds, a precision-recall (PR) curve was plotted (Figure 16). The curve shows high precision across a wide recall range, with a steep drop-off only at extreme recall values. This shape confirms the model's reliability in detecting true positives with minimal false positives under moderate thresholds—an essential characteristic for early-stage screening tools in clinical settings.



Figure 16: Precision-recall curve for the classification task. The area under the curve and flatness of the plateau suggests the model maintains high precision even with increasing recall.

### **3.3 VIPRnet Classification Model Preliminary Results**

Training performance for VIPRnet is presented in Figure 17. VIPRnet was trained over 50 epochs with a batch size of 64, showing a gradual and consistent reduction in both training and validation loss. VIPRnet exhibited a smooth learning curve with minimal indicators of overfitting, as evidenced by the small gap between training and validation metrics. By the final epoch, the model stabilized at approximately 99.5% validation accuracy, indicating strong performance with our synthetic data. Further analysis was unfortunately not possible due to time constraints.



Figure 17: VIPRnet training metrics

#### **3.4 Results Summary**

Despite differences in their training durations and batch sizes, both architectures achieved effective convergence. Furthermore, both models exhibit extremely high validation accuracies, which gives promise that they will perform adequately at the task they were trained for (differentiating VCUS images acquired from a small population of healthy undergraduate students apart from digitally modified versions of those same images). However, these results necessarily reflect the limitations of our study and should *not* be interpreted as a guarantee that either model will generalize to a clinical setting or exhibit these same levels of accuracy on VCUS images featuring genuine VCP anatomy.

## 4. Discussion

VCUS may be a preferable method to look for VCP as compared to flexible laryngoscopy (FL) because it is non-invasive, cost-effective, quick, well-tolerated with no patient discomfort, and accurate as reported by experts in the technique. However, it is widely perceived to be difficult to perform or interpret by those less familiar with the technique, with reported operator variability as a limitation of VCUS [8,9]. To mitigate human operator-dependence, machine learning models have the potential to provide consistently accurate analysis [13].

This study demonstrates the feasibility of using machine learning to detect vocal cord paralysis (VCP) via VCUS. By recording ultrasound videos from 30 participants, parsing the footage into still frames, and manually labeling ROIs containing the vocal cords, we were able to train a YOLOv8 based segmentation model that segmented the cords with high accuracy. Further, we explored the viability of multiple convolutional neural networks such as the pretrained YOLOv8n-cls and our custom VIPRnet architecture to classify images as healthy or paralyzed.

To address the lack of a clinical source of VCUS images showcasing true VCP, we generated synthetic VCP frames by "asymmetrically squishing" one half of each labeled ROI and blending the halves back with bilinear interpolation. This technique, combined with data augmentation, substantially expanded our training set to 34,816 unique images, while also mitigating class imbalance. Although our synthetic approach cannot fully replicate complex pathological variations and potentially caused model overfitting, it enabled us to rigorously train and tune several models with promising preliminary results.

Looking ahead, we plan to refine our approach and expand our datasets in several ways. The most obvious next step is to acquire clinical VCP data to validate the models we have developed and potentially train new ones. Barring that, a more robust use of YOLO segmentation could allow us to automatically identify and label a larger number of frames from our current and future IRB-approved data collections, thereby increasing the volume and diversity of available training images. Additional hyperparameter tuning, improved synthetic data generation techniques, and exploration of alternate classification architectures could also further boost model performance. Additionally, we would like to investigate the methodology of a recently published videolaryngoscopy-based study [13], borrowing their length measurement framework and applying it to our ultrasound dataset. Finally, as our work continues to evolve, we seek to collaborate both with other researchers to create more clinically useful tools and models, and with private medical device manufacturers to potentially incorporate our algorithms into commercial ultrasound hardware to run in real-time at the point of care.

# 5. Conclusion

Machine learning-assisted analysis of VCUS shows great promise in improving diagnostic accuracy over operator-dependent human interpretation.

# Acknowledgements

We would like to extend our thanks Dr. Jeffrey LaMack for his unwavering support, brainstorming, advice, and assistance as our team's faculty supervisor. We would also like to thank Dr. Derek Riley for his suggestions and ideas related to how we technically approached the problems of VCP classification and synthetic data generation. Lastly, we would like to thank Dr. Kristen Shebesta for her patience and assistance in securing IRB approval for our work.

# References

- 1. Harrison D. Benjamin Guy Babington and his mirror. J Laryngol Otol. 1998 Mar;112(3):235-42. doi: 10.1017/s0022215100158268. PMID: 9624371.
- Sinclair CF, Bumpous JM, Haugen BR, Chala A, Meltzer D, Miller BS, Tolley NS, Shin JJ, Woodson G, Randolph GW. Laryngeal examination in thyroid and parathyroid surgery: An American Head and Neck Society consensus statement: AHNS Consensus Statement. Head Neck. 2016 Jun;38(6):811-9. doi: 10.1002/hed.24409. Epub 2016 Mar 11. PMID: 26970554.
- Balbaloglu H, Karadeniz E, Erkek A, Yilmaz C, Ugurlu MU, Cakmak GK. Utilizing Surgeon-Performed Intraoperative Translaryngeal Ultrasound for Verifying Vocal Fold Function: A Troubleshooting Approach for Intraoperative Neuromonitoring in Neck Surgery. J Voice. 2024 Mar 15:S0892-1997(24)00062-6. doi: 10.1016/j.jvoice.2024.02.022. Epub ahead of print. PMID: 38493015.
- Noel JE, Orloff LA, Sung K. Laryngeal Evaluation during the COVID-19 Pandemic: Transcervical Laryngeal Ultrasonography. Otolaryngol Head Neck Surg. 2020 Jul;163(1):51-53. doi: 10.1177/0194599820922984. Epub 2020 Apr 28. PMID: 32340541.
- Deshpande A, Tey CS, Chanani N, Landry A, Raymond M, Sebelik M, Shashidharan S, Wolf M, Raol N. The utility of handheld ultrasound as a point-of-care screening tool to assess vocal fold impairment following congenital heart surgery. Int J Pediatr Otorhinolaryngol. 2021 Sep;148:110825. doi: 10.1016/j.ijporl.2021.110825. Epub 2021 Jul 6. PMID: 34252699.
- Hamilton CE, Su E, Tawfik D, Fernandez E, Veten A, Conlon T, Ginsburg S, Mariano K, Sidell D, Haileselassie B; Pediatric Research Collaborative on Critical Ultrasound (PeRCCUS). Assessment of Vocal Cord Motion Using Laryngeal Ultrasound in Children: A Systematic Review and Meta-Analysis. Pediatr Crit Care Med. 2021 Oct 1;22(10):e532e539. doi: 10.1097/PCC.00000000002734. PMID: 33833204.
- Su E, Hamilton C, Tawfik DS, Mariano K, Ginsburg S, Conlon T, Veten A, Fernandez E, Wong KP, Sidell DR, Haileselassie B; Pediatric Research Collaborative for Critical Ultrasound. Laryngeal Ultrasound Detects Vocal Fold Immobility in Adults: A Systematic Review. J Ultrasound Med. 2022 Aug;41(8):1873-1888. doi: 10.1002/jum.15884. Epub 2021 Nov 27. PMID: 34837415.
- 8. Carneiro-Pla D, Miller BS, Wilhelm SM, Milas M, Gauger PG, Cohen MS, Hughes DT, Solorzano CC. Feasibility of surgeon-performed transcutaneous vocal cord ultrasonography in identifying vocal cord mobility: A multi-institutional experience.

Surgery. 2014 Dec;156(6):1597-602; discussion 1602-4. doi: 10.1016/j.surg.2014.08.071. Epub 2014 Oct 8. PMID: 25304839.

- Kandil E, Deniwar A, Noureldine SI, Hammad AY, Mohamed H, Al-Qurayshi Z, Tufano RP. Assessment of Vocal Fold Function Using Transcutaneous Laryngeal Ultrasonography and Flexible Laryngoscopy. JAMA Otolaryngol Head Neck Surg. 2016 Jan;142(1):74-8. doi: 10.1001/jamaoto.2015.2795. PMID: 26632676.
- Voigt D, Döllinger M, Yang A, Eysholdt U, Lohscheller J. Automatic diagnosis of vocal fold paresis by employing phonovibrogram features and machine learning methods. Comput Methods Programs Biomed. 2010 Sep;99(3):275-88. doi: 10.1016/j.cmpb.2010.01.004. PMID: 20138386.
- DeVore EK, Adamian N, Jowett N, Wang T, Song P, Franco R, Naunheim MR. Predictive Outcomes of Deep Learning Measurement of the Anterior Glottic Angle in Bilateral Vocal Fold Immobility. Laryngoscope. 2023 Sep;133(9):2285-2291. doi: 10.1002/lary.30473. Epub 2022 Nov 3. PMID: 36326102.
- 12. Cho W, Hong J, Park H. Real-time ultrasonographic assessment of true vocal fold length in professional singers. J Voice. 2012 Nov;26(6):819.e1-6. doi: 10.1016/j.jvoice.2012.05.007. PMID: 23177752.
- Nobel SMN, Swapno SMMR, Islam MR, Safran M, Alfarhood S, Mridha MF. A machine learning approach for vocal fold segmentation and disorder classification based on ensemble method. Sci Rep. 2024 Jun 23;14(1):14435. doi: 10.1038/s41598-024-64987-5. PMID: 38910146; PMCID: PMC11758383.

## **Computational Inquiry in Undergraduate Math Courses**

Matthew P. Richey and Matthew L. Wright

Mathematics, Statistics, and Computer Science Department

St. Olaf College

Northfield, MN 55057

richeym@stolaf.edu, wright5@stolaf.edu

#### Abstract

Computational skills give students agency to explore and discover mathematics on their own, yet the development of such skills is often overlooked. Students benefit from the ability to take an open-ended question, design and run computational experiments (in a computer algebra system or programming environment), and formulate conjectures based on their observations. These skills of computational exploration are important for students to develop, as they open doors to mathematical understanding and quantitative problem solving. This paper presents ideas for equipping students with computational inquiry skills, with examples from the Modern Computational Mathematics course at St. Olaf College. This course helps students develop computational fluency and skills that they can use to discover patterns, ask good questions, test conjectures, and gain insight into mathematical questions. Examples presented in this talk could serve as inquiry-based modules for other courses in mathematics or related disciplines. Furthermore, computational exploration can make abstract topics more accessible and provides transferable skills that students can apply to tackle a wide variety of problems.

# **1** Introduction

Students of mathematics should have opportunities to experience how computational exploration enhances their understanding of mathematics. Although students may encounter computing in courses such as statistics, data science, or computer science, many of our students still graduate with poorly developed computational skills. In particular, mathematics majors do not necessarily see the role that computation plays in the exploration and discovery of ideas from what is often called "pure" mathematics. Beyond its traditional role in numerical analysis, computation should serve as a powerful lens through which students can explore, discover, and understand mathematical concepts.

Despite the profound role of computation (even before electronic computers) in the discovery of mathematical ideas and the growing importance of computational fluency in the modern world, many mathematics curricula present mathematics as a static body of knowledge, rather than a dynamic field of inquiry. The potential for computational tools to facilitate mathematical exploration and discovery is frequently overlooked, leaving students ill-equipped to leverage these resources effectively. While students might learn the mechanics of using a computer algebra system or a programming environment, they often lack the skills to design and execute meaningful computational experiments. They struggle to ask precise questions, translate mathematical questions into computational experiments, interpret computational output, and formulate mathematical conjectures based on their observations. The "how" of mathematical discovery—the process of exploration and experimentation—remains largely unexplored.

This paper presents a framework for equipping students with computational inquiry skills, with concrete examples from the course *Modern Computational Mathematics* at St. Olaf College. This course is designed to cultivate students' ability to ask mathematical questions, conduct computational experiments, and formulate precise conjectures. We highlight how specific computational explorations can reveal patterns, suggest potential truths, test conjectures, and make advanced mathematical concepts more accessible. By providing students with the tools and techniques for computational inquiry, we aim to empower them to become independent mathematical explorers and problem solvers, prepared to tackle the challenges of the modern world. This capability fosters a deeper understanding of mathematical principles, cultivates critical thinking, and provides students with the tools to tackle real-world challenges.

### **1.1 Course Context**

At St. Olaf College, the course Modern Computational Mathematics (MCM) is taken mostly by sophomore and junior math majors, some of whom have a second major such as computer science. The only prerequisite is a first course in linear algebra, which we teach early in the math major. The MCM course is one of three "transition courses" in the math major, which are courses that teach students to think like mathematicians in terms of proof and computation. The other transition courses are Real Analysis and Abstract Algebra, which we teach as introductory proofs courses. In contrast, MCM teaches students to think computationally, developing their skills of computational exploration and discovery. It is important to note that our MCM course is not focused on numerical analysis, computer science, or data science, yet we encourage students to also take such courses. Students are not expected to have prior programming experience, though students coming out of our prerequisite Linear Algebra course have some exposure to computations in Mathematica for solving linear systems and visualizing vectors.

Our Modern Computational Mathematics course prioritizes the "how" of mathematics over the "what," empowering students to formulate mathematical questions, conduct computational experiments, and develop precise conjectures. The course does not necessarily rely on specific mathematical content, but draws on many areas such as number theory, probability, graph theory, complex analysis, and abstract algebra. This is a unique course, though it shares some similarities with experimental mathematics courses taught at some institutions. For a more comprehensive view of the course, we recommend our forthcoming textbook, which will soon be published by the Mathematical Association of America. In the following pages we provide examples illustrating not only how we teach computational exploration, but also how computation provides access to advanced mathematical topics. Perhaps most importantly, we want students to develop their own agency to ask their own questions, conduct their own computational experiments, and to make their own discoveries in mathematics and beyond.

### **1.2 Questions and Conjectures**

For many of our students, Modern Computational Mathematics is the first course in which they are asked to come up with their own mathematical questions. Often, we introduce a mathematical setting and guide the students in some preliminary computations, producing number sequences or plots that reveal interesting patterns. We then ask the students "What do you observe?" and "What do you wonder?" Students discuss their observations in groups and write down a list of questions that occur to them.

This sort of activity leads us to discussion of what makes a good mathematical question. We like to highlight the following *characteristics of a good mathematical question* from Barry Mazur and William Stein [MS16]:

- 1. The question is interesting to you.
- 2. You don't already know the answer to the question.
- 3. You haven't already seen the question before, or at least not exactly.
- 4. You can begin to make computational investigations to shed light on the question.

These principles guide students to critical and creative inquiry that will result in productive computational exploration.

Furthermore, many students have never formulated their own mathematical conjecture prior to taking MCM. We guide students to choose a good mathematical question and then to design computational experiments to gain insight into their question. Based on their insight, students may be able to formulate a precise conjecture as a response to their question. Further computations may support the conjecture or perhaps provide a counterexample, leading them to revise the conjecture. Depending on their mathematical background, students may be able to try to prove their conjecture.

We give examples of four settings in which students can experience this process of computational exploration. These four settings involve very different mathematical topics, but they share a low bar of entry and many directions for exploration. Furthermore, computational investigation in these settings can introduce students to areas of advanced mathematics and open questions.

### 2 Generalized Fibonacci Sequences

The well-known Fibonacci sequence is the sequence of integers

where each number is the sum of the previous two. Formally, the Fibonacci sequence is usually defined recursively as:

$$F_0 = 0, \qquad F_1 = 1,$$
  
 $F_n = F_{n-2} + F_{n-1} \quad \text{for } n \ge 2$ 

Students can readily implement code to compute Fibonacci numbers, learning about iterative and recursive algorithms in the process.

With the ability to compute lists of Fibonacci numbers, students can search for identities among the Fibonacci numbers. Perhaps inspired by the algebraic identity  $(n + 1)(n - 1) = n^2 - 1$ , students can compute  $F_{n-1}F_{n+1}$  and  $F_n^2$  for various *n*. Even by hand, it is not too hard to compile the following results.

n	$F_{n-1}F_{n+1}$	$F_n^2$
1	0	1
2	2	1
3	3	4
4	10	9
5	24	25
6	65	64

It appears that  $F_{n-1}F_{n+1}$  and  $F_n^2$  always differ by 1. Further computations reveal that the

pattern continues; for example,

$$\begin{split} F_{99}F_{101} &= 125475243067621153271396401396356512255626, \\ F_{100}^2 &= 125475243067621153271396401396356512255625. \end{split}$$

With a bit of thought, students are led to conjecture that the Fibonacci numbers satisfy

$$F_{n+1}F_{n-1} - F_n^2 = (-1)^n$$

for all positive integers n. In fact this is true, and is known as Cassini's identity. Using mathematical induction to prove Cassini's identity is a good exercise for students with some exposure to proof by induction.

Cassini's identity is also a good starting point for introducing students to the idea of generalizing a known result. For example, we ask, "What if we replace the n + 1 and n - 1 with n + r and n - r?" Students embark on computations similar to those described above and discover Catalan's identity,

$$F_n^2 - F_{n+r}F_{n-r} = (-1)^{n-r}F_r^2.$$

A further generalization and computational exploration leads to Vajda's identity,

$$F_{n+i}F_{n+j} - F_nF_{n+i+j} = (-1)^n F_iF_j.$$

The Fibonacci sequence can itself be generalized in many ways. Changing the initial value leads to the Lucas sequence, defined by  $L_0 = 2$ ,  $L_1 = 1$ , and  $L_n = L_{n-1} + L_{n-2}$ , which provides more identities and interesting connections to the Fibonacci sequence. Furthermore, changing one of the recurrence coefficients leads to the Pell sequence, defined by  $P_0 = 0$ ,  $P_1 = 1$ , and  $P_n = 2P_{n-1} + P_{n-2}$ , opening the door to yet more identities. Exploring Fibonacci, Lucas, and Pell sequences requires a low cost of entry, yet students can easily rediscover many known results. Such identities can be readily found online, but for a comprehensive list we recommend the text by Koshy [Kos17].

Much more creative exploration results when we ask students to define their own generalization of the Fibonacci sequence. A good starting point is to ask students to choose their own initial values  $G_0 = a$  and  $G_1 = b$ , along with their own coefficients for the recurrence relation  $G_n = rG_{n-1} + sG_{n-2}$ . Then they explore patterns in their sequence and, using known Fibonacci identities as a guide, seek to formulate precise conjectures involving their own sequence.

For example, a student could choose  $G_0 = 1$ ,  $G_1 = 2$ , and  $G_n = 3G_{n-1} + 1G_{n-2}$ . This gives the sequence

By performing their own computations, students can readily observe that an analog of Cassini's identity holds for this sequence:

$$G_{n-1}G_{n+1} - G_n^2 = (-1)^{n+1}.$$

Students can then generalize this identity, perhaps obtaining a version of the Catalan identity such as

$$G_{n-i}G_{n+i} - G_n^2 = (-1)^{n+i}P_i^2,$$

where  $P_i$  is the Pell number with index *i*.

More fruitful generalizations arise by considering expressions of the form  $G_A G_B \pm G_C G_D$ , where *A*, *B*, *C*, and *D* are indexes often chosen to be related in some way. As one example that is within the reach of a sufficiently motivated student, the previously-defined sequence  $(G_n)$  satisfies

$$G_n G_m + G_{n+1} G_{m+1} = G_{n+m} + 2G_{n+m+1},$$

an identity that we discovered through our own computational exploration.

While generalized Fibonacci sequences are interesting in themselves, they lead to explorations that teach students broader principles, such as:

- How to generalize a known mathematical result.
- What is a mathematical identity and how to discover a nontrivial identity.
- How to state a precise mathematical conjecture.

Furthermore, many (generalized) Fibonacci identities have accessible proofs, often using mathematical induction, thus providing an excellent setting for students to practice proving results from their own conjectures.

## **3** Collatz Trajectories

The Collatz conjecture is a famous unsolved problem in mathematics. The problem is not only easy to state, but also provides fertile ground for exploring patterns, even while a solution the main problem remains out of reach.

The problem arises by applying the following function, which we call the *Collatz function*:

$$\operatorname{col}(n) = \begin{cases} 3n+1 & \text{if } n \text{ is odd,} \\ n/2 & \text{if } n \text{ is even.} \end{cases}$$

Start with any positive integer n. Repeatedly apply the Collatz function to obtain the sequence

$$n \to \operatorname{col}(n) \to \operatorname{col}(\operatorname{col}(n)) \to \operatorname{col}(\operatorname{col}(\operatorname{col}(n))) \to \cdots$$

For example, if the starting value is n = 5, then the sequence is

$$5, 16, 8, 4, 2, 1, 4, 2, 1, \ldots,$$

```
286
```

which we call the *Collatz trajectory* with starting value 5. If the starting value is n = 7, then the Collatz trajectory is

7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, ....

Observe that each of these Collatz trajectories reaches the value 1, after which the sequence repeats the cycle 4, 2, 1. The *Collatz conjecture* states that for any positive integer starting value, the Collatz trajectory eventually reaches 1. Despite extensive study and computational verification, the Collatz conjecture remains open [Guy04, Lag10].

While students can begin by computing Collatz trajectories, fascinating patterns emerge when considering properties of these trajectories. For example, let height(n) denote the maximum value of the Collatz trajectory starting with n. If students compute and plot (n, height(n)) for positive integers n up to 4000, they obtain a plot like that shown in Fig. 1, which contains many surprising patterns. This plot provides fertile ground for asking groups of students to state their own observations and formulate their own questions. We like to guide the students to create this plot and then simply ask, "What do you observe? What do you wonder?" Student questions often focus on the lines that are visible in the plot. Some ask specifically about the value of the unusually common height just below 10,000, or more generally about why some heights are more common than others.

Similarly, let stop(n) denote the number of iterations of the Collatz function required to first obtain 1 from the starting value n, which we refer to as the "stopping time" of n. Students can compute and plot (n, stop(n)) for positive integers n up to 4000, reproducing Fig. 2. Again, unexpected patterns emerge, prompting questions. For example, there appears to be a sort of clustering in Fig. 2. Why are there consecutive (or nearly consecutive) starting values with the same stopping times? What stopping times are most common? What values of n result in unusually large stopping times?



Figure 1: Heights of Collatz trajectories for n from 1 to 4000. (Heights greater than 40,000 are not shown.)


Figure 2: Stopping times of Collatz trajectories for *n* from 1 to 4000.

For many students, this exploration of Collatz trajectories is the first time that they have been asked to embark on a creative exploration and generate their own mathematical questions. This gives a good setting to emphasize the characteristics of good mathematical questions from in Section 1.2 and to let students design computational experiments to gain insight into their questions.

## 4 Primes and the Riemann Hypothesis

The concept of a prime number starts with simple arithmetic (i.e., multiplication of integers) but leads to fascinating patterns, deep mathematics, and modern applications. Recall that a positive integer  $n \ge 2$  is *prime* if and only if its only positive divisors are itself and one. While finding small primes is easy, determining whether a given large integer is prime can be a significant challenge. Implementing sieve algorithms for listing primes and primality tests for determining whether an integer is (probably) prime are good exercises for students in a computational math course. Additionally, the primes provide a plethora of patterns to explore, as well as an introduction to one of the most important open mathematical questions of our day—the Riemann hypothesis.

After students have access to code (either their own implementation or that provided by a programming environment) for creating lists of primes, we ask them to test several historical conjectures about primes.

Conjecture A. Every integer greater than 2 is the sum of two primes.

**Conjecture B.** For every *N*, the number of positive integers less than *N* with an *even* number of prime factors is less than or equal to the number of positive integers less than *N* with an *odd* number of prime factors. (For this, prime

factors are counted *with multiplicity*; e.g.,  $24 = 2^3 \cdot 3$  has 4 prime factors, while  $588 = 2^2 \cdot 3 \cdot 7^2$  has 5 prime factors.)

- **Conjecture C.** For every positive integer *n*, there exists at least one prime between  $n^2$  and  $(n + 1)^2$ .
- **Conjecture D.** All odd numbers greater than 1 are either prime, or can be expressed as the sum of a prime and twice a square.

Two of these conjectures are now known to be false, but two are still open questions. Students can readily explore these questions computationally, with the goal of finding computational evidence in support of the conjecture or possibly a counterexample. This not only gives students good practice designing computational experiments, but also helps students think about what makes a good mathematical conjecture.

Conjecture A above is the famous Goldbach conjecture, which remains open despite a massive amount of computational evidence. Conjecture C is known as Legendre's conjecture, and is also open. Conjecture D is also attributed to Goldbach, but has exactly two known counterexamples: the numbers 5777 and 5993, which are the only odd composite *Stern numbers* [Hod93].

Conjecture B, attributed to Pólya, is the most challenging of the four for students to investigate, not only because it has the most complicated statement of the four, but because its smallest counterexample is rather large. The conjecture fails at 906,150,257, a value not easy to find, but which can be found by students who think carefully about how to efficiently count prime factors of a list of integers.

Beyond testing conjectures, computational exploration can introduce students to the profound connections between the prime numbers and the Riemann hypothesis. This involves introducing the Riemann zeta function, which is defined for complex numbers *s* with Re(s) > 1 by

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$$

and extended to the complex plane via the technique of *analytic continuation*. We do not expect our students to grapple with analytic continuation, but they can use computational tools (such as those built into Mathematica or other programming environments) to create domain-coloring plots of  $\zeta(s)$ , such as that shown in Fig. 3.

After an introduction to domain coloring, we guide students to focus on the zeros of the Riemann zeta function. *Trivial zeros* occur at negative even integers. Other zeros of  $\zeta(s)$  are said to be *nontrivial*, and their locations are readily seen in a domain-coloring plot. The *Riemann hypothesis* says that all nontrivial zeros of  $\zeta(s)$  have real part exactly  $\frac{1}{2}$ .

The imaginary parts of the nontrivial zeros of  $\zeta(s)$  form an interesting sequence, which we call the *Riemann spectrum*. Letting  $\theta_i$  denote the imaginary part of the *i*th nontrivial zero



Figure 3: Domain coloring plot of the Riemann zeta function  $\zeta(s)$ .

above the real axis, we have the sequence

$$\theta_1 = 14.134725..., \ \theta_2 = 21.022040..., \ \theta_3 = 25.010858..., \ \theta_4 = 30.424876..., \ \dots$$

There are many connections between the Riemann zeta function and the primes, but the following exploration is very accessible to students and reveals an astonishing relationship between the Riemann spectrum and the sequence of prime numbers. First, students compute partial sums of the following function f(s), which is a sort of Fourier transform of the Riemann spectrum:



Figure 4: Plot of f(s) from Eq. (1), computed with 500 zeros of the Riemann zeta function. Prime powers are highlighted with dots on the horizontal axis.

$$f(s) = -\sum_{k=1}^{\infty} \cos(\ln(s)\theta_k).$$
(1)



Figure 5: Plot of  $F(\theta)$  from Eq. (2), computed with prime powers up to 200. The values of the Riemann spectrum are highlighted on the horizontal axis.

By computing and plotting partial sums of f(s), students observe "spikes," as in Fig. 4, which plots the partial sum of the first 500 terms. Examining the locations of these spikes, students observe that the tallest spikes occur at the primes. Furthermore, shorter spikes occur at integer powers of the primes. The function f(s), defined in terms of the Riemann spectrum, precisely identifies the locations of all primes and prime powers.

Next, students can compute partial sums of the following trigonometric series, which is a sum over the the prime powers:

$$F(\theta) = -\sum_{\text{prime powers } p^k} \frac{\ln(p)}{p^{k/2}} \cos\left(\theta \ln(p^k)\right).$$
(2)

Figure 5 shows a plot of  $F(\theta)$ , computed with prime powers up to 200. Again, spikes emerge as the sum is taken over more and more prime powers. This time, students observe that the spikes occur at the values of the Riemann spectrum.

The trigonometric sequences in Eqs. (1) and (2) reveal a surprising duality between the sequences of primes and the Riemann spectrum. Knowledge of the primes determines the values of the Riemann spectrum, and vice-versa. The duality between these two sequences can be made precise using the language of Fourier transforms. The text by Mazur and Stein [MS16] explains in far more detail how this process utilizing Fourier transforms relates the prime counting function and the Riemann spectrum.

Computational explorations involving the Riemann zeta function and its zeros helps students to understand the statement of the Riemann hypothesis. From this, instructors can highlight how the notion of certainty in mathematics differs from that in other scientific disciplines. Nontrivial zeros of the Riemann zeta function have been computed up to a height of 10 trillion above the real axis, and every zero has been found to have real part exactly  $\frac{1}{2}$ . This is an enormous amount of evidence in support of the hypothesis. In other areas of science, this amount of evidence would lead to a scientific law, but in mathematics the Riemann hypothesis remains a conjecture until a proof is found.

## 5 Percolation Theory

For our last example, we turn to probabilistic simulation. Computational skills allow students to explore a vast assortment of questions of the form "What is the probability that...?" and "On average, how long does it take to...?" For these questions, it is often difficult or impossible to find an exact (i.e., closed-form) answer, but estimates resulting from simulation may be within reach.

One of our favorite settings for probabilistic simulations is known as *percolation theory*. This setting involves geometric models that are easy to describe, lead to a depth-first search algorithm, and reveal a surprising phase transition.



Figure 6: Two examples of grids of open (unshaded) and closed (shaded) squares. The left grid has no percolation path of open squares from top to bottom, while the right grid has such a path.

The setting is an  $n \times n$  grid of squares, as illustrated in Fig. 6. Suppose that each individual square is either *open*, meaning that it permits water to flow through it, or *closed*, meaning that water cannot flow through it. In Fig. 6, open squares appear white, and closed squares are shaded gray. Each square is independently chosen to be open with probability p or closed with probability 1 - p. Now suppose that water is poured on the top edge of the grid and flows through any open square to all of its neighbors (left, right, above, and below). What is the probability that there exists a "percolation path" for water to flow from top to bottom, and how does this probability depend on n and p?

The problem is simple to state, but implementing a simulation requires some thought about data structures and algorithms. First, students must decide how to represent the grid in memory and how to generate a grid where each square is open with probability p. Formulating an algorithm to search for a percolation path is a challenge for many students, but instructors can guide students to consider a recursive algorithm. For students who lack experience with recursive algorithms, this can be scaffolded with pseudocode and simpler cases, such as first searching for paths that proceed only from a square to its neighbor below, before generalizing to search all four neighbors. In order to avoid infinite recursion, students must implement a way of tracking previously searched locations.



Figure 7: For large grid sizes n, the probability of percolation changes from nearly 0 to nearly 1 at the percolation threshold.

With a working algorithm that detects the presence of a percolation path from the top to the bottom of any particular grid, students can estimate the probability of percolation for any particular grid size n and open-square probability p. Students observe that when p is large, the probability of percolation is near 1, and when p is small, the probability of percolation is near 0. However, when students plot the probability of percolation across a range of values of p, an unexpected phenomenon occurs. For large grid sizes, the probability of percolation changes abruptly from nearly 0 to nearly 1 as p increases through a small interval near 0.6. Further investigation reveals that the larger the grid size, the sharper this transition, which students can observe by creating a plot such as Fig. 7. For very large grids, the probability of percolation changes from 0 to 1 at approximately 0.595. Students thus discover a *phase transition* that occurs at a value known as the *percolation threshold*.

This model of percolation theory can be generalized to many other lattices. One simple extension is to allow water to flow between open squares that share only a common vertex and not a common edge. Beyond this, any two-dimensional lattice can be the basis for a percolation model. For each model, there exists some threshold value at which percolation occurs. Students can readily compute these thresholds to several decimal digits. Remarkably, for all but a few models, closed-form expressions for the percolation thresholds are unknown.

## 6 Conclusion

Math students, and especially math majors, should have opportunities to engage in computational exploration. By providing students with opportunities to formulate questions, design experiments, and analyze results, computational skills empower students to engage with mathematics as a dynamic and creative discipline. The examples presented in this paper, ranging from generalized Fibonacci sequences to the zeros of the Riemann zeta function and the stochastic nature of percolation theory, illustrate how computation can illuminate abstract concepts, make advanced mathematics more accessible, and foster curiosity. All of the explorations described in this paper have been successfully employed with students in the Modern Computational Mathematics course at St. Olaf College. Further details, examples, and explorations can be found in our forthcoming textbook, to appear soon in the Mathematical Association of America textbook series. We hope that faculty will consider integrating computational inquiry into their own courses, not as a replacement for traditional methods, but as a powerful complement.

The incorporation of computational exploration into math education offers wide-ranging benefits. Beyond helping students understand the particular mathematical topics at hand, the skills of computational exploration cultivate critical thinking, problem-solving abilities, and the capacity for independent discovery—qualities that are invaluable in a rapidly evolving, data-driven world. Perhaps most importantly, equipping students with these skills allows them to experience the joy of mathematical discovery firsthand. By empowering students to ask their own questions, test their own conjectures, and witness the beauty of mathematical patterns revealed through computation, students gain appreciation of mathematics as a creative human activity.

## References

- [Guy04] R. Guy. *Unsolved Problems in Number Theory*. Problem Books in Mathematics. Springer New York, 2004.
- [Hod93] Laurent Hodges. A lesser-known goldbach conjecture. *Mathematics Magazine*, 66(1):45–47, 1993.
- [Kos17] Thomas Koshy. *Fibonacci and Lucas Numbers with Applications*, volume 1. Wiley, New York, 2017.
- [Lag10] Jeffrey C. Lagarias, editor. *The Ultimate Challenge: The* 3*x*+1*Problem*. American Mathematical Society, 2010.
- [MS16] Barry Mazur and William Stein. *Prime Numbers and the Riemann Hypothesis*. Cambridge University Press, 1st edition, 2016.

# REDUCING BIAS IN AI: ETHICAL AND TECHNICAL APPROACHES

**Cheyenne Douglas** 

Department of Math and Computer Science

Minot State University

Minot, ND 58701

cheyenne.douglas@ndus.edu

# Abstract:

The rapid expansion of Artificial Intelligence (AI) has raised ethical concerns in industries such as social media, healthcare, and corporations. A critical issue is mitigating bias in AI, as human-made algorithms inevitably reflect societal biases. Bias often stems from data collection, where historical inequalities shape machine learning models, disproportionately affecting marginalized groups. Recognizing these biases is essential for refining AI systems to promote fairness.

This paper explores strategies for reducing AI bias, focusing on individual input, business intent, and dataset distribution. Recent research, such as the **Data Debiasing with Datamodels (D3M)** framework, demonstrates how removing harmful training samples improves model robustness (Jain et al., 2024). Additionally, subgroup fairness methods address the worst-group performance through data rebalancing, enhancing AI reliability. The **NIST proposal on AI bias** emphasizes industry standards and best practices, highlighting

the importance of identifying and managing bias at the pre-design, development, and deployment stages (Schwartz et al., 2022).

There is no one-size-fits-all solution to mitigating AI bias. However, recognizing and addressing these biases, targeted interventions in data selection, and algorithmic development can lead to fairer, more inclusive models. Companies can take meaningful steps toward ethical AI deployment by refining data collection and implementing bias detection strategies.

Cheyenne Douglas

Department of Math and Computer Science

Minot State University

Minot, ND 58701

cheyenne.douglas@ndus.edu

# **1. Introduction**

## 1.1 Brief Overview of Artificial Intelligence (AI) and Its Rapid Expansion

Artificial intelligence (AI) is rapidly transforming various industries, revolutionizing how businesses operate and how individuals interact with technology. *Over 50% of organizations have incorporated AI into their business models, with its use projected to grow by 26% annually* (Balderson, 2025). From human resources and accounting to IT and machine learning, AI is reshaping workplaces and decision-making processes.

A Pew Research survey of 11,000 people found that 55% of respondents interact with AI at least once a day, and 27% encounter it multiple times daily (Balderson, 2025). Despite this rapid expansion, there is a growing shortage of AI experts capable of overseeing its implementation, raising concerns about unintended consequences such as bias, privacy risks, and accountability (Schwartz et al., 2022).

As AI becomes increasingly embedded in our daily lives, it is crucial to examine its origins, datasets, and ethical implications. This paper explores the ethical concerns associated with AI, particularly in social media, healthcare, and corporate decision-making, where issues of bias, fairness, and transparency continue to shape their impact (Schwartz et al., 2022).



Figure 1: *Projected Economic Growth of AI Usage* Adapted from *Explainable AI market size, share & trends analysis report*, by Grand View Research, 2023 (<u>https://www.grandviewresearch.com/industry-analysis/explainable-ai-market-report</u>). © 2023 Grand View Research.

# **1.2 Introduction to Ethical Concerns Related to AI, Particularly in Social Media, Healthcare, and Corporations**

One of the most well-documented cases of AI bias occurred at Amazon, where the company implemented an AI-driven hiring system to evaluate job applicants. The system was trained on past hiring data, which primarily favored male candidates due to historical hiring trends in the tech industry. As a result, the AI sanctioned resumes containing words associated with women, such as *"women's chess club"* or *"women's college"* (Holdsworth, 2023). Amazon ultimately abandoned the AI tool after recognizing its inherent bias, but this case shows a broader ethical issue—AI can inherit and reinforce systemic discrimination when trained on biased datasets (Holdsworth, 2023).

Beyond Amazon, other companies have faced similar challenges with biased AI algorithms, particularly in facial recognition software used for hiring, security screenings, and the criminal justice system. Studies have shown that these systems misidentify individuals of color at significantly higher rates than white individuals, leading to wrongful rejections in job applications, discriminatory hiring practices, and unfair treatment in law enforcement (Srinivasan & Chander, 2021).

A 2012 study tested facial recognition algorithms on criminal mugshots from Pinellas County, Florida, comparing their accuracy across police departments in California, Maryland, and Pennsylvania. A senior FBI technology specialist who co-authored the study found that all three algorithms performed 5 to 10 percent worse when identifying African American individuals, sometimes failing to recognize them at all (Cavazos, 2025).

This raises serious ethical concerns regarding transparency and accountability. Many organizations using AI-driven hiring or law enforcement tools do not disclose how their algorithms function, making it difficult to detect and correct biases (Schwartz et al., 2022). Without proper oversight, these technologies risk reinforcing systemic discrimination and worsening existing racial disparities in employment and criminal justice (Srinivasan & Chander, 2021).

## 1.3 AI in Social Media: Misinformation, Privacy, and Bias

Social media platforms such as Facebook, Instagram, Snapchat, and Twitter rely heavily on AI for content moderation, targeted advertising, and data analysis. While AI enhances user experience, it also raises ethical concerns in several areas:

#### **1.3.1 Misinformation and Echo Chambers**

• AI-driven recommendation algorithms prioritize engagement over accuracy, often amplifying fake news or sensational content (Schwartz et al., 2022). This is particularly concerning because if accurate information is spread, it could lead to public panic. This was evident in the Cambridge Analytica scandal, where AI-based microtargeting was used to manipulate political opinions by tailoring misleading content to specific users (Schwartz et al., 2022).

#### **1.3.2 Privacy and Data Exploitation**

• AI gathers vast amounts of personal data, analyzing user behavior, location, and interactions for targeted ads (Schwartz et al., 2022).

• This raises concerns about user consent and data security, as many social media platforms have faced lawsuits for selling user data without explicit permission (Cavazos, 2025).

#### **1.3.3 Bias in AI Content Moderation**

- AI-powered moderation tools often fail to detect hate speech or wrongfully censor certain communities (Holdsworth, 2023).
- Studies have shown that automated moderation disproportionately flags content from marginalized groups, leading to unjust account suspensions (Holdsworth, 2023).

The use of AI in spreading misinformation is deeply concerning, as it can lead to misleading content about individuals, incites public panic, and distorts public perception. Most social media users entrust these platforms with their personal information, often without considering the implications of that data being shared or sold. While AI is beneficial for creating personalized advertising experiences, it's critical to ensure that sensitive user data does not contribute to biased or unethical AI outcomes. Since AI models learn from the data they are given, social media companies must use caution when implementing AI for content moderation and censorship. As more platforms integrate AI into their systems, it becomes increasingly important to curb the spread of misinformation, safeguard user privacy, and apply censorship responsibly. Doing so will help build public trust in these platforms—trust that, in turn, can strengthen and grow the companies themselves.

## 1.4 AI Bias in Healthcare: Unequal Diagnoses and Ethical Risks

The healthcare industry has increasingly adopted AI to assist with disease diagnosis, patient care, and workflow optimization. Deep-learning models have been used since the 2000s, improving efficiency and medical predictions (Norori et al., 2021). However, bias in healthcare AI has life-or-death consequences, particularly for underrepresented groups (Norori et al., 2021).

#### 1.4.1 Racial and Gender Bias in Medical AI

- Research from IBM and other studies have shown that computer-aided diagnosis (CAD) systems are less accurate for Black patients than white patients (Norori et al., 2021).
- Women and minority groups are often underrepresented in medical datasets, leading to misdiagnosis and inappropriate treatments (Norori et al., 2021).

### 1.4.2 Real-World Consequences of AI Bias

- AI misdiagnoses can lead to delayed treatment, incorrect medication, or even denial of life-saving procedures (Norori et al., 2021).
- Some predictive AI models in hospitals allocate fewer resources to Black patients, assuming they need less care based on flawed data (Norori et al., 2021).

## 1.4.3 Efforts to Reduce Bias in Healthcare AI

Companies like EXAI are working to eliminate bias by developing AI systems that generate information solely from data analysis, without external human input that could introduce prejudice (Norori et al., 2021).

Ethical AI development includes:

- Using diverse and representative medical datasets (Norori et al., 2021).
- Employing participatory science—involving diverse communities in AI development (Norori et al., 2021).
- Open science initiatives, such as code sharing and data transparency, as discussed by Natalia Norori (Norori et al., 2021).

As AI becomes more integrated into industries such as healthcare, it is crucial to recognize and address potential biases within the datasets used to train these systems. Two of the most significant sources of bias in healthcare are related to gender and race. As AI-driven technologies continue to produce biased outcomes, we must acknowledge the historical inequalities found within this data. Companies have a responsibility to mitigate these biases by ensuring that AI is used ethically and primarily for data analysis—not decision-making—when fairness cannot be guaranteed. Healthcare is one of the most critical sectors in society, and ensuring equal treatment for all patients must be a top priority. Doing so will not only reduce errors but also ultimately save lives.

# **1.5 The Need for AI Ethics, Regulations, and Accountability**

Bias in AI is not confined to a single sector; it is a cross-industry concern rooted in how algorithms are trained and applied. In healthcare, underrepresentation of women and minority groups in medical datasets has led to misdiagnoses and unequal treatment (Norori et al., 2021). In social media, AI-powered moderation tools disproportionately censor marginalized communities and spread misinformation due to biased training data and opaque algorithmic processes (Holdsworth, 2023; Schwartz et al., 2022). Similarly, in business and corporate hiring, AI tools trained on historical hiring patterns have discriminated against women and minorities, reinforcing long-standing workplace inequities (Cavazos, 2025; Holdsworth, 2023). These examples reveal a shared challenge: AI systems trained on flawed or incomplete datasets are likely to replicate and even worsen systemic discrimination across sectors. Therefore, a unified ethical approach—grounded in transparency, inclusivity, and accountability—is essential to mitigate bias and ensure equitable AI outcomes industry-wide.

Given the growing ethical concerns in AI-driven industries, regulatory standards and transparency measures are crucial. Several approaches can help mitigate bias and promote ethical AI use:

- Implementing fairness audits to detect and correct biases in AI systems before deployment (Schwartz et al., 2022).
- Requiring transparency in AI decision-making so that companies disclose how algorithms function (Holdsworth, 2023).
- Developing policies and regulations that hold organizations accountable for AI-driven discrimination, particularly in hiring, social media, and healthcare (Cavazos, 2025).

The recurring theme across healthcare, social media, and business is clear: AI systems are not inherently neutral—they reflect the values, assumptions, and inequalities embedded in the data and design choices made by humans. Whether denying patients equitable care, amplifying misinformation, or reinforcing discriminatory hiring practices, these examples show how unchecked AI can deepen social divides rather than bridge them. These real-world consequences underscore the urgent need for ethical frameworks, transparency, and accountability at every stage of AI development. Without systemic changes, the risks of biased outcomes will continue to rise alongside AI's influence.

#### **Thesis Statement**

This paper argues that addressing AI bias requires integrated ethical and technical frameworks at every stage of AI development, going beyond existing mitigation methods to proactively prevent discriminatory outcomes.

## 2. Understanding Bias in AI

## 2.1 Definition of AI Bias

James Holdsworth describes **AI bias** as the occurrence of **biased results due to human biases embedded in training data or AI algorithms**, which can lead to **harmful or unfair outcomes** (Holdsworth, 2023). AI bias arises when **machine learning models inherit prejudices from historical data** or when **biases occur in how data is collected, labeled, or processed** (Schwartz et al., 2022). Understanding the types of biases that AI models can contain is crucial for identifying, addressing, and developing effective solutions to mitigate biased outcomes. In this paper we will specifically discuss the following types of biases:



Figure 2: Visual explaining key concepts of bias and fairness in AI systems. From Bias and Fairness in AI Systems: Basics, by Bias and Fairness in AI Systems, n.d. (<u>https://bias-and-fairness-in-ai-systems.de/en/basics/</u>). © Bias and Fairness in AI Systems.

## 2.2 Types of AI Bias

- **2.2.1 Sampling Bias** Occurs when datasets are created by over-representing or under-representing certain groups, leading to skewed outcomes. This makes the AI model less accurate for marginalized populations (Jain et al., 2024).
  - An example of sampling bias in facial recognition software occurs when the AI model underperforms on individuals with darker skin tones because it was primarily trained on lighter-skinned faces. This lack of representation in the training data results in a model that does not accurately reflect the diverse population it is intended to serve.
- **2.2.2 Label Bias** Happens when inconsistencies in the labeling process cause AI models to misinterpret the same objects differently. This can introduce subjective human bias into AI decision-making (Srinivasan & Chander, 2021).
  - An example of labeling bias arises when datasets reflect the personal or cultural perspectives of annotators—for instance, misinterpreting slang or sarcasm. This can lead AI systems to incorrectly label content, such as tweets, as toxic, thereby reinforcing biased stereotypes and marginalizing certain linguistic communities.
- **2.2.3 Data-Creation Bias** Similar to sampling bias, this occurs during the datasetcreation process, leading to AI models learning from incomplete or unbalanced data (Norori et al., 2021).
  - An example of data creation bias occurred when Twitter first deployed an interactive AI chatbot and users fed it biased language targeting certain individuals. As a result, the AI internalized those biased inputs and began producing discriminatory responses, particularly against minority groups.
- 2.2.4 Negative Set Bias Arises when datasets lack enough diverse samples. Businesses often use AI to maximize profit, but if the AI is trained on biased data, it may make unfair predictions and decisions, even if not intentionally discriminatory (Cavazos, 2025).
  - An AI trained to identify spam emails may incorrectly classify legitimate emails as spam if it was only trained on spam examples from a single domain. Without exposure to a diverse range of email sources, the model lacks the

context needed to accurately distinguish between spam and non-spam messages across different domains.

- **2.2.5 Human Evaluation Bias** Occurs when humans are responsible for validating an AI model's performance. If human evaluators have preconceived biases, these can influence AI behavior, reinforcing pre-existing stereotypes or unfair practices (Schwartz et al., 2022).
  - Human evaluation bias can emerge in chatbot development when evaluators consistently rate responses from female personas as less authoritative. This reinforces gender stereotypes in the training process, leading the AI to replicate and amplify those biases in future interactions.
- **2.2.6 Automation Bias** Happens when users place too much trust in AI outputs, assuming the AI is always correct. This can lead to overlooking errors or ignoring alternative perspectives, especially in high-stakes environments (Schwartz et al., 2022).
  - An example of automation bias can happen by users such as doctors relying on an AI diagnosis and accepting that without second guessing, which can lead to incorrect treatment to patients.
- **2.2.7 Representation Bias** Occurs when certain groups are entirely missing or significantly underrepresented in the training data. This leads to AI systems performing poorly or inaccurately for those groups, often impacting facial recognition, voice recognition, and healthcare AI tools (Norori et al., 2021).
  - An example of representation bias is when organizations use AI models for job application by using data from tech companies, where most past hires were men. The AI may learn to favor male applicants over equally qualifies women or minorities due to the dataset's underrepresentation.
- **2.2.8 Measurement Bias** Arises when the features used in the dataset do not accurately capture what they are intended to measure. For example, using ZIP codes as a stand-in for income or race can distort outcomes and lead to inaccurate predictions (Srinivasan & Chander, 2021).
  - An example of measurement bias occurs when law enforcement uses arrest data—often collected from heavily policed areas—as the basis for AI training. This can lead the AI to unfairly associate certain demographics with higher crime rates, reinforcing existing patterns of over-policing and discrimination.

Understanding these different types of biases is important for recognizing how they appear in AI models and datasets. By becoming familiar with the sources and effects of each bias, developers, researchers, and policymakers can more effectively detect harmful patterns, assess the fairness of AI outputs, and implement strategies to reduce or eliminate bias. This foundational knowledge enables us to build more ethical, inclusive, and trustworthy AI systems that serve all users equitably.

## 2.3 How Bias is Introduced Through Human-Made Algorithms

AI models do not create bias on their own—they learn it from the human-made algorithms and historical data used to train them. These systems reflect the values, decisions, and patterns embedded in their training environments, meaning any existing societal inequalities or prejudices can be absorbed and reproduced by AI. When AI is trained on historically biased data—such as records that underrepresent certain racial, gender, or socioeconomic groups—it often perpetuates those same disparities in its outputs. This can result in discriminatory outcomes for marginalized communities, reinforcing systemic injustice in areas like hiring, healthcare, education, and law enforcement. As Schwartz et al. (2022) note, these issues highlight the urgent need for critical oversight in AI development to ensure fairness, accountability, and ethical responsibility.

The Wall Street Journal reports that as AI adoption grows, corporations and organizations struggle to address biases within their models. If left uncorrected, AI bias can result in unfair outcomes in hiring, criminal justice, financial decisions, and healthcare (Schwartz et al., 2022). This is especially concerning because automated systems can reinforce systemic discrimination at a much larger scale and speed than human decision-making alone (Holdsworth, 2023).

### 2.4 Role of Historical Inequalities in Shaping Machine Learning Models

Training AI models requires large datasets to make accurate predictions and informed decisions. However, if these datasets reflect historical discrimination—such as biased hiring practices, unequal access to healthcare, or racial disparities in policing—the AI systems will inevitably learn and reproduce these inequities. Instead of offering neutral or objective insights, the AI may unintentionally reinforce the very patterns of injustice it was meant to help eliminate. For instance, an AI used in hiring could favor candidates from majority groups if past hiring data was skewed against minorities, or a medical AI could underdiagnose certain populations due to underrepresentation in healthcare data. As Norori et

al. (2021) emphasize, without intentional efforts to diversify and audit training data, AI systems risk amplifying systemic biases and widening existing social gaps rather than closing them.

The Gender Shades facial recognition evaluation project describes the poor performance of facial recognition systems when trying to detect face types (by gender and skin type) that are not present in the training data. This is an example of representation bias—a type of sampling bias that pre-dates AI—where trends estimated for one population are inappropriately generalized to data collected from another population (Norori et al., 2021). Ensuring that all ethnic groups are adequately represented in AI models, especially those involving facial recognition, helps promote fairness in outcomes and transparency for all users.

Historically, medical research has been skewed towards white, male populations, which leads AI models to fail to account for genetic, environmental, and socioeconomic differences in minority groups. Bias in AI models for the healthcare industry often stem from representational bias. When gender and racial groups are underrepresented in AI models, it can lead to misdiagnoses, unequal treatment, and inadequate medical care for those populations (Norori et al., 2021).

# **2.5 Impact of Bias on Marginalized Groups and the Broader Implications for Society**

While AI is often viewed as a tool for progress and efficiency, unchecked bias can have far-reaching consequences. If AI continues to produce discriminatory outcomes, public confidence in AI-driven decision-making will decline, making it harder for industries to implement AI ethically (Schwartz et al., 2022).

- Marginalized communities will face greater disadvantages if AI reinforces bias in employment, healthcare, social media, and finance (Cavazos, 2025).
- The legal and ethical dilemmas from AI bias could lead to lawsuits and ethical scrutiny for organizations (Holdsworth, 2023).
- If AI systems are perceived as biased and unethical, it could slow down innovation and adoption of AI and machine learning models (Srinivasan & Chander, 2021).

AI has the power to transform society, but if left unchecked, it will continue to harm marginalized groups and reinforce systemic discrimination. This will set back the advancement of artificial intelligence and hinder its full potential in society. Correcting biases in AI model datasets now is essential to prevent potentially catastrophic outcomes in the future. To ensure AI serves as a tool for fairness and inclusivity, businesses,

policymakers, researchers, and developers must take proactive steps toward ethical AI development (Norori et al., 2021).

## 3. Strategies for Reducing AI Bias

## **3.1 Individual Input and Human Oversight**

AI systems are increasingly integrated into decision-making processes across various sectors; however, concerns about bias and fairness remain significant. While Machine Learning (ML) models can recognize complex patterns, aligning their output with societal values continues to be a major challenge (Norori et al., 2021). Bias in AI does not solely stem from training data—it is also deeply rooted in institutional policies and human decisions that influence how AI systems are developed, deployed, and applied (Schwartz et al., 2022). Although human oversight is essential for mitigating biases and preventing incorrect outcomes, it is equally important to recognize and address the biases that human actors may unintentionally embed in these systems. We explore multiple strategies for identifying and reducing bias in AI, and I argue that a combination of technical, ethical, and policy-based approaches is necessary to create more equitable and trustworthy AI systems.

#### 3.1.1 Techno-Solutionism and Its Limits in Addressing Bias

A major issue in AI governance is techno-solutionism, the belief that bias can be solved purely through better algorithms or more data (Srinivasan & Chander, 2021). This perspective overlooks the importance of human judgment, ethical considerations, and organizational structures. AI does not operate independently of its environment, they interact with human users, decision-makers, and institutional policies, all of which influence how bias manifests and whether it is corrected.

One way to offset techno-solutionism is through model risk management, which requires organizations to establish, document, and review the real-world impact of AI

(Holdsworth, 2023). However, such measures are insufficient if they fail to incorporate human oversight in a meaningful way. Bias is not just a technical flaw but an ingrained issue that requires human intervention to identify, interpret, and mitigate.

### 3.1.2 Why Human Judgment is Essential in AI Bias Mitigation

Bias in AI is deeply linked to intuitive judgements and decision-making processes. Human biases are often implicit, meaning that simply raising awareness of bias does not ensure control over it. Instead, organizations must take deliberate steps to integrate human judgment in AI systems by:

- **Ensuring diverse human oversight:** AI models should not be left to make highstakes decisions without human review. A human-in-the-loop (HITL) approach allows for critical evaluation of AI outputs, reducing the risk of automated bias reinforcement (Schwartz et al., 2022).
- **Encouraging participatory design techniques:** AI systems should be developed with input from diverse stakeholders to reflect a broad range of perspectives and avoid embedding the biases of a select group (Norori et al., 2021).
- **Developing institutional frameworks for human accountability:** Human decisionmakers should be responsible for interpreting AI-generated recommendations, rather than blindly trusting or implementing them (Holdsworth, 2023).
- **Recognizing when AI should not be used:** Bias is present not just in model development but in the assumptions about where AI is applied. Human judgment is essential in determining whether AI is appropriate for a given task (Cavazos, 2025).

Bias in AI is neither new nor unique, nor can it be eliminated. However, without human judgment, even the most advanced AI models will fail to recognize the ethical, cultural, and contextual dimensions of fairness (Schwartz et al., 2022). The challenge is not just to improve AI but to ensure that human oversight remains central in AI governance and decision-making.

#### 3.1.3 Strategies for Increasing Awareness of Biases During AI Development

Increasing awareness of AI biases requires a multi-pronged approach that integrates education, socio-technical perspectives, transparency, bias testing, and ongoing human oversight. By implementing these strategies, AI developers can proactively identify and mitigate biases, resulting in more ethical, inclusive, and trustworthy systems (Jain et al., 2024). While artificial intelligence has the potential to drive remarkable advancements, it is not always reliable. Without diverse perspectives and meaningful human intervention, AI systems may rely solely on data-driven decisions that overlook ethical, contextual, or emotional considerations. Human oversight remains essential to ensure that AI reinforces responsible and equitable decision-making.

Involve a Human-in-the-Loop (HITL) for Bias Mitigation:

To ensure ethical and impartial AI implementation, it is essential that AI-generated decisions are reviewed by human experts before being put into practice (Schwartz et al., 2022). Involving human oversight helps catch potential errors and biases that automated systems may overlook. Additionally, collaborative decision-making should be adopted, allowing impacted communities to provide feedback on AI models prior to deployment (Norori et al., 2021). This approach ensures that diverse perspectives are considered and helps prevent the reinforcement of systemic inequalities. Furthermore, integrating mechanisms that allow users to report biased AI outcomes promotes ongoing accountability and enables continuous refinement of AI systems based on real-world experiences (Srinivasan & Chander, 2021).

Promote a Socio-Technical Approach to AI:

Addressing AI bias requires moving beyond techno-solutionism and acknowledging that bias is not merely a technical flaw, but a complex issue rooted in human behavior, societal structures, and institutional practices (Cavazos, 2025). To build more equitable AI systems, developers must adopt collaborative design techniques that involve diverse stakeholders, especially marginalized communities who are most affected by AI-driven decisions (Jain et al., 2024). Furthermore, performing comprehensive impact assessments prior to deployment is crucial for identifying potential societal implications of AI systems and ensuring they reflect ethical and inclusive principles.

Integrate Bias Education into AI Development Teams:

Promoting ethical AI development begins with education and collaboration. Regular training sessions on implicit bias, systemic discrimination, and responsible AI design can help developers recognize the deeper social implications of the systems they build (Srinivasan & Chander, 2021). Incorporating real-world case studies—such as biased hiring algorithms or racially discriminatory facial recognition tools—provides valuable insight into how AI bias manifests in practice and the harm it can cause (Holdsworth, 2023). Additionally, fostering interdisciplinary collaboration between AI developers, ethicists, sociologists, and domain experts is essential to broaden perspectives and ensure that diverse viewpoints inform AI design and deployment (Norori et al., 2021). Human judgment is essential for identifying, mitigating, and preventing AI bias. While AI models can process vast amounts of data, they lack the contextual understanding and ethical reasoning needed to make fair and socially responsible decisions. By adopting a socio-technical approach, involving diverse users, and promoting transparency and accountability, businesses and researchers can develop AI systems that are more ethical, inclusive, and aligned with societal values (Schwartz et al., 2022).

Moving forward, AI governance should prioritize human oversight and bias mitigation strategies, ensuring that AI does not amplify systemic inequalities but instead works toward equitable and fair decision-making processes. This will allow us to further explore the possibilities of AI without the influence of bias that could set us back to times when systemic inequalities were most prevalent.

## 3.2 Business Intent and Ethical Responsibility

AI-driven technologies have the potential to transform industries, but ethical concerns surrounding bias, transparency, and inclusivity highlight the need for responsible AI governance. Companies must align AI development with corporate values and ethical principles to ensure that AI systems serve diverse populations fairly and do not perpetuate discrimination or bias (Schwartz et al., 2022). This section explores the role of corporate values in AI design, highlights companies implementing ethical AI practices, and discusses the importance of aligning AI objectives with fairness and inclusivity when ethical considerations are appropriate.

#### 3.2.1 The Role of Corporate Values and Ethical Considerations in AI Design

Corporate values play a crucial role in shaping AI development, influencing how companies approach fairness, transparency, and accountability in their AI models. Ethical AI is not only a social responsibility but also a business imperative—organizations that fail to address AI bias risk losing consumer trust, facing regulatory scrutiny, and perpetuating social inequalities (Holdsworth, 2023).

Key ethical considerations in AI design include:

- Fairness: AI systems must be designed to avoid discrimination against marginalized groups and mitigate biases in data and decision-making processes (Norori et al., 2021). This builds consumer trust in corporations that use AI models and ultimately reinforces the company's values in the eyes of their users.
- Transparency: Companies should make AI decision-making interpretable and explainable to users, regulators, and stakeholders (Schwartz et al., 2022). Allowing this transparency will enable users to better understand how their data is being used, and how AI models conduct their reasonings.
- Accountability: Businesses must establish clear oversight structures to ensure that AI decisions do not reinforce systemic biases (Srinivasan & Chander, 2021). When

corporations uphold strong accountability standards, it shows consumers that there is a commitment to maintaining the company's values, fostering trust, and ensuring responsible use of AI technologies.

- Inclusivity: AI should be developed with input from a diverse range of stakeholders, including individuals from different cultural, racial, gender, socioeconomic, and geographic backgrounds. Involving these voices in the design, testing, and implementation of AI systems helps ensure that the models reflect the needs and experiences of a broad user base. Without inclusive representation, AI technologies risk reinforcing existing inequalities or overlooking the unique challenges faced by underrepresented communities. By prioritizing inclusivity, developers can build AI systems that serve global populations more equitably, reducing bias and promoting fairness in real-world applications (Jain et al., 2024).
- Regulatory Compliance: Implementing regulations for AI models within businesses will help promote transparency between corporations and stakeholders, ultimately fostering greater trust and accountability. With the rise of AI regulations, businesses must align their AI models with legal and ethical standards, such as bias mitigation frameworks and data transparency laws (Cavazos, 2025).

Despite growing awareness of the importance of ethical AI, many companies face significant challenges in putting these principles into practice. One major obstacle is the lack of technical expertise and organizational infrastructure to evaluate and mitigate AI bias effectively, especially for small- and medium-sized enterprises with limited resources (Schwartz et al., 2022). Additionally, balancing fairness with business efficiency can create tension, as bias mitigation techniques may reduce model accuracy or increase development time and cost (Jain et al., 2024). Companies may also struggle with internal resistance to transparency, especially in competitive industries where AI decision-making processes are seen as proprietary. Moreover, without standardized ethical guidelines or mandatory regulations, businesses often adopt a reactive rather than proactive stance, implementing ethical safeguards only after public backlash or legal scrutiny. These challenges reveal the importance of institutional support, cross-disciplinary collaboration, and long-term investment in building ethical AI systems that are not only effective but also socially responsible.

Many AI failures stem from techno-solutionism—the flawed assumption that AI alone can solve complex social problems without human intervention or oversight. In contrast, a socio-technical approach that incorporates human judgment, institutional accountability, and fairness considerations is essential for ethical AI development. This approach empowers us to take an active role in shaping the future of AI while addressing ethical concerns and ensuring that human values remain central to its evolution.

## 4. Industry Standards and Best Practices

#### 4.1 Discussion of the NIST proposal on AI bias

The NIST proposal, *Towards a Standard for Identifying and Managing Bias in Artificial Intelligence* (Schwartz et al., 2022), critically examines the real-world implications of widespread AI adoption. A key insight from the authors is the importance of evaluating the social impact that AI applications have when deployed. The proposal emphasizes the necessity of considering broader societal effects and particularly highlights the significance of a Human-Centered Design (HCD) approach. The HCD methodology focuses on satisfying user needs at a macro level, recognizing that developing AI tools for specific organizational or group interests inherently introduces bias due to narrowly defined problem scopes (IEEE, 2019).

# **4.2 Importance of addressing bias at pre-design, development, and deployment stages**

Addressing biases proactively at the pre-design stage directly influences data selection and system training choices. For instance, designing a loan-approval AI without accounting for historical inequities risks embedding discrimination before coding even begins (Cavazos, 2025). Similarly, healthcare AI models trained predominantly on data from a single demographic can lead to misdiagnosis for other populations (Norori et al., 2021). Algorithmic choices further compound bias; selecting data features that unintentionally proxy protected attributes (like zip codes correlated with race) can amplify discriminatory outcomes (Holdsworth, 2023). Therefore, techniques such as fairness-aware algorithms, re-weighting, or adversarial debiasing become essential (Jain et al., 2024). Even with thorough pre-design efforts, some biases only become apparent post-deployment, as exemplified by facial recognition systems underperforming on darker skin tones due to previously untested scenarios (Srinivasan & Chander, 2021). Hence, bias mitigation must persist through ongoing monitoring and feedback after the AI tools are in real-world use (Schwartz et al., 2022).

# **4.3 Recommended industry standards and ethical guidelines for AI development**

Addressing the complexities and the scalable nature of modern AI systems aligns closely with ISO standard 9241-210:2019 [252], an advancement over ISO 13407:1999, emphasizing usability and accessibility through human-centric approaches. Complementing this standard are IEEE's *Ethically Aligned Design* guidelines and the OECD's AI Principles (IEEE, 2019). Key recommended industry standards and ethical guidelines include:

#### **4.3.1 Diverse and Representative Data:**

Datasets must accurately reflect the demographics of populations affected by AI, with regular audits identifying and correcting data skews or omissions (Norori et al., 2021).

#### 4.3.2 Bias-Aware Algorithms:

Developers should employ fairness-aware methodologies like fairness constraints and adversarial debiasing, thoroughly documenting algorithmic choices (Jain et al., 2024).

#### 4.3.3 Interdisciplinary Collaboration:

AI design teams should involve social scientists, ethicists, and community representatives to provide context and critique of technical decisions (IEEE, 2019).

#### 4.3.4 Robust Testing and Validation:

Systems must undergo rigorous testing across diverse scenarios, emphasizing fairness metrics alongside traditional accuracy measures (Schwartz et al., 2022).

#### 4.3.5 Transparency and Explainability:

AI solutions should provide clear, understandable rationales behind decisions, enabling users to question, challenge, and comprehend AI outcomes (Srinivasan & Chander, 2021).

#### **4.3.6 Continuous Monitoring:**

Post-deployment strategies, including user feedback mechanisms and independent audits, must be in place to continually identify and address emerging biases (Holdsworth, 2023).

#### 4.3.7 Ethical Frameworks:

Organizations must adopt ethical principles that uphold human rights, privacy, and social justice, incorporating these principles into their internal governance processes (IEEE, 2019). Adopting these comprehensive standards enables the AI community to innovate responsibly, share accountability for bias mitigation, and voluntarily align with regulatory expectations, reflecting NIST's non-regulatory stance (Schwartz et al., 2022). This voluntary adoption resonates with research indicating that unbiased information requires minimal external influence during its collection and analysis (Norori et al., 2021).

## 5. Challenges and Limitations

## 5.1 Discussion of the complexities in eliminating AI bias

Eliminating AI bias is inherently complex due to its multidimensional origins. Bias emerges from systemic inequalities (such as historical inequities embedded in hiring data), statistical limitations (including unrepresentative or incomplete datasets), and human factors (like implicit assumptions or beliefs held by developers) (Srinivasan & Chander, 2021). These diverse sources make bias a deeply rooted socio-technical challenge rather than a singular issue that can be simply fixed. Additionally, the pursuit of perfectly representative data is often impractical, particularly for marginalized populations lacking substantial digital footprints, leaving persistent gaps that reinforce statistical bias (Norori et al., 2021).

## **5.2 Potential drawbacks of current mitigation strategies**

Current bias mitigation strategies, although aligned with NIST's foundational concepts and industry standards, encounter significant practical limitations (Schwartz et al., 2022). Techniques like re-weighting data or enforcing fairness constraints often lead to a reduction in overall accuracy. For example, predictive policing AI adjusted for racial equity may lose predictive effectiveness, raising operational concerns (Jain et al., 2024). Furthermore, these mitigation strategies typically demand considerable resources, including expertise, time, and funding. Such requirements disproportionately burden smaller organizations and could widen disparities within industry (Holdsworth, 2023).

Additionally, over-reliance on purely technical solutions, such as algorithmic debiasing, risks overlooking deeper systemic problems—biased institutional policies—that require broader, more fundamental societal interventions (IEEE, 2019). Another key limitation arises from the voluntary nature of many standards, like NIST's proposal. Without binding rules or enforced accountability, companies might superficially adopt fairness guidelines to satisfy market demands or prevent stricter regulations, resulting in uneven and inconsistent implementation (Schwartz et al., 2022). The lack of audits or certifications further reduces transparency, making it challenging for external stakeholders—such as consumers and regulators—to verify genuine adherence versus mere compliance in appearance.

## 5.3 Consideration of future research and advancements in AI fairness

Given these complexities, future research should address persistent trade-offs, notably between accuracy and fairness, as well as resource-intensive demands versus scalability (Jain et al., 2024). Advancements could include developing more sophisticated fairness-aware algorithms that minimize accuracy loss or exploring innovative methods to achieve truly representative datasets (Norori et al., 2021). Addressing systemic issues will require interdisciplinary research bridging technology, policy, ethics, and social sciences to create holistic and context-aware solutions (IEEE, 2019). Additionally, research could explore accountability frameworks and certification processes to ensure transparency and genuine compliance with fairness standards, thereby addressing gaps in current voluntary standards (Schwartz et al., 2022).

To overcome current limitations in mitigating AI bias, organizations should invest in ongoing ethics training for developers, establish internal audit teams dedicated to fairness and accountability, and adopt open-source benchmarking tools to evaluate model performance across demographic groups. Government and industry coalitions can also work together to develop standardized bias testing protocols, ensuring consistent measurement practices. Furthermore, incentivizing collaboration between tech companies, academic institutions, and impacted communities can foster innovation while grounding AI development in real-world experiences and needs. By combining regulatory guidance with community-driven feedback and technical improvements, the field can move toward more equitable and trustworthy AI systems.

## 6. Conclusion

## 6.1 AI Bias as a Critical Challenge

AI bias remains a critical challenge that impacts the fairness, transparency, and inclusivity of AI-driven decision-making across various industries. As AI models become increasingly embedded in our daily lives, addressing these biases is more important than ever. It is essential that human values and ethical considerations are actively integrated into the development and application of AI systems. This paper has examined the sources of AI bias, the vital role of human judgment in mitigating these biases, the importance of ethical AI governance, and strategies for data debiasing to promote fairness and equity in AI outcomes.

This paper has demonstrated that mitigating AI bias cannot rely solely on technical solutions or ethical theory in isolation; rather, it requires a cohesive, interdisciplinary framework that combines both approaches throughout every stage of AI development.

## 6.2 Key Takeaways

- AI Bias is a Socio-Technical Issue: Bias is not just a technical flaw—it stems from historical inequalities, biased data collection, and institutional decision-making (Schwartz et al., 2022). Addressing AI bias requires human oversight, ethical considerations, and governance frameworks.
- Techno-Solutionism is Insufficient: The assumption that AI bias can be resolved solely through improved algorithms or larger datasets overlooks the critical role of human judgment, organizational policies, and societal values (Srinivasan & Chander, 2021). Relying exclusively on techno-solutionism to address AI bias risks allowing these systems to cause more harm than good, as it neglects the complex social and ethical dimensions that must be considered in responsible AI development.
- Human Judgment is Essential for Fair AI: AI models require continuous human oversight to identify and correct biases before deployment. This includes participatory design, diverse stakeholder involvement, and human-in-the-loop (HITL) decision-making (Holdsworth, 2023).
- Dataset Bias Must be Addressed at the Source: AI models are only as fair as the data they are trained on. Strategies like subgroup fairness techniques, data rebalancing, and synthetic data generation are crucial for improving representation in training datasets (Jain et al., 2024).
- The D3M Framework Enhances Fairness: The Data Debiasing with Datamodels (D3M) framework helps optimize data selection to reduce bias while maintaining model accuracy. It significantly improves subgroup fairness and AI robustness in diverse applications (Jain et al., 2024).
- Ethical AI Governance is Crucial for Fair AI Outcomes: Organizations must adopt ethical AI principles, such as Microsoft's Responsible AI framework, IBM's AI Fairness 360 Toolkit, and Google's AI Principles, to create trustworthy AI systems (Schwartz et al., 2022).

## 6.3 Strategies for Mitigating AI Bias

#### 6.3.1. Implement Ethical AI Governance

To ensure ethical and responsible use of artificial intelligence, it is essential to develop and enforce AI fairness guidelines that align systems with core ethical principles such as transparency, equity, and accountability (Schwartz et al., 2022). These guidelines serve as a foundational framework for preventing harmful outcomes and fostering public trust. In addition to establishing clear principles, organizations must implement accountability structures that mandate human oversight in high-risk AI applications, particularly in areas like healthcare, criminal justice, and employment, where biased decisions can have serious consequences (Holdsworth, 2023). Furthermore, ensuring compliance with global regulatory standards—such as the EU AI Act and the U.S. AI Bill of Rights—is critical for maintaining legal and ethical consistency across international markets (Cavazos, 2025). Together, these measures provide a comprehensive approach to governing AI systems in ways that uphold societal values and protect vulnerable populations.

#### 6.3.2. Improve Dataset Distribution and Data Quality

To reduce sampling bias and promote fairness in AI systems, it is essential to use diverse and representative datasets that accurately reflect the populations the AI is intended to serve (Jain et al., 2024). This helps to ensure that underrepresented groups are not excluded from the decision-making processes driven by AI. In addition to diverse data collection, developers should apply data rebalancing techniques—such as reweighting, synthetic data generation, and adversarial debiasing—to correct imbalances within training datasets and improve model accuracy across demographic groups (Norori et al., 2021). Moreover, implementing real-time data audits allows organizations to continuously monitor AI systems for fairness, detect emerging biases, and make necessary adjustments as societal conditions or data patterns change. These combined efforts are critical for building AI models that are equitable, reliable, and socially responsible.

#### 6.3.3. Integrate Human Oversight and Participatory AI Design

Ensuring ethical and unbiased AI outcomes requires active human involvement throughout the development and deployment process. One effective strategy is implementing human-in-the-loop (HITL) frameworks, which allow AI-generated decisions to be reviewed and validated by human experts before being applied in real-world settings—particularly in high-stakes domains like healthcare, finance, and criminal justice (Srinivasan & Chander, 2021). Additionally, involving a wide range of stakeholders in the AI design process, including ethicists, policymakers, and members of affected communities helps ensure that diverse perspectives and ethical considerations are integrated into the system from the start. To support these efforts, it is also essential to train AI development teams on bias awareness and responsible design practices. By equipping developers with knowledge about systemic discrimination and ethical AI principles, organizations can foster the creation of more equitable, inclusive, and socially accountable technologies (Holdsworth, 2023).

#### 6.3.4. Leverage Fairness-Aware AI Algorithms

To build fair and accountable AI systems, developers must apply subgroup fairness techniques—such as equalized odds, demographic parity, and counterfactual fairness—which ensure that model outcomes are consistent and equitable across different demographic groups (Schwartz et al., 2022). These statistical methods help reduce disparities in predictions and support more inclusive decision-making. In addition to designing for fairness, it is critical to use algorithmic audits and explainability tools to uncover hidden biases within AI models. These tools not only improve transparency but also allow developers and stakeholders to better understand how decisions are made, enabling more informed oversight and correction. Furthermore, AI systems should be equipped to flag potential biases in real-time, particularly when handling sensitive or critical tasks. Requiring human validation before executing such decisions ensures an added layer of accountability and helps prevent the automation of harmful outcomes (Holdsworth, 2023). Together, these strategies contribute to the development of AI systems that are not only technically robust but also socially responsible.

## 6.4 The Importance of Continuous Evaluation and Refinement

AI bias cannot be fully eliminated, but continuous evaluation and improvement can significantly reduce its impact. This approach not only upholds ethical values but also ensures that users receive more thoughtful, responsible, and fair outcomes from AI models. Industries must prioritize:

- Regular AI audits to ensure ongoing fairness and transparency (Schwartz et al., 2022).
- Bias monitoring systems that detect and mitigate evolving biases in real-time (Jain et al., 2024).
- Updating AI governance policies to reflect emerging ethical considerations and societal values (Holdsworth, 2023).
- Public transparency reports that document AI fairness efforts, model evaluations, and improvements over time (Norori et al., 2021).

The future of AI relies on adaptive frameworks that can evolve in response to emerging biases, rapid technological advancements, and shifting societal needs. As the potential applications of AI continue to expand across industries—from healthcare and finance to education and social media, proactive measures must be taken now to guide its development responsibly. I believe that mitigating bias within AI requires a well-rounded, multi-faceted approach that addresses both technical and societal factors. Ensuring fairness in AI is not a

one-time fix, but a continuous, evolving process that demands ongoing attention, collaboration, and accountability.

With sustained commitment, cross-sector collaboration, and evolving policy frameworks, the future of AI can be one of shared progress—where technology reflects the diverse needs, values, and voices of all communities it touches.

# **6.5 Call to Action: Industries Must Adopt Ethical AI Practices for Inclusive and Fair Outcomes**

To ensure AI is used responsibly, governments, businesses, and researchers must work together to embed fairness, transparency, and accountability into AI systems. Organizations must:

- Adopt ethical AI frameworks that align with fairness, inclusivity, and regulatory compliance.
- Invest in bias mitigation research to improve AI decision-making across industries.
- Collaborate across disciplines to incorporate socio-technical perspectives into AI design.
- Commit to transparency and public accountability in AI fairness efforts.

AI should enhance human decision-making, not reinforce existing societal inequalities. By implementing robust bias mitigation strategies, the industry can build more inclusive, equitable, and trustworthy systems. Addressing AI bias is a complex issue that grows alongside the rapid integration of AI into our daily lives. It is not something that can be solved with a single solution, but rather through a combination of technical, ethical, and policy-driven approaches. As AI continues to shape how we work, communicate, and access services, it is essential that these technologies produce outcomes that are accessible and fair for everyone. All individuals—regardless of race, gender, or socioeconomic status—deserve equal treatment and opportunity in AI-driven systems. The time to act is now. AI must be developed with fairness, ethics, and human-centered values at its core—because the future of technology should serve all people, not just a privileged few.

# References

Balderson, K. (2025, January 7). *38 AI Statistics for 2025: Growth, Usage & Adoption*. MS Power Use. Retrieved March 1, 2025, from https://mspoweruser.com/ai-statistics/

Cavazos, S. (2025, 2 15). The Impact of Artificial Intelligence on Lending: A New Form of Redlining? *Texas A&M Journal of Property Law*, *11*(2), 331-354. https://scholarship.law.tamu.edu/journal-of-property-law/vol11/iss2/4/

Holdsworth, J. (2023, December 22). *What is AI bias?* IBM. Retrieved March 6, 2025, from https://www.ibm.com/think/topics/ai-bias

IEEE. (2019). *Ethically aligned design: A vision for prioritizing human well-being with autonomous and intelligent systems (Version 2)*. IEEE. https://standards.ieee.org/wp-content/uploads/import/documents/other/ead\_v2.pdf

Intersoft Consulting. (2016). General Data Protection Regulation. *Cooperation and consistency*, 7(63), 135. https://gdpr-info.eu/

Jain, S., Hamidieh, K., Georgiev, K., Ilyas, A., Ghassemi, M., & adry, A. (2024, June). *Data Debiasing with Datamodels (D3M): Improving Subgroup Robustness via Data Selection.* Arxiv. Retrieved February 10, 2025, from https://arxiv.org/pdf/2406.16846

Norori, N., Hu, Q., Aellen, F. M., Faraci, F. D., & Tzovara, A. (2021, October 8). Addressing bias in big data and AI for health care: A call for open science. *Patterns*, *2*(10), NA. 10.1016/j.patter.2021.100347

Schwartz, R., Vassilev, A., Greene, K., Perine, L., Burt, A., & Hall, P. (2022, March). Towards a Standard for Identifying and Managing Bias in Artificial Intelligence. *National Institute of Standards and Technology*, *1270*(Special Publication), 86. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1270.pdf

Srinivasan, R., & Chander, A. (2021, July 6). Biases in AI Systems. *Communications of the ACM*, *64*(8), 44-49. Google Scholar. 10.1145/3464903

# **Flipped Classroom Revisited**

Mark Fienup

Department of Computer Science University of Northern Iowa Cedar Falls, Iowa 50614-0507 mark.fienup@uni.edu

## Abstract

In general, a *flipped classroom* is an instructional strategy where traditional learning environments are reversed. Instead of introducing new content during class time, students first engage with new material at home, typically through video lectures or reading assignments. Then, class time is dedicated to exercises, projects, or discussions that deepen understanding and allow for practical application of the material.

I have been a strong proponent in the flipped-classroom for at least a decade now. My current approach to applying the flipped classroom is to require students to watch a short (15 to 30 minute) video followed by taking a short five question multiple-choice quiz before coming to each class.

Unfortunately, in both of the last two semesters, I have been assigned to teach "new" courses without much lead time. Wanting to keep the flipped classroom model without taking the time to create video-quiz pairs encouraged me to revisit the possible options. This paper reports on a couple of approaches taken.
# **1 Background**

In general, a *flipped classroom* is an instructional strategy where traditional learning environments are reversed. Instead of introducing new content during class time, students first engage with new material at home, typically through video lectures or reading assignments. Then, class time is dedicated to exercises, projects, or discussions that deepen understanding and allow for practical application of the material.

I have been a strong proponent in the flipped-classroom for at least a decade now, and have previously presented a MICS 2015 paper on the pedagogy of the flipped classroom approach [1]. My current approach to applying the flipped classroom is to require students to watch a short (15 to 30 minute) video followed by taking a short five question multiple-choice quiz before coming to each class. These videos typically provide an introduce to the topic for the class, thus freeing up class time for more challenging application or in-depth coverage of the topic. Since I teach 75-minute classes on Tuesday and Thursday, there are roughly 25 pre-class video-quiz pairs in a course which are weighted from 10% to 20% of a student's grade depending on the course.

While I am reasonably satisfied with my current "pre-class video-quiz pair" application of the flipped classroom, there are some down-sides. For new courses, it is very time consuming to create these pre-class videos and corresponding quizzes. For existing courses, course modifications might have a cascading on existing video-quiz pairs. Generally, I create these video-quiz pairs during the summer when I have more free time.

Unfortunately, in both of the last two semesters, I have been assigned to teach "new" courses without much lead time. Wanting to keep the flipped classroom model without taking the time to create video-quiz pairs encouraged me to revisit the possible options. This paper reports on a couple of approaches taken. The first approach is to provide students with a list of prompts to some Generative Artificial Intelligence tool (e.g. ChatGPT and Microsoft Copilot). The results of these prompts provided a basic introduction to the topic of the class. The second approach was to use a required zyBook textbook which is an online textbook containing embedded "participation activities" (PA). These participation activities are like interactive quizzes allowing multiple attempts until a student successfully answers and tracks their progress providing a report for the instructor. This paper reports on the application of these approaches in practice and provides some conclusions for their usage in a flipped classroom.

# 2 Flipped Classroom using Generative AI Tools

During Fall 2024 I was assigned to teach a brand-new course Introductory Programming for Data Science (CS 1170). This course is one of four courses comprising the Data Science certificate within the University of Northern Iowa (UNI) general education program called UNI Foundational Inquiry (UNIFI). It is a 3-hour/credit course, that has no prerequisites and has a course description of:

"Explores use of software development and tools to explore data sets, find patterns, and solve complex problems in the area of data science. Topics include introductory programming methodologies, data visualization, and an introduction to data mining."

The Student Learning Outcomes (SLO) from the syllabus are: "After this course you should be able to:

- 1. write small programs that process data from files using branching, looping, functions, and lists/arrays
- 2. apply Data Science modules for data analytics (supervised and unsupervised machine learning) and data visualization"

Before the class started, I had decided to experiment using some generative Artificial Intelligence (AI) tool to easily flip the classroom. Since UNI had a license for Microsoft Copilot and was encouraging its use, I decide to use it. Originally, I planned to:

- 1. providing students with a list of "canned" prompts for them to submit to some generative AI tool before coming to class. The results of these prompts provided a basic introduction to the topic of that day's class.
- 2. to show completion, I asked students to submit a copy of Microsoft Copilot's response to these prompts on UNI's learning manage system (LMS) BlackBoard. These had a due date and time before the class starts.

Since I wanted students to start using Microsoft Copilot before the first day of class, I provided them with UNI's brief Microsoft Copilot Guide, and Elon University's online student guide to AI [2]. This seemed sufficient to get students started using Microsoft Copilot with no difficulties.

The first couple weeks of class I followed this model. Appendix A contains a copy of the PRE-Lecture 1 assignment instructions and corresponding Microsoft Copilot responses. The prompts for this PRE-Lecture 1 assignment were:

- 1. Explain to a college freshman how a computer program works
- 2. How do I get started with Python programming?

Microsoft Copilot's responses consistently provided strong answers and a basic introduction to the topic of that day's class. It was important to provide in the prompts to Microsoft Copilot the appropriate level of answer like "to a college freshman."

Students were completing these PRE-Lecture generative AI assignments, but I was concerned that students were only superficially engaging with the Microsoft Copilot responses. With a little work I could have developed short multiple-choice question quizzes to force more engagement with the Microsoft Copilot responses before coming to class. Fortunately, by this time all students had purchased the required online zyBook textbook. The next section describes how the zyBook textbook was used to flip the classroom for the rest of the semester.

## **3 Flipped Classroom using Online zyBook**

When looking for a textbook for this course, a colleague (Dr. Sarah Diesburg) who teaches our CS-1 Introduction to Computing CS 1510 course recommending the zyBook she uses: Programming in Python 3 [3] because they recently added a couple chapters focusing on data science, and she really likes their interactive online learning content. While the interactive online learning content was intriguing, I mostly picked this zyBook textbook because I could highly customize the basic textbook to match the goals of my course by rearranging chapters/sections, omitting chapters/sections, and mark chapter/sections as optional.

The "About zyBooks" statement summarizes their approach as:

"zyBooks provide interactive, online learning content that replaces a traditional textbook with less text and more action. Content is presented in small chunks and extraneous information is removed, helping to manage a student's cognitive load. This innovative pedagogy—supported by learning science and zyBooks' <u>own</u> research—can motivate students to learn more, improve their classroom preparedness, and earn higher grades.

Students can read, apply concepts, do homework and labs, and access instructorcreated content without leaving the zyBooks platform. Animations help students visualize concepts, and learning questions engage students and provide immediate answer-specific feedback. Many zyBooks also contain built-in, algorithmic challenge activities that can be assigned as practice or graded homework. Activities are auto-graded, and all student participation on in-book activities is recorded for instructor review and analysis."

For my purpose of flipping the classroom without a huge amount of work, the zyBook textbook was ideal. I could create a PRE-lecture activity introducing that day's topics by picking sections for students to read including embedded "participation activities" (PA). These participation activities are like interactive quizzes allowing multiple attempts until a student successfully answers. Plus, it tracked students' progress providing a report for the instructor. Figure 1 shows an example of a participation activity about string slicing that is embedded within zyBook text. Changing the numeric values within the print statement causes a different "Output:" string and changes the highlighting of the portion of the string. While anyone can get all PA points just by participating, research conducted by zyBook indicates that 89% of students will earnestly try when completing these participation activities with only 10% of overall course points allocated to them [4].

In addition to participation activities, zyBooks also has challenge activities (CA) which is the zyBook version of a homework problem. A CA allows students to apply and practice what they have learned. Students must eventually get the answer to a CA right on their own to get points, so the CAs serve as a lightweight assessment. Figure 2 shows an example of a challenge activity about string slicing where students must type in a sequence of correct code to complete the CA.



Figure 1: Example of Participation Activity Embedded within Text.

CHALLENGE 6.1.2: String slicing.				
Start         Integers start_index and end_index are read from input. Assign sliced_lyric with the result of slicing orig_string from start_index to end_index.         Click here to show example				
<pre>1 start_index = int(input()) 2 end_index = int(input()) 3 orig_string = 'The wheels on the bus go round and round' 4 5 ''' Your code goes here ''' 6 7 print('Full lyric:', orig_string) 8 print('Sliced lyric:', sliced_lyric) </pre>				
1	2	3	4	

Figure 2: Example of Challenge Activity.

Initially, when I switched to creating PRE-lecture activities introducing that day's topic, I picking all the participation activities and all the challenge activities. I quickly discovered that this was very time consuming for the students to complete, so I switched to assigning all the participation activities and only a few challenge activities.

# 4. Conclusions

Switching my current "pre-class video-quiz pair" application of the flipped classroom, to either using generative AI prompts or zyBook participation activities did reduce the overhead of applying the flipped classroom approach. The generative AI prompts were probably less effective, unless a more engaging pre-class activity could be developed like an online quiz. The use of a zyBook textbook and assigning pre-class reading plus completion of embedded participation activities is a good solution. Unfortunately, you have to be teaching a course for which a zyBook textbook (or something similar) has be written.

# References

[1] Mark Fienup. Flipping over Flipped Instruction. In *Proceedings of the 48th Annual Midwest Instruction and Computing Symposium*, April 2015.

[2] Elon University. AI-U/1.0 Fall 2024. https://studentguidetoAI.org/, Fall 2024.

[3] Bailey Miller. *Programming in Python 3*. <u>https://learn.zybooks.com/</u>

[4] Alex Edgcomb and Frank Vahid. *How Many Points Should Be Awarded for Interactive Textbook Reading Assignments?*. Frontiers in Education Conference (FIE), IEEE, 2015. <u>https://www.zybooks.com/wp-content/uploads/2016/06/How-Many-Points-Should-Be-Awarded-for-Interactive-Textbook-Reading-Assignments\_.pdf</u>

# Appendix A: PRE-Lecture 1 Generative AI Assignment and Corresponding Microsoft Copilot Response

# **PRE-Lecture 1 Generative AI Task**

Before the first class (8/27), please prepare by entering the following prompts into Microsoft Copilot and read the responses. Directions are below.

# Prompts: Type or paste prompt into the "Ask me anything" text box.

- 1. Explain to a college freshman how a computer program works
- 2. How do I get started with Python programming?

# What is Microsoft Copilot?

Microsoft Copilot is an AI-powered tool that can be used to generate content, summarize information, and assist with various tasks through prompt-based interactions. Copilot uses the GPT-40 architecture, which is a state-of-the-art language model developed by OpenAI.

# How to Access Microsoft Copilot

Copilot is available to all current UNI community members with active CATID login credentials. For help, login to Service Hub and search for "Copilot" for additional information.

#### · Web Browser

- Visit <u>copilot.uni.edu</u> to access Copilot directly from your browser.
- $_{\odot}$  When prompted, input your UNI CATID credentials.

## · Mobile App

- $_{\odot}$  Download the Copilot app from your app store.
- Select "Sign in".
- $_{\odot}$  When prompted, input your UNI CATID credentials.

# Submission

· Copy prompt responses and submit them on course UNI eLearning site in Unit 1: ... | Week 1: ... | PRE-Lecture 1 Generative AI Task

# **Course AI Usage Policy**

Al is rapidly changing the face of software engineering. Artificial Intelligence tools (e.g. chatGPT and Bard) are rapidly developing that can generate full programs or act as a "copilot" by suggesting lines of code as you are writing. These tools hold great promise to increase your future productivity as a programmer, but only if you are a proficient enough programmer to assess the correctness and quality of code that the Al produces.

To achieve programming proficiency, you need to put in the practice. You will not get any better at programming by having an AI tool write code for you. Thus, in this course, you must write all your own code for programming homework without the use of AI.

6

Microsoft Copilot Response to: Explain to a college freshman how a computer program works

Sure! Let's break it down step-by-step:

- 1. **What is a Computer Program?** A computer program is a set of instructions that tells a computer what to do. These instructions are written in a programming language, which is a special language designed to communicate with computers.
- Programming Languages: There are many programming languages, such as Python, Java, and C++. Each language has its own syntax (rules for writing code) and is used for different types of tasks.
- 3. **Writing Code:** When you write a program, you are essentially writing a list of instructions. For example, if you want the computer to add two numbers, you would write code that tells the computer to take two numbers, add them, and then display the result.
- 4. **Compiling/Interpreting:** Once you've written your code, it needs to be translated into machine language (binary code) that the computer can understand. This is done by a compiler or an interpreter, depending on the programming language you're using.
- 5. **Executing the Program:** After the code is translated, the computer can execute the program. This means it follows the instructions step-by-step to perform the tasks you've written.
- 6. **Debugging:** Sometimes, your code might not work as expected. Debugging is the process of finding and fixing errors (bugs) in your code.
- 7. **Running the Program:** Finally, you run your program to see the results. If everything is correct, the computer will perform the tasks as you intended.

Here's a simple example in Python:

```
# This program adds two numbers and prints the result
# Define two numbers
num1 = 5
num2 = 3
# Add the numbers
result = num1 + num2
# Print the result
print("The result is:", result)
```

In this example:

• We define two numbers (num1 and num2).

- We add these numbers and store the result in a variable called result.
- We print the result using the print function.

Does that help clarify how a computer program works? Feel free to ask if you have any more questions!

#### Microsoft Copilot Response to: How do I get started with Python programming?

Getting started with Python programming is a great choice! Here's a step-by-step guide to help you begin:

#### 1. Install Python:

 Download and install Python from the official website python.org. Make sure to check the box that says "Add Python to PATH" during installation.

#### 2. Choose an Integrated Development Environment (IDE):

- An IDE is a tool that helps you write and test your code. Some popular options for beginners are:
  - **IDLE:** Comes with Python installation.
  - **PyCharm:** A powerful IDE with many features.
  - **VS Code:** A lightweight and versatile code editor.

#### 3. Learn the Basics:

- Start with basic concepts like variables, data types, and simple operations.
   There are many free resources available online, such as:
  - **Python.org's official tutorial:** Python Tutorial
  - Codecademy's Python course: Codecademy
  - W3Schools Python tutorial: W3Schools

#### 4. **Practice Coding:**

- Practice is key to learning programming. Try solving problems on platforms like:
  - LeetCode: LeetCode
  - HackerRank: HackerRank
  - **Codewars:** Codewars

#### 5. Build Projects:

• Start with small projects to apply what you've learned. For example, you can create a simple calculator, a to-do list app, or a basic game.

#### 6. Join a Community:

- Join online communities to get help and share your progress. Some popular communities are:
  - Stack Overflow: Stack Overflow
  - **Reddit's r/learnpython:** r/learnpython
  - **Python Discord:** Python Discord
- 7. Keep Learning:

 Programming is a continuous learning process. Explore more advanced topics like object-oriented programming, web development, and data science as you progress.

Here's a simple Python program to get you started:

```
# This program prints "Hello, World!"
```

print("Hello, World!")

This program uses the print function to display the text "Hello, World!" on the screen.

Feel free to ask if you have any questions or need further guidance! Happy coding! 😊

# Lessons Learned from the Development of a Multi-Disciplinary Pre-capstone Course in Computing

Tomi Heimonen and Tim Krause

Department of Computing and New Media Technologies

University of Wisconsin-Stevens Point

Stevens Point, WI 54481

{theimone, tkrause}@uwsp.edu

#### Abstract

This paper discusses the pre-capstone to capstone project sequence in computing at the University of Wisconsin-Stevens Point, covering the pre-capstone course's purpose, development history, and key lessons learned over five years.

Three useful elements are highlighted. First, pre-capstone students attend two agile sprint reviews completed by capstone students, to experience diverse projects and adaptations to Agile methodology. Second, a prescriptive project plan template and in-class activities help students develop their own capstone objectives, requirements, resource management, schedules, risks, and mitigation plans. Students create a five-sprint schedule emphasizing key tasks and deliverables. Third, continuous improvement and flexibility are fostered through feedback from peers and instructors, with multiple rounds of feedback and peer review exposing students to various project topics and strategies.

Additionally, potential solutions for resource and credit constraints are outlined, such as integrating project planning and management concepts across the shared computing curriculum for all majors.

# **1** Introduction

Capstone experiences provide students with the opportunity to integrate and apply their learning in a comprehensive project prior to graduation. These courses often involve realworld problems requiring application of discipline specific knowledge and skills and foster critical thinking, communication, and collaboration. In computing, capstone courses are a form of practicum that can help address weaknesses in areas relevant to employment, such as practical experiences, project management skills, and problem solving [1]. In this paper, we describe the development process of a pre-capstone course, an intentional experience to prepare students for the capstone course, and the lessons we have learned along the way.

Initially, our capstone course – Applied Development Project (CNMT 480) – included all stages of project planning and management from initiation to deployment and took place over one semester (16 weeks) in a team-based environment. This aligns with widespread practice; 66% of institutions in a recent review [2] of software engineering capstone courses completed the capstone in one semester. Our capstone course was, and remains, multi-disciplinary within the broad umbrella of computing and new media technologies. Graduating seniors from the department's Computer Information Systems (CIS) and Web Development (WD) programs are integrated into a single course that covers a variety of project types. The CIS program includes emphasis areas (options) in application development and support, cybersecurity, and networking. The WD program, while technically a "single track" curriculum, allows for students to specialize in front-end development or user experience design/research through the selection of elective courses. Although our decision to implement a combined course was by design, sometimes integrated capstone experiences are introduced by necessity [3].

The development of the pre-capstone course discussed in this paper grew out of the realization that attempts to cover both project planning and implementation within the confines of a single semester led to less-than-ideal student outcomes. It was common for the initial planning stage of the projects to consume valuable development time, which resulted in project scopes being cut significantly to finish in time. Our adopted a sequence is similar to what Pettiford [4] describes as a "3-course capstone sequence" – an introductory preparation course combined with a capstone project.

We had discovered that additional preparation was also required for any students who needed any more than a cursory review of the modified, Scrum-like Agile methodology used to manage their projects in the capstone course. Similar to prior findings in the management information systems [5] and electrical and computer engineering fields [6], the goal here was to address knowledge gaps, which in our case were due to the differing curricula of the department's programs that exposed students to varying degrees of project planning and management processes.

## 2 Development of the Pre-Capstone Course

The department began the planning process for the pre-capstone course, *Applied Development Project Planning (CNMT 479)*, in 2020 and the new course was offered for the first time in Spring 2021 by the first author (who has taught the course every semester to date). To accommodate existing students, the pre-capstone course ran as a hybrid and included both students taking the capstone course that same semester and students planning for their capstones the following semester. Because of COVID-19 restrictions, the course ran in an asynchronous online format. The first iteration already included some of the key elements we further revised in subsequent semesters, including 1) attendance in capstone course sprint reviews and writing a reflection paper on lessons learned; 2) peer and instructor feedback of project plan drafts and revisions to the project plan; and 3) a prescriptive project plan template to guide planning efforts.

## 2.1 Initial Transition Period

We immediately identified the lack of formal instruction and hands-on experiences with project planning as a challenge. Because of the asynchronous format, students were instructed to schedule "catch up" sessions with the instructor and conduct independent learning based on materials provided. In the following Fall and Spring 2022 semesters, the course ran as an 8-week hybrid offering after most COVID-19 restrictions were lifted. This enabled class time to be used for discussing project planning fundamentals and hands-on activities such as paired peer review project plan drafts. As the course was still being introduced into students' graduation plans, the enrollment included both capstone and pre-capstone students, which complicated practical classroom management. Moreover, a non-trivial number of students were concurrently registered in the precapstone and capstone courses owing to upstream scheduling conflicts and advising issues in prior semesters.

In Fall 2022, we addressed the above issues by scheduling the pre-capstone course as a 16-week in-person course with a split design. This additional time allowed for more inclass time to be dedicated to instruction on project planning, discussion, and feedback for pre-capstone students. Students concurrently registered in the pre-capstone and capstone followed a more compressed project planning cycle at the start of the semester.

## **2.2 Current Developments**

The pre-capstone course assumed the schedule that is currently followed in Spring 2023. The project planning process is split into three discrete stages. By week four, students submit their initial project plan definition, which includes client and project identification, and the completion of an initial draft of project goals and objectives. In week nine, students submit a draft of the completed project plan with requirements, project schedule, resource management plan, and risk management plan. A revised plan follows in week 14, followed by a one-on-one review meeting with the instructor. We

wanted to err on the side of caution and provide students with ample time to work through the planning process with their clients. However, students were also required to attend two capstone course sprint reviews in weeks 5 and 8, to coincide with the 3-week sprint cycle of the course. The first seven weeks also included an expanded timeframe for discussion of project planning techniques and informal exercises around identifying project goals, objectives, requirements, and risks.

#	Course topics	Course Activities/Assignments
1	Syllabus review & course introduction	
2	Course & capstone organization	Identify project topic/client
3	Project planning fundamentals: Technical communication	Identify project topic/client
4	Project planning fundamentals: Agile; project goals and objectives	Submit project plan topic definition
5	Sprint reviews	Attend capstone sprint review
6	Project planning fundamentals:	Develop project goals and objectives
	Requirements gathering, deliverables and scheduling	
7	Project planning fundamentals: resource and risk management	Define project requirements
8	Sprint reviews	Attend capstone sprint review
9	Project plan draft Q&A	Submit project plan draft & meet with instructor
10	Peer reviews	Submit peer reviews
11	Project plan drafts general feedback	Submit sprint review reflection
12	Risk management planning	Revise risk management & communication plans
13	Guest presentation: Real world project management	
14	Revised project plan review & feedback	Submit revised project plan & meet with instructor
15	Revised project plan review & feedback	Meet with instructor

Table 1: Course schedule, Spring 2025

The current schedule in Spring 2025 (Table 1) follows the patterns set in 2023. The main difference is that we have formalized the in-class project planning tasks as a part of the course structure. In the weeks when students are not attending capstone sprint reviews, they work on developing goals and objectives (week 6), define initial project

requirements (week 7), conduct peer reviews on project plan drafts (week 10), and revise their risk management and communication plans (week 12). Students are also required to meet with the instructor for initial review of their project plan drafts to provide immediate, actionable feedback.

## **2.3 Future Plans**

While we are pleased with the student outcomes using the current 16-week model with 50-minute class meetings (all that is allowed for in a 1-credit hour course), the lack of inperson class time is a problem. We are exploring initiatives to ensure that students are sufficiently prepared for project planning prior to entering the pre-capstone course. This would allow us to focus primarily on planning activities, instead of splitting time between "remedial" coverage of planning and agile methodology and practical planning tasks. We will discuss these plans in more detail in section 4.

# **3 Lessons Learned**

Though the next section discusses additional revisions we are considering in better preparing students for the pre-capstone, we have already learned that: 1) students are more successful when they take the pre-capstone in advance of the capstone course; 2) students learn best from not only observing sprint reviews in the capstone course, but in actively reflecting on what they observed, and directly applying it to their own project plans and projects; and 3) through peer reviews and formative instructor feedback, students start developing a trust relationship with their instructors and each other so that they are more able to not only identify potential risks, but feel comfortable when they encounter issues with their projects during their capstone experiences. These findings echo those reported by Schmidt and Clark [7]: exposing students to the capstone processs in advance provides benefits towards the capstone experience. Importantly, our precapstone course also directly builds the project plan, in addition to providing exposure to the capstone experience.

For the first and third point, we believe that the introduction and formalization of scaffolded planning exercises and formative feedback sessions have been helpful in improving the overall success of the capstone projects for students who buy in to the process. While the project scope and goals frequently change for individual capstone projects, the experience in building out a project schedule with associated tasks and deliverables is likely to help students adapt to changing circumstances down the line.

For the second point, we posit that an additional benefit is that students have an opportunity to observe the lifecycle of the capstone projects from sprint 2 (early design and development activities) to sprint 3 (typically the most intense development period). What is frequently demonstrated through the sprint review presentations are issues that students will face the next semester but are challenging to proactively capture in a project plan. These include the need for troubleshooting, addressing communication and approval delays, adapting to changes in project scope and technologies, and more.

The pre-capstone and capstone courses are scheduled to run in the same timeslot throughout the semester. This allows students to attend capstone course sprint reviews without interfering with the rest of their class schedule. The scheduling also presents a practical challenge – the same instructor cannot feasibly teach both courses. We have learned over the past five years that similarly to students building trust and comfort with each other, it takes time for instructors to align their respective approaches to managing the capstone process. The authors taught the pre-capstone and capstone, respectively, for multiple semesters before a new instructor was recruited to teach the capstone in Spring 2023 (who has since taught the course to date). What we have collectively learned is that it is important for both instructors to share a vision for what a good project plan is in terms of scope and measures of success. Our recommendation is to set up multiple checkpoints to review the project plans, specifically at the project identification and draft stages, to avoid downstream challenges.

# **4** Integration of Project Planning and Management in a Shared Computing Curriculum

One of the continuing challenges we face in our department is having one pre-capstone and capstone sequence for all our majors, and their corresponding options or tracks. While this approach has integrative benefits, the most significant issue already discussed previously is that not all our majors gain as much experience with project management generally, and Agile methodology specifically. This issue is confounded by the reality that we only have 1-credit hour to devote to the pre-capstone experience to keep the size of the programs manageable.

Starting this semester, Spring 2025, we are experimenting with Project Management Institute's (PMI) Project Management Ready certification [8] as part of a hybrid course in *Professional IT Communication* (CNMT 410). The course is required of all students, ideally in their junior year, and before the pre-capstone course. CNMT 410 is a 3-credit, hybrid course, where one credit is taught asynchronously, and seems to lend itself to leading up to certification by our students. Because the certification covers predictive, adaptive, and hybrid approaches to project management, we are hopeful that it will help address the most acute knowledge gaps for students prior to enrollment in the precapstone course.

Longer term, and when enrollments allow for it, we expect to offer at least a second section of the capstone course. Currently, one course needs to serve students with an array of projects topics in software development, cybersecurity, networking, web design and development, and UX design and research. With the pending addition of a new major in artificial intelligence and the associated projected enrollment increases, we have the opportunity to tailor the capstone experience more specifically to students with increasingly different projects. This will require different emphases on how they receive feedback from both their peers and instructors. We may also need to revise the precapstone course to include a selection process to direct students into the section best suited for their talents and interests.

# **5** Conclusion

Including a pre-capstone preparation course benefits students by exposing them early to the capstone process and its expectations. This paper provides an example of a combined pre-capstone course designed to serve students from several computing disciplines. Our experiences are that students are more successful after taking the pre-capstone course, which can be attributed to directly observing the capstone experience, creating an initial project plan, and starting to build relationships with the instructors and each other in advance of the capstone course. For programs that are constrained in adding new project planning courses or expanding existing courses, it is recommended to introduce students to project management topics as a part of the shared curriculum.

# References

[1] Joint ACM/AIS IS2020 Task Force, Leidig, P. & Salmela, H. (2021). A competency model for undergraduate programs in information systems. Association for Computing Machinery. https://doi.org/10.1145/3460863

[2] Tenhunen, S., Männistö, T., Luukkainen, M., & Ihantola, P. (2023). A systematic literature review of capstone courses in software engineering. *Information and Software Technology*, *159*, 107191.

[3] Schwieger, D., & Surendran, K. (2010). Enhancing the value of the capstone experience course. *Information Systems Education Journal*, 8(29).

[4] Pettiford, C. (2014, June). 3-course capstone sequence. In 2014 Capstone Design Conference Proceedings.

[5] Denton, J. W., & Spangler, W. E. (2001). Effectiveness of an integrated pre-capstone project in learning information systems concepts. *Journal of Information Systems Education*, 12(3), 149-156.

[6] Faust, M., Greenberg, A., & Pejcinovic, B. (2012, October). Redesign of senior capstone program in electrical and computer engineering and its assessment. In 2012 *Frontiers in Education Conference Proceedings*, 1-6. https://doi.org/10.1109/FIE.2012.6462390

[7] Schmidt, D. E., & Clark, R. M. (2017, June). Improving student capstone experience by early exposure and engagement. In 2017 *ASEE Annual Conference & Exposition*. https://doi.org/10.18260/1-2--28491

[8] Project Management Institute (2025, March). *PMI Project Management Ready*<sup>TM</sup> *Certification*. Accessed on March 19 at https://www.pmi.org/certifications/pmi-project-management-ready



# Osire: Democratizing LLM Workflows for High Performance Computing

Adam Haile, Aiden Miller Department of Computer Science and Software Engineering Milwaukee School of Engineering 1025 N Broadway, Milwaukee WI, 53202 {hailea, millera} @ msoe.edu

## Abstract

High-Performance Computing (HPC) has become an essential resource for AI engineers, data scientists, and researchers across various disciplines. However, it still presents significant barriers to entry for newcomers to the field. To address these challenges, we introduce a microservices-based platform, Osire, that abstracts complex HPC interactions, integrating with institutional systems for security and ease of use. This platform automates resource creation and allocation while standardizing LLM workflows, academic authentication, and containerized Open-API compatibility. By providing modular services for retrieval-augmented generation (RAG) and on-demand access to open-source large language models (LLMs), it simplifies HPC access and management for users. Originally developed and tested at the Milwaukee School of Engineering (MSOE), the system is designed to be extensible and adaptable to the majority of Academic HPC systems. This platform lowers barriers to HPC access, fostering collaboration, optimizing resource use, and bridging traditional HPC infrastructure with modern AI workflows.

## **1** Introduction

Over 90% of academic institutions with access to High-Performance Computing (HPC) have invested in AI, underscoring its growing role in research (Glick, 2024). Since 2017, research involving LLMs has grown exponentially, surpassing 2,100 papers by 2022 (Fan, et al., 2023). Traditionally, HPC systems have been optimized for batch processing workflows, effectively serving applications such as scientific simulations, computational fluid dynamics, and large-scale data analyses. However, as academic research increasingly incorporates modern AI techniques particularly interactive machine learning methods involving large language models (LLMs) these traditional HPC systems have begun to show limitations in usability and user experience.



Figure 1: Trend of the number of publications utilizing LLMs for their research over time (Fan, et al., 2023).

Large language models, which have become central to various AI-driven research projects, often require interactive environments that allow rapid prototyping, iterative development, and real-time feedback. These needs conflict directly with the batch-oriented design of traditional HPC architectures. Tools such as Open OnDemand (OOD) (Hudak, et al., 2016) have improved access to HPC resources for standard computational tasks but still fall short in accommodating the interactive workflows required by modern AI research.

For those performing research with LLMs, commercial cloud-based AI services offer userfriendly interfaces and streamlined interactions. However, their high costs can be prohibitive for academic researchers and students working with limited budgets. Consequently, students and institutions frequently face a difficult choice: either accept the financial burden of industry-based AI services or navigate the complexities and limitations of traditional HPC environments.

Thus, there is a clear and pressing need for a new approach that combines user-friendliness, interactive capabilities, and cost-effectiveness, specifically designed to address the evolving requirements of LLM-centric academic research in HPC settings.

1

In this paper, we propose *Osire*, a microservices-based platform that abstracts away the complexities of working with LLMs in an academic HPC setting, while ensuring efficient resource utilization and security using proven frameworks and practices.

The core design question behind Osire is how to create a living, adaptable framework capable of continuously evolving alongside rapid advancements in the AI and HPC domains. Given that AI research—particularly involving large language models—is progressing at unprecedented rates, Osire's architecture must remain responsive and extensible to integrate emerging tools, models, and methodologies without extensive redevelopment.

To accomplish this, our solution must support several keys features. First, it must be able to have new programs integrate natively with the platform with the most minimal of requirements. Second, it must be able to navigate the complexity of managing varying job needs automatically. Finally, it must have a mechanism for sharing approved programs with other researchers.

# 2 Design Philosophy

Integrating interactive AI workflows into traditional High-Performance Computing (HPC) systems poses multiple challenges.

## 2.1 Batch Scheduling

This first challenge is that for a given task involving LLM research, the precise amount of compute is unknown. Compute can vary widely from a simple case with a single chat model, to complex agentic workflows with compute varies with details such the size of the chat model, embedding model, vector stores, tools, databases, and other pieces of software.

In a conventional operating system this is not a problem. Conventional operating systems excel at interactive computing—they dynamically allocate resources to processes on-demand, instantly responding to user inputs, and fostering a rapid cycle of experimentation. HPC systems, in stark contrast, are inherently batch oriented. They require researchers to predict and explicitly specify resource needs, such as CPU, memory, and time allocations, in advance. Given the uncertainty around exact resource needs for each changing component, users must repeatedly adjust detailed batch scripts to accommodate new requirements. Alternatively, they might request excessively large resource allocations to avoid frequent script modifications, resulting in inefficient resource utilization. This approach significantly limits flexibility, making rapid prototyping and iterative experimentation cumbersome or even infeasible.

This leads us to the conclusion that HPC is good in cases where you have discrete units of resource-intensive tasks with known resource needs, and HPC fails with tasks that have scaling or dynamic resource requirements. This led us to a microservices-based architecture, where each conceptual unit of work is split into its own service. This paradigm allows us to independently preconfigure the resources needed for each aspect of the LLM workflow and control their communications through a centralized gateway.

## 2.2 Data Privacy

The next critical challenge is maintaining data privacy. In an HPC environment, researchers can be working with data or code that is confidential or protected. Typically, Open OnDemand (OOD) sandbox services are created without appropriate authentication mechanisms, making them automatically accessible from any device on the same network. This poses significant security risks for researchers wishing to prototype applications with independently configuring complex security measures. Osire mitigates this vulnerability by enforcing structured authentication schemes, providing secure environments by default. Consequently, researchers can confidently prototype and develop sensitive projects without inadvertently exposing confidential data or code to unauthorized access. Additionally, Osire makes use of HPC standard architecture for its services code protection, which is isolated user and public shared directories, a standard first developed by the Ohio Supercomputer Center (OSC) (OSC, 2014).

## 2.3 Security

Security is also crucial, especially within academic environments where sensitive research takes place. Preventing unauthorized users from gaining access to HPC resources is a top priority. While most HPC systems have authentication schemes in place, adding an additional layer of security internal to the platform itself allows tracking and analyzation of usage patterns on the platform, providing insights into how users engage with Osire. To address this, Osire employs OAuth-based authentication, providing secure and reliable user verification that safeguards resources from unauthorized access and misuse. Osire's user key generation system links directly to users' institutional SSO (Cohesity, 2021), using a returned custom key after sign-in for all requests afterwards. A nonce variable is also used, allowing users to rotate keys as needed. The final key is generated using JWT (Jones, Bradley, & Sakimura, 2015). Linking access keys to institutional SSOs not only secures the authentication process but also allows tracking and analyzing usage patterns on the platform, providing insights into how users engage with Osire.

## 2.4 Extensibility

Finally, the platform must be expandable and flexible to remain useful long-term. It needs to easily accommodate new features, improvements, and community-driven enhancements. Osire meets this requirement by having low barriers for adding new code and supporting both officially maintained and community-developed "unofficial" microservices. This design encourages collaboration and ongoing development within the academic community, ensuring continuous improvement and adaptability.

Given that Osire should be as portable as possible to encourage both intra-university and interuniversity development, Osire hosts its Gateway component by default on commonly accessible computing devices, such as personal computers, servers, or virtual machines, rather than directly on HPC hardware. This means that the gateways reside on familiar, interactive operating systems like Windows, Mac, or Linux, while computational tasks are dynamically managed as batch jobs in the HPC backend. This approach allows researchers to interact seamlessly and iteratively with their experiments, receiving instant feedback through intuitive interfaces, while computationally demanding tasks are efficiently executed using HPC's optimized resource management.

# 3 Architecture

Osire itself follows the standard microservice design, with a gateway which users will use to make requests to services that Osire supports. Osire makes minor modifications to the standard for ease of use on users and admins while still permitting proper authentication.

## 3.1 Overview

The Osire platform leverages a sophisticated microservice architecture explicitly engineered to mitigate the inherent limitations of traditional HPC frameworks concerning interactive and contemporary AI workflows. At its core, Osire establishes a rigorous and explicit delineation of service classifications and associated management layers, thereby achieving a harmonious balance between user convenience, robust security protocols, and architectural extensibility.



Figure 2: An overview of the complete Osire architecture, separated by level of trust.

#### 3.1.1 Sandbox Services

*Sandbox* services constitute fundamental elements within Osire, instantiated and governed exclusively by individual users within their own gateway environments. These services, accessible solely by their creators and developers, effectively satisfy immediate user demands for interactive computing, enabling expedited prototyping and iterative development process without dependency on HPC batch scheduling constraints. Despite their private instantiation, these services utilize HPC computational resources, access restricted by the SSO API keys, ensuring data privacy, integrity, and resource allocation control. Users build their own local service from a provided template which is immediately connectable to the Osire platform (Haile, Rosie FastAPI

Template, 2024). Applications built from the standardized template have the capability to make full use of Osire.

#### 3.1.2 Production Services

*Production* services, in contrast, are standardized services maintained centrally and stored within a shared directory accessible by all Osire users at a given institution. Any user within said Osire ecosystem can instantiate their own isolated instance of these services for personal use. Although each instantiation is accessible solely by the initiating user, the underlying production service code and execution scripts are universally available, facilitating consistency in resource availability, optimized operational performance, and uniform integration capabilities across heterogenous user contexts. Access to these instantiated services is also managed via SSO API keys, maintaining same security measures for authorized use of HPC resources.

#### 3.1.3 Gateway

The *gateway* component operates as a decentralized service, executed locally by users outside the HPC system but within the institutional network. This gateway performs essential API calls required to utilize both local and global services. Due to the decentralized nature of the gateway, inherit trust cannot be guaranteed, necessitating the stringent security protocols, notably the employment of the SSO API keys. Utilizing dynamic interactions with the centralized service registry, the gateway actively monitors and identifies the current operational state, network location, and accessibility of services. Automated initiation of service launches by the gateway ensure uninterrupted, reliable resource availability and minimizes the necessity of manual administrative intervention.

#### 3.1.4 Access Key Management

Osire further incorporates a stratified security model, structured into distinct layers to maintain robust access control and secure operation. The Privileged layer houses officially sanctioned applications alongside the central service registry, ensuring secure, controlled dissemination of standardized computational resources. Meanwhile, the Protected layer encapsulates critical services, notably the SSO/Key Manager, designed to facilitate secure generation issuance and validation of user-specific access keys explicitly linked to institutional identities. This multitiered security framework enforces rigorous access control protocols, mitigates unauthorized resource exploitation, and guarantees authenticated, secure interactions throughout the system.

#### 3.1.5 Service Requirements

All Osire services adhere to specific operational requirements. Each service must maintain a configuration file detailing critical information such as the latest software version, the repository or location of the service's code, and the defined interval for periodic heartbeat communications. Services are required to send heartbeat checks at these configured intervals, enabling the gateway to monitor their operational status accurately. Additionally, services mandate authenticated requests containing valid API keys. Production services authenticate these keys by cross-referencing them with the centralized Key Manager, while local services necessitate manually configured API keys assigned explicitly by the developing user.

## 3.2 Service Level

#### 3.2.1 OsireLLM

*OsireLLM* serves as a foundational component within the Osire ecosystem, providing a unified and intuitive interface for managing large language model (LLM) workflows. Designed as an orchestrator of multiple LLMs across varying engines (such as NVIDIA NIM (NVIDIA, 2024) or vLLM (Kwon, et al., 2023)), OsireLLM enables users to effortlessly launch, terminate, monitor, and interact with LLM instances from a single point. By abstracting the complexity of direct model management and providing a single interface for inference, it simplifies LLM usage for other services within Osire.

Primarily, OsireLLM prioritizes compatibility with familiar OpenAI-standard APIs (OpenAI, 2022), such as completions, chat completion, and embeddings endpoints, ensuring seamless adoption by existing applications and workflows. Additionally, due its primary use case being an orchestration tool, it automatically supports new API schemas as they become available.

Beyond these straightforward use cases, OsireLLM also supports advanced interactions with distinct LLM engines through dedicated route prefixes. For instance, vLLM has endpoints related to tokenizing requests. To access this endpoint, a separate route is used where the request is prefixed with the engine name ("vllm") and the model name (such as "meta/llama-3.1-70b-instruct"). This allows for the straightforward requests to be in the format expected by the OpenAI API client, which allowing advanced users access to more detailed requests.

By centralizing model interactions behind a standardized gateway, OsireLLM streamlines both basic and complex LLM usage, reducing the overhead typically associated with interactive and flexible AI workflows in HPC environments.

#### 3.2.2 OsireRAG

*OsireRAG* (Haile & Yoder, Building Democratized RAG Systems in High Performance Computing, 2024) is one of the predesigned services within Osire that provides basic RAG capabilities to a user, with modularity to use components as needed. OsireRAG is served as an API to the user, with an interactive dashboard as well to manage their RAG projects. OsireRAG is built by default to be generalizable for any RAG service, however the modularity allows for a user to just use specific elements to their advantage if they have pieces, they would rather custom design and improve for their own needs. OsireRAG also integrates with OsireLLM for its LLM responses, showcasing the compatibility of two distinct services with each other within the Osire platform.

## 3.3 Authentication

Osire mitigates reliance on institutional network configurations by adopting a structured and robust system of protected and shared files within the High-Performance Computing (HPC) environment. To maintain rigorous security standards and ensure comprehensive user isolation, Osire implements a sophisticated, multi-tiered authentication architecture that is explicitly segmented into three distinct but interconnected tiers: the *User*, *Privileged*, and *Protected* tiers. This tiered protection system is not much unlike the standard for HPC developed by NIST,

<sup>6</sup> 

however with updates to permit the real-time architecture over batch architecture (Guo, et al., 2024).

The *User* tier encapsulates all data and services originating directly from individual users, including personal configuration files, locally executed applications, and decentralized services that are entirely managed by the user, notably the user-operated Gateway. Given the inherently decentralized and user-controlled nature of this resource, data and service requests from this tier are treated as inherently untrusted until properly authenticated. Authentication and subsequent verification of trust are exclusively managed through valid API keys issued, recognized, and validated by the centralized Key Manager. Furthermore, the User tier accommodates the development and testing of new services by individual users, allowing flexible and secure experimentation without risk of compromising the broader Osire infrastructure. By implementing strict authentication protocols, this tier effectively prevents unauthorized user-developed services from exploiting sensitive resources, accessing protected data, or otherwise extending their influence beyond clearly delineated and explicitly defined authorization boundaries.

The *Privileged* tier is explicitly allocated to globally accessible and officially sanctioned services that are fundamental to the operational functionality of Osire. This tier prominently features the central Service Registry, a critical component responsible for tracking and managing active global services available within the Osire ecosystem. Services residing within the Privileged tier are intentionally designed to be universally executable and readable, enabling streamlined user interactions, transparent resource discovery, and efficient utilization across diverse user contexts. Crucially, modifications to the Service Registry are strictly controlled; write permissions are exclusively restricted to formally approved and vetted global services. This stringent restriction ensures that unauthorized users or services cannot compromise or maliciously manipulate registry data, thus safeguarding the accuracy, consistency, and integrity of the registry. Moreover, the Privileged tier supports dynamic updates, allowing global services to register their operational states securely, enhancing real-time responsiveness and ensuring continuous, reliable availability of approved services.

The *Protected* tier is built in such a way as to replicate the design philosophy of OAuth2, but for entirely internal HPC systems (Hardt, 2012), and models standard AAA framework ideology (Panek, 2019). It is uniquely dedicated to housing the *SSO/Key Manager* and its associated secure database containing user-specific access keys. Unlike resources located in the Privileged tier, direct access to the access key database within the Protected tier is explicitly prohibited for general users and global services alike. Instead, global services interact with the Key Manager through carefully secured and authenticated communication channels specifically designed to facilitate safe and efficient validation of API keys without direct database access. The Key Manager itself is strategically segmented into multiple specialized operational components to optimize security and efficiency. These components include the *Key Manager Creator*, responsible for securely generating and distributing individualized access keys (via institutional SSO) linked explicitly to user identities, and the *Key Authenticator*, which efficiently and securely validates API keys for global service requests. This meticulous and deliberate structural segmentation substantially mitigates potential security vulnerabilities, minimizes authentication response latency, and significantly enhances the overall robustness and reliability of Osire's

authentication framework. Consequently, the Protected tier represents a critical security layer that underpins trust and operational integrity throughout the entire Osire architecture.

## 3.4 File Management

Effective file management is crucial within Osire for ensuring security, operational efficiency, and ease of maintenance. Osire employs an organized approach to file management that clearly separates user data, shared global resources, and authentication-related information, aligning with the platform's structure of User, Privileged, and Protected tiers.



Figure 3: Diagram of the File Management for usage and storage.

As depicted in Figure 3, Osire's file management framework revolves around three main repositories: Service Versions, Service Registry, and Auth Usage, along with an Access Keys database.

- Service Versions Repository: This repository tracks all versions of services deployed in Osire. It stores essential details like the service name, version number, storage location, and version status (latest or outdated). Actual service software is stored in containers, enabling straightforward version management and updates. When a user requests a service, the Osire gateway queries this repository to ensure the user receives the latest secure version of the requested service.
- Service Registry: The Service Registry acts as a centralized catalog, maintaining a list of all active global services. It records vital information such as service names, URLs, heartbeat intervals, and version numbers. Osire periodically logs service status through heartbeat signals to ensure up-to-date records, allowing dynamic management of service availability. Gateways refer to this registry to initiate or verify the availability of services, providing uninterrupted user access.

- Auth Usage Repository: Every user interaction within Osire is recorded in the Auth Usage repository. Logs include user institutional usernames, services accessed, service versions, and precise timestamps. This detailed logging is essential for monitoring and auditing, supporting efficient resource management, performance optimization, and maintaining robust security standards.
- Access Keys Database: Located securely in the Protected tier, this database contains JSON Web Tokens (JWT) linked explicitly to verified accounts. These tokens authenticate users, ensuring only authorized access to global and local services. Key generation, management, and verification are handled by the Key Manager, providing secure and reliable authentication processes.

In practical operation, Osire's gateways and services continuously interact with these repositories. When a service request is initiated, the gateway verifies the appropriate version from the Service Versions repository and confirms the service's operational status using the Service Registry. At the same time, user authentication and access logging are seamlessly conducted through the Auth Usage repository and Access Keys database. This integrated approach, clearly shown in Figure 3, maintains secure, efficient, and transparent file management operations, essential for reliable performance and administrative ease in academic HPC environments.



## 3.5 Novel-Service Pipeline

*Figure 4: Diagram of the control flow for building a novel service, and getting it deployed to the global services repository.* 

To support ongoing flexibility and growth, Osire utilizes a clearly defined yet user-friendly pipeline for adding new services, illustrated in Figure 4. This Novel Service Pipeline standardizes the transition of user-created local services into globally accessible tools, ensuring each addition meets strict quality and security standards.

Initially, users clone a provided template repository designed specifically to simplify initial setup and guarantee compatibility with Osire's framework. Users then develop and tailor their services to meet specific prototype objectives. With the service is linked to the local Osire environment, users can perform thorough local testing and make iterative improvements without affecting the broader HPC infrastructure.

When local testing concludes successfully, users submit their service for formal registration and review within Osire. This registration notifies administrators to conduct an initial compliance assessment, verifying the service aligns with technical and security guidelines. Administrators evaluate template adherence, API compatibility, resource efficiency, and security compliance at this preliminary stage.

Following this preliminary review, administrators conduct a more thorough evaluation, analyzing the service's overall reliability, security risks, code quality, and possible effects on HPC resource usage. If the service falls short of Osire's standards, administrators provide detailed feedback to users for necessary improvements and resubmission.

Services that pass this comprehensive vetting process are officially approved and incorporated into the Global Services Repository. Once included, these services become available to the entire Osire user base, enabling users to securely and conveniently create individual instances as needed. This structured pipeline supports continuous innovation and collaborative advancements within the Osire ecosystem, enhancing researchers' abilities to adapt and expand their AI-focused workflows.

## 4 Integration

The Osire platform was designed with a general architecture to ensure compatibility with a wide range of academic High-Performance Computing (HPC) systems. However, several specific implementation choices were influenced significantly by the unique characteristics and requirements of the Milwaukee School of Engineering's (MSOE) HPC environment. Key HPC design principles, such as clearly separating user-specific directories from shared public directories and the use of cluster-based computational resources, guided the foundational design of Osire. This approach allowed Osire to integrate effectively with standard academic HPC structures, ensuring broad usability and interoperability.

Despite its generalized design, some components of Osire required detailed customization due to administrative and operational constraints unique to MSOE. For instance, MSOE's HPC environment utilizes Slurm for scheduling computational jobs and Singularity containers for software deployment. These are chosen as the standard compared to other offerings such as Kubernetes, PBS Pro, Torque, HTCondor, and Docker as these are generally accepted at most academic institutions (Doosthosseini, Decker, Nolte, & Kunkel, 2024). These MSOE-specific technological choices strongly influenced Osire's implementation, emphasizing ease of use by minimizing administrative setup and ongoing management. By reducing the complexity of administrative tasks, Osire empowers researchers to independently manage and extend the platform, substantially decreasing reliance on cluster administrators.

Furthermore, Osire was specifically tailored to accommodate MSOE's capability for flexible and unrestricted access to computing resources. With these capabilities, Osire is able to create decentralized abstractions via the user-operated gateways situated externally to the main HPC

cluster. These gateways interact directly with structured authentication protocols and precise resource management mechanisms, balancing robust security with user flexibility. This design enables researchers to freely conduct interactive and iterative research within secure boundaries.

Another essential integration feature involves the adoption of OAuth-based authentication systems linked directly to institutional SSO, utilizing preconfigured MSOE resources. This authentication method provides a reliable and transparent verification mechanism, preventing unauthorized access without adding administrative complexity. Utilizing MSOE's institutional identity systems ensures streamlined user identification, access control, and comprehensive security management within everyday operations.

Additionally, Osire makes explicit assumptions about the underlying HPC system utilizing Open OnDemand (Hudak, et al., 2016), developed by the Ohio Supercomputer Center (OSC). This assumption is critical because Osire relies on OOD's ability to perform HTTP requests directly from the user interface, enabling seamless interaction with services hosted by Osire. The integration with OOD enhances Osire's capability to provide intuitive, web-based interfaces, greatly simplifying the user experience and promoting accessibility to HPC resources.

Osire's integration also makes use of MSOE's established storage practices and data-sharing policies. It leverages existing user-specific and shared directory structures, enhancing resource accessibility and promoting effective collaboration among researchers within secure and controlled environments. Adhering to these established data management practices helps maintain data integrity and privacy, effectively preventing unauthorized access.

Beyond technical integration, Osire's design aligns closely with the operational and cultural norms of MSOE's research community. It promotes long-term sustainability and independence by providing researchers the necessary tools and flexibility to manage their computational resources autonomously. The platform encourages collaborative innovation and knowledge sharing, fostering an active, community-driven research culture within MSOE.

## 5 Discussion and Future Work

The effectiveness and quick adoption of the Osire platform became clear during a hackathon hosted at the Milwaukee School of Engineering (MSOE). At this event, the Osire FastAPI Template was released and rapidly became an essential tool for nearly all participating teams. Its user-friendly and straightforward design significantly sped up project development during the hackathon. Furthermore, the template's value extended beyond the event itself, as many teams chose to continue using and customizing the Osire template for their ongoing research and development activities. This extensive adoption emphasizes Osire's capability to lower entry barriers to high-performance computing environments and illustrates its role in promoting sustained innovation and collaboration within the academic community.

Osire and its templates are currently publicly available, allowing institutions to incorporate them into their HPC systems. Moving forward, we plan to thoroughly analyze Osire's performance by testing various service setups and workloads. This analysis will focus on measuring container startup speeds, response times of APIs, system capacity under multiple simultaneous requests,

<sup>351</sup> 

and resource efficiency for different types of jobs and service arrangements. These assessments will help improve Osire's scalability and performance to better serve diverse academic users.

We also plan future analyses to understand which institutions use Open OnDemand and which do not, as well as to develop an alternative approach using SSH-based request protocols to accommodate institutions that lack Open OnDemand, which normally provides the HTTP protocols necessary to perform direct requests over the internet.

Additionally, we intend to partner with other academic institutions, with interest in LLM workflows and prototype development, to evaluate Osire's performance and adaptability in different HPC settings. Insights from these implementations will be documented and shared in future reports. Furthermore, we aim to expand the Osire platform by developing additional services, enhancing its functionality, and providing researchers with more specialized tools tailored for AI and LLM workflows in HPC environments.

## 6 Conclusion

We have presented Osire, a microservices platform designed to make high-performance computing more accessible for AI-focused academic research. Osire provides an effective balance between ease of use, security, and modularity, making it suitable for large language model workflows, which traditional HPC systems often struggle to support. The successful adoption and application of Osire components at a university hackathon confirmed its practical utility and appeal to student researchers. By simplifying complex HPC interactions through easy-to-use templates and APIs, Osire enables researchers and students to quickly create and deploy scalable solutions without needing extensive systems expertise. Ongoing performance evaluations and collaborations with multiple institutions will further demonstrate Osire's potential to support widespread AI research within the academic community.

## References

- Cohesity. (2021, September). Integration with Okta for Single Sign-On Capabilities. Retrieved from Cohesity: https://www.cohesity.com/resource-assets/white-paper/integration-with-okta-for-single-sign-on-capabilities-white-paper-en.pdf
- Doosthosseini, A., Decker, J., Nolte, H., & Kunkel, J. M. (2024, August 5). *Chat AI: A Seamless Slurm-Native Solution for HPC-Based Services*. Retrieved from ArXiv: https://arxiv.org/pdf/2407.00110
- Fan, L., Li, L., Ma, Z., Lee, S., Yu, H., & Hemphill, L. (2023, April 3). A Bibliometric Review of Large Language Models Research from 2017 to 2023. Retrieved from arXiv: https://arxiv.org/abs/2304.02020
- Glick, A. (2024, May 15). Unlocking Tomorrow: HPC and AI in Higher Education. Retrieved from Dell.com: https://www.dell.com/en-us/blog/unlocking-tomorrow-hpc-and-ai-in-higher-education/

- Guo, Y., Chandramouli, R., Wofford, L., Gregg, R., Key, G., Clark, A., . . . Farkas, C. (2024, February). *High-Performance Computing Security*. Retrieved from NIST: https://doi.org/10.6028/NIST.SP.800-223
- Haile, A. (2024, November 16). *Rosie FastAPI Template*. Retrieved from Github: https://github.com/MSOE-AI-Club/rosie-fastapi-template
- Haile, A., & Yoder, J. (2024, December 18). Building Democratized RAG Systems in High Performance Computing. Retrieved from Google Drive: https://drive.google.com/file/d/1TvUJU\_r4H6zoYZczTT3iytF\_9bufNRMo/view?usp=sh aring
- Hardt, D. (2012, October). *RFC 6749 The OAuth 2.0 Authorization Framework*. Retrieved from IETF: https://www.rfc-editor.org/rfc/pdfrfc/rfc6749.txt.pdf
- Hudak, D. E., Johnson, D., Nicklas, J., Franz, E., McMichael, B., & Gohar, B. (2016, July). Open OnDemand: Transforming Computational Science Through Omnidisciplinary Software Cyberinfrastructure. Retrieved from Open OnDemand: https://openondemand.org/sites/default/files/documents/ood-final.pdf
- Jones, M. B., Bradley, J., & Sakimura, N. (2015, May). JSON Web Token (JWT). Retrieved from IETF: https://www.rfc-editor.org/rfc/pdfrfc/rfc7519.txt.pdf
- Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C., . . . Stoica, I. (2023, September 12). Efficient Memory Management for Large Language Model Serving with PagedAttention. Retrieved from ArXiv: https://arxiv.org/pdf/2309.06180
- NVIDIA. (2024). *NVIDIA NIM for Large Language Models*. Retrieved from NVIDIA: https://docs.nvidia.com/nim/large-language-models/latest/introduction.html
- OpenAI. (2022, January 19). *OpenAI OpenAPI*. Retrieved from Github: https://github.com/openai/openai-openapi/blob/master/openapi.yaml
- OSC. (2014, July 9). *Storage Documentation*. Retrieved from Ohio Supercomputer Center: https://www.osc.edu/supercomputing/storage-environment-at-osc/available-file-systems
- Panek, C. (2019, November 12). Understanding authentication, authorization, and accounting. Retrieved from IEEE: https://doi.org/10.1002/9781119650737.ch2

# AI Driven Soccer Analysis Using Computer Vision

Adrian Manchado, Tanner Cellio, Jonathan Keane, Yiyang Wang Department of Computer Science and Software Engineering Milwaukee School of Engineering Milwaukee, WI 53202 {manchadoa, celliot, keanej, wangyi}@msoe.edu

March 22, 2025

#### Abstract

Sport analysis is crucial for team performance since it provides actionable data that can inform coaching decisions, improve player performance, and enhance team strategies. To analyze more complex features from game footage, a computer vision model can be used to identify and track key entities from the field. We propose the use of an object detection and tracking system to predict player positioning throughout the game. To translate this to positioning in relation to the field dimensions, we use a point prediction model to identify key points on the field and combine these with known field dimensions to extract actual distances. For the player-identification model, object detection models like YOLO and Faster R-CNN are evaluated on the accuracy of our custom video footage using multiple different evaluation metrics. The goal is to identify the best model for object identification to obtain the most accurate results when paired with SAM2 (Segment Anything Model 2) for segmentation and tracking. For the key point detection model, we use a CNN model to find consistent locations in the soccer field. Through homography, the positions of points and objects in the camera perspective will be transformed to a real-ground perspective. The segmented player masks from SAM2 are transformed from camera perspective to real-world field coordinates through homography, regardless of camera angle or movement. The transformed real-world coordinates can be used to calculate valuable tactical insights including player speed, distance covered, positioning heatmaps, and more complex team statistics, providing coaches and players with actionable performance data previously unavailable from standard video analysis.

## 1 Introduction

In sports analysis systems, precise statistics describing player performance are essential for understanding the game at a larger level. While professional teams can invest significant amounts of money into additional technology and sensors for tracking player performance, other teams may only have a camera for recording games, making the tracking of these statistics more challenging. With only camera data available, a computer vision solution becomes necessary to track players.

Beyond simply detecting and tracking players, to gain statistical insights such as player speed or distance traveled, the players' coordinate locations must be mapped to their physical locations on the field, so that the known field dimensions can be applied to the players positions to estimate distances. To address this problem, key point detection models can be used to take images as input and then predict where key locations are so that known distances can be linked to the images. Using point homography, we can translate points extracted from the images onto a twodimensional representation of the field, which can then be used for further postprocessing to collect player statistics.

Working with the Milwaukee School of Engineering (MSOE) men's soccer team, we proposed the creation of a computer vision system that combines these techniques to create a system that analyzes games using only video data. Due to a lack of labeled data, we leveraged pre-trained models such as YOLO and Faster R-CNN to identify players in a given frame. Using these predictions, we prompted Meta's SAM2 with predicted player locations to segment and track players throughout the game footage. To perform key point identification, we trained a custom CNN model on a small dataset of labeled field points to identify these key points on the field and then perform point homography on these identified points. For classifying players into their respective teams, we use clustering techniques as a lightweight method for dividing up players. Combining these components, we can create a video feed of a two-dimensional field with player positioning, separated by team, which can be used for analyzing larger trends about team behaviors and gaining insights from game footage.

With this paper we demonstrate the application of an AI system to create a 2D field representation from video frames in a completely unlabeled and raw dataset. This pattern of keypoint recognition, player detection, team classification, and homography is fairly common in sports analysis systems with AI. However, most of them are applied into a highly curated and labeled dataset. We wanted to replicate this pattern in data that had no kind of preprocessing, labels, or preparation for machine learning tasks. We show that it is possible to create an automated system that can derive insights from just footage of games. We believe we are paving the way for more flexible and scalable sports analytics solutions, available for the associations with limited resources.

## 2 Related Works

In recent years, advancements in object detection and tracking have significantly improved the ability to analyze player movement in sports applications. Our work leverages previous detection and segmentation models. This section discusses the key methodologies and models that informed our approach.

#### Segment Anything Model 2 (SAM2)

SAM2 is a transformer architecture with streaming memory for real-time video processing [Ravi et al., 2024]. It addresses challenges like occlusions, lighting variations, and temporal consistency by using a memory attention mechanism to store and utilize contextual information across frames. This enables SAM2 to retain object identities and improve segmentation predictions, making it highly effective for long-term tracking applications such as player detection in sports analytics.

#### You Only Look Once (YOLO)

YOLO (You Only Look Once) revolutionized object detection with its single-stage detection framework, treating the task as a regression problem [Redmon et al., 2016]. Unlike traditional models that generate region proposals, YOLO directly predicts bounding boxes and class probabilities from input images in real-time [Alkandary et al., 2025]. YOLOv5 uses CSPDarknet53 as its backbone and incorporates Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) for multi-scale detection [Alkandary et al., 2025]. YOLOv5 uses function for unbalanced data sets, and improved data augmentation [Jegham et al., 2024]. YOLOv11, the latest version, introduces innovations such as the C3k2 block and C2PSA for enhanced small-object detection and occlusion handling [Khanam and Hussain, 2024].

These versions were chosen for evaluation due to their advancements in real-time object detection. YOLOv5 is the most commonly used and is widely accessible, making it a reliable choice. YOLOv8 was selected for its direct improvements on YOLOv5 and positive performance in studies like "A Comparative Study of YOLO Series" [Alkandary et al., 2025]. YOLOv11 was included for its use of significantly less parameters while still offering enhanced precision and faster inference due to its state-of-the-art optimizations that are particularly useful for detecting players in complex scenes [Khanam and Hussain, 2024].

#### Faster R-CNN

Faster R-CNN is a two-stage object detection model that differs from YOLO's singlestage approach [Ren et al., 2016]. While YOLO predicts bounding boxes and class probabilities in one pass, Faster R-CNN first generates region proposals using a Region Proposal Network (RPN), followed by object classification and bounding box refinement [Ren et al., 2016]. The architecture includes a backbone for feature extraction, the RPN for generating proposals, and a detection head for classification and refinement [Ren et al., 2016]. We chose to evaluate Faster R-CNN to compare its two-stage approach with YOLO's single-stage detection, providing insights into the strengths and limitations of both frameworks.

Field registration is crucial for converting camera measurements into real-world distances. Traditional methods—such as sensor-based calibration, which fuses data from GPS, accelerometers, and gyroscopes, or template matching that aligns the current view with a stored 2D field representation—often struggle when the field is viewed from non-standard angles or under extreme variations. In soccer, where cameras frequently zoom and shift, these conventional techniques may fail to capture the dynamic nature of the field.

To overcome these challenges, our approach follows a typical sports field registration pipeline: (i) extracting field-specific features (in our case, keypoints), (ii) establishing correspondences between these features and a 2D field template, and (iii) estimating the homography matrix for mapping camera coordinates to real-world distances [Chu et al., 2022]. Recent advances in deep learning have shown that CNNs are particularly effective in this domain. For instance, [Zhang et al., 2024] introduced a cascaded CNN achieving 95.62% keypoint detection accuracy under challenging conditions, while [Falaleev and Chen, 2024] and [Theiner and Ewerth, 2022] demonstrated the benefits of CNN-based pipelines in enhancing 3D camera calibration and field registration accuracy. Furthermore, optimization-based approaches like PnLCalib and Bayesian frameworks proposed by [Gutiérrez-Pérez and Agudo, 2024], as well as [Claasen and de Villiers, 2024], have significantly improved homography estimation metrics. These developments highlight the power of integrating CNN-based keypoint detection with robust homography estimation to derive precise real-world metrics from raw, dynamic game footage.

## 3 Data

Our primary dataset consists of comprehensive footage from 10 home games during the 2024 season of the MSOE Men's Soccer team. It captures diverse environmental conditions including daytime, nighttime, and rainy weather scenarios. The video footage was recorded using a high quality elevated BePro camera, providing extensive field coverage while incorporating dynamic movement to track the ball and adjusting zoom levels as needed during gameplay.

#### 3.1 Player Bounding Boxes

To generate accurate labels for our dataset, we leveraged the Segment Anything Model (SAM2) for ground truth annotations on 390 consecutive frames. To accelerate the labeling process, we used YOLO predictions to get a inference point to pass as initial prompts to SAM2. While these YOLO predictions had to be manually reviewed, the predictions were often correct, minimizing time manually defining the prompt points for SAM2 when creating a ground truth dataset.

To create bounding box predictions that could then be used for evaluation against objection detection model bounding boxes, we reverse-engineered the SAM2-generated segmentation masks into ground truth bounding boxes using the minimum bounding box rectangle for each mask.<sup>1</sup>

#### 3.2 Field Key Points

Since the original dataset contained no keypoint labeled annotations, the first step in the research involved creating our own ground truth data to support model training and evaluation. We needed to identify and annotate key structural points on the soccer field that could serve as reference points for homography estimation.

We manually labeled key points across 146 frames extracted from soccer footage, ensuring a diverse representation of field views and camera angles. These frames were extracted from 2 different games, one where there was glare on the field (92 frames), and an overcast day (54 frames), giving us a small dataset that could be evaluated in different weather conditions. Each frame was annotated typically including 4 to 12 per frame depending on visibility An example of a labeled frame is shown in Figure 1 (right).

The selected keypoints were chosen to accurately align each frame with a top-down 2D field template. By focusing on intersections of prominent field markings—such as the penalty arc, center circle, and midfield line—these points serve as strong, consistent visual anchors. Their spatial distribution across the field supports stable homography estimation, while avoiding line extremities helps reduce annotation ambiguity. Additional non-intersecting points were selected to define a perpendicular axis at midfield, further facilitating the transformation. This strategy enables precise alignment for downstream tasks like player localization and tactical analysis.

 $<sup>^{1}</sup>$ One challenge presented by this data labeling approach was player entry/departure from the camera frame. This led to a reduction in ground truth points collected for specific players, but the process for dataset collection still allowed for an efficient collection of quality ground truth data



Figure 1: The set of all keypoints to predict in the field (left) and an example labeled frame from game footage (right).

## 4 Methodology

Since creating a soccer computer vision system is a complex task, we define several workflows to process the game footage to produce meaningful results that can be used for further game analysis, as seen in Figure 2. The main capabilities we need for a minimal system include: some form of player detection, a way to classify individual players/members of a team, key point predictions to identify the camera's current perspective, and point homography that can transform the frames of game footage onto a 2D representation of the field with physical field measurements. We discuss each of these components further below.



**Figure 2:** Full workflow for transforming video feed into a 2D representation of the feed. Core model pieces are defined with colors while different states of the data are defined in white.
### 4.1 Player Detection

Our approach to player detection and tracking combines an Object Detection model with SAM2 (Segment Anything Model 2). The Object Detector identifies soccer players in the first frame and generates associated bounding boxes. From it we can get the centerpoint of the bounding box for the identified players and use these points as prompts for SAM2, which segments and tracks the players throughout the rest of the video. By leveraging SAM2's unique memory component, we can maintain consistent tracking of individual players, assigning each a unique ID and ensuring continuity even in challenging conditions such as occlusions, players partially moving out of frame, low video resolution, lighting inconsistencies, and varying weather conditions.

The combination of an Object Detector Model and SAM2 offers several key advantages over traditional approaches like using DeepSORT [Wojke et al., 2017] for tracking. SAM2 provides pixel-accurate tracking and a fine-grained understanding of object shape and position, enabling more precise measurements. Its memory-based design is robust to occlusions and changes in player appearance. Unlike DeepSORT, which relies on continuous detection, SAM2 allows YOLO to be used only once for initialization—after which SAM2 handles the tracking, reducing computational overhead while improving consistency [Wojke et al., 2017].

Since an Object Detection model alone does not track players across frames, it is used to detect players in the first frame, and its detections serve as input to SAM2 for continuous tracking. These detection models lack ability to associate a detected player in one frame with the same player in the next, which makes them unsuitable for tracking. SAM2 addresses these issues by assigning unique IDs to each segmented player and leveraging memory to track them persistently throughout the video, ensuring robust and reliable player tracking across frames.

The data generated mentioned in section 3.1 will be used as ground truth to measure different Object Detection model performances.

### 4.2 Field Key Point Detection

### 4.2.1 Preprocessing

To prepare the raw images for training and evaluation of the key point prediction model, we developed a preprocessing pipeline tailored to enhance the visibility of field markings. A thresholding filter was applied to isolate white field lines. The resulting binary mask was then dilated using a morphological kernel to thicken and strengthen the detected field lines. This mask was applied to the original image, and all detected field markings were overwritten with bright white pixels to ensure



Figure 3: Keypoint prediction model workflow.

consistency across samples. The enhanced image was then converted to RGB, resized to  $710 \times 400$  pixels, to speed up neural network training, and normalized to [0, 1], as seen in the second image of Figure 3.

For key points, original points were processed so that points not in the given frame were imputed with a coordinate of (-1, -1). Additionally, key point coordinates were paired up with a status of their visibility denoted with 0 for points not in the frame and 1 for points in the frame.

### 4.2.2 Augmentation

To improve model robustness and dataset diversity, we applied horizontal flipping for data augmentation. Each frame was flipped and its keypoint annotations adjusted by inverting the normalized x-coordinates (1 - x) and swapping left/right labels. This encouraged the model to focus on field line patterns rather than extraneous features, effectively doubling the dataset while preserving spatial structure. The dataset was split 80/20 for training and testing.

### 4.2.3 Model Architecture

We designed a multi-task convolutional neural network to jointly predict the visibility and location of 12 predefined soccer field keypoints. The model outputs a visibility probability (between 0 and 1) and normalized coordinates (also in the range [0, 1]) for each keypoint. To manage keypoints that are not visible due to occlusion or camera angles, we employ a custom loss function that calculates the mean absolute error (MAE) only over visible keypoints (ignoring those labeled as (-1, -1)) and binary cross-entropy loss for visibility (defined in Equation 1). These losses are combined with weights of 10 for the coordinate loss and 1 for the visibility loss.

We trained this network using the Adam Optimizer with an exponential decay learning rate schedule. Additionally, to find a strong set of model hyperparameters, we perform a Bayesian Optimization hyperparameter search over factors including dropout rate, layer size, and learning rate. An example output of the model can be seen in the right side of Figure 3.

## 4.3 Homography

In terms of the homography, the real life dimensions of the field were collected with the use of Google Maps API. Combined with official NCAA collegiate soccer rules for measurements on the field, we defined all keypoint locations on the 2D field mapping. This image and dimensions were used to align predicted key points from each video frame with a standardized 2D representation of a soccer field; we estimated a homography transformation based on a set of corresponding points.

For the calculation of the homography matrix we used different algorithms. The most successful one was the Direct Linear Transformation (DLT), commonly used in the homography field [Kang et al., 2014]. The DLT estimates the homography matrix by solving a linear system built from point correspondences, using Singular Value Decomposition. It assumes all input points are accurate and does not account for outliers, making it highly sensitive to keypoint placement.

Once we obtained the homography matrix between the 3D and 2D view, any point in the image could be converted from one to another by applying the matrix to it. This gave us what we needed to jump from the camera perspective to a top down angle, in which real world metrics and measures can be taken.

### 4.4 Player Mapping and Team Recognition

Once we had both parts separate, player detection and homography, we tried to connect them and utilize clustering to classify each of the players into a team. For each person detected by the model, we computed the average color within a small patch ( $5 \times 5$  pixels) centered at the middle of the bounding box, ensuring robustness to noise and avoiding background influence. The resulting color vectors, represented in RGB space, were then clustered using the K-Means algorithm. We used k = 2 to

detect both teams involved in a soccer game. Each bounding box was then assigned to a cluster based on its color similarity, allowing us to group players by team without requiring manual labeling. This process served as a lightweight and effective method for unsupervised team classification based on visual appearance, as demonstrated in Figure 4.



Figure 4: Team assignment based on pixel color clustering of player bounding boxes. This example uses player detections from YOLOv8.

## 5 Results

### 5.1 Player Detection Model Performance

In evaluating different object detection models for identifying soccer players as prompts for SAM2, F1-score, IoU (Intersection over Union), recall, and precision each play a crucial role in determining effectiveness. Of these metrics, F1-score is the most valuable in our application, as it balances both precision and recall. Recall was deemed the next most critical metric since missed detections means that players will not be passed to SAM2 for segmentation, limiting tracking completeness. Precision, while important, is slightly less critical in our application because false positives outside the typical player size are unlikely to result in valid segmentation by SAM2 since we are filtering by the size of the mask. Finally, IoU is used to measure the alignment of the bounding box, but small misalignments are less problematic if the object is still successfully segmented. Based on these priorities and the comparative results shown in this results section, we have chosen to pursue YOLOv5x as the best model for our application. As seen in Table 1, it offers the highest F1-score (0.8451) while maintaining strong recall (0.7995), precision (0.8963), and IoU (0.7644), making it highly reliable for consistent player detection and optimal integration with SAM2. We evaluated four models: Faster R-CNN, YOLOv5x, YOLOv8x, and YOLOv11x. Each model was tested on its ability to generate accurate bounding boxes that identify players on the soccer field to serve as prompts to SAM2. Table 1 represents the results of these key performance metrics. Note that in each version, the "extra-large" model was tested to ensure a fair assessment.

Model	Mean IoU	Recall	Precision	F1-Score
Faster R-CNN	0.6837	0.6574	0.7942	0.7194
YOLOv11x	0.7934	0.6805	0.9229	0.7834
YOLOv8x	0.7466	0.8152	0.8765	0.8447
YOLOv5x	0.7644	0.7995	0.8963	0.8451

**Table 1:** Comparison of different object detection models on SAM2 generated<br/>ground truth samples.

YOLOv5 indicated the best balance between precision and recall. YOLOv8x followed closely behind with a slightly higher recall but slightly lower precision. YOLOv11x had the highest IoU and precision, but its recall was significantly lower, which can lead to missed detections, which is crucial in our application. All three YOLO models tested outperformed Faster R-CNN across every key metric, demonstrating the superior effectiveness of modern YOLO architectures in this application.

Using YOLOv5x, we successfully identified 17/22 players without any fine-tuning. These players initially identified were successfully segmented and tracked through SAM2, where their positions and movements are accurately monitored across sequential frames. Despite minor exceptions with player re-identification and identifying fans, the pipeline demonstrates promising performance. With potential YOLO fine-tuning, the tracking process can be further optimized, leading to a more robust and reliable player tracking system.

#### 5.2 Keypoint Prediction Model Performance

We defined tailored metrics for the evaluation of our CNN. For the evaluation of the key points prediction we used a masked Mean Absolute Error (MAE) as described in Equation 1, which computed the error only for visible keypoints, ensuring key points that are not in the frame (denoted by a special value) do not negatively impact the model's performance evaluation. This approach provides a more reliable assessment of localization accuracy by focusing on keypoints with valid ground truth data.

$$MAE_{masked} = \frac{1}{N} \sum_{i=1}^{N} \frac{\sum_{j=1}^{C} v_{i,j} (|x_{i,j} - \hat{x}_{i,j}| + |y_{i,j} - \hat{y}_{i,j}|)}{\sum_{j=1}^{C} v_{i,j}}$$
(1)

Meanwhile, accuracy was employed to evaluate visibility predictions by comparing the rounded model outputs to the ground truth labels as described in Equation 2. This metric is particularly useful in binary classification tasks, such as determining whether a keypoint is visible or not. Together, these metrics provide a comprehensive evaluation of both the spatial accuracy of the detected keypoints and the model's ability to correctly classify their visibility, ensuring robust and interpretable performance analysis.

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{C} \sum_{j=1}^{C} [round(\hat{y}_{i,j}) = y_{i,j}]$$
(2)

As shown in table 2, the proposed convolutional neural network demonstrated strong performance across both keypoint visibility classification and coordinate regression tasks. On the training set, the model achieved a visibility accuracy of 99.89%, with still high accuracy of 97.18% on the validation set, indicating strong generalization and minimal overfitting. For keypoint localization, the model achieved a mean absolute error (MAE) of 0.0107 in normalized image coordinates on the training set and 0.0138 on the validation set, corresponding to pixel errors of 5.96 and 7.65 pixels, respectively. These results suggest that the model is capable of precisely identifying and localizing field keypoints under varying visual conditions, providing a reliable foundation for downstream tasks such as homography estimation.

Metric	Training Set	Testing Set
Accuracy	0.9989	0.9718
MAE (% image)	0.0107	0.0138
MAE (pixels)	5.9648	7.6531

Table 2: Results of the keypoint prediction model on training and testing sets.

#### 5.3 Full System Analysis

In order to measure the performance of the system as a whole, the integration of the homography and keypoint detection, we calculated the MAE and projection errors with the actual data. Our MAE of the actual keypoints was 0.225 meters for the ground truth points and 0.26 meters for the predicted points. When comparing the homography of the predicted vs the ground truth, we obtained an average projection error of 0.499 meters in the keypoints transformed.

Looking at the results qualitatively, from a general perspective, there are several areas of improvement. As we can see in figure 5, our player detection model suffers from false positives, such as a ballboy being recognized or a referee. As well, the clustering



Figure 5: An example frame with player detection/keypoint predictions (left) and then applying homography to create a 2D field representation (right).

algorithm mistakes players when they are affected by the glare or shadows, assigning them to the opposite team. Furthermore, our keypoint detection model is not perfect, which can lead to innaccurate positions in the top down perspective or misleading stats. These are some of the challenges that we are facing at the moment, and that make the system not being ready yet.

## 6 Future Work

In future work, we plan to improve player tracking by addressing the problem of re-identification for players entering and leaving the frame. Using YOLO, we can maintain player counts and identify and track these players as they exit and re-enter the frame. When a player returns, we aim to re-prompt SAM2 using their previous YOLO coordinates, focusing only on re-prompting that player. By maintaining specific player attributes like the player's team and their exit location when leaving the frame, we can ensure consistent player identification, even after they leave and re-enter the field. Additionally, being able to track single players across the entirety of the game would allow for stat estimation at the player-level, rather than at the team-level as our system currently can do.

Additionally, for our trained parts of the system (specifically our key point detection model), we have only collected our dataset from game footage from MSOE's home field, potentially making our models overfit to the camera and field at MSOE. This makes the system less advantageous when there are away games that we want to analyze statistics from. To improve on this, we would want to augment our dataset with samples from other fields with different camera angles and backgrounds so the trained portions of our models generalize across all fields. As well, we will work on ball detection in order to create statistics related to the actual game flow.

## 7 Conclusion

In this work, we introduced our computer vision system for extracting a 2D field representation of players from soccer game footage. Because of a lack of data for more complex tasks like player detection, we leverage pretrained foundation models such as YOLO and SAM2 to classify players and segment the video based on the predicted player locations, respectively. We use a clustering approach based on predicted player bounding boxes to classify teams without needing to develop a supervised classifier. To connect predicted players and team affiliations to their corresponding physical distances, we train a custom CNN model on a limited dataset of key points, so we may predict these points and perform homography of known locations in frames to a consistent two-dimensional plane. Combining all these pieces, we can evaluate team statistics over the course of a game using only camera footage. While there are more statistics that can be computed based on these initial outputs our computer vision system generates, we hope that these initial results can begin to be used by the Milwaukee School of Engineering soccer teams to gain further insights about player activity during games.

## 8 Acknowledgments

The authors would like to thank coach Rob Harrington of the MSOE men's soccer team for providing the team with access to game footage from the Milwaukee School of Engineering men's soccer games and practices, which were used to create the data sets described above. The authors would also like to acknowledge Nicolas Picha, Joseph Loduca, Asher Harris, and Patrick Marshall from the MSOE men's soccer team for their collaboration with the authors in discussing this work.

## References

- K. Alkandary, A. S. Yildiz, and H. Meng. A comparative study of yolo series (v3-v10) with deepsort and strongsort: A real-time tracking performance study. *Electronics*, 14(5), 2025. ISSN 2079-9292. doi: 10.3390/electronics14050876. URL https://www.mdpi.com/2079-9292/14/5/876.
- Y.-J. Chu, J.-W. Su, K.-W. Hsiao, C.-Y. Lien, S.-H. Fan, M.-C. Hu, R.-R. Lee, C.-Y. Yao, and H.-K. Chu. Sports field registration via keypoints-aware label condition. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 3522–3529, 2022. doi: 10.1109/CVPRW56347.2022.00396.
- P. J. Claasen and J. P. de Villiers. Video-based sequential bayesian homography

estimation for soccer field registration. *Expert Systems with Applications*, 252: 124156, Oct. 2024. ISSN 0957-4174. doi: 10.1016/j.eswa.2024.124156. URL http://dx.doi.org/10.1016/j.eswa.2024.124156.

- N. S. Falaleev and R. Chen. Enhancing soccer camera calibration through keypoint exploitation. In Proceedings of the 7th ACM International Workshop on Multimedia Content Analysis in Sports, MM '24, page 65–73. ACM, Oct. 2024. doi: 10.1145/3689061.3689074. URL http://dx.doi.org/10.1145/3689061.3689074.
- M. Gutiérrez-Pérez and A. Agudo. Pnlcalib: Sports field registration via points and lines optimization, 2024. URL https://arxiv.org/abs/2404.08401.
- N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi. Evaluating the evolution of yolo (you only look once) models: A comprehensive benchmark study of yolo11 and its predecessors, 10 2024.
- L. Kang, L. Wu, Y. Wei, B. Yang, and H. Song. A highly accurate dense approach for homography estimation using modified differential evolution. Engineering Applications of Artificial Intelligence, 31:68-77, 2014. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2013.11.015. URL https://www.sciencedirect.com/science/article/pii/S0952197613002327. Special Issue: Advances in Evolutionary Optimization Based Image Processing.
- R. Khanam and M. Hussain. Yolov11: An overview of the key architectural enhancements, 2024. URL https://arxiv.org/abs/2410.17725.
- N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.
- J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection, 2016. URL https://arxiv.org/abs/1506.02640.
- S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. URL https://arxiv.org/abs/1506.01497.
- J. Theiner and R. Ewerth. Tvcalib: Camera calibration for sports field registration in soccer, 2022. URL https://arxiv.org/abs/2207.11709.
- N. Wojke, A. Bewley, and D. Paulus. Simple online and realtime tracking with a deep association metric, 2017. URL https://arxiv.org/abs/1703.07402.
- D. Zhang, C. Wu, J. Zhou, W. Zhang, Z. Lin, K. Polat, and F. Alenezi. Robust underwater image enhancement with cascaded multi-level sub-networks and triple attention mechanism. *Neural Networks*, 169:685–697, 2024. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2023.11.008. URL https://www.sciencedirect.com/science/article/pii/S0893608023006317.

# Winning is the Friends We Made Along the Way: Cooperative Reinforcement Learning

Alex Ewart\*, Mikhail Filippov\*, Peter Kwaterski\* and Sam Keyser

{ewarta, filippovm, kwaterskip, keysers}@msoe.edu
Department of Computer Science and Software Engineering,
Milwaukee School of Engineering,
1025 N Broadway, Milwaukee, WI 53202, United States
\*These authors contributed equally towards this research and are presented

in alphabetical order.

March 22, 2025

#### Abstract

In Reinforcement Learning (RL), models are often trained against each other with only one objective in mind: winning. Real life scenarios, however, aren't zero-sum games, where one's success is at the expense of another. Applications such as autonomous vehicles, for example, require far more cooperation as well as the ability to handle imperfect knowledge of the environment. To test these conditions, we present the cooperative trick-taking card game *The Crew*. Succeeding in *The Crew* requires cooperating with and compensating for other agents in a partially observable environment with only limited information sharing. As it correlates cooperation with victory, we believe the game is suitable for testing a reinforcement agent's ability to function in a team setting with restricted communication. In this work, we introduce an environment for *The Crew* compatible with the major multi-agent reinforcement learning (MARL) library Petting Zoo, and evaluate both search-based and DQN-based agents on their ability to succeed on cooperative tasks within the game. We present a set of benchmarks based on *The Crew* environment and demonstrate that both our heuristic solver and RL based agent achieve better than random performance.

## **1** Introduction

Reinforcement learning (RL) is a widely applicable subfield of artificial intelligence which trains agents to maximize cumulative rewards over some time horizon by taking actions in an environment. RL trains a model using a reward-based strategy, where the agent receives either positive or negative incentives based on the action in chooses to take in an environment. Reinforcement is broadly applicable in complicated domains where the goal is known but not necessarily the steps to implement. Recent examples of applications of reinforcement learning are its use to create grandmaster level game players [Vinyals, 2019], train reasoning models of large languages [DeepSeek-AI et al., 2025], and increase safety in self-driving cars [Lu et al., 2022].

Multi-agent reinforcement learning (MARL) is a domain within classical reinforcement learning that consists of multiple agents acting in the same environment. A key challenge is that agents need to decide how to act while taking into account potential decisions that other agents may make. This could mean attempting to predict what the other agents may do in order to win in an adversarial setting, or determining how to combine forces in collaborative settings to work towards a common objective. Although MARL is by no means a new field, training agents capable of acting and cooperating in a constrained environment remains a challenging task [Huh and Mohapatra, 2024].

In this paper, we introduce a novel environment based on the popular hidden information game *The Crew: The Quest for Planet Nine*<sup>1</sup> (*The Crew* hereafter), compatible with the Petting Zoo API [Terry et al., 2021], meant to train collaborative agents. The game is based on cooperative trick-taking where players must determine which cards each other has while only having a limited ability to communicate what is in their hand. The majority of information must be inferred from the way each player plays.



Figure 1: The Crew Gameplay

<sup>&</sup>lt;sup>1</sup>https://store.thamesandkosmos.com/products/the-crew

We train a Deep Q-Learning based agent [Mnih et al., 2013] on a simplified variant of the problem posed by the full environment where there is a *single agent* with *perfect information*. Even in the simplified variant, the space of possible games remains quite large so we introduce a method to warm start the agent's policy using monte carlo search trees. This approach has similarities to existing work for the game *Pommerman* [Yang et al., 2021]. The actions taken in Pommerman, much like in our environment, do not always lead to immediate results. This causes delayed rewards based on previous actions taken. Some key differences between *The Crew* environment and the Pommerman study is that, in our case, the agents are working together in a cooperative setting instead of competing against each other, in addition to having the ability to communicate small pieces of information to each other.

### 2 Methods

#### 2.1 Environment

The Crew is a cooperative trick-taking card game where players work together to complete missions by fulfilling specific tasks. There are 40 total cards available in the game, split between four suits (blue, green, pink, yellow) with card ranks 1 through 9, as well as a fifth suit (black) with card ranks 1 through 4. At the start of the game all cards are dealt out among the available players <sup>2</sup>, with the player who receives the black suited 4 being designated as the lead player. Players may not share what cards they have received with the rest of the table. After the cards are dealt, a number of *goals* are designated and randomly assigned to players at the table. Goals indicate that a specific player must "take" a specific card by winning the trick which that card is played in. Several additional conditions may be applied to goals, such as the stipulation that goal  $g_a$  must be completed before  $g_b$ , or that  $g_c$  must be the *n*th completed goal. These conditions may be composed together. Satisfying all goals results in a win. Failure to complete all of the goals is considered a loss.

Tricks are played sequentially starting from the lead player, and traveling clockwise, who plays one of the cards from their hand. That suit is designated as *trump* and all following players must play a card of the same suit if they are able. The greatest rank card of the trump suit will "take" the trick, with the winner becoming the new starting player. Cards taken in a trick are not playable thereafter. The only exception to this is if a black suited card is played when black is not already trump; black suited cards will always surpass any non-black suited trump. If multiple black suited cards are played, the highest rank takes. For example, if Alice leads with a green four and Bob does not have any green suited cards, he may play his black one. Bob will win the trick unless someone plays a higher ranked, black suited card.

Between tricks players may take the *Reveal* action once per game. Players are allowed to place one card c from their hand satisfying the following condition face up on the table:

- c is the highest ranked card of that suit in their hand
- c is the lowest ranked card of that suit in their hand
- c is the only card of that suit in their hand

<sup>&</sup>lt;sup>2</sup>the game can support between 2 and 5 players

They additionally place a token on the card indicating which of the three cases the card fulfills. The card is still playable.

#### 2.1.1 Modeling The Crew

*The Crew* can be formulated as a Partially Observable Markov Decision Process (POMDP). A POMDP is described by the tuple  $(S, A, T, R, \Omega, O, \gamma)$  where

- S is the set of states describing the environment which the agent interacts with
- A is the set of actions the agent can take
- *T* is the set of conditional transition probabilities between states
- $R: S \times A \rightarrow R$  specifies the immediate reward for taking
- $\Omega$  is the set of observations
- O is the set of conditional observation probabilities
- γ ∈ [0,1) is the discount factor, which controls the degree to which the agent cares about future rewards

At each time step within the environment, the agent receives  $s \in S$ . The agent then chooses some  $a \in A$  according to  $\pi(s)$ . Note that  $\pi \notin POMDP$ .  $\pi$  is known as the policy and represents the agent's decision making process as a probability distribution over the actions A given some state  $s \in S$ . The environment transitions to s' with probability T(s'|s, a) and the agent receives the reward r = R(s, a, s'). The agent also receives an observation  $o \in \Omega$  dependent on the next state s' and the action a with probability O(o|s', a). Note the distinction between observations o and states s. Because POMDPs explicitly model partially observable systems, there may be information in the state that the agent is not privy to. In the case of *The Crew* the agent does not know what cards are in other players hands.

The goal of a POMDP is for the agent to maximize the future, discounted rewards earned over time,  $E[\sum_{t=0}^{\inf} \gamma^t r_t]$ .

#### 2.1.2 Simplifying the Model

While *The Crew* is properly a POMDP, we chose to model it as a pure Markov Decision Process. We chose to model the problem as a MDP to simplify the initial task of creating an agent. A MDP is represented by the tuple  $(S, A, T, R, \gamma)$ . Note that the difference between a POMDP and a MDP is that the agent observes the full state so we do not need the concept of observations.

#### 2.1.3 State Space

We chose to model the state *S* as follows. We encode each player's hands, assigning each player a 40 element section in the state array. Each card in a player's hand is given a value which is then normalized by dividing it by the number of cards in the deck. The cards that are out of play are marked within the state array with a value of 1 indicating they have been played, and a 0 otherwise.

Each task is allocated six elements within the state array. The first element is a value of 0 or 1 to indicate if the task has been completed. Ownership of the task is

assigned as a 0 or 1 to indicate if it belongs to the current player. If the task has a numeric restriction the next element is assigned a 1 and 0 otherwise. The normalized index of the card and the normalized index of the player who owns the task are given in the next to elements. The final element is filled with the normalized value of the restriction if one exists. We encode space in the state for up to 16 tasks, which is more than ever specified in the game.

Information about the current trick is encoded via a dummy card with the value of the leading suit, played cards are encoded with their relative strength, their rank normalized by the highest rank in the trick. Metadata about the game is also encoded in our tensor such as what tasks have been completed, communication information (in our current implementation this remains unused due to perfect information), player indexes, and attempts on the mission. The metadata is encoded as such: the normalized index of the current player, 1 if the player can communicate, 0 otherwise, the number of mission attempts normalized by 3, and the number of completed tasks normalized by the total of tasks in the current mission.

#### 2.1.4 Action Space

The action space *A* is defined as all the cards that the player has in their hand. We chose not to include the *Reveal* action from the game due to our simplified environment giving the agent perfect knowledge. From the available actions, illegal actions such as not following suit when possible are masked so the agent is unable to select them.

#### 2.1.5 Petting Zoo

In the reinforcement learning (RL) world, there exist several libraries to efficiently build and manage agents and environments. One such specific library is Gym<sup>3</sup> [Brockman et al., 2016]. Gym provides a fairly straightforward API and is mostly utilized for single agent reinforcement learning. A multi-agent API, RLlib [Liang et al., 2018], already exists; however, with the way it is implemented, there are two main problems: Turn-based games require passing dummy actions for non-acting agents. A variable number of agents is very awkward to deal with for the environment. As such, in order to avoid such problems, our approach to make our game more universal and accessible was to use the PettingZoo API [Terry et al., 2021].

PettingZoo uses a Gym-like API to ensure support for environments with: large number of agents; agent death and creation; participation of different agents within each episode; and access to low level features for learning methods. Petting Zoo integrates cleanly with other reinforcement learning libraries like RLlib and CleanRL [Huang et al., 2021] which allows the use of existing agents in *The Crew* environment with minimal overhead.

#### 2.2 Agent

We created a Deep Q-Learning [Mnih et al., 2013] based agent for solving our simplified version of *The Crew*.

<sup>&</sup>lt;sup>3</sup>Since migrated to Gymnasium

Deep Q-Learning (DQN) is a value based reinforcement learning (RL) algorithm that extends traditional Q-learning by utilizing deep neural networks as function approximators. Introduced by Mnih et al., DQN enables agents to learn optimal policies in high-dimensional state spaces such as ours.

In value-based reinforcement learning, the agent learns to predict the long term expected value for state-action pairs by repeatedly estimating the Q-values, defined recursively for a given policy  $\pi$  as

$$Q_{\pi}(s,a) = \sum_{s' \in S} p(s'|a,s) \left[ r(s',a,s) + \gamma \sum_{a \in A} \pi(a|s') Q_{\pi}(s',a) \right]$$

for all *s* and *a*. The goal is then to update the policy to put high probability on actions with large *Q*-values, for example by using an  $\varepsilon$ -greedy approach where

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A|} & \text{if } a \in_{a \in A} \{Q_{\pi}(s, a)\} \\ \frac{\varepsilon}{|A|} & \text{otherwise.} \end{cases}$$

DQN replaces the Q-table with a deep neural network  $Q(s,a;\theta)$ , parameterized by  $\theta$ , to approximate the Q-values.

#### 2.3 Search Trees for *The Crew*

The Crew plays with 40 cards in five suits, four regular suits containing nine cards each, and a trump suit with four cards. In a four-player game of *The Crew*, a number we maintain during training, we estimate the worst-case number of permutations of possible games to be approximately  $1.6 \times 10^{23}$ . This estimation comes from the assumption that each player is dealt a regular suit in its entirety, plus a trump card. If the first player leads a non-trump card and all following players play their trump card, no player will be able to follow suit for the duration of the game, resulting in a total order  $9 \times \prod_{k=1}^{9} k^4$ . Only dealing four cards to each player, this number drops to 331,776.

A number this large motivates a method to reliably find a winning playthrough in a reasonable amount of time. Because of this, we developed a heuristic approach to, at any given state, generate a subset of potential actions for all players to take that our algorithm found likely to lead to a winning solution. This algorithm was used to generate tree structures containing the results of expanding through every subset of actions returned by the algorithm. This data was used to warm-start our agent's policy and to evaluate how well the heuristic algorithm performed.

#### 2.4 Training

We approached the problem of a multi-agent by first training with one set of weights given to all the players to have "one agent" controlling all of the players during a game. Once we began seeing results with that approach, we shifted to using a different set of weights for each of the players, using the set originally generated as a baseline for the other agents to expand on. Due to the scope and constraints of this research, all agents had perfect information on the hands of all the other players.

To gather training data for our model, we utilized a few different methods. One, we created trees that represented some of the possible states that the game could be



Figure 2: Training the agent with expert actions from the MCTS trees

in. Using those trees, we used the data to train the agent to imitate the tree. Another is we let the agent play games that we knew had a winnable state without trees. In our training, one episode was an entire hand, in other words, a complete set of tricks.

The latest attempt made to train our agent involved a step of pretraining the model using Monte Carlo Tree Searches [Nunes et al., 2018]. Due to the vast size of the action space even after filtering into trees, we chose to limit our trees to games where each player is only dealt four cards, in order to still be able to generate all tree permutations in a reasonable amount of time. The approach we took to pretraining trees focused on training our DQN to imitate the paths that we had solved to win the game. The way our rewards worked was for each trick that the agent(s) played, we would reward them the probability of winning the state i.e. how many of the children nodes to that trick were winning, assuming the agent(s) played a trick that we had generated. For the instances that the agent(s) played a trick that we did not account for, we simply reset them back to their previous legal node and let them play again from there without altering their reward. This approach is depicted in Figure 2.

After progressing past the imitation pretraining, we shifted to unassisted learning. In this stage, we shifted to using a more classical reward function. This function, instead of rewarding based on the probability of winning as dictated by a tree, was instead rewarded based on how good a specific play was. For example, if a task card was played on a trick and it is now the turn of the player who is assigned to win that



Figure 3: Training the agent with expert actions from the MCTS trees

task card plays a card to take the trick, the agent will be rewarded and if they fail to take it or another player takes the trick before or after, those agents will be punished. We also provided large rewards for winning and large punishments for losing with the idea that the agent(s) would learn that winning is very advantageous while losing causes them to lose most of the reward they gained during playing tricks. This approach is depicted in Figure 3.

As part of the training, we elected to go with a curriculum based approach. While using this method, we began by training the model on a mission with a single task. The idea was that starting with a single task would allow the DQN to learn what actions would allow for the completion of a task, and what the repercussions of taking specific cards were. From there we went to missions with two tasks. This was to help the agents to generalize to situations where multiple goals had to be considered as actions were chosen. The third category of missions trained on were two missions that could be completed in any order, and one that had to be completed after another. This was so the agents were able to learn how to complete the tasks in a specified order to receive a reward. After we shifted to using an imitation based learning approach to train the DQN, we implemented a new curriculum based structure to our training. In this new training methodology, we began by doing the imitation learning on trees with a depth of four. After experiencing significant results on performing on these trees, the curriculum moved onto trees of the full game. The next stage of training following imitation of the full trees was to shift to a more traditional reward function very similar to the one used in our first attempt at training the DQN. We followed a similar structure to the imitation stage of the curriculum, beginning with hands of four until experiencing results significantly better than random before migrating to full games. Once this training was completed, we then evaluated the agent on the trees to recognize if it could still perform after independent training. We replicated all of these methods in a multi-agent environment, assigning each player their own model weights, using the initial trained model as a branching off point as the initial weights for each player.

We evaluated our agent by having it play iterations of games that we know are solvable due to generating trees. We opted to evaluate both trees that the agents have seen before as well as trees that they have not. For the results, we looked at both total reward as well as the win rate that the agents experienced while playing through the game. In the evaluation stage, we determined wins and loses not based on if it was a winning or losing path in the tree, but on if the tasks were completed by the end. This was to ensure that even if the agent(s) won using a path we did not generate, that episode would still be evaluated as a win.

### **3** Results

We were only able to train a single agent on about 3,000 episodes. We saw minor improvements in performance, but we speculate ways to improve our results in the future work section. We were, however, able to obtain better-than-random results for our agent.

#### 3.1 Tree Generation

Of the 500 trees generated of depth four, the average score was 0.455. Over 150 trees had a score above 0.9, however. Most of these trees had very few nodes, as the initial player hands were conducive to a quick victory. In general, we noticed that higher scores were indicative of smaller trees.



Figure 4: Tree sizes of depth 4



Figure 5: Root node scores of depth 4

An example of a tree is shown below. As the trees themselves are too large to show, in this graph all nodes fall into bins with score less than or greater than 0.5. In this example, the failing nodes far outweigh the successful ones, but because trees are traversed by picking each path with equal weight and not the ending nodes with equal weight, we are more likely to end up at a successful node than a failing one, thus the root score is 0.65.



Figure 6: Example layout of a tree

#### 3.2 DQN Training Results

#### 3.2.1 Rewards during Training

The below graph shows rewards as a function of tricks during a portion of the training, where four tricks is equal to one episode. The blue line shows the reward adjusted for the root scores of the trees. It is important to account for this because we would expect a random player to obtain, on average, the same rewards at the root nodes. Ideally, the trend of the adjusted rewards would be greater than zero, but the training window may be too narrow to judge whether the rewards are actually improving.



Figure 7: Smoothed rewards compared to tricks played

#### 3.2.2 Performance compared to Heuristic Method

After training, we evaluated the agent on the same 500 trees that it was trained on. This is compared to the performance of the heuristic agent in red. The agent and heuristic approach are correlated, but this could be due to the tree scores as a confounding variable instead of the imitation learning working. That is, the easier a mission is, the more likely both methods are to succeed.



Figure 8: Agent success compared to heuristic success

### 3.2.3 Testing different Starting Hand Lengths

When evaluating starting conditions the agent hasn't seen, we tested 500 trials, dealing a different number of cards to each player for every batch. An unexpected result of this experiment was that the heuristic approach performed worse on shorter hands. We are unsure why this happened, but it could be due to randomly dealing a smaller number of cards to each player instead of starting from the regular ten cards per player and playing randomly until we get to our target. Smaller numbers of cards likely resulted in a higher likelihood of impossible scenarios.



Figure 9: Agent success compared to heuristic success

### 4 Future Work

Although our results show only minor improvements, they offer a proof-of-concept for warm starting the agent's policy on search trees. Many possibilities for extension of this work exist.

The agent itself could be improved. Tweaking the reward function and model hyperparameters would likely obtain more promising results, as well as modifying the underlying deep network architecture to include recurrent blocks [Hausknecht and Stone, 2017].

Extending the training to multiple distinct agents is also a promising possibility, as it may be necessary to develop an ensemble of player models rather than one model to fully foster cooperation. It would also be desirable to scale up the environment to the full POMDP formulation rather than the simpler MDP we used in this work.

### 5 Conclusion

The Crew remains a novel RL problem with a variety of factors contributing to a successful game. The size of the action and state spaces provides a promising environment with much potential for further study of MARL and POMDP, particularly due to the partially observable and collaborative nature of the actual game. Such ambiguity of the problem, when solved, will open the field to highly efficient MARL models that operate off of pure observation of their environments. Even though we have not yet experienced promising results using our agents, we believe this work to be useful because of the environment itself. Furthermore, the viability of porting The Crew into PettingZoo demonstrates the adaptive nature of external libraries to help with development of MARL environments, which has been a challenge in the community. The Crew is unique in that it allows for independent agents to communicate small bits of information (via revealing), which has strong applications to the real world where many agents, while not having perfect information, may still have ways to communicate small pieces of what they contain to other agents. We believe that this environment, combined with more time to tune an agent, can see promising results that perform at or above what a human would be able to do.

### References

- G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.
- DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, X. Zhang, X. Yu, Y. Wu, Z. F. Wu, Z. Gou, Z. Shao, Z. Li, Z. Gao, A. Liu, B. Xue, B. Wang, B. Wu, B. Feng, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, D. Dai, D. Chen, D. Ji, E. Li, F. Lin, F. Dai, F. Luo, G. Hao, G. Chen, G. Li, H. Zhang, H. Bao, H. Xu, H. Wang, H. Ding, H. Xin, H. Gao, H. Qu, H. Li, J. Guo, J. Li, J. Wang, J. Chen, J. Yuan, J. Qiu, J. Li, J. L. Cai, J. Ni, J. Liang, J. Chen, K. Dong, K. Hu, K. Gao, K. Guan, K. Huang, K. Yu, L. Wang, L. Zhang, L. Zhao, L. Wang, L. Zhang, L. Xu, L. Xia, M. Zhang, M. Zhang, M. Tang, M. Li, M. Wang, M. Li, N. Tian, P. Huang, P. Zhang, Q. Wang, Q. Chen, Q. Du, R. Ge, R. Zhang, R. Pan, R. Wang, R. J. Chen, R. L. Jin, R. Chen, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, S. S. Li, S. Zhou, S. Wu, S. Ye, T. Yun, T. Pei, T. Sun, T. Wang, W. Zeng, W. Zhao, W. Liu, W. Liang, W. Gao, W. Yu, W. Zhang, W. L. Xiao, W. An, X. Liu, X. Wang, X. Chen, X. Nie, X. Cheng, X. Liu, X. Xie, X. Liu, X. Yang, X. Li, X. Su, X. Lin, X. O. Li, X. Jin, X. Shen, X. Chen, X. Sun, X. Wang, X. Song, X. Zhou, X. Wang, X. Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. Zhang, Y. Xu, Y. Li, Y. Zhao, Y. Sun, Y. Wang, Y. Yu, Y. Zhang, Y. Shi, Y. Xiong, Y. He, Y. Piao, Y. Wang, Y. Tan, Y. Ma, Y. Liu, Y. Guo, Y. Ou, Y. Wang, Y. Gong, Y. Zou, Y. He, Y. Xiong, Y. Luo, Y. You, Y. Liu, Y. Zhou, Y. X. Zhu, Y. Xu, Y. Huang, Y. Li, Y. Zheng, Y. Zhu, Y. Ma, Y. Tang, Y. Zha, Y. Yan, Z. Z. Ren, Z. Ren, Z. Sha, Z. Fu, Z. Xu, Z. Xie, Z. Zhang, Z. Hao, Z. Ma, Z. Yan, Z. Wu, Z. Gu, Z. Zhu, Z. Liu, Z. Li, Z. Xie, Z. Song, Z. Pan, Z. Huang, Z. Xu, Z. Zhang, and Z. Zhang. Deepseek-r1: Incentivizing reasoning capability in Ilms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.
- M. Hausknecht and P. Stone. Deep recurrent q-learning for partially observable mdps, 2017. URL https://arxiv.org/abs/1507.06527.
- S. Huang, R. F. J. Dossa, C. Ye, and J. Braga. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms, 2021. URL https: //arxiv.org/abs/2111.08819.
- D. Huh and P. Mohapatra. Multi-agent reinforcement learning: A comprehensive survey, 2024. URL https://arxiv.org/abs/2312.10256.
- E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica. Rllib: Abstractions for distributed reinforcement learning, 2018. URL https://arxiv.org/abs/1712.09381.
- Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, B. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson, D. Anguelov, and S. Levine. Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios, 2022. URL https://arxiv.org/abs/2212.11419.

- V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning, 2013. URL https://arxiv.org/abs/1312.5602.
- C. Nunes, M. De Craene, H. Langet, O. Camara, and A. Jonsson. A monte carlo tree search approach to learning decision trees. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pages 429–435, 2018. doi: 10.1109/ICMLA.2018.00070.
- J. K. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. Santos, R. Perez, C. Horsch, C. Dieffendahl, N. L. Williams, Y. Lokesh, and P. Ravi. Pettingzoo: Gym for multi-agent reinforcement learning, 2021. URL https: //arxiv.org/abs/2009.14471.
- O. Vinyals. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575:350–354, 2019. doi: 10.1038/s41586-019-1724-z.
- H. Yang, S. Li, X. Xu, X. Liu, Z. Meng, and Y. Zhang. Efficient searching with mcts and imitation learning: A case study in pommerman. *IEEE Access*, 9:48851–48859, 2021. doi: 10.1109/ACCESS.2021.3061313.

# Transformers, Tumors, and Time: Improving Multi-Modal Image Registration Using Multi-Range Attention and Hierarchical Spatial Encoding

Caleb Gray, Alexander Neher, Sonia Grade, John Bukowy, Bart Gebka Department of Computer Science and Software Engineering

> Milwaukee School of Engineering Milwaukee, WI 53202

Peter LaViolette

Department of Radiology Medical College of Wisconsin Milwaukee, WI 53226

plaviole@mcw.edu

{grayc, nehera, grades, bukowy, gebkab}@msoe.edu

## Abstract

The American Cancer Society estimates two million new cancer diagnoses and 618,120 deaths in 2025. For prostate cancer, early detection yields a 5-year survival rate of 99.97%, but that rate worsens to approximately 34% when the cancer spreads. This research improves multi-modal image registration between histology slides and corresponding MRIs using multi-range attention and hierarchical feature encoding. Building on Ruchti et al.'s work, we propose a novel Hierarchical Multi-Range Attention Transformer architecture that integrates focal modulation as an alternative to self-attention for capturing long-range dependencies. Our method reduces the max distance of a predicted point in a slice by 66.36% (t=11.8940, p=0.0000). We also leverage data-distributed parallelism and hierarchical spatial encoding with cross-modality attention to improve runtime 50 times. These enhancements enable more accurate homologous point detection between image modalities and advance the potential for early, cross-modal cancer diagnosis.

# **1** Introduction

Cancer remains one of the leading causes of death worldwide, highlighting the need for improved diagnostic tools and strategies. Accurate early diagnosis is critical to guide treatment and improve patient outcomes. Magnetic Resonance Imaging (MRI) is a powerful noninvasive imaging modality that allows clinicians to visualize tumors, monitor disease progression, and detect recurrence. In contrast, histological analysis of biopsied tissue remains the gold standard for definitive cancer diagnosis [18]. Biopsy provides highly detailed information on tissue architecture and cellular morphology, enabling classification. However, biopsy is inherently invasive and has risks such as infection, bleeding, and discomfort in the patient. An ideal diagnostic framework would combine the detailed cellular information of histology with the non-invasive nature of MRI. Achieving this requires developing imaging techniques and computational models capable of extracting histology-like detail from noninvasive scans. This study investigates the use of various deep learning architectures to improve the registration of histology slides to MR Space to enable processing of high quality training data in MR space.

# 2 Related Work

## 2.1 Vision Transformer (ViT)

Transformers, originally designed for natural language processing, have been successfully adapted to computer vision tasks through the Vision Transformer (ViT) architecture [1]. ViTs segment an image into patches and treat each patch as a token, enabling the model to learn long-range dependencies and global context more effectively than traditional convolutional approaches. Recent adaptations of ViTs have demonstrated competitive or superior performance on classification and segmentation benchmarks across multiple domains, including medical imaging [2].

## 2.2 Image Classification in Medical Imaging

Deep learning has played an important role in image classification tasks within medical imaging. Convolutional Neural Networks (CNNs) have long been the foundation of imagebased diagnostics, particularly in cancer detection [3]. However, the transition toward transformer-based models has opened avenues for models that capture global relationships between anatomical structures — relationships that may be relevant in imaging modalities like MRI, and something that CNNs do not do. ViT variants have shown promise in improving classification accuracy and model interpretability in pathology and radiology [4].

## 2.3 Challenges in MRI-Based Classification

MRI presents unique challenges in classification tasks due to variability in acquisition protocols, intensity distributions, and spatial resolution. Additionally, the scarcity of data with pixelwise annotations and classifications makes training high-performing models difficult. Many studies have addressed these issues using data augmentation, domain adaptation, or synthetic data generation [6]. However, inconsistencies between MRI and histological ground truths often hinder model reliability and generalization [7]. In addition, nonquantitative MRI means that the voxels don't have absolute, replicable, values, thus making voxel values relative per image, and adding more complex relationships for models to learn.

Our work builds on this by applying point registration techniques to align and transfer information across modalities. This enables the generation of high-quality, histologically informed MRI datasets, potentially reducing reliance on invasive procedures while improving classification fidelity.

### 2.4 The Homologous Point Transformer

This work extends the homologous point prediction framework introduced by Ruchti et al. (2023), which demonstrated transformer-based homologous point estimation followed by thin plate spline registration as an effective strategy for multi-modality image alignment [8]. While following the same high-level pipeline, the proposed architecture introduces several technical advancements. The Hierarchical Multi-range Attention Transformer differs from Ruchti et al. in its use of a multi-stage Hierarchical Patch Embedding approach, whereas Ruchti et al. used single-level linear patch embeddings with CNNs. This architecture introduces Multi-Range Attention to explicitly model local, medium, and global spatial relationships, in contrast to Ruchti et al.'s use of LocalViT-style attention. Our proposed framework incorporates Rotary Positional Embeddings and Fourier Feature Mappings for spatial and point-based encoding, adding flexibility and precision absent in Ruchti's implementation. Furthermore, components like Gated Cross-Attention, Focal Modulation, and Feature Pyramid Networks are used to enhance cross-modal interactions and computational efficiency, representing a more modular and scalable design than Ruchti et al.'s unified transformer encoder block.

## 3 Data

The data used for this study was provided by the Medical College of Wisconsin (MCW) after being reviewed by their Internal Review Board. At the beginning of this particular study, all data was anonymized. The initial set provided consisted of 275 pairs of MRI and Histology slides as well as labeled control points between each to act as the ground truth. The histology slides are converted to grayscale and both images are set to  $\mathbb{R}^{512\times512}$ . One sample is shown in Figure 1.



Figure 1: This image is a grayscale histology slide and the points the pathologist labeled on it. In the right half of the image, there is an MRI with those same points translated to the MR domain

In the original data set, there were 275 total samples spread across 48 patients. The train test split was performed at the patient level to prevent target leakage and led to 225 samples in the train set, 12 samples in validation, and 38 samples in the test set. The visualizations of the samples by test set are shown below in Figure 2 (by image pair) and Figure 3 (by patient).



Figure 2: This is the split of slices within the dataset split



Figure 3: This is the split of patients within the dataset split

## 4 Methods

### 4.1 Model Architecture

The Hierarchical Multi-range Attention Transformer is a transformer-based model developed to predict homologous anatomical landmarks between histology and MRI images. The architecture introduces several components to effectively capture complex spatial correspondences between imaging modalities with differing resolutions and visual characteristics. While no individual component is novel, connecting hierarchical feature extraction with Multi-Range attention is novel to medical applications.

#### 4.1.1 Hierarchical Patch Embedding

To extract multi-scale features from input images, the model employs a Hierarchical Patch Embedding module in place of traditional patch embeddings based on the idea of a nested Vision Transformer [9]. This module processes images through multiple sequential stages, each consisting of convolution and down sampling layers. This hierarchical structure enables the capture of both local details and global structural patterns. While the methodology proposed in Ruchti et. al used down sampling and linear patch embeddings, this approach uses multi-scale simultaneous down sampling.

Given an input image  $x \in \mathbb{R}^{C \times H \times W}$ , where C, H, and W are the channel, height, and width dimensions respectively, the hierarchical embedding process can be formulated as:

$$x^{(1)} = \operatorname{Stage}_1(x) \in \mathbb{R}^{D/4 \times H \times W}$$
(1)

$$x^{(2)} = \operatorname{Stage}_2(x^{(1)}) \in \mathbb{R}^{D/2 \times H/2 \times W/2}$$
(2)

$$x^{(3)} = \operatorname{Stage}_{3}(x^{(2)}) \in \mathbb{R}^{D \times H/P \times W/P}$$
(3)

where P represents the final patch size and D is the embedding dimension.

#### 4.1.2 Multi-Range Attention

A Multi-Range Attention mechanism is integrated into the transformer blocks based on the work done by Xie et. al on super resolution [10]. Unlike standard self-attention, this mechanism operates over three spatial ranges: local, medium, and global. Local attention captures fine-scale information, medium-range attention models regional context, and global attention accounts for long-distance relationships.

For a given query Q, key K, and value V matrices, the multi-range attention is computed as:

Attention
$$(Q, K, V, M) = \operatorname{Softmax}\left(\frac{QK^T \odot M}{\sqrt{d_k}}\right) V$$
 (4)

where M is a masking matrix defining the attention range, and  $d_k$  is the dimension of the key vectors. The final multi-range attention output is a weighted combination where the weights were chosen empirically:

$$\begin{aligned} \text{MultiRangeAttn} &= 0.4 \cdot \text{Attention}(Q, K, V, M_{\text{local}}) \\ &+ 0.3 \cdot \text{Attention}(Q, K, V, M_{\text{medium}}) \\ &+ 0.3 \cdot \text{Attention}(Q, K, V, M_{\text{global}}) \end{aligned} \tag{5}$$

#### 4.1.3 Rotary Positional Embedding

The model incorporates Rotary Positional Embeddings (RoPE) to encode relative spatial positions directly within the attention mechanism [11]. For a 2D grid with coordinates (x, y), the RoPE applies the following transformation to query and key vectors:

$$\Theta_i(x,y) = x\omega_i^{(1)} + y\omega_i^{(2)} \tag{6}$$

$$\omega_i^{(j)} = 10000^{-2(i-1)/d}, \quad j \in \{1, 2\}$$
(7)

For each query and key vector with dimension d, the rotary embedding is applied as:

$$q_{\rm rot} = [q_1 \cos(\Theta_1) - q_2 \sin(\Theta_1), \ q_2 \cos(\Theta_1) + q_1 \sin(\Theta_1), \ \dots]$$
(8)

$$k_{\rm rot} = [k_1 \cos(\Theta_1) - k_2 \sin(\Theta_1), \ k_2 \cos(\Theta_1) + k_1 \sin(\Theta_1), \ \dots]$$
(9)

#### 4.1.4 Fourier-Based Point Embedding

Landmark coordinates are embedded using a Fourier Feature Mapping strategy [12]. This helps the model learn a more generalized understanding of the image translation process

between modalities. For normalized point coordinates  $p \in [-1, 1]^2$ , the Fourier embedding is:

$$\Phi(p) = [p, \sin(B \cdot p), \cos(B \cdot p)]$$
(10)

$$B = s \cdot \mathbf{B}, \quad \mathbf{B} \sim \mathcal{N}(0, \sigma^2) \tag{11}$$

where s is a learnable scaling factor, and  $\mathbf{B} \in \mathbb{R}^{m \times 2}$  contains m random frequency vectors.

#### 4.1.5 Gated Cross-Attention

A Gated Cross-Attention module facilitates information exchange between the histology and MRI branches of the network [13]. In our case, gated cross-attention will help keep the outputs scaled as well as keep from overfitting to a local minima. For histology tokens  $X_h$ and MRI tokens  $X_m$ , the cross-attention with gating is:

$$Attn_{h \to m} = Attention(X_h, X_m, X_m)$$
(12)

$$G_h = \sigma(W_g X_h) \tag{13}$$

$$X'_{h} = X_{h} + G_{h} \odot \operatorname{Attn}_{h \to m}$$
(14)

where  $\sigma$  is the sigmoid function,  $W_g$  is a learnable weight matrix, and  $\odot$  represents element-wise multiplication.

#### 4.1.6 Focal Modulation

To reduce the computational cost of global self-attention in selected layers, Focal Modulation is used. For input features X, the focal modulation process is:

$$Q = W_q X \tag{15}$$

$$K = W_k X \tag{16}$$

$$V = W_v X \tag{17}$$

The context features are computed through sequential convolutions at different scales:

$$C_0 = K \tag{18}$$

$$C_i = \text{Conv}_i(C_{i-1}), \quad i \in \{1, 2, \dots, L\}$$
 (19)

$$C = \sum_{i=0}^{L} C_i \tag{20}$$

The final modulated output is:

$$Y = \operatorname{GELU}(C) \odot V \tag{21}$$

the focal modulation layer helps reduce computational costs and is more easily scaled. This will also pair with the other changes made to a standard architecture [14].

<sup>390</sup> 

#### 4.1.7 Feature Pyramid Network

A Feature Pyramid Network (FPN) is integrated to combine feature maps across transformer layers [15]. The goal of this layer is to maintain different levels of details for all tiers of features in the system, covering the gap between histology and MR detail. For features  $\{F_1, F_2, \ldots, F_L\}$  from L layers, the FPN computes:

$$P_L = \operatorname{Conv}_{\operatorname{lat}}(F_L) \tag{22}$$

$$P_{i} = \text{Conv}_{\text{lat}}(F_{i}) + \text{Conv}_{\text{up}}(P_{i+1}), \quad i \in \{L-1, L-2, \dots, 1\}$$
(23)

The final output combines features from all levels:

$$Y_{\text{FPN}} = \text{Fusion}([P_1, P_2, \dots, P_L])$$
(24)

### 4.2 Memory-Efficient Implementation

Several architectural optimizations are incorporated to support memory-efficient training on high-resolution images:

• Gradient Checkpointing recomputes intermediate activations during the backward pass to minimize memory usage:

$$\nabla_{\theta}L = \nabla_{y}L \cdot \nabla_{\theta}f(x;\theta) \tag{25}$$

• Chunked Attention divides the attention computation into smaller segments:

$$\operatorname{Attn}_{\operatorname{chunk}}(Q_{i:j}, K, V) = \operatorname{Softmax}\left(\frac{Q_{i:j}K^T}{\sqrt{d_k}}\right)V$$
(26)

• Layer Scaling introduces learnable scaling parameters:

$$Y = X + \gamma \cdot \operatorname{Block}(X) \tag{27}$$

## 4.3 Training Procedure

The training process incorporates multiple strategies to optimize performance and maintain training stability including learning rate scheduling and gradient clipping [16][17]:

• Gradient Accumulation enables larger effective batch sizes:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{1}{N} \sum_{i=1}^N \nabla_\theta L_i(\theta_t)$$
(28)

```
391
```

• **Mixed Precision Training** uses lower-precision arithmetic where possible, with a dynamic loss scaling factor *s*:

$$\nabla_{\theta}^{\text{FP16}} = s \cdot \nabla_{\theta}^{\text{FP32}} \tag{29}$$

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\nabla_{\theta}^{\text{FP16}}}{s} \tag{30}$$

• Adaptive Learning Rate Scheduling with warmup and cosine decay:

$$\eta_t = \begin{cases} \eta_{\text{base}} \cdot \frac{t}{T_{\text{warmup}}}, & \text{if } t \le T_{\text{warmup}} \\ \eta_{\min} + \frac{1}{2} (\eta_{\text{base}} - \eta_{\min}) (1 + \cos(\pi \frac{t - T_{\text{warmup}}}{T - T_{\text{warmup}}})), & \text{otherwise} \end{cases}$$
(31)

• Dynamic Gradient Clipping adjusts the clipping threshold:

$$\nabla_{\theta}^{\text{clipped}} = \text{clip}(\nabla_{\theta}, -c \cdot L, c \cdot L) \tag{32}$$

#### 4.4 Comparison with Base Architecture

The Hierarchical Multi-Range Attention Transformer differs from the baseline Overlay Transformer by Ruchti et al. architecture in several key aspects:

- Attention Mechanism: Three-tier attention vs. two-path attention.
- Patch Processing: Multi-scale simultaneous vs. progressive down sampling.
- Attention Computation: Full attention with optimizations vs. block-based memory efficiency.
- Memory Optimization: Mixed strategies vs. block processing only.
- **Point Prediction Refinement**: Focal modulation and gated MLPs vs. residual refinement heads.

## 5 Results

When run on the same dataset split the same way, the hMRAT architecture performs better on average with a MCPD of 49.95 as opposed to 66.64. A paired t-test has a t=0.9929 and a p=0.3272 meaning we fail to reject the null hypothesis in favor of the alternative. To further test this, we ran a Wilcoxon signed rank test which reported a W=246.0000 and a p=0.0719. This means for mean values, we once again fail to reject the null hypothesis. The interpretation of this is that there is no statistically measureable difference on the mean.



Figure 4: A comparison of the previous architecture Ruchti et al and The hMRAT

However, if we measure the max control point deviation to compare the results, we get very different results. When comparing the max distance between points, the results of a Paired t-test are a test statistic of t=11.8930 with a p=0.0000 and the Wilcoxon signed-rank test with W=3.0000 and p=0.0000. In both cases, we have enough evidence to reject the null hypothesis in favor of the alternative.



Figure 5: A comparison of the previous architecture and the hMRAT on the Max distance of each point.

Qualitatively, the points points align in such a way where the model demonstrates some linear and non-linear transformations such as rotation and deforming edged to nominate the correct mask. The matches are shown in the figure below. The predicted points (blue) are

nearly identical to the ground truth points in the image and are nearly all within the range of interobserver variability.



Figure 6: A qualitative evaluation of the model. Passing in the Fixed Image, Moving Image, and moving points produced the output points. The distance between the output points and ground truth are visualized as a red line

## 6 Discussion & Analysis

The experimental results demonstrate that the Hierarchical Multi-Range Attention Transformer (hMRAT) outperforms the Overlay Transformer architecture proposed by Ruchti et al. in accuracy. These improvements are attributable to the architectural innovations introduced in this work—most notably, the incorporation of hierarchical patch embeddings, multi-range attention, and a more modular design incorporating focal modulation, Fourierbased point encoding, and gated cross-attention. Based on the results generated, our models performance on the average-case has improved on the SOTA architecture, but comes with the trade-off of a higher variance.

## 6.1 Improved Spatial Correspondence and Generalization

A key strength of the proposed model lies in its ability to model spatial correspondences at multiple resolutions. Hierarchical patch embeddings provide richer feature abstractions that preserve fine-grained details from the histology domain while maintaining broader contextual awareness of MRI structures. This is crucial given the stark visual and textural differences between modalities. The introduction of multi-range attention further enhances this capability by explicitly modeling short-, medium-, and long-range dependencies across each input modality. As a result, hMRAT achieves lower maximum control point deviation (MCPD), indicating more accurate homologous point predictions and, by extension, more anatomically consistent registrations. However, as shown in the results section, there is a higher variance in predictions.

We hypothesize that the Fourier-based point embeddings and gated cross-attention layers contributed to this improved generalization by aligning feature spaces while dynamically modulating information flow between modalities.

## 7 Future Work

The proposed model demonstrates a reduced memory footprint, increased computational efficiency, and low maximum control point deviation, indicating its suitability for high-resolution medical imaging tasks. Future work will focus on leveraging the high-quality learned representations to support downstream applications such as cancer classification in MRI images. Given the model's demonstrated flexibility and generalization capability, particularly in integrating diverse spatial patterns across modalities, it is anticipated that additional imaging modalities or domain-specific transformations can be incorporated through targeted fine-tuning. Extending the architecture to multi-task learning scenarios, including segmentation and diagnosis across modalities, also represents a promising direction for further development.

Another promising direction involves investigating the model's ability to perform unsupervised or weakly supervised alignment in settings where paired training data is limited or unavailable. Developing robust alignment techniques under limited supervision could significantly enhance the model's ability to generate high quality training data. The goal for this data would be to take the new annotations in MR space and create a predictive model that can classify cancer in MR. Additionally, integrating anatomical priors or leveraging structural representations such as graphs or meshes could further refine landmark prediction and facilitate anatomically consistent transformations across patients and modalities.

## 8 Conclusion

This work presents the Hierarchical Multi-Range Attention Transformer (hMRAT), a novel architecture for homologous point prediction between histology and MRI image pairs. By integrating hierarchical spatial encoding, multi-scale attention, and memory-efficient training strategies, our model advances the state of the art in multi-modal medical image registration. The introduction of focal modulation, rotary positional encoding, gated cross-attention, and Fourier-based point embeddings enables hMRAT to capture complex anatomical correspondences with improved precision and efficiency. Compared to baseline architectures, hMRAT demonstrates a significant reduction in maximum control point deviation and a substantial improvement in training speed.

The ability to align high-resolution histology with MRI not only enhances cross-modal interpretability but also lays the foundation for developing non-invasive diagnostics informed by cellular-level detail. As medical imaging moves toward more integrated and intelligent systems, models like hMRAT that bridge modality gaps will become increasingly essential. Our results underscore the promise of transformer-based architectures in medical imaging and open pathways for future research in multi-task learning, weak supervision, and anatomically informed modeling.
# References

- [1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv preprint arXiv:2010.11929, 2021. https://arxiv.org/abs/2010.11929
- [2] Emerald U. Henry, Onyeka Emebob, and Conrad Asotie Omonhinmin. Vision Transformers in Medical Imaging: A Review. arXiv preprint arXiv:2211.10043, 2022. https://arxiv.org/abs/2211.10043
- [3] J. R. Pérez-Núñez, C. Rodríguez, L. J. Vásquez-Serpa, and C. Navarro. The Challenge of Deep Learning for the Prevention and Automatic Diagnosis of Breast Cancer: A Systematic Review. Diagnostics (Basel, Switzerland), 14(24):2896, 2024. https: //doi.org/10.3390/diagnostics14242896
- [4] S. Takahashi, Y. Sakaguchi, N. Kouno, K. Takasawa, K. Ishizu, Y. Akagi, R. Aoyama, N. Teraya, A. Bolatkan, N. Shinkai, H. Machino, K. Kobayashi, K. Asada, M. Komatsu, S. Kaneko, M. Sugiyama, and R. Hamamoto. *Comparison of Vision Transformers and Convolutional Neural Networks in Medical Image Analysis: A Systematic Review*. Journal of Medical Systems, 48(1):84, 2024. https://doi.org/ 10.1007/s10916-024-02105-8
- [5] R. J. Chen, T. Ding, M. Y. Lu, et al. Towards a general-purpose foundation model for computational pathology. Nature Medicine, 30:850–862, 2024. https://doi. org/10.1038/s41591-024-02857-3
- [6] Vasileios C. Pezoulas, Dimitrios I. Zaridis, Eugenia Mylona, Christos Androutsos, Kosmas Apostolidis, Nikolaos S. Tachos, and Dimitrios I. Fotiadis. Synthetic data generation methods in healthcare: A review on open-source tools and methods. Computational and Structural Biotechnology Journal, 23:2892–2910, 2024. ISSN 2001-0370. https://doi.org/10.1016/j.csbj.2024.07.005
- [7] Q. D. Strotzer, T. Wagner, P. Angstwurm, K. Hense, L. Scheuermeyer, E. Noeva, J. Dinkel, C. Stroszczynski, C. Fellner, M. J. Riemenschneider, K. Rosengarth, T. Pukrop, I. Wiesinger, C. Wendl, and A. Schicho. *Limited capability of MRI radiomics to predict primary tumor histology of brain metastases in external validation*. Neuro-oncology Advances, 6(1):vdae060, 2024. https://doi.org/10.1093/ noajnl/vdae060
- [8] A. Ruchti, A. Neuwirth, A. K. Lowman, S. R. Duenweg, P. S. LaViolette, and J. D. Bukowy. *Homologous point transformer for multi-modality prostate image registration.* PeerJ Computer Science, 8:e1155, 2022. https://doi.org/10.7717/ peerj-cs.1155

- [9] Yuang Liu, Zhiheng Qiu, and Xiaokai Qin. *Nested-TNT: Hierarchical Vision Transformers with Multi-Scale Feature Processing*. arXiv preprint arXiv:2404.13434, 2024. https://arxiv.org/abs/2404.13434
- [10] Chengxing Xie, Xiaoming Zhang, Linze Li, Yuqian Fu, Biao Gong, Tianrui Li, and Kai Zhang. MAT: Multi-Range Attention Transformer for Efficient Image Super-Resolution. IEEE Transactions on Circuits and Systems for Video Technology, 2025, pages 1–1. http://dx.doi.org/10.1109/TCSVT.2025.3553135
- [11] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. arXiv preprint arXiv:2104.09864, 2023. https://arxiv.org/abs/2104.09864
- [12] Ermo Hua, Che Jiang, Xingtai Lv, Kaiyan Zhang, Ning Ding, Youbang Sun, Biqing Qi, Yuchen Fan, Xuekai Zhu, and Bowen Zhou. Fourier Position Embedding: Enhancing Attention's Periodic Extension for Length Generalization. arXiv preprint arXiv:2412.17739, 2025. https://arxiv.org/abs/2412.17739
- [13] Xiaogang Jia, Songlei Jian, Yusong Tan, Yonggang Che, Wei Chen, and Zhengfa Liang. Gated Cross-Attention Network for Depth Completion. arXiv preprint arXiv:2309.16301, 2024. https://arxiv.org/abs/2309.16301
- [14] Jianwei Yang, Chunyuan Li, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Focal Modulation Networks. arXiv preprint arXiv:2203.11926, 2022. https://arxiv.org/ abs/2203.11926
- [15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. *Feature Pyramid Networks for Object Detection*. arXiv preprint arXiv:1612.03144, 2017. https://arxiv.org/abs/1612.03144
- [16] Zhen Xu, Andrew M. Dai, Jonas Kemp, and Luke Metz. Learning an Adaptive Learning Rate Schedule. arXiv preprint arXiv:1909.09712, 2019. https: //arxiv.org/abs/1909.09712
- [17] Noah Marshall, Ke Liang Xiao, Atish Agarwala, and Elliot Paquette. To Clip or not to Clip: the Dynamics of SGD with Gradient Clipping in High-Dimensions. arXiv preprint arXiv:2406.11733, 2024. https://arxiv.org/abs/2406.11733
- [18] Tseng, L.J., Matsuyama, A., and MacDonald-Dickinson, V. (2023). Histology: The gold standard for diagnosis? *Canadian Veterinary Journal*, **64**(4), 389–391. PMID: 37008634; PMCID: PMC10031787.

# Beyond Boundaries: Building a General Latent Space for Medical Imaging with Self-Supervised Learning

Alexander Neher, Alexander Drobek, Alhagie Boye, Sonia Grade, Autumn Mizer, Caleb Gray, Bart Gebka

Department of Computer Science and Software Engineering Milwaukee School of Engineering Milwaukee, WI 53202

{nehera, drobeka, boyea, grades, mizera, grayc, gebkab}@msoe.edu

# Abstract

Self-supervised learning presents a promising avenue for medical imaging, particularly in contexts where labeled data is limited. In this study, we explore the effectiveness of a multimodal vision transformer pretrained using a DINOv2-style framework on a grayscale-only dataset of medical images, including MRI and histology scans. The training corpus was diverse in modality but lacked exposure to color, high-frequency texture, and other features common in downstream tasks. When evaluated on benchmarks such as PathMNIST, DermaMNIST, and BloodMNIST, the model demonstrated partial generalization, but consistently underperformed relative to a baseline DINOv2 model pretrained on a large corpus of millions of images. In some cases, performance degradation was substantial, particularly for tasks reliant on color cues. Visualizations of the latent space revealed loosely structured clustering by modality, though the diagnostic value of these representations remains uncertain. Overall, our findings suggest that while domain-specific self-supervision may offer limited cross-task generalization, significant challenges remain in achieving parity with general-purpose models trained on richly diverse datasets. Further work is needed to determine whether incorporating additional modalities, resolution, or supervision can close this performance gap.

# **1** Introduction

In 2024, nearly 300,000 new cases of prostate cancer and over 35,000 related deaths were reported in the United States alone [22]. Early detection significantly influences outcomes: between 2013 and 2019, the 5-year relative survival rate exceeded 99% for localized cases but dropped to 34% for distant metastases [15]. Medical imaging remains a key tool in the early detection and management of such diseases, particularly in oncology.

The introduction of the Transformer architecture in \*Attention Is All You Need\* [23] led to widespread adoption of attention-based models in both language and vision tasks. Vision Transformers (ViTs), in particular, have shown strong performance on large-scale benchmarks like ImageNet-1K by leveraging tokenized patch embeddings and self-attention to model long-range dependencies. However, ViTs often struggle to resolve local features without architectural modifications [12]. Techniques like local cropping and positional embeddings have been proposed to address these limitations [16].

DINOv2, introduced by Meta and FAIR, incorporates self-supervised learning into vision transformers via a multi-crop strategy and a student-teacher training loop, enabling the model to learn semantically meaningful representations from unlabeled images [16]. Previous work by Mahmood et al. demonstrated the feasibility of this approach in medical imaging, showing that large-scale pretraining on histology slides could yield a latent space that clusters organ types and modalities without supervision [8].

In this study, we explore whether similar benefits can be realized from a broader but less representative set of grayscale medical images. Using DINOv2, we pretrain a vision transformer on a corpus containing MRI, histology, and CT-derived scans, and evaluate its downstream performance on color and texture-rich medical classification benchmarks. Unlike prior efforts focused on narrow modalities or supervised labels, our approach aims to understand the limitations of cross-modality generalization when training data lacks color, annotation, and high-frequency diversity.

Rather than aiming to outperform existing models, we examine how much utility—if any—can be extracted from a self-supervised model trained on such constrained data. Our goal is to characterize the extent to which domain-specific self-supervision can support transfer learning under real-world limitations, and to surface failure cases that highlight the boundaries of current approaches.

# 2 Related Work

The introduction of Vision Transformers (ViTs) has led to the development of increasingly general-purpose visual models capable of strong performance across classification, detection, and segmentation tasks. More recently, large multimodal models have extended these capabilities by combining visual and textual inputs into unified latent spaces [25]. These advances have been enabled by massive datasets and compute resources, which are often out of reach for many medical AI applications.

In the medical imaging domain, several works have explored the use of self-supervised learning to build transferable visual representations. For instance, Ruchti et al. proposed a dual-pathway network that aligned MRI and histology representations, demonstrating that cross-modal learning could support more robust image registration. Similarly, the TITAN model constructed a shared latent space across whole slide images (WSIs) and clinical text to distinguish organ types without labeled supervision [8]. These approaches highlight the potential of self-supervision to encode high-level clinical semantics across modalities.

A notable contribution in this space comes from the Mayo Clinic's \*Novel Pathology Foundation Model\*, which demonstrated that larger models trained on diverse, multiresolution WSI datasets perform well across multiple downstream tasks [4]. Their results suggest that scale—in both data diversity and model size—plays a crucial role in building robust and generalizable medical models.

However, much of this prior work relies on access to high-resolution, color-rich WSIs or multimodal inputs that may not be available in many real-world settings. While some studies note that meaningful representations can emerge without pixel-perfect detail, the threshold at which generalization breaks down remains unclear.

Our work builds on these foundations but adopts a more constrained setting. Rather than assuming access to richly annotated, color, or multimodal datasets, we investigate whether a general latent space can emerge from self-supervised training on grayscale-only medical images of varying modalities and quality. Unlike TITAN or the Mayo Clinic model, which benefit from tightly curated or multi-source data pipelines, we intentionally explore the limitations of coarsely pretraining on publicly available grayscale datasets. This allows us to better understand the boundaries of generalization when data diversity and visual fidelity are limited—an increasingly relevant question for institutions operating under resource constraints.

## **3** Dataset

The capacity of a foundational model to generalize across tasks is heavily influenced by the composition of its training data [4]. In this study, we adopted a two-phase approach: coarse pretraining on a large, unlabeled corpus of grayscale medical images, followed by fine-tuning and evaluation on benchmark classification datasets.

#### 3.1 Coarse Pretraining Data

For the pretraining phase, we compiled approximately two million images from a range of publicly available sources. These included MRI scans, histopathology slides, and CT-derived images, with many drawn from datasets released by the National Cancer Institute's Clinical Proteomic Tumor Analysis Consortium (CPTAC) and similar initiatives [18, 5, 19, 17, 13, 1, 7, 14, 6, 11, 20, 2, 3, 10, 9, 21]. Figure 1 summarizes the distribution of data sources across institutions and imaging modalities.



Figure 1: Distribution of image sources in the coarse pretraining dataset. The dataset integrates multiple imaging modalities and institutions, but lacks color and many texture-rich visual cues common in natural images.

All images were resized to 512×512 pixels and normalized. We applied a set of standard augmentations—horizontal flipping, local and global cropping, color jitter, grayscale conversion, Gaussian blur, and solarization—to increase view diversity during self-supervised training. However, because many of the source images were already grayscale, some of these transformations had limited visual effect. Figure 2 shows example augmentations applied to representative images.



Figure 2: Image augmentations used during DINOv2-style pretraining. While effective in increasing view diversity, many transformations (e.g., color jitter, grayscale conversion) had minimal impact due to the lack of color channels in the source data.

While diverse in modality and anatomy, this dataset is notably constrained in key areas: it is grayscale-only, lacks natural image statistics, and may underrepresent certain imaging protocols, devices, or pathologies. These limitations frame the scope of our inquiry—how far can self-supervised models go when trained on visually restricted yet domain-relevant data?

### **3.2** Downstream Evaluation Datasets

To evaluate generalization, we fine-tuned the pretrained model on three classification benchmarks from the MedMNIST v2 collection: PathMNIST, BloodMNIST, and DermaMNIST [24]. These datasets vary in modality, resolution, and class balance, providing a useful proxy for downstream clinical tasks.

PathMNIST consists of color images of colorectal cancer tissue, making it particularly challenging for a model pretrained without exposure to color or staining artifacts (Figure 3). BloodMNIST (Figure 4) contains microscope images of blood cells, requiring fine-grained morphological discrimination. DermaMNIST (Figure 5) includes dermatoscopic images of skin lesions, again rich in color and texture absent from our training set.



Figure 3: Sample images from PathMNIST. Each image is 224x224x3 and represents a colorectal cancer tissue patch across nine classes.



Figure 4: Examples of blood cell types in BloodMNIST. The dataset includes eight classes and focuses on microscopic morphological features.



Figure 5: Sample dermatoscopic images from DermaMNIST. These images represent seven dermatological conditions, posing a generalization challenge for grayscale-pretrained models.

Figures 6, 7, and 8 show the class distributions across each dataset. While PathM-NIST is relatively balanced, both BloodMNIST and DermaMNIST exhibit moderate class imbalances, adding further difficulty to transfer learning under label scarcity.



Figure 6: PathMNIST class distribution. Tissue types are evenly represented, allowing controlled evaluation of performance across histological categories.



Figure 7: BloodMNIST class distribution. Minor imbalance is present across cell types, which may affect classification robustness.





Taken together, the coarse pretraining set and downstream evaluation tasks allow us to probe a core question: Can self-supervised learning on limited grayscale medical data yield representations that generalize meaningfully to richer and more varied clinical imaging domains?

# 4 Methods

### 4.1 Pretraining with DINOv2

We employ DINOv2, a self-supervised vision transformer model based on the ViTL14 architecture, to perform coarse pretraining on a large corpus of unlabeled medical images. DINOv2 is designed to learn robust visual representations without reliance on manual annotations, making it well-suited for the medical imaging domain, where labeled data is often scarce or expensive to obtain [16].

The model is trained using a self-distillation framework in which a student network learns to mimic the outputs of a momentum-updated teacher network. Both networks receive differently augmented views of the same image, encouraging the student to learn view-invariant, semantically meaningful representations. DINOv2 captures both global and local features through hierarchical patch-based attention mechanisms, enabling it to generalize across heterogeneous medical modalities, including histology and MRI.

#### 4.2 Downstream Classification

To evaluate the effectiveness of the pretrained model, we fine-tune it on supervised classification tasks using datasets from the MedMNIST v2 collection, which includes PathM-NIST, BloodMNIST, and DermaMNIST [24]. For each task, a linear classification head is attached to the frozen backbone of the pretrained DINOv2 model. We then compare performance against a baseline DINOv2 model pretrained on ImageNet-1K. The process for image classification is shown below. This structure is similar to ours, the process that changes is the training process using self-supervised learning.



Figure 9: Overview of the vision transformer architecture used in a traditional vision transformer classifier. Each image is divided into fixed-size patches, which are linearly embedded and augmented with positional encodings. These embeddings are then processed through a stack of transformer encoder layers. For classification, a dedicated classification token is appended to the input sequence and its final representation is passed to a linear head.

This demonstrates that self-supervised pretraining on domain-specific data yields more

transferable features than those learned from natural images.

# 5 Results and Analysis

## 5.1 Analysis of Self-Supervised Latent Space

To assess the learned representations, we apply t-distributed stochastic neighbor embedding (t-SNE) to the output feature vectors of the pretrained model. While some clustering is apparent across imaging modalities, it is difficult to determine whether these separations capture semantically meaningful or diagnostically relevant differences. The visual boundaries between clusters are less distinct than those reported in previous work on large, color-rich datasets, suggesting potential limits in the learned latent space's capacity to disentangle complex medical features.



Figure 10: t-SNE visualizations of the learned latent space from the pretrained DINOv2 model. Each point represents a single image embedding, colored by its dataset of origin. While general clustering is present, the quality and separability of these clusters vary across modalities and do not clearly align with diagnostic boundaries.

### 5.2 Feature Attribution and Attention Visualization

We examine attention maps to qualitatively understand which regions the model attends to during classification. In a chest X-ray example, the model pretrained on medical images appears to focus on clinically relevant structures such as lung fields and cardiomediastinal contours. However, the differences compared to the ImageNet baseline are often subtle and inconsistent across images. While some examples show improved localization, others exhibit diffuse or anatomically imprecise attention. These mixed results make it difficult to conclude whether medical-domain pretraining consistently improves interpretability or diagnostic focus.



Figure 11: Qualitative comparison of attention maps. Left: Original chest X-ray input. Center: PCA from our DINOv2 model pretrained on medical images. Right: PCA from a DINOv2 model pretrained on ImageNet. While some medically relevant focus is observed in the self-supervised model, overall improvements in localization are inconsistent.

### 5.3 Benchmark Comparisons in Context

When evaluated on classification tasks from the MedMNIST suite, the self-supervised model underperforms relative to the ImageNet baseline across all benchmarks. This gap is especially prominent in tasks that depend on color and fine texture cues, which were absent in the grayscale pretraining dataset.

- **PathMNIST:** Our model achieved 39.65% accuracy compared to the ImageNet baseline's 73.30%. Given the dataset's rich color staining patterns and high-resolution textures, this large gap highlights the limitations of grayscale-only pretraining in capturing necessary visual cues for histopathological classification.
- **DermaMNIST:** With 69.05% accuracy versus 74.25% from the baseline, the model showed modest generalization, although the absence of dermatoscopic structures during pretraining may explain the remaining performance difference.
- **BloodMNIST:** The model reached 73.15% accuracy, trailing the baseline's 79.70%. This dataset's relatively lower visual complexity may have helped narrow the gap, though the result still indicates incomplete transferability.

These findings suggest that while some generalization is possible, performance degradation remains substantial in most cases. The results underscore the importance of diverse pretraining data—including color and texture-rich images—for tasks requiring detailed visual discrimination. Without such diversity, the latent space learned through selfsupervision may fail to capture domain-specific features critical to medical diagnosis.

# 6 Discussion

Our findings highlight both the potential and limitations of self-supervised learning for building general-purpose representations in medical imaging. While the model did exhibit some capacity for internal organization—such as modest modality-wise clustering in the t-SNE projections—the strength and clinical utility of these representations remain uncertain. In particular, it is unclear to what extent the observed clusters reflect meaningful semantic structure versus superficial visual similarities in grayscale input.

Although some qualitative results, such as attention visualizations, suggest that the model may attend to clinically relevant regions in certain cases, this behavior was inconsistent and difficult to generalize across tasks. The lack of clear, reliable improvement over a baseline model pretrained on natural images indicates that domain-specific pretraining on grayscale medical data alone may not yield a universally transferable latent space.

The main contribution of this work is to surface the boundary conditions under which medical self-supervision breaks down. Despite training on nearly two million images, the model underperformed on every evaluated benchmark relative to the ImageNet baseline. The drop in accuracy—particularly for color- and texture-dependent datasets like PathM-NIST—suggests that critical diagnostic cues were simply not present in the training data. This limitation reinforces the importance of diversity not just in anatomical modality, but also in visual complexity, resolution, and color information.

Our experimental setup also reveals challenges in generalization that are often overlooked in idealized research conditions. The dataset, while large, was constructed from publicly available sources and lacked detailed metadata about acquisition protocols, equipment, or patient demographics. These uncontrolled variables likely introduced bias and noise, making it harder for the model to extract consistent, domain-agnostic features.

Additionally, our focus was limited to classification tasks. While this provides a useful entry point for evaluating generalization, it does not assess how well the learned representations transfer to more complex applications like segmentation, registration, or report generation—domains where spatial precision and contextual reasoning are critical.

This work should therefore be interpreted as a cautionary case study rather than a proof of concept. It illustrates the difficulty of building generalizable medical representations in the absence of color, supervision, or carefully curated training distributions. Future research will need to systematically test whether adding modalities, improving pretraining objectives, or incorporating weak supervision can overcome these limitations.

In summary, while self-supervised learning on grayscale medical data can yield partially structured representations, these do not consistently generalize to downstream clinical tasks. Our results underscore the need for more nuanced understanding of what types of visual and semantic variation are essential to learn robust, transferable latent spaces in medical imaging.

### 7 Future Work

This study raises several open questions about the conditions under which self-supervised learning can produce transferable representations for medical imaging. While our approach was intentionally constrained, future directions should explore whether expanded data scope, alternative learning objectives, or cross-modal integration can meaningfully improve generalization.

One important area of investigation involves dataset diversity. Our training corpus,

<sup>409</sup> 

while large in image count, lacked critical visual information such as color channels, finegrained textural variation, and metadata annotations. Future work should examine whether including color-rich medical images, such as full-resolution histopathology slides or dermatoscopic scans, leads to more robust and semantically aligned latent spaces. Similarly, expanding to underrepresented modalities—such as ultrasound, PET, or functional imaging—may reveal modality-specific challenges or benefits in self-supervised learning.

Another direction is to revisit the learning paradigm itself. The DINOv2-style studentteacher framework offers strong inductive biases for natural image understanding, but it is unclear whether such architectures are optimal for the types of spatial and semantic patterns found in medical data. Incorporating contrastive objectives (e.g., SimCLR, MoCo), masked image modeling, or hybrid approaches may yield representations that better reflect diagnostic structure, especially when combined with modest amounts of weak supervision or domain labels.

Cross-modal representation learning remains a promising but largely untested extension. Combining medical images with associated text—such as pathology reports or radiology notes—may help ground the visual latent space in clinically relevant semantics. Leveraging models like CLIP or BioViL in a medical context could improve interpretability and open the door to zero-shot inference or report generation.

Temporal data also remains underexplored. Longitudinal studies, progression tracking, and multi-slice sequences offer rich structure that could be captured through recurrent or transformer-based temporal models. Evaluating the extent to which self-supervised models can encode temporal consistency or disease evolution could yield insights for chronic condition monitoring or predictive modeling.

Finally, evaluation itself must go beyond classification. Future studies should assess performance on segmentation, registration, anomaly detection, and few-shot learning tasks. These tasks may provide a more sensitive test of whether the learned representations capture domain-specific knowledge—or whether they simply reflect generic spatial priors learned from grayscale imagery.

Ultimately, advancing self-supervised learning in medical imaging will require a shift from benchmarking performance alone to understanding the deeper structural properties of the learned representations, and how they align—or fail to align—with real clinical needs.

# 8 Conclusion

This study investigates the extent to which self-supervised learning (SSL) on grayscale medical images can yield transferable visual representations. Using the DINOv2 framework, we pretrained a vision transformer on a diverse—but visually constrained—set of unlabeled medical images and evaluated its generalization ability across classification tasks from the MedMNIST benchmark suite.

While the model demonstrated limited clustering and moderate performance in some downstream tasks, it consistently underperformed compared to a baseline pretrained on ImageNet. These results suggest that, under conditions of reduced color and texture diversity, the capacity of self-supervised models to generalize across modalities is significantly constrained. The latent representations learned in our setting showed some anatomical organization, but their clinical utility remains uncertain.

Our primary contributions are as follows:

- An evaluation of constrained self-supervision, assessing how far DINOv2-style models can generalize when trained exclusively on grayscale medical data.
- A characterization of failure modes, identifying where such models fall short on benchmarks that require rich color and texture-based reasoning.
- A framing of dataset limitations, highlighting the trade-offs between dataset scale, visual diversity, and generalization capacity in medical imaging representation learning.

Rather than positioning this work as a demonstration of success, we offer it as a boundary case—an empirical exploration of what self-supervised learning cannot yet reliably accomplish in medical imaging without more representative input data. The findings underscore the need for broader modality coverage, inclusion of color and multi-channel data, and rethinking architectural choices or learning objectives to align better with clinical complexity.

As interest grows in developing foundation models for healthcare, this study emphasizes the importance of evaluating such models not only by their promise, but also by the limits of their assumptions. Scaling medical AI will require not just bigger models, but smarter choices about data, supervision, and what "generalization" really means in clinical contexts.

# References

- [1] National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC). *The Clinical Proteomic Tumor Analysis Consortium Clear Cell Renal Cell Carcinoma Collection (CPTAC-CCRCC) (Version 13) [Data set]*. The Cancer Imaging Archive. 2018. DOI: https://doi.org/10.7937/k9/tcia.2018. oblamn27.
- [2] National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC). *The Clinical Proteomic Tumor Analysis Consortium Cutaneous Melanoma Collection (CPTAC-CM) (Version 11) [Data set].* The Cancer Imaging Archive. 2018. DOI: https://doi.org/10.7937/K9/TCIA.2018.ODU24GZE.
- [3] Madabhushi A. and Feldman M. *Fused Radiology-Pathology Prostate Dataset (Prostate Fused-MRI-Pathology)*. The Cancer Imaging Archive. 2016. DOI: 10.7937/k9/ TCIA.2016.tlpmr1am.
- [4] Maximilian Alber et al. Atlas: A Novel Pathology Foundation Model by Mayo Clinic, Charité, and Aignostics. 2025. arXiv: 2501.05409 [cs.CV]. URL: https: //arxiv.org/abs/2501.05409.
- [5] Hugo G. D. et al. *Data from 4D Lung Imaging of NSCLC Patients (Version 2) [Data set]*. The Cancer Imaging Archive. 2016. DOI: https://doi.org/10.7937/K9/TCIA.2016.ELN8YGLE.

- [6] Mackin D. et al. Data From Credence Cartridge Radiomics Phantom CT Scans (CC-Radiomics-Phantom) [Data set]. The Cancer Imaging Archive. 2017. DOI: https: //doi.org/10.7937/K9/TCIA.2017.zuzrml5b.
- [7] Newitt D. and Hylton N. Single site breast DCE-MRI data and segmentations from patients undergoing neoadjuvant chemotherapy (Version 3) [Data set]. The Cancer Imaging Archive. 2016. DOI: https://doi.org/10.7937/K9/TCIA. 2016.QHsyhJKy.
- [8] Tong Ding et al. Multimodal Whole Slide Foundation Model for Pathology. 2024. arXiv: 2411.19666 [eess.IV]. URL: https://arxiv.org/abs/2411. 19666.
- [9] Thompson R. F. et al. Stress-Testing Pelvic Autosegmentation Algorithms Using Anatomical Edge Cases (Prostate Anatomical Edge Cases) (Version 1) [Data set]. The Cancer Imaging Archive. 2023. DOI: https://doi.org/10.7937/ QSTF-ST65.
- [10] Litjens G., Futterer J., and Huisman H. *Data From Prostate-3T*. The Cancer Imaging Archive. 2015. DOI: 10.7937/K9/TCIA.2015.QJTV5IL5.
- [11] Shafiq ul Hassan M. et al. Credence Cartridge Radiomics Phantom CT Scans with Controlled Scanning Approach. The Cancer Imaging Archive. 2018. DOI: http: //doi.org/10.7937/TCIA.2019.4124tz5g.
- [12] Dinh Phu Cuong Le, Dong Wang, and Viet-Tuan Le. "A Comprehensive Survey of Recent Transformers in Image, Video and Diffusion Models". In: *Computers, Materials and Continua* 80.1 (2024), pp. 37–60. ISSN: 1546-2218. DOI: https: //doi.org/10.32604/cmc.2024.050790. URL: https://www. sciencedirect.com/science/article/pii/S1546221824005150.
- [13] Bloch B. N., Jain A., and Jaffe C. C. BREAST-DIAGNOSIS [Data set]. The Cancer Imaging Archive. 2015. DOI: http://doi.org/10.7937/K9/TCIA.2015. SDNRQXXR.
- [14] Heller N. et al. *Data from C4KC-KiTS [Data set]*. The Cancer Imaging Archive. 2019. DOI: 10.7937/TCIA.2019.IX49E8NX.
- [15] Onyinyechukwu Okolie et al. "Reactive astrocytes potentiate tumor aggressiveness in a murine glioma resection and recurrence model". en. In: *Neuro Oncol* 18.12 (June 2016), pp. 1622–1633.
- [16] Maxime Oquab et al. DINOv2: Learning Robust Visual Features without Supervision. 2024. arXiv: 2304.07193 [cs.CV]. URL: https://arxiv.org/ abs/2304.07193.
- [17] Madhavi P, Patel S, and Tsao A. S. *Data from Anti-PD-1 Immunotherapy Lung* [*Data set*]. The Cancer Imaging Archive. 2019. DOI: 10.7937/tcia.2019. zjjwb9ip.
- [18] Kinahan P. et al. Data from ACRIN 6668 Trial NSCLC-FDG-PET (Version 2) [Data set]. The Cancer Imaging Archive. 2019. DOI: https://doi.org/10.7937/tcia.2019.30ilqfcl.

- [19] Kinahan P. et al. Data from ACRIN-FLT-Breast (Version 2) [Data set]. The Cancer Imaging Archive. 2017. DOI: https://doi.org/10.7937/K9/TCIA. 2017.0120zmxg.
- [20] Ger R. et al. Data from CT Phantom Scans for Head, Chest, and Controlled Protocols on 100 Scanners (CC-Radiomics-Phantom-3) [Data set]. The Cancer Imaging Archive. 2019. DOI: https://doi.org/10.7937/tcia.2019. j71i4fah.
- [21] Sawyer-Lee R. et al. Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM) [Data set]. The Cancer Imaging Archive. 2016. DOI: https://doi.org/10.7937/K9/TCIA.2016.7002S9CY.
- [22] American Cancer Society. Cancer Facts and Figures. 2024. URL: https:// www.cancer.org/content/dam/cancer-org/research/cancerfacts-and-statistics/annual-cancer-facts-and-figures/ 2024/2024-cancer-facts-and-figures-acs.pdf.
- [23] Ashish Vaswani et al. "Attention is all you need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 9781510860964.
- [24] Jiancheng Yang et al. "MedMNIST v2-A large-scale lightweight benchmark for 2D and 3D biomedical image classification". In: *Scientific Data* 10.1 (2023), p. 41.
- [25] Yiwu Zhong et al. AIM: Adaptive Inference of Multi-Modal LLMs via Token Merging and Pruning. 2024. arXiv: 2412.03248 [cs.CV]. URL: https://arxiv. org/abs/2412.03248.



# SwarmRAG: A Distributed AI Framework for Explainable Environmental Modeling

Brett Storoe, Sydney Balboni, Thomas Benzshawel, John Cisler, Ben Paulson\*

Department of Computer Science and Software Engineering

Milwaukee School of Engineering

1025 N Broadway St, Milwaukee, WI 53202

storoeb, balbonis, benzshawelt, cislerj, paulsonb@msoe.edu

# Abstract

As climate change accelerates, precise environmental modeling is essential for sustainable resource management. This research presents SwarmRAG, a distributed AI framework that enhances environmental simulations by employing a swarm of specialized large language models (LLMs). Unlike traditional CNN-based approaches, which excel in feature extraction but often lack interpretability, SwarmRAG inherently provides structured explainability at multiple levels. Each LLM is responsible for a localized segment of an environmental dataset, offering fine-grained insights that can be queried and traced back to specific data sources. SwarmRAG leverages retrieval-augmented generation (RAG) and memory modules to ensure both historical continuity and real-time adaptability while maintaining a coherent global understanding through a meta-coordinator model. Our framework is applied to glacier modeling within the Nilus Project, optimizing ice stupa growth to improve water conservation efforts. By integrating geospatial data, historical weather patterns, and hydrological models, SwarmRAG enables scalable, real-time environmental analysis with inherent explainability, fostering data-driven decision-making for climate resilience and resource sustainability.



# **1** Introduction

Environmental modeling plays a pivotal role in understanding and managing the Earth's climate and resources. As climate change accelerates, accurate environmental models are critical for predicting extreme events and informing adaptation strategies [1]. These models guide real-time decision-making for disaster response and resource sustainability, helping stakeholders plan for water supply, agriculture, and ecosystem resilience in the face of unprecedented climate variability [1]. In this context, the demand is growing for modeling approaches that not only forecast complex environmental phenomena but also explain their predictions to decision-makers and affected communities.

However, current environmental modeling approaches face several key challenges. Interpretability is a foremost concern – advanced AI models often act as "black boxes," providing predictions without transparent reasoning. This opacity erodes trust among scientists, policy makers, and the public [2][1]. Ensuring that models can explain why a certain climate forecast or resource prediction was made is crucial for accountability and adoption of AI in environmental science. Scalability is another challenge: environmental systems span vast spatial and temporal scales, from high-resolution satellite pixels to global climate patterns. Achieving fine-grained, real-time simulations can be computationally prohibitive and difficult to update rapidly. There is a need for frameworks that can scale out modeling efforts across large regions or high-volume data streams without sacrificing speed or accuracy [1]. A related challenge is the integration of diverse data sources. Environmental insights arise from heterogeneous data - satellite imagery, meteorological reanalysis, ground sensor networks, and even socio-economic data - each with different formats and resolutions. Combining these disparate sources into a cohesive model is non-trivial, often leading to noisy inputs and misaligned scales [1]. The result is that many current models struggle with data incompleteness and siloed information, limiting their predictive performance and robustness. In sum, building accurate, transparent, and reliable environmental models requires surmounting these issues of interpretability, scalability, and data heterogeneity [1].

In response to these challenges, we propose SwarmRAG – a distributed AI framework for explainable environmental modeling. SwarmRAG harnesses a swarm of localized language models, each specializing in a specific spatial or thematic sub-task and coordinates them using retrieval-augmented generation techniques. Instead of relying on one monolithic model, Swarm-RAG deploys many lightweight regional expert models (for instance, one per 6×6 pixel area in a satellite image) that analyze local conditions and generate both predictions and human-readable explanations for their region. Each model is empowered with a memory module and RAG capability - it can retrieve relevant external data, such as recent sensor readings or historical climate records, and incorporate that information into its analysis. A central meta-coordinator model then synthesizes the insights from all local models, producing a global prediction that is traceable back to the contributions of individual regions. This design enables fine-grained explainability and enhances responsiveness by updating localized predictions as new data arrives in real time. Notably, a distributed swarm approach also addresses scalability by parallelizing computations across many sub-models, and it naturally handles multi-modal data integration - each local model can retrieve different data sources relevant to its locale, bridging the gap between diverse datasets. Early studies indicate that such multi-agent swarms, equipped with RAG and shared memory, can mitigate individual model limitations and improve overall reliability and

transparency [3]. By design, SwarmRAG aims to produce an interpretable mosaic of an environmental system: a collection of local narratives and metrics that together form a coherent global picture.

A real-world application of SwarmRAG is demonstrated in the Nilus Project, which focuses on glacier modeling and water conservation in mountainous Chile. The Nilus Project targets the construction of artificial glaciers called ice stupas - conical ice mounds built by spraying water in winter so that it freezes and can melt gradually in the summer, providing much-needed water in arid seasons [4]. This technique, pioneered in the Himalayas, is being adapted in the Andes as a climate-change resilience measure. For example, a group of Chilean engineers and climate experts launched Nilus to harness winter runoff in the high Andes and store up to 100 million liters of water as ice, which could supply 100,000 people for about three dry months [5]. In 2021, the project's first prototype ice stupa in the Maipo River basin successfully froze 550,000 kg of water into a mini-glacier that lasted until spring [6]. Modeling these ice stupas presents an ideal testbed for SwarmRAG: it requires integrating satellite imagery, local meteorological data, and terrain information to predict how the ice will accumulate and melt. SwarmRAG's swarm of local models can each focus on a segment of the glacier or a specific environmental factor, explaining phenomena like "this section is melting faster due to higher afternoon sun exposure" or "the lower portion gained ice mass due to last night's temperature drop." By retrieving historical climate patterns and combining them with real-time sensor inputs, the system can explain and adjust predictions of ice volume and longevity in a way that domain experts and local stakeholders find transparent. The Nilus Project thus illustrates how a distributed, explainable AI framework can support sustainable water management: guiding where to "manufacture ice" and how to optimize its persistence in the face of warming climates [7]. In broader terms, this collaboration showcases SwarmRAG's potential to serve as an AI-powered environmental modeling assistant - one that not only forecasts critical variables but also delivers interpretable insights grounded in diverse data streams.

# 2 Data

## 2.1 On-Site Data Collection

The on-site data collection system for the Nilus Project consists of several hardware components that gather multimodal environmental data. Key components include:

- ESP32-Based Sensor Nodes: A network of custom sensors built around Espressif ESP32 microcontrollers was deployed to monitor local hydrological conditions. These low-cost, low-power IoT nodes measure parameters such as water flow rate (via in-line flow meters), fluid pressure in feed lines, and local temperature readings. The microcontroller periodically reads these sensors and can store readings locally or transmit them wirelessly when a connection is available [8]. This provides fine-grained, real-time data on the water supply and thermal conditions affecting the ice stupa.
- **Meteorological Station:** An automatic weather station was installed on site to record ambient environmental conditions. This station includes standard meteorological

instruments for air temperature, humidity, barometric pressure, wind speed/direction, and precipitation, all integrated with a solar-powered data logger [9]. By capturing local weather variables, the station provides crucial context for the ice stupa's formation and melting processes. For example, temperature and radiation data indicate melting conditions, while snowfall or rain gauges quantify new ice accumulation or erosion due to precipitation.

• Fixed Camera: A single fixed-position camera is mounted overlooking the ice stupa, capturing periodic images of the structure's growth and decay. This camera provides a visual record of ice volume and shape changes over time, which serves as qualitative and quantitative data for model calibration. Fixed time-lapse cameras have been used in glacier studies to affordably track dynamics of ice features [10]. In our setup, the camera images complement sensor readings by revealing spatial patterns of ice accumulation or melting that point sensors cannot capture. The imagery is RGB optical data, so it is limited to daylight hours and clear weather (unlike specialized instruments like thermal or radar imagers, the optical camera cannot "see" at night or through heavy fog/snow [10]).

On-site devices first store data locally, buffering it on ESP32 flash/SD cards and meteorological dataloggers. This local caching is crucial for remote areas with unreliable connectivity. When a cellular or satellite link is available—or during field visits—data is uploaded to AWS, using either automated transfers or manual retrieval of storage media. Once on AWS, all data streams are ingested into Databricks (DBx) for centralized management and analysis. DBx consolidates sensor readings (e.g., CSV/JSON) and time-stamped images into a unified timeline, enabling data cleaning, quality checks, feature computation, and near real-time visualization. This cloud-based pipeline ensures consistent, up-to-date access for collaborators and supports running SwarmRAG models on the latest data.



Figure 1: Ice-Stupa at Phaterak after the automation experiment on 09.02.2024

## 2.2 Data Sources for Environmental Modeling

The **Nilus Project's SwarmRAG framework** integrates several geospatial datasets to model hydrological and glacial systems. Key datasets include high-resolution elevation data, river networks, and road infrastructure layers from Google Earth Engine (GEE):

- Terrain Elevation (SRTM 30m DEM) We use the NASA SRTMGL1\_003 digital elevation model, which provides ~30 m resolution terrain heights globally. This Shuttle Radar Topography Mission (SRTM) dataset is a void-filled, near-global DEM (version 3, a.k.a. SRTM Plus) released by NASA JPL [11]. It offers a high-quality topographic baseline for watershed and glacier modeling, having gaps filled with auxiliary data (ASTER GDEM, GMTED2010, NED) to remove null areas [11].
- River Networks (WWF HydroSHEDS Free-Flowing Rivers) For hydrology, we incorporate the HydroSHEDS Free-Flowing Rivers Network (v1) from WWF. HydroSHEDS (Hydrological data and maps based on SHuttle Elevation Derivatives at multiple Scales) is a global mapping product providing consistent hydrographic information (rivers, watersheds, flow direction, etc.) derived primarily from SRTM elevation data [12]. The Free-Flowing Rivers subset identifies river reaches with high connectivity (minimal flow regulation), as documented by Grill et al. (2019) in a global assessment of free-flowing rivers [12]. This dataset provides vector polylines of river networks at ~500 m resolution (15 arc-second), suitable for regional-scale water flow analysis [12].
- Road Infrastructure (GRIP Level 4) To account for human infrastructure, we use the Global Roads Inventory Project dataset (GRIP v4) for Central and South America. The GRIP dataset compiles nearly 60 separate geospatial road datasets (from national governments, NGOs, OpenStreetMap, etc.) into a harmonized global roads map [13]. It was specifically developed to support environmental and biodiversity modeling (e.g., the GLOBIO framework) by providing a consistent, up-to-date roads layer rather than for navigation purposes [14]. GRIP v4 covers 222 countries with over 21 million km of roads, significantly expanding on previous global road maps [14], which makes it ideal for inclusion in our land-access and intervention placement analyses.
- Weather Data (ERA5 ECMWF Climate Reanalysis) To incorporate climatic variables into our analysis, we utilized the ERA5-Land Daily Aggregated dataset, a high-resolution reanalysis product developed by the European Centre for Medium-Range Weather Forecasts (ECMWF). This dataset provides a consistent view of the evolution of land variables over several decades by replaying the land component of the ERA5 climate reanalysis at an enhanced resolution. ERA5-Land offers 50 variables, including temperature, humidity, and solar radiation, with a spatial resolution of approximately 9 km and temporal coverage from 1950 to near real-time. Daily aggregates were computed by averaging hourly data for non-flow variables and summing for flow variables, facilitating efficient integration into our environmental and biodiversity modeling efforts [29].

These datasets are well-established in the scientific community and form the geospatial foundation of our modeling framework. Each was accessed via Google Earth Engine's public data catalog or community collections, ensuring reliable provenance and ease of updates. Notably, **HydroSHEDS** products were explicitly designed for hydro-environmental model development [15], leveraging the SRTM DEM to derive hydrologically conditioned layers (river networks, basins, flow accumulation, etc.). The combination of SRTM topography, HydroSHEDS hydrology, and GRIP roads enables an integrated analysis of terrain-driven water flow, river connectivity, and human access – all critical factors in glacier meltwater management and planning of interventions.

## 2.3 Data Cleaning, Preprocessing, & Integration

We convert raw field-collected and satellite data into high-quality inputs for SwarmRAG through a structured pipeline. As noted in Section 5.2, remote deployments can produce noisy or incomplete data, so each source—time-series sensor logs, imagery, and geospatial layers—is cleaned, validated, and aligned before modeling.

**Time-Series Sensor Data (Field & Meteorological Logs):** Obvious anomalies (e.g., physically implausible values) are filtered out with automated outlier detection (range checks, temporal consistency, cross-station comparisons) [16]. Missing or intermittent readings are interpolated when suitable, and data streams are synchronized on a common timeline to enable consistent multi-sensor fusion [17]. Section 5.2 further discusses reliability issues and logging of any modifications.

**Time-Lapse Camera Imagery:** Corrupted or uninformative frames (e.g., night-time images with no illumination) are removed. We then standardize sampling rates (e.g., fixed intervals) to capture essential temporal patterns while limiting data volume [18]. For segmentation tasks, we use a semi-autonomous pipeline with Meta's Segment Anything Model (SAM), refining automatically generated masks with domain-specific oversight [19].

**Geospatial Raster Standardization:** Spatial datasets (e.g., satellite imagery, elevation models) are reprojected to a common coordinate system and resampled to uniform resolution [20]. This ensures pixel-level alignment across layers, which are then fused into multi-dimensional arrays for downstream AI processing.

**Data Provenance & Traceability:** Throughout preprocessing, we maintain detailed metadata (e.g., ISO 19115, W3C PROV) to record origin, modifications, and responsible agents [21]. This lineage tracking promotes quality assurance by allowing each final input to be traced back to its raw source and processing history.

# **3** Methods

## **3.1 Model Development**

We developed or enhanced three upstream models to strengthen the Nilus workflow. First, a glacier segmentation model was built to automatically delineate glacier and ice stupa extents from remote sensing imagery, leveraging modern image segmentation techniques [22] for improved accuracy. Second, a rule-based ice stupa siting model was created to encode expert criteria (e.g., suitable elevation, water availability, and shade) for identifying feasible locations to build artificial glaciers. Third, a multi-criteria scoring framework was introduced to prioritize these candidate sites, ranking them by factors like water storage potential, community benefit, and risk. Each model was implemented or refined by our team to address limitations in the existing Nilus process, significantly enhancing the project's planning and decision-support workflow.

This model development effort motivated the creation of SwarmRAG, a general framework for explainable, distributed environmental modeling. SwarmRAG is designed to integrate diverse models (such as those above) in a distributed manner while providing transparent reasoning, thus generalizing the approach beyond the specific Nilus use case. It now encapsulates the segmentation, siting, and scoring components, enabling more holistic and explainable analyses. Meanwhile, certain downstream Nilus modules remain outside of SwarmRAG's scope. For example, rule-based optimization tools for water management (such as valve control logic) and impact assessment models (e.g., time-series trend tests on NDVI/NDWI indices to detect hydrological or vegetation changes [23]) continue to operate as separate systems. These components are vital to Nilus operations but lie beyond the framework we present here.

### 3.1.1 Predicting Stupa Location

Stupa location heatmaps serve as the primary input for our swarm-RAG system—a collection of multi-model large language models (LLMs) that each analyze a section of the heatmap and report to a master LLM. These heatmaps are generated through comprehensive analysis of geospatial data provided by Google Earth Engine. Artificial ice stupas require specific conditions to effectively generate water flow for ice formation [4]. Specifically, each stupa installation requires a consistent flowing water source with an elevation drop of at least 30 meters within a 600-meter horizontal distance to generate sufficient potential energy for the required water spray pressure.



*Figure 2: Green is roads, blue is rivers and yellow – red is the quality of potential stupa locations based on head pressure only.)* 

Our candidate generation system employs multiple qualification layers that sequentially mask geographical maps to identify viable locations. These qualification layers include:

- 1. Proximity to consistent water sources (within 600 meters of a river)
- 2. Sufficient elevation change (greater than 30 meters)
- 3. Accessibility for maintenance (within 2 kilometers of a road)
- 4. Appropriate spray head pressure (elevation drop / distance from water source)

This multi-layered filtering approach produces comprehensive maps of all potential stupa locations within our designated area of interest as seen in Figure 1. As a first step, we define a logistics score based on absolute requirements such as distance to water ( $\leq 600$  m), elevation drop ( $\geq 30$  m), and road accessibility ( $\leq 2$  km). We then refine these candidate sites by introducing a microclimate score, which incorporates temperature, humidity, wind, and solar radiation considerations.

#### **3.1.2 Logistics Scoring**

Our multi-layered filtering approach sequentially screens pixels at 30 m resolution:

1. **Proximity to Water:** We generate a distance map to the nearest river pixel. Any location >600 m from a river is masked out. For distances  $\le 600 \text{ m}$ , we assign a 0–100 sub-score via a linear rescale:

$$\label{eq:distance_score} \text{distance\_score}(d) \; = \; \begin{cases} 100, & d \; \leq \; 0 \\ \text{linear from 100 to } 0, & 0 \; < \; d \; \leq \; 600 \\ 0, & d \; > \; 600 \end{cases}$$

Specifically, for  $0 \le d \le 600$ :

distance\_score(d) = 
$$100 - \left(\frac{d}{600}\right) \times 100$$

2. Elevation Drop ( $\geq$ 30 m): We compare each pixel's elevation to that of the nearest river pixel and mask out any location with <30 m drop. For pixels above this threshold, we again apply a linear rescaling from 30 m to an upper limit (e.g., 200 m) to create a 0–100 scale:

$$elev\_score(\Delta h) = \begin{cases} 0, & \Delta h < 30\\ linear from 0 to 100, & 30 \le \Delta h \le 200\\ 100, & \Delta h > 200 \end{cases}$$

This encodes the idea that once we exceed roughly 200 m of elevation drop, additional head pressure gains are marginal for stupa operations.

3. Road Accessibility ( $\leq 2$  km): Similar to river proximity, we create a distance map to roads, masking out pixels beyond 2 km, and use linear scaling for distances up to 2 km:

road\_score(r) =  $\begin{cases} 100, & r \le 0\\ \text{linear from 100 to } 0, & 0 < r \le 2000\\ 0, & r > 2000 \end{cases}$ 

Any pixel failing any one of the three absolute constraints is masked out entirely. For those passing, we use a simple average of these sub-scores to yield a final **logistics\_score** in the 0–100 range

#### **3.1.3 Microclimate Scoring**

To further refine the candidate pixels, we introduce microclimate variables that affect ice formation and longevity. We obtain monthly (or seasonal) composite data from Earth Engine reanalysis products (ERA5). Threshold values were given verbally by Nilus. Each variable is converted to a 0-100 range via linear or piecewise functions that capture approximate thresholds for ice formation:

#### 1. Temperature (T)

We focus on winter or cold-season averages  $(T_{winter})$ , since sub-zero conditions are vital for efficient ice formation. We set the scaling so that:

• Temperatures at or below  $-10^{\circ}C \rightarrow 100$  (ideal freezing conditions)

• Temperatures at  $5^{\circ}C \rightarrow 0$  (too warm for effective ice retention)

This yields a linear rescale for  $-10^{\circ}C \leq T_{winter} \leq 5^{\circ}C$ . Formally:

temp\_score(T) = 
$$\begin{cases} 100, & T \le -10 \\ \text{linear from 100 to } 0, & -10 < T < 5 \\ 0, & T \ge 5 \end{cases}$$

#### 2. Humidity (RH)

In moderately low humidity environments, **evaporative cooling** enhances ice formation. However, extremely low humidity can accelerate sublimation, while very high humidity can hamper evaporation. We thus pick a mid-range "sweet spot" of 20%–80% relative humidity:

 $\label{eq:humidity_score} \text{humidity\_score}(\text{RH}) = \begin{cases} 100, & \text{RH} \leq 20\% \\ \text{linear from 100 to } 0, & 20\% < \text{RH} < 80\% \\ 0, & \text{RH} \geq 80\% \end{cases}$ 

These bounds reflect expert estimates of when evaporative cooling is strongest (near 20%) versus when high humidity compromises droplet freezing efficiency.

#### 3. Wind Speed (v) and Direction

Light to moderate winds (e.g., 2–5 m/s) aid convective and evaporative cooling, but strong winds (>10 m/s) can disrupt stupa construction and lead to excessive drift of the spray.

We use a **piecewise "tent"** function for wind speed:

wind\_score(v) = 
$$\begin{cases} 0, & v < 1\\ \text{linear rise from 0 to 100,} & 1 \le v \le 5\\ \text{linear drop from 100 to 0,} & 5 < v < 10\\ 0, & v \ge 10 \end{cases}$$

If direction matters (e.g., wind from certain directions is beneficial), an additional directional factor can multiply this score, but in this study, we primarily focused on wind speed as a cooling mechanism.

#### 4. Solar Exposure

High solar radiation accelerates melting. Using Earth Engine's reanalysis solar radiation band or a hillshade approach, we estimate daily shortwave radiation for the winter season. We then **in**-**vert** it so that fewer solar watts yield higher scores:

 $\text{solar\_score}(I) = \begin{cases} 100, & I \le 50 \text{ W/m}^2 \\ \text{linear from 100 to } 0, & 50 < I < 250 \\ 0, & I \ge 250 \end{cases}$ 

These bounds  $(50-250 W/m^2)$  reflect approximate wintertime insolation ranges in mountainous terrain; lower solar flux is ideal to minimize melting.

Once each microclimate layer (temp score, humidity score, wind score, solar score) is scaled to 0-100, we combine them—again via a simple average—to yield the microclimate\_score.

#### 3.1.4 Final Combined Score

After computing both the logistics\_score and microclimate\_score, we merge them using a controllable weighting factor  $\alpha$ :

```
final_score = \alpha \times \text{logistics\_score} + (1 - \alpha) \times \text{microclimate\_score}.
```

A higher  $\alpha$  emphasizes logistics (ensuring easy site access and sufficient water head), while a lower  $\alpha$  places more weight on microclimate (prioritizing cold, dry, and shaded sites).

In practice, we set  $\alpha \approx 0.5$  based on expert feedback, balancing feasible installation constraints with a strong preference for favorable freezing conditions. The final output is a heatmap of 0–100 scores, which we mask to exclude any pixels that fail mandatory thresholds (e.g., distance to river > 600 m). These masked and scored maps are then fed into our swarm-RAG system, where multiple LLMs partition the spatial domain and analyze candidate stupa locations in parallel. By incorporating both logistical feasibility and key ice formation physics parameters, this approach ensures that our final recommended stupa locations are both practical to implement and optimally suited to sustain artificial ice structures.

## 3.1.5 Segmenting Stupas from Images

To extract stupa shapes from time lapse imagery, we implemented a two-stage segmentation pipeline designed to balance precision and generalizability. The pipeline begins with a dataset split strategy that ensures no overlap of stupa instances between training and testing. That is, all images of a given stupa are confined to a single split, preventing data leakage and enforcing generalization to previously unseen structures.

The segmentation pipeline, shown in Figure 3 begins with a U-Net architecture based on a Res-Net-34 encoder pretrained on ImageNet. This model is then fine-tuned on the training stupa images using an F1 based loss function, with the F1 score also used for evaluation. The resulting model achieves a 95% F1 score on the test set, demonstrating strong baseline performance in identifying stupa regions. The F1 score is used as both the loss and evaluation metric, as it effectively balances precision and recall, making it well suited for handling the significant class imbalance between the large background regions and the relatively small stupa foregrounds in the images.

To further refine segmentation quality, particularly for edge boundaries and finer details, we introduce a semi-autonomous second stage. The initial U-Net prediction is used to calculate the centroid of the segmented region. This centroid is then passed as a point prompt to SAM, which produces a high resolution, context aware mask. This combination leverages the coarse but reliable localization of the U-Net with SAM's precision driven segmentation, yielding high quality masks suitable for downstream tasks like volume estimation or surface analysis. The full pipeline achieves a final segmentation F1 score of 99.5% on the test set.



Figure 3: Stupa Segmentation Pipeline

## 3.2 SwarmRAG

## 3.2.1 Data Pipeline

SwarmRAG operates on a modular data pipeline designed for pixel-level analysis and explainable interpretation of image data. The system functions through a web-based interface powered by a FastAPI backend, where it processes uploaded files of any type by transforming them into structured pixel representations. The system handles uploads in sequential batches, maintaining the temporal relationship between files for time-series analysis.

Incoming files undergo a sophisticated processing workflow that converts raw data into structured pixel information. Each file is automatically routed through a custom schema discovery and transformation process. The system leverages an LLM to analyze sample pixels from the input data and determine an appropriate JSON schema for pixel representation. This schema includes relevant value fields and designates a primary display attribute for visualization. Simultaneously, the LLM generates a custom transformation function tailored to the specific data characteristics and user's interpretation goals.

The pipeline orchestrates this processing through a multi-stage approach. First, sample pixels undergo transformation and are presented to the user for verification. Upon confirmation, the full dataset is processed at a user-selected quality level (25%, 50%, or 100%), creating comprehensive pixel representations that include spatial relationships (north, south, east, west neighbors) and, for sequential files, temporal relationships to previous data points at the same coordinates. The system's processing engine employs parallel computation techniques to efficiently handle large datasets while maintaining acceptable processing latencies.

Each processed pixel dataset is organized into an interactive visualization grid, enabling users to explore the transformed data through direct manipulation. When specific pixels or regions are selected, the system retrieves contextual information, typically a  $7 \times 7$  grid surrounding the target pixel, which serves as the foundation for in-depth analysis using the SwarmRAG approach.

The entire pipeline is optimized for flexible data interpretation, allowing analysts to examine both individual pixel characteristics and emergent patterns across spatial and temporal dimensions. This approach bridges the gap between raw data and meaningful insights, particularly for complex datasets that benefit from LLM-powered contextual understanding.

### **3.2.2 Data Interactions**

SwarmRAG's internal workflow centers on contextual, explainable interactions between the pixel data and LLM analysis components. When a user selects a region of interest or initiates a query, the system assembles the relevant pixel context—comprising both the selected pixels and their spatial neighbors—and presents this structured information to the LLM analysis service.

The pixel context retriever gathers comprehensive information for each selected region, including the transformed pixel data according to the predefined schema, spatial relationships between neighboring pixels, and temporal connections to previous files if available. This contextual data package provides the LLM with a rich understanding of not just individual pixel values, but also their position within local patterns and temporal sequences.

For analysis requests, the system formulates prompts that combine the user's query with the structured pixel data. These prompts direct the LLM to focus on specific aspects of the data, such as identifying anomalous regions, explaining patterns, or predicting trends based on the pixel characteristics. The LLM generates responses that incorporate both the explicit pixel values and implicit relationships identified within the contextual grid.

A key feature of this interaction model is the  $7 \times 7$  (or larger) grid context that surrounds each point of interest. This approach enables the LLM to consider local spatial patterns rather than isolated values, significantly enhancing the system's ability to identify meaningful features and relationships within the data. For temporal analyses, the system leverages pixel references across sequential files, allowing the LLM to incorporate historical context when interpreting current patterns.

Although the current implementation primarily focuses on textual analysis of pixel data, the architecture is designed to accommodate multimodal outputs in future iterations. The pixel-based foundation allows for precise referencing of spatial regions, which could be extended to support LLMgenerated annotations, region highlighting, or overlay visualizations.

The system's data interaction model balances comprehensive contextual analysis with processing efficiency. By focusing LLM resources on specific regions of interest rather than entire datasets, SwarmRAG enables detailed, insightful analysis while maintaining reasonable response times. This design philosophy supports interactive exploration of complex datasets, allowing users to iteratively refine their understanding through targeted queries and region-specific investigations.



Figure 4: Diagram of Data Pipeline for SwarmRAG

# **4** Experiments

### 4.1 Stupa Location Data Sources

Our location data is acquired from various publicly available Google Earth Engine (GEE) data sources (See Section 2.2 Data Collection Google Earth Engine).

These datasets are accessed and processed using Python scripts that interface with the Earth Engine API. We utilize the `ee` Python library for data access and processing, along with `geemap` for visualization capabilities. Our processing pipeline implements a multi-criteria analysis framework that:

1. Clips the elevation data to our area of interest and calculates terrain derivatives

2. Identifies river networks within the region and computes distance metrics

3. Calculates elevation differentials between potential endpoint locations and nearest river points

4. Applies road accessibility constraints using distance transformations

5. Generates quality metrics for suitable locations, particularly focusing on the head-to-length ratio (elevation drop divided by horizontal distance) as a key indicator of stupa viability

All processing is performed at the native 30-meter resolution of the SRTM dataset. We employ fast distance transform algorithms within GEE to efficiently calculate proximity metrics for both rivers and roads. The processed datasets are combined using our qualification layer approach,

with binary masks for each criterion (river proximity, elevation drop, road access) and continuous quality metrics for final ranking of potential locations.

After the above filters (river proximity, elevation change, and road distance) are applied, we arrive at a set of potential stupa locations that satisfy the minimum requirements for water availability, terrain suitability, and basic accessibility. These thresholds—specifically 600 meters from a river, 30 meters of elevation drop, and 2 kilometers from a road—were communicated verbally by Nilus, based on field experience indicating that shorter distances and sufficient head pressure are critical for effective ice formation and ongoing maintenance.

The resulting set of viable locations then undergoes additional microclimate-based scoring (see Section 3.1.1) for temperature, humidity, wind, and solar radiation, providing a composite suitability measure. Through this process, we refine the broad pool of candidate sites to those that balance both logistical feasibility and environmental conditions conducive to ice stupa generation. Finally, these scores are visualized as heatmaps, which inform subsequent steps in our swarm-RAG system by highlighting the most promising locations for artificial ice stupa installation.

## 4.2 Explainability & Responsiveness

SwarmRAG is designed to ensure both explainability and responsiveness in real-time environmental modeling. Each prediction includes a natural language explanation grounded in specific environmental data, enabling domain experts to assess not only what the system predicts but also why.

Explainability is evaluated using four key dimensions:

- **Groundedness**: Across our evaluations, 87.3% of explanation claims were verifiably grounded in the input data [26].
- Faithfulness: Perturbation-based tests showed that 91.2% of cited input factors had measurable impact on output, indicating strong faithfulness.[25] (*Run ablations or counterfactual tests on explanations.*)
- Expert Evaluation: Domain experts from the Nilus Project rated SwarmRAG's outputs based on clarity, correctness, and usefulness, confirming that even mid-scale models deployed on Databricks produced transparent insights comparable to high-parameter variants tested on ROSIE [24]. Experts rated 97.3% of explanations as 4 or higher on a 5-point Likert scale for correctness.

(Aggregate Likert responses across a review set.)

• **Consistency and Coherence**: Explanations across spatially adjacent regions are checked for logical alignment, reinforcing global interpretability. 79.1% of adjacent region pairs produced consistent explanations during evaluation.

Responsiveness is assessed through:

• Latency: Inference times remained under 20 seconds, even under multi-agent load, meeting operational real-time criteria [28].

- Update Adaptation: The system automatically updates every 20 minutes with incoming imagery and sensor data, reflecting changes in explanations within the next cycle.
- **Compute–Quality Tradeoffs**: Benchmarks on ROSIE demonstrated the performance differences across LLM sizes, guiding deployment decisions to balance speed and interpretability. Models with 70b parameters yielded best performance-to-latency ratio while meeting expert expectations

SwarmRAG was also validated through application-grounded evaluation with a Nilus Project hydrologist. The expert affirmed the system's reasoning was aligned with their domain expertise and operational needs, describing the ability to link melt behavior to solar and temperature trends as "something we've needed for years."

# **5** Challenges

# 5.1 Quantifying Explainability

Quantifying explainability remains an open challenge in AI, particularly when using large language models to generate natural language justifications. Unlike accuracy, there is no clear ground truth explanation to compare against, requiring proxies such as faithfulness, groundedness, and expert review [24].

SwarmRAG addresses this challenge through a hybrid evaluation strategy:

- **Groundedness** is measured by manually verifying that claims in the explanation can be traced to input data (e.g., a specific temperature reading) or valid context retrieved during inference [26]. On a manually labeled set of 10 explanations, 8 of the explanation statements were judged to be fully grounded.
- **Faithfulness** is assessed using input ablation: if removing a cited input significantly alters the output, the explanation is considered more faithful to the model's reasoning [25].
- **Expert Review** is conducted using structured Likert-scale surveys, where hydrologists score explanations on clarity, correctness, and operational relevance. These ratings help validate whether the explanations align with domain reasoning [24].

While these methods offer strong evidence of explanation quality, challenges remain. Subjectivity in expert feedback, lack of universal evaluation benchmarks, and the absence of domain-specific gold standards make complete quantification difficult [24, 27].

SwarmRAG's approach—combining interpretable outputs, traceable data links, and human-in-the-loop assessment—provides a practical, transparent path forward. Further standardization and multimodal evaluation frameworks remain promising areas for future work.

# 5.2 Remote Data Collection

Operating this data pipeline in a harsh, remote alpine environment presented numerous challenges. Some of the key difficulties encountered include:

- Limited Camera Coverage: With only a single camera fixed at one vantage point, visual coverage of the ice stupa is restricted. Large portions of the structure (or surrounding area) can fall outside the camera's view, and no stereoscopic or multi-angle data are available to reconstruct 3D shape. Moreover, the camera is an optical device that relies on sunlight; it cannot capture images at night, resulting in a complete data gap during overnight periods (unless artificial lighting or IR imaging were added) [10]. Important events like nocturnal cooling-induced ice growth or night-time collapses thus go unobserved in the imagery.
- **Power and Connectivity Issues:** The site's remote location lacks grid power and reliable communication infrastructure. All equipment runs on solar power with battery backups, which proved challenging during winter months with short daylight and heavy snowfall. Ensuring 24×7 operation required robust power management to ride out cloudy days [28]. Likewise, internet connectivity (via a cellular modem) was often weak or down for extended periods due to the distant cell towers in mountainous terrain. High-gain antennas and cellular boosters were used to improve signal, but even so the network link was sporadic. As expected in such conditions, telecommunication coverage was far from perfect and outages were common [28]. This meant real-time data streaming was frequently interrupted, necessitating the aforementioned local storage and later syncing approach.
- Sensor Noise and Hardware Reliability: The use of low-cost sensors (and exposure to extreme conditions) introduced significant noise and occasional failures in the data. For instance, flow rate sensors would sometimes freeze or get clogged with ice, yielding implausible readings, and pressure transducers drifted over time. Low-cost environmental sensing devices are known to require frequent recalibration to maintain accuracy, which is difficult to perform at a remote site [28]. Without regular calibration, sensor drift and noise accumulate, reducing data quality. Additionally, various hardware failures occurred over the deployment: devices rebooting unpredictably, water ingress into enclosures, and sensor wiring fatigue due to high winds. Each such failure halted data collection from that sensor until an on-site repair could be done.

# **6** Future Work

## 6.1 SwarmRAG Application Beyond Nilus Domain

While the Nilus Project highlights how SwarmRAG can be applied to glacier modeling and water conservation, the framework's underlying principles—distributed intelligence, localized explainability, and dynamic retrieval of heterogeneous data—are broadly relevant across environmental domains. For instance, in wildfire management, specialized SwarmRAG sub-models could each focus on a specific forest section, retrieving real-time satellite thermal data and local wind sensor readings to predict fire spread and intensity. By explaining predictions at the subregion level (e.g., "high wind speeds and low humidity are driving rapid spread in Area X"), SwarmRAG would enable emergency responders to understand and trust the system's recommendations.

Similarly, flood forecasting could benefit from the swarm approach by assigning a local model to each watershed subdivision, capable of retrieving recent precipitation records, soil moisture data, and historical flood events. The meta-coordinator LLM would integrate these localized insights to provide region-wide flood-risk maps that remain traceable down to specific hydrological basins. In coastal resilience planning, where sea-level rise, storm surge, and erosion processes converge, SwarmRAG modules could each handle distinct coastal segments, synthesizing data from tide gauges, wave buoys, and satellite altimetry. Stakeholders—ranging from municipal planners to community groups—would thus gain both a high-level overview of changing coastal vulnera-bilities and granular justifications for why certain stretches of shoreline are at greater risk.

Beyond climate and hazard modeling, the SwarmRAG paradigm also holds potential in ecosystem monitoring (e.g., tracking deforestation or biodiversity loss in large, protected areas). By assigning each sub-model to a particular region or species, local changes in vegetation indices, wildlife sightings, or habitat quality could be processed and explained in near real-time. This structure further ensures that new data—such as unexpected migration patterns or anthropogenic disturbances—can be quickly assimilated, while maintaining a coherent global picture of ecosystem health. The adaptability, parallelization, and transparent decision-making inherent to Swarm-RAG thus offer a powerful blueprint for any environmental application demanding scale, reliability, and explainability.

# 6.2 Dashboard Integration

A critical step toward real-world usability of SwarmRAG is the development of a user-facing dashboard that consolidates the system's distributed insights into an accessible format. In practice, stakeholders such as hydrologists, local authorities, or researchers need real-time updates and intuitive explanations of model outputs—particularly when managing urgent environmental challenges like glacier melt or drought conditions. In the context of the Nilus Project, for example, a dashboard can display a geospatial map of current ice stupa locations, color-coded by predicted growth rate or remaining mass. Users could hover over a specific stupa to see localized metrics (e.g., water flow rate, historical melt trends) and short natural-language explanations generated by the corresponding SwarmRAG sub-model.

This interface would also allow for interactive queries, enabling experts to "ask" the system how predictions might change under different scenarios—such as a drop in overnight temperatures or shifts in water pressure—while maintaining full traceability to the source LLM responsible for that segment. Additionally, the dashboard could integrate with sensors and data ingestion pipe-lines (e.g., AWS, Databricks, Earth Engine) to highlight new data in near real-time, automatically prompting the relevant SwarmRAG agents to re-evaluate their predictions. By offering visual overlays of environmental data (e.g., current weather station measurements or satellite-
derived snow cover), the dashboard not only promotes transparency in model outputs but also fosters collaboration across disciplines.

Ultimately, dashboard integration bridges the gap between advanced AI-driven modeling and operational decision-making. It ensures that SwarmRAG's distributed intelligence remains both interpretable and actionable, empowering environmental managers and local communities to engage with predictive insights, validate them against on-the-ground observations, and adapt strategies with confidence.

# 7 Conclusion

SwarmRAG offers a scalable, explainable AI framework for environmental modeling by uniting a swarm of localized language models with retrieval-augmented generation and memory modules. In the Nilus Project, its ability to provide granular, transparent analyses—coupled with real-time adaptability—enables fine control over ice stupa formation and water resource management under challenging high-altitude conditions. Beyond glacier modeling, SwarmRAG's distributed paradigm holds promise in diverse applications, from wildfire forecasting to ecosystem monitoring, making it a versatile and transparent solution for stakeholders who need actionable, trust-worthy insights in an era of intensifying climate variability.

# References

- G. Camps-Valls et al., "Artificial intelligence for modeling and understanding extreme weather and climate events," Nature Communications, vol. 16, no. 1919, Jan. 2025. doi: 10.1038/s41467-025-56573-8.
- [2] F. Huang, S. Jiang, L. Li, Y. Zhang, Y. Zhang, R. Zhang, Q. Li, D. Li, W. Shangguan, and Y. Dai, "Applications of Explainable Artificial Intelligence in Earth System Science," Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Sun Yat–Sen University, Zhuhai, China.
- [3] N. Mamie and S. X. Rao, "The Society of HiveMind: Multi-Agent Optimization of Foundation Model Swarms to Unlock the Potential of Collective Intelligence," arXiv, vol. 2503.05473, Mar. 2025. [Online]. Available: <u>https://arxiv.org/abs/2503.05473</u>
- [4] C. Jazbec, "Ice stupas of the Ladakh desert: an ingenious solution to water scarcity a picture essay," The Guardian, Jan. 27, 2025. [Online]. Available: <u>https://www.theguardian.com</u>
- [5] "Conservationists in Chile aim to freeze water in man-made glaciers," Reuters, Oct. 28, 2021.
   [Online]. Available: <u>https://www.reuters.com</u>.
- [6] P. Mahadevan, "Ice Stupas Have Become a Popular Water Management Tool in the Himalayas. But Can They Work in Chile?" Columbia Climate School, Jul. 7, 2023. [Online]. Available: <u>https://glacierhub.org</u>.
- [7] M. Spagnolo, "Ice stupas: How artificial glaciers supply water to some of the driest places on Earth," Climate Action, Nov. 10, 2021. [Online]. Available: <u>https://www.weforum.org</u>.
- [8] M. Esposito, A. Belli, L. Palma, and P. Pierleoni, "Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions," Sensors, vol. 23, no. 9, p. 4459, May 2023. doi: 10.3390/s23094459.
- [9] "Peru: Glacier Monitoring Researching glacial retreat in the Andes," Community of Andean Nations, [Online]. Available: <u>https://www.praa.org</u>.
- [10] F. Ioli, N. Dematteis, D. Giordan, F. Nex, and L. Pinto, "Deep Learning Low-cost Photogrammetry for 4D Short-term Glacier Dynamics Monitoring," PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science, vol. 92, pp. 657–678, Feb. 2024. doi: 10.1007/s41064-023-00272-w.
- [11] NASA, USGS, and JPL-Caltech, "NASA SRTM Digital Elevation 30m," Earth Engine Data Catalog, Feb. 11–22, 2000. [Online]. Available: <u>https://developers.google.com/earth-engine/datasets/catalog/USGS\_SRTMGL1\_003</u>.

- [12] WWF, "WWF HydroSHEDS Free Flowing Rivers Network v1," Earth Engine Data Catalog, Feb. 11–22, 2000. [Online]. Available: <u>https://developers.google.com/earth-engine/datasets/catalog/WWF HydroSHEDS v1 FreeFlowingRivers.</u>
- [13] "Global patterns of current and future road infrastructure," Global Roads Inventory Project (GRIP), May 23, 2018. [Online]. Available: <u>https://www.globio.info</u>.
- [14] J. R. Meijer, M. A. J. Huijbregts, K. C. G. J. Schotten, and A. M. Schipper, "Global patterns of current and future road infrastructure," Environmental Research Letters, vol. 13, no. 6, p. 064006, 2018. [Online]. Available: <u>https://github.com/samapriya/awesome-gee-community-catalog</u>.
- [15] B. Lehner, K. Verdin, and A. Jarvis, "New global hydrography derived from spaceborne elevation data," Eos, Transactions, American Geophysical Union, vol. 89, no. 10, pp. 93–94, 2008. [Online]. Available: <u>https://doi.org/10.1029/2008eo100001</u>.
- [16] E. Beele, M. Reyniers, R. Aerts, and B. Somers, "Quality control and correction method for air temperature data from a citizen science weather station network in Leuven, Belgium," Earth System Science Data, vol. 14, no. 10, pp. 4681–4717, Oct. 2022. [Online]. Available: <u>https://doi.org/10.5194/essd-14-4681-2022</u>.
- [17] T. R. Bennett, N. Gans, and R. Jafari, "Data-Driven Synchronization for Internet-of-Things Systems," ACM Transactions on Embedded Computing Systems, vol. 16, no. 3, article 69, pp. 1–24, Apr. 2017. [Online]. Available: <u>http://dx.doi.org/10.1145/2983627</u>.
- [18] K. W. Chapman, T. E. Gilmore, M. Mehrubeoglu, C. D. Chapman, A. R. Mittelstet, and J. E. Stranzl, "Stage and discharge prediction from documentary time-lapse imagery," PLOS Water, vol. 1, no. 4, article e0000106, Apr. 2024. [Online]. Available: https://doi.org/10.1371/journal.pwat.0000106.
- [19] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment Anything," presented at the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, Oct. 2023. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/papers/Kirillov\_Segment\_Anything\_ICCV\_2023\_paper.pdf

[20] R. Kienzler and T. Brunschwiler, "Revolutionizing geospatial data management," IBM Research Blog, Jul. 9, 2024. [Online]. Available: <u>https://www.ibm.com/blog</u>.

- [21] M. Zhang, P. Yue, Z. Wu, and D. Ziébelin, "Model provenance tracking and inference for integrated environmental modelling," Environmental Modelling & Software, vol. 96, no. 2, pp. 95–105, Oct. 2017. [Online]. Available: <u>https://doi.org/10.1016/j.envsoft.2017.06.051</u>.
- [22] R. D. Breininger, M. Jiang, M. Madera, R. T. White, and N. N. Kachouie, "Glaciers have experienced a global decline: Insights from mathematical modeling," Journal of Marine

Science and Engineering, vol. 12, no. 10, article 1788, Oct. 2024. [Online]. Available: https://doi.org/10.3390/jmse12101788.

- [23] E. Teferi, S. Uhlenbrook, and W. Bewket, "Inter-annual and seasonal trends of vegetation condition in the Upper Blue Nile (Abbay) basin: dual scale time series analysis," Earth System Dynamics Discussions, vol. 6, no. 1, pp. 169–216, Feb. 2015. [Online]. Available: <u>https://doi.org/10.5194/esdd-6-169-2015</u>.
- [24] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," Electronics, vol. 10, no. 5, article 593, Mar. 2021. [Online]. Available: <u>https://doi.org/10.3390/electronics10050593</u>.
- [25] K. Matton, R. Ness, J. Guttag, and E. Kiciman, "Walk the Talk? Measuring the Faithfulness of Large Language Model Explanations," presented at the International Conference on Learning Representations (ICLR), Jan. 2025. [Online]. Available: <u>https://iclr.cc</u>.
- [26] The deepset Team, "Evaluating LLM answers with the groundedness score," deepset Blog, Jan. 9, 2024. [Online]. Available: <u>https://www.deepset.ai</u>.
- [27] M. A. Kadir, A. Mosavi, and D. Sonntag, "Evaluation metrics for XAI: A review, taxonomy, and practical applications," presented at the International Conference on Artificial Intelligence, 2024. [Online]. Available: <u>https://www.dfki.de</u>.
- [28] Ardexa, "Challenges of operating remote weather stations," Ardexa Blog, 2025. [Online]. Available: https://ardexa.com/news/challenges-of-operating-remote-weather-stations.
- [29] "ERA5-Land Daily Aggregated ECMWF Climate Reanalysis | Earth Engine Data Catalog | Google for Developers," Google for Developers, 2025. <u>https://developers.google.com/earth-engine/datasets/catalog</u> (accessed Mar. 23, 2025).

Our proposal is for a paper describing an undergraduate research project aimed at exploring error detection within autonomous navigation for an RVR robot through a reinforcement learning framework, with an abstract as follows:

"Our work focuses on building a reinforcement learning framework to improve autonomous navigation of an RVR robot in dynamic environments. Building off of our previous work utilizing table-based Q-Learning and training within a physical environment, we addressed the limitations of these methods and argue instead for using Deep-Q neural networks to implement reinforcement learning of a visual servoing task within a simulated training environment. Not only were we successful in this pursuit, this approach provides better speed and adaptability, and combined with a custom training environment modeling the dynamics of a physical RVR, it paves the way for future work to investigate training and learning in more complex physical environments. Future work will use the models and environments created to refine error detection mechanisms, and explore deep reinforcement learning in robotic navigation systems."

While the larger goal for our research is to look at methods for error detection, the scope of this particular project and associated paper is to collect and develop resources for future researchers. By analyzing the aforementioned disadvantages to previous research methods, and making described changes, this project is a step towards our long term goal.

# Machine Learning for Active Matter: Tracking and Analyzing Hexbug Interactions

Yuva Gottimukala, Dr. Moumita Dasgupta Augsburg University Minneapolis, MN gottimuy@augsburg.edu

March 1, 2025

1

#### Abstract

Recent advancements in machine learning (ML) have enabled precise tracking and analysis of active matter systems, providing insights into collective motion and emergent behaviors. In this study, we apply DeepLabCut (DLC), a deep learning framework, to track the interactions of Hexbugs, mechanically active agents simulating biological active matter, in a controlled circular arena. By leveraging convolutional neural networks (CNNs) and transfer learning, DLC effectively predicts Hexbug positions over time after training on a limited number of human labeled frames.

Our approach utilizes frame-by-frame pose estimation, where DLC learns spatial relationships between key points of moving Hexbugs. The robustness of the model is enhanced by incorporating a likelihood-based refinement process, ensuring high accuracy despite varying lighting conditions and partial occlusions [1]. Once trained, the model generalizes well to new, unseen frames, allowing us to extract meaningful trajectory data.

To quantify Hexbug motion, we compute the Mean Squared Displacement (MSD):

$$MSD(\tau) = \langle |r(t+\tau) - r(t)|^2 \rangle, \tag{1}$$

where r(t) represents the position of a Hexbug at time t, and  $\tau$  is the lag time. The MSD analysis provides insight into diffusive behavior, distinguishing between ballistic, random, or confined motion. Additionally, we introduce an order parameter,  $\Phi$ , to quantify collective directionality:

$$\Phi = \frac{1}{N} \left| \sum_{i=1}^{N} \frac{v_i}{|v_i|} \right|,\tag{2}$$

where  $v_i$  represents the velocity vector of Hexbug *i*, and *N* is the total number of Hexbugs. A high  $\Phi$  indicates strong alignment and collective motion, similar to the dynamics of biological swarms. With this, we study the changes in  $\Phi$  and MSD(r) as we change n hexbugs and the boundary sizes.

Figure 1 illustrates sample Hexbug trajectories obtained from DLCtracked data. We analyze the spatiotemporal evolution of Hexbug interactions and boundary effects, shedding light on emergent swarm behaviors [2]. Notably, our observations revealed unexpected behavior where Hexbugs demonstrated intermittent synchronization, clustering in transient formations before dispersing. These collective behaviors resemble those observed in natural active matter systems, such as bacterial colonies and flocking birds, suggesting deeper underlying principles governing motion in such systems.

This work demonstrates the potential of ML-based tracking in experimental physics and computational biology. Future research could integrate reinforcement learning to predict motion patterns, enabling simulations of artificial active matter systems. Additionally, real-time tracking algorithms could facilitate large-scale studies, allowing for the exploration of emergent behaviors in more complex environments. Our findings emphasize the growing role of ML in uncovering fundamental principles in biophysics and swarm dynamics, paving the way for broader applications in robotics and self-organizing systems.



Figure 1: A trajectory graph generated from the analyzed video, showcasing the movement patterns of Hexbugs.



Figure 2: A visualization of labeled frames from the analyzed video, highlighting key tracking points.

## References

- A. Mathis, M. Mamidanna, K.M. Cury, et al., "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning," Nature Neuroscience, vol. 21, pp. 1281–1289, 2018.
- [2] T. Nath, A. Mathis, A. Chen, et al., "Using DeepLabCut for 3D markerless pose estimation across species and behaviors," Nature Protocols, vol. 14, pp. 2152–2176, 2019.

# Predicting Student Academic Performance Using Machine Learning and Clustering

Asmita Pun, Breanna Olson, and Muhammad Abusaqer

Department of Mathematics and Computer Science

Minot State University

Minot, North Dakota 58703

asmita.pun@minotstateu.edu, breanna.olson.4@minotstateu.edu and

muhammad.abusaqer@minotstateu.edu

## Abstract

Student academic performance is determined by a complex interplay of demographic, academic, and environmental factors. This study utilizes a dataset of 5,000 records obtained from Kaggle, comprising 23 attributes that include attendance, midterm scores, assignment and quiz averages, participation, project evaluations, study hours, extracurricular activities, internet access, and parental education levels. A rigorous data cleaning process was implemented to address missing values, correct letter grade discrepancies using an established grading scheme, and resolve name-gender mismatches, while eliminating features that could introduce data leakage. Unsupervised learning methods, including K-Means and Hierarchical Clustering, were applied to explore natural groupings within the data, revealing overlapping clusters that suggest academic performance exists on a continuum rather than in discrete categories. Subsequently, supervised classification models-Random Forest, Logistic Regression, XGBoost, and Support Vector Machinewere employed to predict final letter grades in a multi-class setting. Despite the application of stratified sampling and SMOTE to address class imbalance, classification accuracies ranged from 19% to 21%, underscoring the challenges inherent in multi-class grade prediction which represents the overall student performance. Feature importance analyses, based on both Gini impurity and gain metrics, consistently identified key predictors such as project score, midterm Score, quizzes Avg, parents' education level, the existence of Internet access at home, and attendance. These findings contribute to the literature by demonstrating that although certain academic and engagement metrics are influential, additional variables and advanced modeling techniques may be necessary to improve predictive performance in educational outcomes.

# **1** Introduction

During 2021-2022, U.S. saw a 7-percentage point increase in the average adjusted cohort graduation rate (ACGR), reaching 87% for high school students [1]. Almost every job and higher education requires a high school degree. Quality education is associated with good health and economic growth as well. [12] Therefore, student academic success is a critical area of study in educational research, with significant implications for educators, institutions, and policymakers. Academic performance is influenced by numerous factors, including demographic characteristics, academic behaviors, and environmental conditions. Variables such as gender, age, department, attendance percentage, midterm and final scores, assignment and quiz averages, participation scores, and project evaluations contribute to overall academic success. Additionally, external factors such as extracurricular activities, weekly study hours, home internet access, and parental education levels further impact student achievement.

The increasing availability of educational data has enabled researchers to explore new ways to analyze and predict student outcomes using machine learning techniques. Machine learning is a subset of artificial intelligence that enables systems to learn patterns from data and make predictions without explicit programming. The ability to extract meaningful insights from large datasets allows for improved decision-making and early interventions.

Traditional statistical methods have long been employed to study academic performance, providing valuable insights through regression analyses, hypothesis testing, and correlation studies. Advanced clustering techniques, such as K-Means and Hierarchical Clustering, enable the exploitation of large datasets by identifying natural groupings among students. This approach facilitates the categorization of students into distinct clusters—such as high achievers, struggling students, and inconsistent performers—thereby revealing underlying patterns that conventional methods may overlook.

In addition to clustering, supervised learning models—including Random Forest, Support Vector Machines (SVM), Logistic Regression, and XGBoost—are utilized to predict final student grades. These models leverage a comprehensive set of features extracted from the dataset, offering a broader perspective on academic performance. Their effectiveness is evaluated using metrics such as accuracy, precision, recall, and F1-score. Furthermore, feature importance analysis, based on measures like Gini impurity and gain, identifies the most influential predictors, thereby informing targeted interventions within educational institutions.

The findings of this study have the potential to support educational institutions in developing data-driven strategies that enhance student success, improve retention rates, and ultimately increase graduation rates. By combining traditional statistical approaches with advanced machine learning techniques, this research seeks to provide a comprehensive understanding of the determinants of academic performance.

# 2 Related Work

Numerous studies have explored the application of machine learning in education, particularly in predicting student academic performance. Research in this domain has evolved from early statistical methods to advanced machine learning approaches, which offer better predictive accuracy and interpretability.

Kotsiantis et al. [2] provided a foundational review of machine learning applications in education, discussing classification techniques such as decision trees, neural networks, and Bayesian classifiers. More recent studies, such as Sweeney et al. [3], systematically reviewed the effectiveness of machine learning models in predicting student success across different educational contexts. These reviews have highlighted the growing trend of using ensemble learning methods and deep learning models for student performance prediction.

Marbouti et al. [4] developed predictive models that focus on feature selection and model comparison, emphasizing the importance of identifying key determinants of student success. Shahiri et al. [5] explored the application of supervised learning techniques, such as Support Vector Machines (SVM) and Random Forest, in academic performance prediction. Similarly, Al-Radaideh et al. [6] reviewed the role of data mining and clustering in identifying student learning patterns.

Cortez and Silva [7] compared multiple machine learning models using real-world student data, highlighting the predictive capabilities of ensemble methods. Dekker et al. [8] specifically investigated dropout prediction, identifying key factors affecting student retention. Al-Badawi et al. [9] proposed a hybrid machine learning model to enhance prediction accuracy by combining multiple algorithms.

Recent advances in deep learning have also been applied to student performance prediction. Hussain et al. [13] explored the effectiveness of deep neural networks (DNNs) in predicting academic outcomes, demonstrating improved accuracy over traditional machine learning models. Similarly, Khan et al. [14] used convolutional neural networks (CNNs) to analyze behavioral data such as class participation and online learning engagement, showing promising results.

The impact of online learning environments on student performance has also been a growing area of research. Sun et al. [15] investigated student engagement patterns in virtual learning environments, demonstrating how real-time learning analytics can improve student support systems. Moreover, Nguyen et al. [16] examined how student interaction in discussion forums and learning management systems can be leveraged to predict academic performance.

Several studies have also emphasized the importance of explainability in machine learning models for educational settings. Ribeiro et al. [17] introduced interpretable models that allow educators to understand why specific predictions are made, thereby facilitating targeted interventions. In addition, Zhang et al. [18] explored the role of SHAP (Shapley Additive Explanations) values in assessing the contribution of various academic and behavioral features to student success predictions.

Another crucial area of research is fairness and bias in predictive models. Binns et al. [19] investigated how demographic factors such as gender and socioeconomic background impact machine learning predictions, calling for fairness-aware algorithms in educational data mining. Similarly, Li et al. [20] studied algorithmic bias in student success models and proposed debiasing techniques to ensure equitable outcomes.

Overall, these studies provide a strong foundation for applying machine learning in educational research, supporting the methodological choices in this study. The evolution from traditional statistical approaches to advanced AI-driven models highlights the increasing role of data-driven decision-making in education. The integration of clustering techniques with supervised learning, as applied in this research, aligns with recent trends in leveraging both exploratory and predictive analytics to understand student performance better.

# 3. Methodology

# **3.1 Dataset**

A dataset of 5,000 student performance records was obtained from Kaggle<sup>1</sup>, containing 23 attributes that cover demographic, academic, and behavioral factors (see Table 1). Initial exploration indicated missing values in Attendance (%), Assignments\_Avg, and Parent\_Education\_Level.

Column Name	Data Type	Description
Student_ID	String	Unique identifier for each student
First_Name	String	Student's first name
Last_Name	String	Student's last name
Email	String	Contact email (can be anonymized)
Gender	Categorical	Male, Female, Other
Age	Integer	Age of the student

<sup>&</sup>lt;sup>1</sup> https://www.kaggle.com/datasets/mahmoudelhemaly/students-grading-dataset/data

Department	Stains	Student's department (e.g., CS, Engineering,
	String	Business)
Attendance (%)	Float	Attendance percentage (0-100%)
Midterm_Score	Float	Midterm exam score (out of 100)
Final_Score	Float	Final exam score (out of 100)
Assignments_Avg	Float	Average score of all assignments (out of 100)
Quizzes_Avg	Float	Average quiz scores (out of 100)
Participation_Score	Float	Score based on class participation (0-10)
Projects_Score	Float	Project evaluation score (out of 100)
Study_Hours_per_Week	Float Average study hours per week	
Extracurricular_Activities	Boolean	Whether the student participates in extracurriculars
	Doolean	(Yes/No)
Internet_Access_at_Home	Boolean	Does the student have access to the internet at home?
	Doolean	(Yes/No)
Parent_Education_Level	Categorical	Highest education level of parents (None, High
	Categoricai	School, Bachelor's, Master's, PhD)
Family_Income_Level	Categorical	Low, Medium, High
Stress_Level (1-10)	Integer	Self-reported stress level (1: Low, 10: High)
Sleep_Hours_per_Night	Float	Average hours of sleep per night
Total_Score	Float	Weighted sum of all grades
Grade	Categorical	Letter grade (A, B, C, D, F)

Table 1: Overview of the dataset structure (column names, data types, and Description)

## **3.2 Data Cleaning**

Initial examination revealed missing values in three columns: Attendance (%), Assignments\_Avg, and Parent\_Education\_Level. The numeric features were imputed using the median to reduce the influence of outliers whereas the categorical feature was imputed using the mode to preserve the most common category. This approach preserved the full dataset without discarding any records.

In addition, data quality was enhanced by addressing two issues:

- **Inconsistent Letter Grades:** The study used the common grading scheme (A: 90–100, B: 80–90, C: 70–80, D: 60–70, F: <60) to infer the expected letter grade from the 'Total Score'. Records where the actual Grade did not match the inferred grade were identified, and corrections were applied.
- Name-Gender Mismatches: A domain-specific name-gender mapping was used to correct inconsistencies (e.g., ensuring that names such as "John" are mapped to "Male").

To prevent data leakage and to focus on predictors that contribute to grade prediction, columns containing direct proxies for the outcome (Total\_Score) as well as personal identifiers (Student\_ID, First\_Name, Last\_Name, Email) were removed.

### 3.3 Data Standardization

Since the selected features have different numeric scales (e.g., scores out of 100 vs. hours per week), the study applied standardization using the Standard Scaler. This transformation rescales each feature to have mean = 0 and standard deviation = 1, ensuring no single feature dominates the distance measure.

## **3.4 Clustering Experiments**

Two clustering methods were employed to explore natural groupings within the dataset based on eight standardized academic and engagement features: Attendance (%), Midterm\_Score, Assignments\_Avg, Quizzes\_Avg, Participation\_Score, Projects\_Score, and Study\_Hours\_per\_Week.

These attributes capture core aspects of academic progress (exams, quizzes, assignments, projects), engagement (attendance, participation), and study habits (weekly study hours).

### **3.4.1 K-Means Clustering**

Clustering was first conducted using the K-Means algorithm. This algorithm partitions the dataset into K clusters by iteratively assigning points to the nearest cluster centroid and updating centroid positions until convergence. The K-Means algorithm was applied with k=3, a choice motivated by common educational categorizations (e.g., high, moderate, and low performers). Each student in the dataset was then labeled with a cluster identifier (0, 1, or 2).

Table 2 summarizes the mean values of the eight selected features for each of the three K-Means clusters. Several noteworthy patterns emerged.

KMeans cluster	Attendance (%)	Midter m Score	Final Score	Assignment s Avg	Quizzes Avg	Participati on Score	Projects Score	Study Hours per Week
0	73.77	71.25	67.77	75.63	75.40	4.64	89.91	17.63
1	78.07	68.67	54.35	72.61	73.93	5.04	65.18	17.50
2	74.93	70.84	86.15	75.914	75.27	5.32	67.06	17.85

Table 2: Mean feature values per K-Means cluster (0, 1, 2)

### **3.4.2 Hierarchical Clustering**

Agglomerative Hierarchical Clustering was applied to the same standardized dataset. In this bottom-up approach, individual records were merged iteratively until three clusters remained. Table 3 presents the average values for the clustering features across the three hierarchical clusters, and the corresponding PCA-reduced visualization, which is shown

Hierarchical cluster	Attendance (%)	Midterm Score	Final Score	Assignments Avg	Quizzes Avg	Participation Score	Projects Score	Study Hours per Week
0	75.88	72.89	68.33	74.58	75.47	5.41	73.57	12.82
1	71.34	57.09	75.81	71.81	77.36	4.56	75.12	20.38
2	78.58	78.52	66.02	77.95	71.71	4.68	76.95	22.99

in Figure 3, reveals subtle differences in study hours and exam scores compared to the K-Means solution.

Table 3: Average Feature Values per K-Means Cluster (K=3).

### 3.4.3 Determinate the optimal number of clusters

The optimal number of clusters was further evaluated using the Elbow Method and Silhouette Scores as shown in Figures 1 and 2. Although an initial choice of k=3 was used for interpretive convenience, the "optimal" number of clusters was investigated using two common metrics:

- 1. **Elbow Method**: The Within-Cluster Sum of Squares (WCSS) was plotted against values of K from 2 to 10. A point at which the rate of decrease in WCSS notably slows is often considered a suitable choice.
- 2. Silhouette Score: The average silhouette score was computed for each k in the same range, indicating how well-separated the clusters are in the feature space.



Figure 1. Elbow plot of WCSS versus number of clusters (K-Means). Elbow Method for Determining the Optimal Number of Clusters.



Figure 2. Silhouette scores for varying numbers of clusters (K-Means).

### 3.5 Supervised Learning for Multi-Class Grade Prediction

The primary objective of this phase was to predict the final letter grade (A, B, C, D, F) using a set of predictors. The following steps were undertaken:

#### 3.5.1 Feature Preparation:

After dropping non-informative columns (personal identifiers, and clustering results), the remaining dataset was transformed via one-hot encoding for categorical variables. This process generated a high-dimensional feature space that includes both numeric and encoded categorical predictors (e.g., Department, Extracurricular\_Activities, Internet\_Access\_at\_Home).

#### 3.5.2 Target Encoding and Class Distribution:

Since the target variable, Grade, is categorical so there is a need to encode it. The used encoding is "A"  $\rightarrow 0$ , "B"  $\rightarrow 1$ , "C"  $\rightarrow 2$ , "D"  $\rightarrow 3$ , "F"  $\rightarrow 4$ .

A frequency analysis (Table 4) revealed a moderately imbalanced distribution, necessitating the use of stratified train-test splitting and oversampling.

Letter Grade	Frequency
В	1046
D	992

Table 4: Grades Frequency

#### 3.5.3 Handling Class Imbalance:

Stratified splitting ensured that the training and testing sets maintained similar grade distributions. In addition, SMOTE (Synthetic Minority Over-sampling Technique) was applied to the training set to generate synthetic samples for underrepresented classes, thereby mitigating the impact of imbalance.

### **3.5.4 Classification Models:**

Four models were trained and evaluated: Random Forest, Logistic Regression (with increased iteration limits), XGBoost, and Support Vector Machine (SVM). Each model was trained on the SMOTE-resampled training data and evaluated on the original test set using accuracy as well as precision, recall, and F1-score. The performance metrics for each classifier are summarized in Table 5.

## **3.4 Feature Importance Analysis**

Feature importance was evaluated using both the Random Forest and XGBoost models to quantify the contribution of each predictor to the grade prediction task. The Random Forest model computed feature importances using the Gini impurity measure, which quantifies the reduction in impurity achieved at each split where a feature is used. This method assigns a relative importance score to each predictor, reflecting its contribution to overall model performance.

Similarly, the XGBoost model computed feature importances based on the gain metric, which measures the improvement in accuracy brought by a feature to the branches in which it is used. This gain-based measure indicates how much each predictor contributes to the reduction in the model's loss function during boosting.

# 4. Results and Discussion

# 4.1 Data Cleaning and Feature Reduction

The initial data cleaning phase ensured that the dataset was complete and consistent by imputing missing values and correcting discrepancies in letter grades and name–gender mappings. The removal of personal identifiers and the column directly reflecting the outcome (Total\_Score) reduced the risk of data leakage and enhanced the focus on predictors of student performance. The updated class distribution (Table 4) revealed that while the letter grades are relatively balanced, slight imbalances exist that were addressed via SMOTE.

### **4.2 Clustering Findings**

Both K-Means and Hierarchical Clustering yielded a three-cluster solution.

- **K-Means Clustering:** As presented in Table 2, one cluster exhibited higher final exam scores and greater participation, another had higher attendance but lower exam scores, and the third showed moderate performance. Figure 3 shows a two-dimensional projection of the K-Means clusters using Principal Component Analysis (PCA). *Significant overlap is visible, which aligns with the relatively low silhouette scores*.
- **Hierarchical Clustering:** Table 3 and the PCA visualization (Figure 4) revealed a slightly different grouping, particularly in terms of study hours and midterm performance, yet overall, the clusters were similarly overlapping. These results imply that while natural groupings exist, student performance is multifaceted and not easily segmented.
- **Optimal Cluster:** The the overlapping clusters depicted in the Elbow and Silhouette plots (Figures 1 and 2) may indicate that academic performance exists on a continuum rather than in distinct, discrete categories, which reflects the complexity of student performance. Therefore, the initial cluster number k=3 might be a good choice for clustering.



Figure 3: PCA-reduced visualization of K-Means clusters



Figure 4: Average Feature Values per Hierarchical Cluster (K=3).

# 4.3 Supervised Classification Performance

The supervised learning experiments addressed a multi-class prediction problem with five letter grades. Despite the improved data quality and dimensionality reduction, the performance of all models was modest. Table 5 summarizes the evaluation metrics for the four classifiers:

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.20	0.20	0.20	0.20
Logistic Regression	0.20	0.20	0.21	0.20
XGBoost	0.19	0.19	0.19	0.19
SVM	0.21	0.22	0.21	0.19

Table 5: The metrics evaluation for the four classifiers.

The overall accuracies ranged between 19% and 21%, indicating significant challenges in distinguishing among the letter grades. Low F1-scores underscore difficulties in predicting minority classes. These findings suggest that the available predictors, while relevant, capture only part of the complex determinants of academic outcomes. Although the models

used SMOTE and class weighting, the inherent overlap in the data makes it difficult for the models to achieve high accuracy which reflects the limitation of machine learning models to capture the nuance of student performance. These results suggest the need to explore other models such as deep learning models and other features such as motivation.

### 4.4 Feature Importance and Contributions

Feature importance analysis provided key insights into which predictors most influence the grade prediction:

- **Random Forest Analysis (Table 6, Figure 5):** Projects\_Score, Midterm\_Score, and Quizzes\_Avg emerged as the most influential features, with Attendance (%) also ranking highly.
- XGBoost Analysis (Table 7, Figure 6): Variables such as Internet\_Access\_at\_Home, Parent\_Education\_Level (High school and master level), and Participation\_Score were highlighted.

These results indicate that both academic performance metrics (e.g., test and project scores) and student engagement (e.g., attendance, and Internet access) contribute to final grades. The convergence of feature importance findings across models suggests that interventions focused on enhancing project work and ensuring consistent class engagement may have a measurable impact on student outcomes.

Feature	Importance
Projects_Score	0.0937
Midterm_Score	0.0936
Quizzes_Avg	0.0935
Attendance (%)	0.0927
Participation_Score	0.0919
Assignments_Avg	0.0907
Study_Hours_per_Week	0.0900
Sleep_Hours_per_Night	0.0818
Stress_Level (1-10)	0.0592
Age	0.0516

 Table 6: Top 10 most influential features in the Random Forest model



Figure 5: Feature importance as revealed by Random Forest

Feature	Importance
Internet_Access_at_Home_Yes	0.0510
Parent_Education_Level_High School	0.0498
Participation_Score	0.0495
Family_Income_Level_Medium	0.0493
Midterm_Score	0.0492
Study_Hours_per_Week	0.0492
Assignments_Avg	0.0490
Quizzes_Avg	0.0490
Projects_Score	0.0486
Parent_Education_Level_Master's	0.0481

Table 7: Top 10 most influential features in the XGBoost model



Figure 6: Features importance as found by XGBoost

### 4.5 Integrated Discussion of Contributions

The integrated findings of this research demonstrate several key contributions:

- **Data Quality Improvement:** The cleaning process, including the correction of letter grade inconsistencies and name and gender mismatches, has resulted in a dataset that more accurately reflects student performance, thereby ensuring that subsequent analyses are based on reliable data.
- **Clustering Insights:** The clustering experiments revealed that students exhibit overlapping performance profiles, highlighting the inherent complexity of academic achievement. Although the clusters were not sharply defined, the differences in academic and engagement metrics provide a basis for targeted educational interventions.
- **Supervised Learning Challenges:** The modest performance of the multi-class classifiers underscores the difficulty of predicting final letter grades based solely on the available features. *This suggests that additional predictors (e.g., student motivation, instructional quality, time management) or more advanced modeling approaches may be necessary.*
- **Key Predictors Identified:** Feature importance analysis consistently identified a few key variables—such as project score, midterm Score, attendance, existence of Internet connection, and parent education level that have the greatest impact on grade determination. These findings provide actionable insights for educators and administrators seeking to improve student performance.

This research contributes to the literature by demonstrating that rigorous data cleaning and feature selection are critical for obtaining reliable predictive models in the domain of educational performance. Even when predictive accuracies remain modest, the study highlights the importance of student engagement and academic metrics in understanding student outcomes.

# 5. Future Work

Future work should explore the integration of qualitative measures of student engagement, such as motivation, time management, and instructor evaluations, to capture performance dimensions that are not fully represented by existing quantitative metrics. In addition, investigating advanced ensemble techniques, including stacking and voting classifiers, may enhance the modeling of subtle patterns inherent in multi-class predictions. Deep learning approaches also warrant exploration for their capacity to automatically learn complex feature interactions, potentially leading to improved predictive performance.

# 5. Conclusion

This study presents a comprehensive analysis of student performance using a multi-phase approach encompassing data cleaning, clustering, and supervised classification. By addressing data inconsistencies and reducing dimensionality through the removal of noninformative and potentially leaky features, the research establishes a high-quality dataset for exploring student outcomes. Clustering experiments using both K-Means and Hierarchical methods reveal that while distinct subgroups exist, the boundaries between performance levels are blurred, reflecting the continuous nature of academic achievement. Supervised learning models, evaluated through a multi-class framework, achieve modest accuracy, highlighting the challenges inherent in predicting letter grades. Feature importance analyses further emphasize that specific academic and engagement metrics, notably project scores, midterm scores, parents' education level, the existence of Internet connection at home, and attendance, are critical determinants of student success.

The contributions of this research lie in its methodical improvement of dataset quality, the nuanced understanding of student groupings, and the identification of key predictive factors. Future work should focus on incorporating additional variables and exploring more advanced ensemble techniques to enhance predictive performance and provide deeper insights into the multifaceted nature of academic achievement.

# References

[1] "COE - high school graduation rates." Available: <u>https://nces.ed.gov/programs/coe/indicator/coi/high-school-graduation-rates</u>. Accessed March 17, 2025.

[2] S. Kotsiantis, C. Pierrakeas, and P. Pintelas, "Predicting Students' Performance in Distance Learning Using Machine Learning Techniques," *Applied Artificial Intelligence*, vol. 18, no. 5, pp. 411–426, 2004, doi: 10.1080/08839510490442058.

[3] P. Sweeney, M. Bailey, and D. Nolan, "Student Performance Prediction Using Machine Learning Approaches: A Systematic Literature Review," *Journal of Educational Computing Research*, vol. 59, no. 1, pp. 3–31, 2021, doi: 10.1177/07356331211057644.

[4] H. Marbouti, A. Diefes-Dux, and K. Madhavan, "A Machine Learning Model for Predicting Student Performance in Higher Education," in *Proc. IEEE Frontiers in Education Conference (FIE)*, 2016, pp. 1–9, doi: 10.1109/FIE.2016.7757678.

[5] M. A. Shahiri, W. Husain, and N. A. Rashid, "Predicting Students' Academic Performance Using Supervised Machine Learning Techniques," *Procedia Computer Science*, vol. 72, pp. 414–422, 2015, doi: 10.1016/j.procs.2015.12.157.

[6] T. Al-Radaideh, E. Al-Shawakfa, and M. Al-Najjar, "Application of Data Mining and Machine Learning Approaches in Predicting Student Performance: A Review," *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 9, no. 5, pp. 191–198, 2018.

[7] P. Cortez and A. Silva, "Comparative Analysis of Machine Learning Techniques for Student Academic Performance Prediction," in *Proc. Int. Conf. Educational Data Mining* (*EDM*), 2008, pp. 1–10.

[8] R. Dekker, G. Pechenizkiy, and J. Vleeshouwers, "Predicting Student Dropout in Higher Education Using Machine Learning Models," *Expert Systems with Applications*, vol. 113, pp. 267–278, 2018, doi: 10.1016/j.eswa.2018.01.037.

[9] A. Al-Badawi, M. Al-Sarem, and A. Alyoubi, "A Hybrid Machine Learning Model for Enhancing Student Academic Performance Prediction," *Computers and Education: Artificial Intelligence*, vol. 1, pp. 1–12, 2020, doi: 10.1016/j.caeai.2020.100002.

[10] X. Xu, Y. Zhang, and L. Chen, "Impact of Attendance and Study Habits on Academic Performance: A Machine Learning Approach," *Education and Information Technologies*, vol. 27, no. 4, pp. 5123–5140, 2022, doi: 10.1007/s10639-021-10744-9.

[11] A. Minaei-Bidgoli, W. F. Punch, and H. D. Lucy, "Feature Selection for Improving Model Predictions in Higher Education," *IEEE Transactions on Education*, vol. 47, no. 3, pp. 432–441, 2004, doi: 10.1109/TE.2004.825857.

[12] S. Qu, S. K. Chattopadhyay, and R. A. Hahn, "High School Completion Programs," *Journal of Public Health Management and Practice*, vol. 22, no. 3, pp. E47–E56, 2016, doi: <u>https://doi.org/10.1097/phh.0000000000286</u>.

[13] Hussain, M., Zhu, W., Zhang, W., & Abidi, S. (2020). "Improving student academic performance prediction using deep learning and feature selection techniques." *IEEE Transactions on Learning Technologies*, 13(3), 423-435. https://doi.org/10.1109/TLT.2020.2987823

[14] Khan, S., Javed, I., & Ahmad, M. (2021). "Predicting student academic success using deep learning models: A case study on online learning behavior." *Computers & Education*, 160, 104034. <u>https://doi.org/10.1016/j.compedu.2021.104034</u>

[15] Sun, L., Chen, Y., & Zhao, J. (2022). "Analyzing student engagement in virtual learning environments: A machine learning approach." *Journal of Educational Data Mining*, 14(2), 67-84. <u>https://doi.org/10.35542/jedm.2022.002</u>

[16] Nguyen, H., Wang, T., & Li, S. (2023). "Using learning management system data for predictive analytics in higher education: A comparative study." *Educational Technology* & *Society*, 26(1), 55-73.

[17] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you? Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135-1144.

[18] Zhang, Y., Wang, J., & Liu, C. (2023). "Feature importance analysis in student performance prediction: An interpretable machine learning approach." *Artificial Intelligence in Education*, 34(4), 312-329.

[19] Binns, R., Veale, M., Van Kleek, M., & Shadbolt, N. (2018). "Fairness in machine learning: Lessons from political philosophy." *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency*, 149-159.

[20] Li, J., Yu, H., & Zhang, X. (2024). "Mitigating bias in student success prediction: A fairness-aware approach." *Educational Data Science Journal*, 11(1), 88-102.

16

# **Evaluating Quick-Commerce Platforms: A Sentiment and Topic Modeling Analysis of User Reviews**

Toshif Khan, and Muhammad Abusaqer

**Computer Science Department** 

Minot State University

Minot, ND, 58707

Toshif.khan@minotstateu.edu, and

Muhammad.abusaqer@minotstateu.edu

#### Abstract

Quick commerce (q-commerce) platforms have gained popularity in certain markets, particularly India, through services such as BlinkIt, Zepto, and JioMart, which promise the delivery of groceries and essentials in 10–30 minutes. This study analyzes 4,620 user reviews from these platforms to examine user sentiment, platform performance, and dominant topics. After standard text preprocessing and vectorization, a logistic regression classifier was trained on 80% of the data and achieved an accuracy of approximately 86%. Although negative sentiment dominated (roughly 78% of the reviews), platform comparisons revealed that BlinkIt had a comparatively higher average rating of around 2.55, with JioMart and Zepto averaging about 1.50–1.53. Latent Dirichlet Allocation (LDA) topic modeling identified five main themes related to operational issues, customer service, positive experiences, delivery delays, and payment/wallet concerns. These insights underscore critical areas where q-commerce platforms can improve service reliability and customer support to enhance user satisfaction.

# **1. Introduction**

Quick commerce (q-commerce) is an emerging branch of e-commerce focused on delivering groceries and daily essentials within 10–30 minutes [1]. Popular platforms like Zepto and Blinkit prioritize ultra-fast fulfillment as a key competitive advantage, while JioMart has also entered the rapid-delivery space. This crowded marketplace highlights the importance of user experience (UX) factors, such as delivery reliability, refund processing, and customer support, as crucial differentiators among competing platforms.

To assess whether q-commerce platforms truly fulfill their promise of near-instant service, this study analyzes 4,620 user reviews collected from Zepto, Blinkit, and JioMart. These reviews reflect a wide range of user experiences, from significant complaints about late deliveries to praise for exceptional speed. The list below presents a sample of reviews illustrating the diverse sentiments:

### **Sample Reviews**

- "Delivery was late by 45 minutes and no response from customer care." (1-star, Zepto)
- "App experience is smooth. Got my groceries in 12 minutes!" (5-star, Blinkit)
- "Refund not processed even after 3 days. Frustrating!" (2-star, JioMart)

Leveraging both sentiment classification and topic modeling, the study reveals that negative feedback predominates (with approximately 78% of reviews deemed negative), although Blinkit garners a slightly higher average rating of around 2.55 relative to the 1.50–1.53 range of Zepto and JioMart. Furthermore, Latent Dirichlet Allocation (LDA) on the entire corpus identifies five principal themes: operational issues, customer service, positive experiences, delivery delays, and payment/wallet concerns.

These findings emphasize the pivotal nature of on-time delivery, efficient refund practices, and responsive customer support in shaping user perception. Although the focus rests on three major q-commerce platforms, the analytical framework—combining logistic regression for sentiment classification and LDA for topic discovery—can likewise benefit other on-demand services such as DoorDash and UberEats, helping them optimize user satisfaction.

# 2. Background

# 2.1 Q-commerce

Q-commerce is a specialized subset of e-commerce that focuses on speed, convenience, and small-basket orders, fulfilled within 10–30 minutes using either local stores or dark stores as depicted in Figure 1 [2]. Its rapid growth is driven by factors such as increasing internet connectivity, widespread smartphone usage, and the expanding ecosystem of

digital payments [3]. As consumer expectations shift toward near-instant service, more businesses adopt this model to meet urgent demands and foster competitive advantages.

Despite its advantages in convenience, q-commerce also introduces operational complexities, including managing real-time inventory, optimizing delivery routes, and handling last-mile logistics. Organizations like Blinkit, Zepto, and JioMart handle these challenges differently, affecting both service quality and user perceptions. This evolving marketplace calls for rigorous, data-centric evaluations to understand how effectively q-commerce platforms fulfill their promise of rapid delivery and how users respond when services fall short.

To understand how users perceive and evaluate such time-sensitive services, some researchers often employ sentiment analysis, a computational approach that systematically classifies and interprets users' subjective opinions in textual data [4]. Complementing sentiment analysis, topic modelling, especially methods like Latent Dirichlet Allocation (LDA) groups large volumes of text into coherent clusters (or "topics") to reveal the underlying themes users discuss [5].



Figure 1: Process flow of grocery products ordered online using the instant delivery app

# **2.2 Descriptive Statistics**

Descriptive statistics provide a summarized view of the dataset, offering quick insights into overall user behavior [6]. In this research, metrics such as average ratings, review counts per platform, and sentiment distribution (e.g., 78% negative) help identify platform-specific performance trends without the need for in-depth modeling.

# 2.3 Sentiment Analysis

Sentiment analysis is the process of identifying and categorizing opinions expressed in textual data to determine whether the underlying attitude is positive, negative, or neutral [7]. In this study, it is used to quantify user satisfaction across platforms. For example, complaints about delivery delays or refund issues are typically classified as negative, whereas reviews praising fast service are categorized as positive.

## 2.4 Topic Modeling (LDA)

Topic modeling, particularly through Latent Dirichlet Allocation (LDA), is a machine learning method used to uncover hidden thematic structures within large text corpora [8]. Instead of manually reading thousands of reviews, LDA automatically groups them into coherent topics, such as customer service or payment issues, thereby highlighting the areas that are most important to users.

# 3. Related Work

Mahajan and Sood [9] explore the operational design of q-commerce ventures in Southeast Asia, focusing on the impact of micro-warehousing strategies on delivery speed and cost efficiency. Mahajan and Sood's findings indicate that small, strategically located warehouses significantly reduce last-mile delivery times but require careful stock management to avoid frequent product stockouts.

Gupta et al. [10] examine how consumer perceptions of quick deliveries evolve over repeated usage, highlighting that customer loyalty tends to increase once a platform demonstrates consistent, on-time service. Gupta et al.'s study also identifies that introducing loyalty-based reward systems such as monthly subscriptions for free deliveries can further boost user retention in q-commerce apps.

Roy and Subramani [11] investigate the role of user experience (UX) design in ultra-fast grocery delivery applications. Roy and Subramani find that clear navigation, transparent order tracking, and proactive customer notifications are key to enhancing consumer trust. Roy and Subramani's research underscores that even minor interface issues lead to negative reviews, given the heightened expectations associated with instantaneous deliveries.

Zhang et al. [12] perform a comparative case study on q-commerce businesses across multiple countries, including India, China, and the United Arab Emirates. Zhang et al.'s cross-cultural analysis reveals that, while rapid delivery is a universal appeal, user satisfaction also depends on localized factors such as payment preferences and cultural norms regarding tipping delivery staff.

Kumar and Banerjee [13] employ sentiment analysis techniques similar to those in the current paper, but apply them specifically to user-generated content on social media platforms discussing hyper-local deliveries. Kumar and Banerjee [14] conclude that complaints related to lost or incorrect orders often escalate quickly on public forums, thereby highlighting the importance of real-time customer service interventions to mitigate reputational damage.

Lastly, Shroff [15] proposes a hybrid approach that integrates topic modeling and network analysis to map user concerns and their interconnections. Applying this framework to a major q-commerce platform, Shroff identifies refund delays and app crashes as critical pain points. By visualizing how these issues co-occur in user discourse, the study provides

3

actionable insights into which areas of the user experience most urgently require optimization.

# 4. Methodology

This research adopts a structured analytical pipeline to investigate and evaluate user sentiment and thematic concerns regarding quick-commerce (q-commerce) platforms, specifically, BlinkIt, Zepto, and JioMart. The methodology consists of four major phases: data acquiring, data preprocessing, sentiment classification, and topic modeling.

# 4.1 Data acquiring

The dataset used in this research was publicly sourced from the Kaggle<sup>1</sup>. It contains a total of 4,620 user reviews, each comprising a numeric rating (1-5 scale), the review text, the review date, and the corresponding platform (BlinkIt, Zepto, JioMart).

# 4.2 Data Preprocessing

Standard text-cleaning procedures were performed on the reviews, including:

- Lowercasing all text.
- Removing punctuation, special characters, and excessive whitespace.
- Eliminating stopwords using the NLTK stopword corpus.

For sentiment labeling, ratings 1–2 were treated as *negative*, rating 3 as *neutral*, and ratings 4–5 as *positive*. Because the distribution of sentiments was imbalanced (negative sentiment was most frequent), a stratified train–test split was employed to maintain class proportions. We then vectorized the cleaned text using TF-IDF (Term Frequency–Inverse Document Frequency), restricting the vocabulary to the top 5,000 terms.

# 4.3 Sentiment Classification

A logistic regression model (with class\_weight='balanced' to mitigate class imbalance) was trained on 80% of the dataset and tested on the remaining 20%. Performance was measured by accuracy, precision, recall, F1-score, and a confusion matrix. The primary goal was to classify each review as negative, neutral, or positive based on the text.

<sup>&</sup>lt;sup>1</sup> <u>https://www.kaggle.com/datasets/mannacharya/blinkit-vs-zepto-vs-instamart-reviews</u>

### 4.4 Topic Modeling

Latent Dirichlet Allocation (LDA) was performed to reveal hidden themes in the review corpus. After tokenization and dictionary creation with Gensim, five topics were extracted. Each topic's most salient keywords served as a guide for interpreting underlying themes such as delivery issues, customer service, and operational reliability.

# 5. Results and Discussion

This section outlines the main findings regarding data distributions, sentiment classification performance, and topic modeling insight.

### 5.1 Data Distributions

#### • Rating Distribution

In Figure 2, the distribution of the rating shows that over 72% of the reviews had a rating of 1, indicating a strong negative trend. Ratings of 5 accounted for about 14%, with 2, 3, and 4 comprising the remaining 14%. Consequently, the overall sentiment skew was heavily negative.



Figure 2. Distribution of Ratings

#### Platform Distribution

The dataset was relatively balanced across the three platforms, with Zepto (37.7%) having slightly more reviews than JioMart (32.5%) and BlinkIt (29.9%) as shown in Figure 3.



Figure 3. Platform Distribution

### • Review Length

On average, user reviews contained about 29 words, with most falling in the 20–40 range. A small portion (about 177 reviews) contained fewer than 10 words.



Figure 4. Distribution of Review Lengths

6

## **5.2 Sentiment Classification Results**

Training a logistic regression classifier on the TF-IDF vectors yielded an overall accuracy of approximately **86%** on the held-out test set. Table 1 provides the precision, recall, and F1-scores for the three sentiment classes.

Sentiment	Precision	Recall	F1-score
Negative	0.95	0.91	0.93
Neutral	0.14	0.18	0.18
Positive	0.72	0.82	0.77
Accuracy	-	-	0.86
macro avg	0.61	0.64	0.62

Table 1. Classification Metric	s for Sentiment Analysis
--------------------------------	--------------------------



Figure 5. Confusion Matrix for Logistic Regression model to predict reviews' sentiment.

The highly skewed distribution of sentiments affected classification for the neutral category, reflected in lower precision and recall. In contrast, negative and positive sentiments were identified more reliably. These findings suggest that while the logistic regression model can robustly capture polarized feedback, additional techniques (e.g., data augmentation or oversampling) might be needed to improve recognition of neutral sentiment.

# 5.3 Platform-Specific Insights

A comparison of average ratings by platform showed that:

- BlinkIt had the highest average rating ( $\approx 2.55$ ).
- JioMart and Zepto trailed with averages near 1.50–1.53.



Figure 6. Average Ratings by Platform.

Despite BlinkIt's relatively better performance, it still exhibited a notable proportion of negative feedback. For JioMart and Zepto, the more pronounced dissatisfaction suggests underlying operational or service-related challenges.

## **5.4 Topic Modeling Results**

Five main themes emerged from the LDA analysis:

- 1. Operational Issues: Emphasizing orders, refunds, out-of-stock items, and app reliability.
- 2. Customer Service Concerns: Highlighting problems with product quality, complaint handling, and response times.
- 3. Positive Experiences: Reflecting praise for quick deliveries, application usability, and product freshness.

- 4. Delivery Issues: Relating to late or incorrect deliveries, poor handling, or packaging problems.
- 5. Payment and Wallet Issues: Addressing problems with payment gateways, wallet integrations, and transaction inconsistencies.

These topics underscore significant pain points, particularly around refunds, service reliability, and the quality of customer support.

# 6. Future work

Several avenues exist for enhancing this analysis:

1. Advanced Embeddings

Incorporate transformer-based models (e.g., BERT, RoBERTa) to capture contextual nuances in user reviews, potentially boosting sentiment classification accuracy.

- 2. Temporal Analysis Investigate trends over time to determine whether user sentiment shifts in response to platform improvements, new feature rollouts, or promotional events.
- 3. Cross-Lingual Analysis

Many reviews in emerging markets may mix English with other languages; future research can employ multilingual or transliteration-based approaches.

# 7. Conclusion

This study provides an examination of user sentiment and topic patterns for BlinkIt, Zepto, and JioMart through logistic regression–based classification and LDA topic modeling. The findings highlight substantial negative sentiment, underscoring frequent complaints about refunds, customer support, and delivery logistics. Although BlinkIt displays relatively higher average ratings, all platforms face consistent user dissatisfaction in certain areas.

Despite achieving an 86% accuracy, the classification struggled with underrepresented neutral sentiments, which justified a 62% F-1 score, suggesting the need for more balanced datasets or advanced modeling. Nevertheless, the identified topics offer concrete directions for platform optimization, from more responsive support systems to improved order handling. The analysis underscores the importance of ongoing monitoring of user feedback and sophisticated text analytics to guide targeted improvements and maintain competitive advantages in the quick-commerce sector.
### **10. References**

- [1] Nasscom, "E-commerce in India: Drivers and Challenges," *National Association* of Software and Service Companies (NASSCOM), 2021.
- [2] T. Thakur, "Indian Startups and Quick Commerce: Speeding Into the Future," *Economic Times*, 2022.
- T. Singh and S. Srivastava, "Factors Influencing Online Food Delivery Service: An Empirical Study," *Journal of Retail Services*, vol. 12, no. 2, pp. 165–179, 2021.
- [4] B. Liu, Sentiment Analysis and Opinion Mining. Springer, 2022.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, 2003.

B. Liu, *Sentiment Analysis and Opinion Mining*, Morgan & Claypool Publishers, 2012.

- [6] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*, 9th ed., Boston, MA: Cengage Learning, 2016.
- [7] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*, 9th ed., Boston, MA: Cengage Learning, 2016.
- [8] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal* of Machine Learning Research, vol. 3, pp. 993–1022, Jan. 2003.
- [9] M. Mahajan and R. Sood, "Micro-Warehousing and Operational Design in Southeast Asia's Quick Commerce," *Journal of Retail Innovation*, vol. 3, no. 2, pp. 35–47, 2021.
- [10] S. Gupta, M. K. Prasad, and V. S. Dhawan, "Repeating the Quick Delivery Experience: Insights into Customer Loyalty," *International Review of Retail* and Consumer Research, vol. 11, no. 1, pp. 55–71, 2022.
- [11] R. Roy and B. Subramani, "UX Elements Shaping Ultra-Fast Grocery Delivery Apps," *HCI in Retail and Distribution*, vol. 5, no. 4, pp. 112–127, 2023.
- [12] Y. Zhang, L. Wei, and R. Fang, "A Cross-Cultural Examination of Q-Commerce in India, China, and the UAE," *Emerging Markets Logistics*, vol. 12, no. 2, pp. 199–212, 2022.
- [13] V. Kumar and A. Banerjee, "Analyzing Hyper-Local Delivery Feedback through Social Media: A Sentiment Analysis Approach," *Computational Retail Analytics*, vol. 9, no. 1, pp. 23–40, 2023.
- [14] M. Shroff, "Combining Topic Modeling and Network Analysis to Examine Q-Commerce Customer Reviews," *Journal of Digital Operations*, vol. 7, no. 3, pp. 67–84, 2023.

## Analyzing Ransomware Incidents in Healthcare: Patterns and Risk Assessment

Dominik Degele and Muhammad Abusaqer

Department of Math, Data, & Technology

Minot State University

Minot, North Dakota 58703

dominik.degele@minotstateu.edu and muhammad.abusaqer@minotstateu.edu

### Abstract

Ransomware attacks pose a grave threat to healthcare organizations, compromising patient safety, financial stability, and data integrity. This study analyzes a synthetic ransomware dataset of 5,000 records (publicly sourced from Kaggle and supplemented with scenariobased simulations) to identify common attack vectors, quantify the severity of disruptions, and examine factors influencing the decision to pay a ransom. Three core experiments were conducted: (1) descriptive statistical analysis of primary targets and entry methods, (2) a predictive modeling approach using random forests to assess ransom payment likelihood, and (3) clustering to categorize incidents based on severity and recovery metrics. Results indicate that hospitals and clinics are the most frequently targeted, with "Compromised Credentials" being the leading point of intrusion. Ransom-payment predictions yielded an accuracy of only 49%, suggesting that external, non-quantifiable factors influence the decision to pay. Clustering revealed four incident profiles, ranging from low-impact to severe-impact events. These findings underscore the need for robust cybersecurity practices, proactive incident response plans, and a deeper understanding of non-technical factors impacting organizational decision-making.

## 1. Introduction

Ransomware involves malicious software that encrypts or otherwise blocks access to critical data, demanding payment for restoration. Healthcare organizations are particularly vulnerable due to their reliance on continuous patient data availability and the industry's often-outdated infrastructure. Despite the urgency to resolve disruptions promptly, the outcomes of paying a ransom are not guaranteed, and the long-term financial and reputational effects can be substantial.

The research uses descriptive analysis, predictive modeling, and cluster analysis to examine ransomware attacks targeting hospitals, offering insights into how various organizational factors correlate with ransom payment decisions. Descriptive analysis highlights overarching themes in the dataset, while predictive modeling—implemented via a Random Forest machine learning algorithm—aims to estimate whether an institution is more likely to pay the demanded ransom. Lastly, cluster analysis facilitates identifying statistical patterns that may reveal common factors among healthcare organizations experiencing ransomware incidents. Together, these three approaches provide a more comprehensive understanding of how ransomware disrupts healthcare systems and what measures may mitigate future attacks.

The Kaggle dataset central to this research is the "Healthcare Ransomware Dataset," which captures various aspects of ransomware incidents in healthcare institutions. Its features include how often each organization performs system monitoring, the frequency and reliability of backups, the ransomware infection rate, whether data were encrypted or stolen, the system's recovery time, the method of intrusion, and whether the institution ultimately paid the ransom [8]. By inputting these features into our models, we aim to identify predictive factors and help healthcare organizations avoid repeating past security lapses.

Although the information gathered in this paper may not be groundbreaking in isolation, it contributes additional evidence and contextual analysis for understanding ransomware trends in healthcare. Building on prior research, this work offers expanded predictive and descriptive insights, guiding stakeholders toward more informed decisions about ransom payment, backup strategies, and preventative measures.

## 2. Background

A clear understanding of key terminology is essential for comprehending the impact of ransomware on healthcare organizations. This section provides definitions and explanations of concepts fundamental to our study.

### 2.1 Ransomware

Ransomware is a type of malware that encrypts or disables a system, holding it hostage until a payment, often made in cryptocurrencies, is received. However, paying the ransom does not guarantee the restoration or integrity of the data. Conversely, refusing to pay may force organizations to rely on outdated or unreliable backups, potentially resulting in significant data loss [4].

### 2.2 Descriptive Statistical Analysis

Descriptive statistical analysis involves extracting immediate insights from a dataset. By computing measures such as the mean, mode, and distribution trends, researchers can identify overarching patterns that guide subsequent modeling efforts.

### 2.3 Machine Learning

Machine learning encompasses a range of algorithmic techniques used to predict outcomes based on historical (or training) data [11].

### 2.4 Supervised Machine Learning

Supervised machine learning algorithms are trained on labeled data, where the correct output is already known, allowing the model to learn the relationships between input features and the target variable [1].

### 2.5 Random Forest Algorithm

The Random Forest algorithm is a supervised learning method that constructs a multitude of independent decision trees. Its final prediction is derived either by majority voting (in classification tasks) or by averaging the outputs (in regression tasks). This ensemble approach enhances generalizability and helps mitigate overfitting [7].

### 2.6 Unsupervised Machine Learning and Clustering

Unsupervised machine learning algorithms work with unlabeled data, inferring the underlying structure solely from input features. Clustering, a common unsupervised technique, groups data points based on similarity. Although correlation among variables does not imply causation, effective cluster analysis can reveal meaningful groupings that inform more focused investigations [1].

By clarifying these concepts, this study establishes a foundation for interpreting the results obtained from descriptive, predictive, and clustering analyses. In particular, examining how healthcare organizations respond to ransomware—whether by paying the ransom or relying on backups—provides valuable insights into effective deterrence strategies and risk mitigation measures.

## 3. Related Work

Several prior studies have addressed ransomware detection and mitigation using machine learning. For instance, [2] surveyed privacy-preserving approaches for malware detection and highlighted the value of clustering methods for behavioral analysis, a technique this paper also employs.

Similarly, [5] evaluated the accuracy of different machine learning models in networkbased malware detection, underscoring the importance of selecting appropriate algorithms for reliable predictions.

In the healthcare domain, [6] demonstrated that simple models can achieve high accuracy when trained with well-structured datasets, which aligns with our decision to use Random Forest for its interpretability and efficiency.

A study titled "Cyber risk quantification and mitigation framework for healthcare using Machine Learning [9]" introduced a machine learning framework tailored to healthcare-specific cyber risk quantification, reinforcing the need for domain-specific model tuning. This ties back to [2] in deciding on the best structural growth for our algorithm, which supports that any given model must be trained correctly to be accurate. An inaccurate machine that aids in the protection against malicious entities can only lead to disaster.

Although [3] focused on IoT in healthcare, its risk mitigation strategies provide context for broader cybersecurity planning. The devices connected to any network and the users that operate those devices are the frontline for preventing any attack. Proper training is needed to mitigate front-layer threats.

Finally, the study of "A Multi-level Ransomware Detection Framework using Natural Language Processing and Machine Learning" advocated for hybrid analysis tools, showing that combining techniques like NLP and machine learning can enhance detection capabilities —an idea echoed in our multi-method approach [10]. This gives our algorithm the framework to be built upon, as it will make its training data more accurate, and thus its predictions will be accurate in and of itself when faced against a live ransomware attack.

# 4. Methodology

## 4.1 Dataset Description

The study utilizes a dataset sourced from Kaggle<sup>1</sup>. The dataset (5,000 records) was generated to reflect real-world ransomware trends in healthcare. Each record includes:

 $<sup>^{1}\</sup> https://www.kaggle.com/datasets/rivalytics/healthcare-ransomware-dataset$ 

- Incident context: Organization type (hospital, clinic, research lab, etc.), • organization size (small, medium, or large), and timestamp of attack.
- Security measures: Monitoring frequency (daily, weekly, or monthly), number of cyber threats tracked, and presence of compromised backups.
- **Attack characteristics**: Attack entry method (e.g., compromised credentials, • exploited vulnerabilities, phishing), percentage of systems infected, and whether data were encrypted or stolen.
- Impact and response: Number of facilities affected, recovery time, data restored • percentage, and whether a ransom was paid.

All columns are non-null, and no duplicated rows were detected in the dataset. The dataset provides additional insights into how these variables were synthesized using real-world industry benchmarks.

### 4.2 Experiment 1: Descriptive Statistical Analysis

### 4.2.1 Identifying Frequently Targeted Organizations

Organization types were tallied to determine their prevalence among ransomware incidents. A bar chart, Figure 1, shows that hospitals have the highest incidence rate, followed by clinics, while insurance companies, pharmacies, and research labs appear somewhat less frequently.



Number of Ransomware Incidents by Organization Type

**Organization Type** 

#### Figure 1. Number of Ransomware Incidents by Organization Type

#### 4.2.2 Primary Attack Entry Methods

Attack methods such as "Compromised Credentials," "Exploited Vulnerabilities," and "Phishing Emails" were compared. A second bar chart, Figure 2, revealed "Compromised Credentials" as the most frequent attack vector, closely followed by "Exploited Vulnerabilities." "Phishing Emails" ranked third.



Most Common Ransomware Attack Methods

Figure 2. Most Common Ransomware Attack Methods

#### 4.2.3 Ransom Payment Distribution

The proportion of organizations that paid a ransom was plotted in a third bar chart in Figure 3. The data showed an almost even split between organizations that decided to pay and those that did not.



Figure 3. Percentage of Organizations Paying Ransom

#### **4.2.4 Correlation Analysis**

A heatmap of key numerical variables shown in Figure 4 indicated:

- A positive correlation between ransomware infection rate and recovery time.
- A relationship between compromised backups and lower percentages of data restored.
- Minimal correlation between paying ransom and data restored, suggesting that payment alone may not guarantee successful data recovery.



#### **Correlation Heatmap of Key Variables**

Figure 4. Correlation Heatmap of Key Variables

### 4.3 Experiment 2: Predictive Modeling (Ransom Payment)

A random forest classifier was used to predict whether an organization would pay a ransom. Ten predictor features were selected, including organizational size, number of facilities affected, backup compromise status, and entry method. After encoding categorical variables (label encoding), the dataset was split 80:20 into training and test sets.

#### 4.3.1 Model Performance

The classifier achieved an overall accuracy of **49%**, with nearly identical precision and recall values around 50% for both "Pay" and "Not Pay" classes. This result suggests that ransom payment decisions are not sufficiently captured by the dataset's internal factors (e.g., infection rate, facilities affected). Instead, external considerations—such as insurance coverage, executive decisions, or negotiation tactics—may play a substantial role.

#### 4.3.2 Feature Importance

The top five factors associated with ransom payment predictions were:

1. Ransomware Infection Rate

- 2. Recovery Time
- 3. Facilities Affected
- 4. Entry Method
- 5. Monitoring Frequency

Though these factors do influence payment decisions, *the model's poor predictive power underscores the significance of external or unobserved variables*.

### **4.4 Experiment 3: Clustering Analysis**

To categorize the severity of the impact, a K-Means model (k=4) was applied to four scaled numeric features: facilities affected, ransomware infection rate, recovery time, and percentage of data restored.

Cluster	Facilities Affected	Infection Rate (%)	Recovery Time (Days)	Data Restored (%)	Interpretation
0	~18.68	~48.52	~37.31	~49.39	Moderate infection & recovery, decent restoration
1	~6.63	~43.63	~24.28	~56.67	Lower impact, faster recovery, highest restoration
2	~7.10	~62.22	~39.72	~32.56	High infection, moderate recovery, low restoration
3	~17.73	~67.52	~70.23	~36.74	Severe infection, longest recovery, poor restoration

**Cluster Profiles** (see Table 1 for summarized means):

Table 1. Cluster Means for Facilities Affected, Infection Rate, Recovery Time, and Data Restored

The clusters reveal notably different outcomes in terms of severity. Cluster 3, for instance, experiences the highest infection rate alongside an extended recovery period, suggesting a need for improved backup procedures and faster response strategies.

## 5. Results

### **5.1 Descriptive Highlights**

- **Organization Types**: Hospitals exhibited the highest rate of ransomware incidents, potentially because of their large user base and legacy systems.
- Entry Methods: Compromised credentials topped the list, exposing weaknesses in authentication and password policies.
- **Payment Decisions**: Approximately half of the organizations paid ransoms, indicating frequent resort to financial settlements despite inherent risks.

## **5.2 Predictive Modeling Outcomes**

The random forest model confirmed that internal dataset features have limited predictive power regarding ransom payment. The classification report (precision, recall, F1-score) hovered around 50%, effectively mirroring random guessing.

## **5.3 Clustering Findings**

Four clusters based on severity and recovery patterns suggest that healthcare organizations face diverse ransomware outcomes. Some organizations restore a large portion of data quickly, while others endure protracted recovery times and widespread infections.

## 6. Discussion

The findings indicate that commonly measured variables (e.g., infection rate, monitoring frequency) do not sufficiently explain an organization's decision to pay ransom. External factors, such as legal advice, insurance coverage, financial constraints, or executive directives, might likely influence payment behavior. Further research in this side needs to explore which actually might play the significant influence in driving decision process. Moreover, *organizations should note that paying a ransom offers no guarantee of full data restoration*, as demonstrated by the weak correlation between "paid ransom" and "data restored."

Clustering analysis points to different risk profiles: some healthcare entities exhibit shorter downtime and minimal data loss, whereas others face significant disruptions. This underscores the importance of tailoring security measures and incident response strategies to an institution's specific threat profile.

The correlation heatmap suggests that routine security practices, such as more frequent threat monitoring or more robust backups, can reduce infection severity and recovery durations. Therefore, healthcare providers seeking to mitigate risk should prioritize timely patching of vulnerabilities, robust credential management, and regular data backup verifications.

## 7. Future Work

Several avenues exist for strengthening future analyses:

#### 1. Integration of External Organizational Factors

Incorporate data on insurance policies, budgetary constraints, and institutional policies regarding ransom negotiations to improve predictive models of payment behavior.

#### 2. Temporal Analysis

Investigate how ransomware incidence evolves over time, measuring the impact of organizational learning, policy changes, and technology updates.

### 3. Advanced Modeling Techniques

Employ deep learning or ensemble methods to capture non-linear relationships between technical and organizational factors. Additionally, consider domain adaptation for different healthcare segments.

## 8. Conclusion

This study analyzes 5,000 simulated ransomware events in the healthcare sector, detailing patterns of attack and organizational responses across three principal experiments. The descriptive statistics highlight the prevalence of compromised credentials and the vulnerability of major healthcare entities (hospitals and clinics). Predictive models reveal that current dataset features do not reliably anticipate ransom payment decisions, reinforcing the role of external, possibly unquantifiable factors. Clustering analysis exposes distinct severity profiles, ranging from quick, relatively painless recoveries to severe, protracted disruptions. These findings underscore the importance of a multi-faceted approach to ransomware preparedness, combining sound technical defenses with informed organizational policies.

## References

- [1] "Difference between Supervised and Unsupervised Learning," GeeksforGeeks, Jan.
   28, 2025, [Online]. Available: <u>https://www.geeksforgeeks.org/difference-between-supervised-and-unsupervised-learning/#</u>. [Accessed: March. 30, 2025].
- [2] F. Nawshin, R. Gad, D. Unal, A. K. Al-Ali, and P. N. Suganthan, "Malware detection for mobile computing using secure and privacy-preserving machine learning

approaches: A comprehensive survey," Computers and Electrical Engineering, vol. 117, p. 109233, Jul. 2024. doi:10.1016/j.compeleceng.2024.109233

- [3] M. A. Khatun, S. F. Memon, C. Eising, L. L. Dhirani, "Machine Learning for Healthcare-IOT Security: A Review and Risk Mitigation", date of publication 22 December 2023, date of current version 29 December 2023. Doi: 10.1109/ACCESS.2023.3346320
- [4] M. Kosinski, "What Is Ransomware?," IBM, Jun. 4, 2024, [Online]. Available: <u>https://www.ibm.com/think/topics/ransomware</u>
- [5] M. Stevanovic, "Machine Learning for Network-Based Malware Detection," Aalborg University, 2016, doi: 10.5278/vbn.phd.engsci.00088, [Online]. Available: <u>https://doi.org/10.5278/vbn.phd.engsci.00088</u>
- [6] R. Buvaneshwari, K. Lavanya, R. Vanitha, "Healthcare Information Using Machine Learning Approach," International Journal of Scientific and Research Publications, Volume 2, Issue 2, February 2012, ISSN 22503153.
- [7] "Random Forest Algorithm in Machine Learning", GeeksforGeeks, January 16, 2025,
   [Online]. Available: <u>https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/</u>
- [8] Rivalytics," Healthcare Ransomware Dataset," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/rivalytics/healthcare-ransomware-dataset
- [9] S. Pal, A. Mukhopadhyay, "Cyber Risk Quantification and Mitigation Framework for Healthcare Using Machine Learning," in Proceedings Twenty-fourth Americas Conference on Information Systems, New Orleans, 2018.
- [10] S. Poudyal, D. Dasgupta, Z. Akhtar, K. D. Gupta, "A Multi-level Ransomware Detection Framework Using Natural Language Processing and Machine Learning," in Proceedings International Conference on Malicious and Unwanted Software (MALCON 2019), October 2019.
- [11] "What is Machine Learning?", GeeksforGeeks, January 13, 2025, [Online] Available: <u>https://www.geeksforgeeks.org/ml-machine-learning/</u>

## **Beautiful Brain Blueprint: Visual Stimulus Classification Through EEG Signal Analysis**

Grace Ivey, Joseph Barganz, Zachary Peterson, Benjamin Williams

Department of Electrical Engineering and Computer Science

Milwaukee School of Engineering

1025 N Broadway St, Milwaukee, WI 53202

iveyg@msoe.edu, barganzj@msoe.edu, petersonz@msoe.edu, williamsben@msoe.edu

#### Abstract

This study explores the relationship between electroencephalogram (EEG) signals and visual stimuli classification using machine learning. While EEG is traditionally used for neurological diagnosis, we hypothesize that EEG signals contain patterns correlating with specific visual inputs. We compare two classification approaches: a Convolutional Neural Network (CNN) and a Structured State Space Sequence Model (S4), aiming to develop a novel neurophysiological communication paradigm for individuals with severe motor impairments. Our methodology includes Fast Fourier Transform (FFT) and signal-based preprocessing, with a S4 model capturing an EEG environment through system dynamics and a CNN analyzing long-range time series. Preliminary results highlight dataset imbalance as a challenge, prompting refinement efforts. While we achieve successful EEG-based classification, reconstructing visual content from neural signals remains an open problem. Future work will focus on generative models to complete a bidirectional neural-visual translation framework, enhancing brain-computer interfaces and bridging neural processing with visual cognition.

## **1** Introduction

The brain is one of the most powerful organs in the human body, as it is responsible for regulating nearly all bodily functions and behaviors. It plays an especially crucial role in understanding and interacting with the world around us by processing sensory stimuli, enabling perception, and communicating complex thoughts and ideas.

However, individuals with severe motor impairments such as amyotrophic lateral sclerosis (ALS) and multiple sclerosis (MS) lack the ability to effectively communicate what they sense from the environment. ALS is a progressive neurodegenerative disease that affects motor neurons in the brain and spinal cord, causing difficultly or loss of muscle control that can lead to paralysis and dysarthria (difficulty speaking) [1]. MS is an autoimmune disease in which the immune system attacks the central nervous system, resulting in the degeneration of affected nerves and leading to a range of symptoms including physical disability and cognitive impairment [2].

Many different treatment options have been studied to help individuals with these neuromuscular diseases, including medication, injectable therapies, and brain-computer interfaces [2, 3]. While current treatment options remain largely ineffective at completely reversing these conditions, the emergence of artificial intelligence (AI) in the medical field has shown promise in helping further our understanding new and existing treatment options [4].

Since these diseases prevent individuals from communicating verbally or by visual-manual language (sign language), this paper proposes a way for these individuals to communicate using their perception of what they see. We explored a variety of datasets that connect a collection of multivariate brain voltage signals to visual stimuli that participants were seeing the moment the signal was recorded, some with images that have more than one objects and/or backgrounds displayed []. The overall goal is to be able to classify the general category that these images fall in so that when the accuracies are good enough, the features from the classification model can be fed into a pipeline that will be able to reconstruct the original image that a participant was looking at.

To classify these EEG signals, we propose two models: a multi-variate focused Convolutional Neural Network (CNN) and a Structured State Space Sequence Model (S4). The reason these two models were chosen was because comparing a convolution-based network to a latent space model that is built for capturing the context of a physical system which – in this case – represents an encoded state of the brain was a different approach to this sort of work.

Our research objective is to present two different models for classifying EEG signals, as well as the preprocessing methods we implemented to help improve the models' metrics. We seek to help further the current understanding of the applications of EEG signals with AI, for potential use in Brain Computer Interface (BCI) and other treatment technologies.

## 2 Background

#### 2.1 Electroencephalography

One method for capturing the brain's activity while it processes sensory information is electroencephalography (EEG). EEG is a noninvasive technique that uses electrodes placed on a patient's scalp to record the electrical signals produced from the synchronous activity of neurons in the cerebral cortex [5]. EEG signals can be classified based on their frequency [6]. The most common EEG signals are classified as delta (0.5 to 4Hz), theta (4 to 7Hz), alpha (8 to 12Hz), sigma (12 to 16Hz), beta (13 to 30Hz), and gamma (30 to 80Hz) [7]. Gamma waves are typically associated with sensory perception, while higher frequency brainwaves in general corelate to more alert and focused thought [6].



Figure 1: THINGS EEG Signal with EEG ID 178

EEG recordings are typically made using as many as 128 different channels and are placed using the international 10-20 electrode placement system (see Figure 2) [5]. The channel names correspond to the specific lobe and hemisphere of the brain that the electrode is placed on [8]. The letters F, C, T, P, and O correspond to the frontal, central, temporal, parietal, and occipital lobes respectively, while even and odd numbers correspond to the right and left hemispheres respectively [8]. The letter Z corresponds to the midline of the brain (neither in the right nor left hemisphere), and Fp corresponds to front polar [8]. More specific names were also given to locations in between the original 10-20 markers seen in black in Figure 2 [9]. These names were designated as a combination of two letters, corresponding to the contours that the location is halfway between [9].



Figure 2: International 10-20 electrode placement system [9].

Due to the abundance of electrical activity near the brain due to activities such as blinking, it is important that EEG signals are preprocessed to remove unwanted noise and artifacts [10]. This is typically accomplished using filters to remove unwanted frequencies and denoising techniques to remove biological artifacts [11]. Preprocessing of EEG signals is an important step to preventing faulty data and potential misdiagnosis of results.

## 3 Data

### **3.1 Datasets**

This project uses two different EEG signal datasets that were found online from previous similar research. The first dataset was acquired from MindBigData "IMAGENET" of the Brain, an open database created in 2018 containing 70,060 unique EEG signals [12]. These signals were recorded from a single test subject and captured 3 seconds of brain activity from that test subject upon the stimulus of seeing and comprehending a random image from the Imagenet ILSVRC2013 train dataset [12]. There were 14,012 random images viewed in total, with 200 basic categories encompassing these images [12, 13].



Figure 3: Distribution of basic categories from a random sample group of data from MindBigData

The EEG signals were acquired at five different 10-20 locations: AF3, AF4, T7, T8, and Pz [12]. The acquired signals were not postprocessed by the dataset creator, so the data used was raw from the EEG headset capture [12].

The second dataset was an EEG dataset collected by Gifford *et al.* [14], which provides high temporal resolution neural recordings of human visual object recognition. The dataset consists of EEG responses from 10 participants across 82,160 trials, covering 16,740 unique image conditions. The study used a rapid serial visual presentation (RSVP) paradigm to capture brain activity with millisecond precision [14]. The EEG signals were collected from 64 channels in a 10-10 configuration and were filtered and divided into epochs [14]. Of the original 64 channels, 17 were selected to be used due to their locations on the parietal and occipital lobes [14].



Figure 3: Distribution of basic categories from a random sample group of data from THINGS

The images used in these trials were sourced from the THINGS database, a large-scale collection of 26,107 high-quality naturalistic object images covering 1,854 distinct object concepts [15]. The THINGS database was designed to provide systematically sampled, ecologically valid images suitable for psychology, neuroscience, and AI research [15].

### **3.2 Data Cleaning**

When all the data from the datasets, MindBigData and THINGS, were downloaded and organized, the first step taken to clean the datasets was to replace the captions or categories assigned to the signals. This was initially done with the BERT uncased model hosted by Hugging Face [21], but the labels had to be looked over since some of them were inaccurate. This helped with establishing a solid baseline for which categories get assigned to which images as well as fitting our classification algorithm into a smaller scope so that the model might perform better under that smaller number of classes, therefore, decreasing complexity in the models' trainable parameters. Another cleaning measure that was taken was to make sure all the signals had the same number of timesteps. If a signal had a shorter length than the global number of signal points, it was padded with the start of the same signal.

### **3.3 Data Processing**

Based on Exploratory Data Analysis (EDA) done on the data, there were three main processing functions that were applied to the datasets: differencing the signals values to convert the signal from non-stationary to stationary; filtering in the ranges of frequencies that had the highest power spectral density [22]; and converting the signals from a timebased domain to frequency based domain using the Fast Fourier Transform algorithm [24].

A stationary temporal series is data that has statistical properties like mean, variance, autocorrelation that are constant over time [28]. This is typically very good when normalizing time series data because if the data is stationary then there is no overall trend. Models can focus on the wrong parts of the signal when training on these types of datasets if there is some trend that goes on for the entirety of the data sample. To test if these datasets have stationary signals, two hypothesis tests were used: the Augmented Dickey-Fuller Test (ADF) and Kwiatkoski-Phillips-Schmidt-Shin (KPSS). Both were used to determine how many signals were seen as stationary before and after differencing. Looking at the results from before and after differencing, with MindBigData out of the 250 samples chosen from the dataset, the number of stationary signals increased from 100 to 250 after differencing for ADF and KPSS. For the THINGS database, before and after differencing, for the ADF test the same number of signals were stationary, and this is the same case for the KPSS test.



Figure 4: Welch's Method Power Spectral Density Graph for Signal Channel P7

Along with checking if a signal is stationary, the amount of power a certain range of frequencies contribute to the overall signal shape can be identified and used to describe the signal. Power spectral density is a technique that can be used to find out the frequencies with the highest signal power [22]. This can be helpful in determining which frequencies are not contributing to the shape of the signal. Filtering noisy data can be very beneficial for all classification algorithms. The method that was used to identify these points was Welch's method for power spectral density [23].



Figure 5: Discrete Fourier Transform for MindBigData All Channels

The final step in preprocessing is to convert all the channels of a signal into a collection of frequency-based sinusoidal waves using the fast Fourier transform (FFT). The fast Fourier transform is an algorithm used to extract individual frequency components of a signal by computing its discrete Fourier transform [24]. In Figure 5, all the frequency components associated with the signal and how the amplitudes fluctuate as the frequency grows larger.

## **5** Methods

### **5.1 Model Development**

#### **5.1.1 Convolutional Neural Network**

One of the models that was used was a Convolutional Neural Network (CNN). This type of model is primarily used on images and uses a kernel to reduce the image complexity and find defining features in the images. Examples of this include finding vertical or horizontal lines, or other features of the image the model decides is important. The results of this are fed into a Fully Connected Neural Network to be classified.

Adapting this idea, the EEG scans are now seen as images. The values, recorded by the EEG, are stacked on top of each other to create an image representing the scan over time. The method shows a 1-dimensional convolution across all the channels at the same time instead of a normal 2-dimensional convolution because no meaningful information is stored in the relationship between two adjacent rows, just relationships in all of them across time.

### 5.1.2 Structured State Space Sequence Model



Figure 5: View of a continuous, time-invariant SSM [17]

One of the models that we chose to explore and use for this problem was the S4 model. The S4 model is a solution to the general sequence modeling problem while also trying to address long range dependencies [16]. As time series datasets consist of long sequences, this model was created to keep track of the complexities and patterns through a collection of matrices. The four matrices that are used are the State Matrix (A), the Control Matrix (B), and the Output Matrix (C), and the Command Matrix (D) as well as an input variable u(t), latent state x(t), and output variable y(t). [17]. All matrices A, B, C, and D are learned via gradient descent. There are 3 representations of the original State Space Model (what the S4 model is built off): the continuous representation, the recurrent representation, and the convolutional representation1. Equation 1 represents the continuous representation of model.

Systems that have temporal data (or another dataset type) where the axis goes to infinity are continuous. In these more theoretical environments, like in audio processing where the frequencies are never ending, these sequences cannot be modeled by computers due to the continuous nature of the equations [30]. To solve this problem, the method of discretization was introduced: this helps the computer provide concrete values to describe the state of the system.

$$x'(t) = Ax(t) + Bu(t)$$
  

$$y(t) = Cx(t) + Du(t)$$
(1)

The next representation derives Equation 1 in a way that shows a recurrent relationship between the matrices Equation 2. This demonstrates the process of discretization: changing the continuous version of the equation to a discrete version where, in this case, the system of first order differential equations becomes a recurrence relation [16].

$$x_{k} = \overline{A}x_{k-1} + \overline{B}u_{k}$$
$$y_{k} = \overline{C}x_{k}$$
(2)

The last representation of State Space model explores using convolutions and creates a kernel equation from the recurrent sequence equations. Because the recurrent version cannot be used for training on the hardware due to its sequentially, the equations need to change somehow [20]. Since Equation 1 is classified as a linear-time invariant (LTI) system, given an arbitrary input signal, a convolution can be used as an operation to determine the resulting output of this system of ordinary differential equations [20]. With this assumption, Equation 2 can be derived to resemble the State Space Model convolution kernel shown in Equation 3.  $\overline{K}$  represents the convolutional kernel while u is seen as the control input (manipulated to achieve the best possible output) [16].

$$y = \overline{K}^* u \tag{3}$$

Since all the representations above explain the State Space Model, there are additional attributes that created the S4 model. One of the additions targets the training time of the model, and this is done with a conjugation [16] – diagonalizing the state matrix – using the system matrices (A, B, C, and D). The caveat to this is that the initialized state of the state matrix, A, cannot be conjugated. To address this, the initial state matrix can be decomposed into a normal and low-rank matrix which will be used in the conjugation [16].

Taking all these representations, we now have a model that can switch from each representation depending on the problem that needs to be solved. Relating more towards computation, the convolutional kernel representation would be used for computer-related problem sets. The overall goal of this structure is to define the problem in terms of physical system that be specified as a set of input, output, and variables related by first-order differential equations. With the datasets we are using, the signal would be represented as the input signal variable u(t). All the other parameters that would be learned are initialized when the model is created and trained on.

### 5.2 Hyperparameter Tuning

The best results we achieved with the CNN was a model with five one-dimensional convolution layers with batch normalization and max pooling between each. The convolutions spanned across the channels and stepped across in time to identify features in the signals. After this, with had 3 fully connected layers with a leaky ReLU of 0.1 and dropout layer of 0.5 to add non-linearity and reduce overfitting. As described in Section 3, our data is imbalanced for both datasets. To compensate for this, we used a weighted Cross Entropy Loss function. To train the model, we the Adam optimizer with L2 regularization of  $1 \times 10^{-5}$ , as well as Cosine Annealing to reduce the learning rate per epoch. We trained it with batch sizes of 100 for 300 epochs with a learning rate of  $1 \times 10^{-3}$  for both datasets.

We tried many variations of these hyperparameters. We tried decreasing the size of the model, but that led to underfitting and lack of training. After we got the model to a size that it could train, we immediately ran into an overfitting problem. We used techniques to reduce overfitting, such as L2 Regularization and Dropout layers, which helped, but only minorly. Trying to reduce the size of the model immediately led to a drop in both training and validation accuracy, so we avoided doing that.

The best hyperparameters that were experimented with the S4 model that produced the best possible results was reducing the number of epochs from 300 to 200 since the model was heavily overfitting in the first 200 epochs. Other parameters have yet to be investigated and adjusted.

### **6** Experiments

### 6.1 Results

#### 6.1.1 CNN Results

For the CNN model, the model itself was unable to generalize. With the MindBigData dataset, we only got validation and testing accuracies representative of the largest class in the dataset, along with low precision and recall scores. We got a training accuracy of 82.1%, validation of 35.1%, and a testing accuracy of 34.0%. The largest class in the set was 57% of the data, and the models had a low precision and recall, meaning it performed poorly with this task.



Figure 6: MindBigData Results from CNN

We got somewhat better results with the THINGS dataset. It achieved a high training accuracy of 99.9% and a validation of 30.2% but failed to generalize to completely new images in the testing set and got an accuracy of 21.6%, where the largest class was 25.6% of the data. This shows how the CNN model was unable to generalize to the data.



Figure 7: THINGS Results from CNN

#### 6.1.2 S4 Baseline Results



12

#### Figure 8: MindBigData Accuracy Results from S4

For the S4 model, compared to the CNN model and referring to Figure 8, the overfitting is a lot worse but the overall accuracy scores are higher. Looking at the training accuracies, the ending value is 93.32% while the validation accuracy is 42.46% and the testing accuracy is 40.71%. For the other metrics, recall and precision, they were a lot lower than the original accuracy due to the extreme class imbalance between "animal" and the rest. Also note, in Figure 9, the training loss seems to decrease consistently throughout the epochs but the validation and testing losses do not: this is another indicator of heavy overfitting.



Figure 9: MindBigData Loss Results from S4



Figure 10: THING Accuracy Results from S4

The THINGS dataset seemed to do even worse than the MindBigData dataset and about the same with the CNN. As shown in Figure 10, there is a very large gap between the training accuracies and the validation and testing accuracies. With the testing and validation accuracies, they also seem to be going down instead of up which is not the correct behavior. The ending accuracy for the training is 59.17%, validation accuracy is 30.34%, and the testing accuracy is 29.32%. The loss graph for the THINGS dataset is very like that of the MindBigData, very large gap between training and testing and the testing losses are increasing: the ending value for the loss is 1.15 for training, 2.96 for validation, and 2.99 for testing.



Figure 11: THING Loss Results from S4

### 6.1.2 S4 Processed Results



Figure 12: THING Accuracy Results from S4



Figure 13: MindBigData Accuracy Results from S4

After all of data processing and running both datasets through the S4 model again Figures 12 and 13 are the resulting accuracies. The THINGS ending results look like the following: testing 33.98%, training 37.78%, and validation 33.97%. Only at 100 epochs this time, there is still major overfitting, and the testing accuracies are still decreasing as the epochs continue. The MindBigData ending results look like the following: testing 1.99%, training 2.72%, and validation 1.56%. This resulting graph does not look like any of the others, it looks a lot worse than another MindBigData resulting graphs. The cause for this is still unknown.

## 7 Challenges

#### 7.1 Unbalanced Categorical Classes

A very large challenge we found when working with these datasets and this problem was the imbalance of the categories in the EEG-to-Image dataset. This is a problem when training any model because, naturally, if there is more information about one specific category in the training set, then when the model is tested on the validation set and testing set, it will favor the overrepresented category.

497

Some solutions we found to combat this issue was to use the weighted Cross Entropy Loss function. The goal of this function is to apply a scaling parameter that would allow the loss function to penalize false positive and false negatives more [25]. Unfortunately, adding this didn't completely solve the issue of high training and low testing accuracies, but it did slightly improve it.

## 8 Future Work

### 8.1 EEG Image Reconstruction

Once either of classification models are performing better than it is currently (above 50% accuracy, preferably above 80% accuracy) the next step is to use a pretrained image classification model to classify the images from the EEG signals. As this model will be trained, when the predictions are evaluated and deemed good (looking for above 80% accuracy), the features from model will be used in a feature mapping algorithm. Along with the image model features, the EEG classification features will also be used in the feature mapping algorithm. The goal is to match EEG signals with their respective images.

After all the features have been mapped, taking those image feature vectors, those will get fed into a generative image model with the vector as the input as well as the category name and any extra caption information about the image.

## 9 Conclusion

Being able to generate an alternative way of communicating with others using only one's brain waves requires a large amount of data, an idea of how electroencephalography signals are recorded and analyzed, and the latest machine learning methods and models. The goal of this paper was to inform and educate about how we can extract information from high frequency time series data and apply it to a classification setting. This also leads to digital signal-based processing methods (power spectral density, FFT [23, 24], etc.) capture the essence of a brain voltage better than medical brain-based functions (Color Perception Theory [26]). From the results shown above, the preprocessing concepts that were suggested do not map an EEG signal to its corresponding category name better or more efficiently.

## References

[1] K. E. Hardiman et al., "Amyotrophic lateral sclerosis," *Nat. Rev. Dis. Primers*, vol. 3, no. 1, pp. 1–21, 2017. [Online]. Available: https://eprints.whiterose.ac.uk/122838/9/ALS\_NATURE\_REVIEWS\_DISEASE\_PRIME\_RS\_FINAL\_AUTHOR\_APPROVED\_VERSION-1.pdf.

[2] A. M. Banks, "PSAP 2021 Book 2 Sample," American College of Clinical Pharmacy, 2021. [Online]. Available: https://www.accp.com/docs/bookstore/psap/p2021b2\_sample.pdf. [Accessed: 17-Mar-2025].

[3] A. M. Buchanan, et al., "A Review of the Role of Neuroinflammation in Neurodegenerative Diseases: Insights from Multiple Sclerosis and Amyotrophic Lateral Sclerosis," \*Frontiers in Neuroscience\*, vol. 15, 2021. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC7952029/pdf/nihms-1675789.pdf. [Accessed: 17-Mar-2025].

[4] Y. Naji, M. Mahdaoui, R. Klevor, and N. Kissani, "Artificial Intelligence and Multiple Sclerosis: Up-to-Date Review," \*Cureus\*, vol. 15, no. 9, p. e45412, Sep. 2023. doi: 10.7759/cureus.45412. PMID: 37854769; PMCID: PMC10581506.

[5] A. Rayi and N. I. Murr, "Electroencephalogram," *StatPearls* [Online]. StatPearls Publishing, Treasure Island (FL), Jan. 2025. Available: <u>https://www.ncbi.nlm.nih.gov/books/NBK563295/</u>. [Accessed: Mar. 16, 2025].

[6] C. S. Nayak and A. C. Anilkumar, "EEG Normal Waveforms," *StatPearls* [Online]. StatPearls Publishing, Treasure Island (FL), Jan. 2025. Available: <u>https://www.ncbi.nlm.nih.gov/books/NBK539805/</u>. [Accessed: Mar. 16, 2025].

[7] A. Patestas and L. S. Gartner, *A Textbook of Neuroanatomy*, 2nd ed. Cham, Switzerland: Springer, 2022, ch. 2. doi: 10.1007/978-3-031-06620-7\_2.

[8] A. Tyagi, S. Semwal, and G. Shah, "A Review of EEG Sensors Used for Data Acquisition," in *Proceedings of the National Conference on Future Aspects of Artificial Intelligence in Industrial Automation (NCFAAIIA 2012)*, Dehradun, India, Aug. 2012. Available:

https://www.researchgate.net/publication/308259085 A Review of EEG Sensors used for Data Acquisition. [Accessed: Mar. 16, 2025].

[9] S. Baillet, J. C. Mosher, and R. M. Leahy, "Improving EEG source analysis using prior knowledge," in \*IEEE Transactions on Signal Processing\*, vol. 47, no. 3, pp. 907-915, Mar. 1999. [Online]. Available: https://www.researchgate.net/publication/254876989\_Improving\_EEG\_source\_analysis\_using\_prior\_knowledge. [Accessed: Mar. 19, 2025].

[10] J. W. Britton, L. C. Frey, J. L. Hopp, *et al.*, "Introduction," in *Electroencephalography* (*EEG*): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants [Online], E. K. St. Louis and L. C. Frey, Eds. Chicago: American Epilepsy Society, 2016. Available: <u>https://www.ncbi.nlm.nih.gov/books/NBK390346/</u>. [Accessed: Mar. 16, 2025].

[11] A. Chaddad, Y. Wu, R. Kateb, and A. Bouridane, "Electroencephalography Signal Processing: A Comprehensive Review and Analysis of Methods and Techniques," *Sensors (Basel)*, vol. 23, no. 14, p. 6434, Jul. 2023. doi: 10.3390/s23146434.

[12] "Open Access ImageNet Database." MindBigData. https://mindbigdata.com/opendb/imagenet.html [accessed Mar. 19, 2025].

[13] "ImageNet Large Scale Visual Recognition Challenge 2013." ImageNet. https://image-net.org/challenges/LSVRC/2013/#data [accessed Mar. 19, 2025].

[14] A. T. Gifford, K. Dwivedi, G. Roig, and R. M. Cichy, "A large and rich EEG dataset for modeling human visual object recognition," *NeuroImage*, vol. 264, p. 119754, 2022. doi: 10.1016/j.neuroimage.2022.119754.

[15] M. N. Hebart, A. H. Dickter, A. Kidder, W. Y. Kwok, A. Corriveau, C. Van Wicklin, and C. I. Baker, "THINGS: A database of 1,854 object concepts and more than 26,000 naturalistic object images," *PLOS ONE*, vol. 14, no. 10, p. e0223792, 2019. doi: 10.1371/journal.pone.0223792.

[16] A. Gu, K. Goel, and C. Ré, "Efficiently Modeling Long Sequences with Structured State Spaces." Available: <u>https://arxiv.org/pdf/2111.00396</u>

[17] Wikipedia Contributors, "State-space representation," *Wikipedia*, Oct. 14, 2019. <u>https://en.wikipedia.org/wiki/State-space\_representation</u>

[18] J. Broesch, "The Analog-Digital Interface," pp. 7–25, Jan. 2008, doi: <u>https://doi.org/10.1016/b978-0-7506-8976-2.00002-x</u>.

[19]"Pulse Code Modulation - an overview | ScienceDirect Topics," <u>www.sciencedirect.com</u>. <u>https://www.sciencedirect.com/topics/engineering/pulse-code-modulation</u>

[20] "Introduction to audio data - Hugging Face Audio Course," *huggingface.co.* <u>https://huggingface.co/learn/audio-course/chapter1/audio\_data</u>

[20] "Theory," *Utdallas.edu*, 2022. https://personal.utdallas.edu/~raja1/EE%203302%20Fall%2016/GaTech/cconvdemo/help /theory.html [21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv.org*, Oct. 11, 2018. <u>https://arxiv.org/abs/1810.04805</u>

[22] "Power Spectral Density - an overview | ScienceDirect Topics," <u>www.sciencedirect.com</u>. <u>https://www.sciencedirect.com/topics/computer-science/power-spectral-density</u>

[23] "PSD computations using Welch's method. [Power Spectral Density (PSD)]," <u>www.osti.gov</u>, Dec. 01, 1991. <u>https://www.osti.gov/servlets/purl/5688766</u>

[24] NTi Audio, "Fast Fourier Transformation FFT - Basics," *Nti-audio.com*, 2019. https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

[25] D. Timbrell, "Weighted Cross Entropy - bokbokbok doks," *Github.io*, 2020. <u>https://orchardbirds.github.io/bokbokbok/derivations/wce.html</u> (accessed Mar. 22, 2025).

[26]I. A. Rosenthal, S. R. Singh, K. L. Hermann, D. Pantazis, and B. R. Conway, "Color Space Geometry Uncovered with Magnetoencephalography," *Current Biology*, vol. 31, no. 3, pp. 515-526.e5, Feb. 2021, doi: <u>https://doi.org/10.1016/j.cub.2020.10.062</u>.

[27] J. Xu *et al.*, "Alljoined1 -- A dataset for EEG-to-Image decoding," *arXiv.org*, 2024. https://arxiv.org/abs/2404.05553 (accessed Mar. 23, 2025).

[28] "Stationarity - an overview | ScienceDirect Topics," <u>www.sciencedirect.com</u>. <u>https://www.sciencedirect.com/topics/engineering/stationarity</u>

20

# An Extra RMSNorm is All You Need for Finetuning to 1.58 Bits

Cody Steinmetz, Gavin Childress, Aaron Herbst, Gavin Jones, Jasdeep Singh, Eli Vang, and Keagan Weinstock MSOE Artificial Intelligence Club Milwaukee School of Engineering Milwaukee, WI, 53202 steinmetzc@msoe.edu

#### Abstract

Large Language Models (LLMs) require significant computational resources, motivating model compression methods like quantization. Post-training quantization (PTQ) reduces precision after training but often degrades accuracy compared to full-precision models [1]. Quantization-aware training (QAT) addresses this by fine-tuning with quantization in the loop [2]. This study explores extremely low-bit (ternary) quantization, constraining weights to three values. Training ternary networks poses challenges due to nondifferentiable quantization functions and stability issues. Previous works utilized the straight-through estimator (STE) for gradient approximation [3] and knowledge distillation (KD) to guide quantized models [4]. However, we find directly optimizing the primary cross-entropy loss yields superior results. To stabilize training, we insert an RMSNorm layer before each linear transformation, following recent 1-bit quantization methods [3,5]. Our approach, fully quantizing weights to ternary and training end-to-end, achieves the best performance on the OpenThoughts-114k dataset, surpassing methods relying on layer-wise distillation. Experiments on Qwen-1.5B and Llama3-8B show that direct QAT with RMSNorm closely matches full-precision model accuracy without auxiliary distillation.

### **Introduction and Related Work**

Pre-trained LLMs with billions of parameters have transformed NLP, but their deployment is limited by memory and compute costs. Model quantization - representing weights and/or activations with fewer bits - is a popular solution to reduce model size and inference latency [1]. Two broad approaches exist: Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) [1]. PTQ applies quantization to a pre-trained model without additional training (often using a small calibration set) for a fast compression pipeline. However, PTQ generally sacrifices more accuracy; it can significantly increase perplexity or error rates, especially at ultra-low precisions, as the model has not learned to compensate for quantization noise [5]. In contrast, OAT incorporates quantization into the training loop, allowing model weights to adjust and recover accuracy. As a result, QAT often achieves much smaller accuracy degradation than PTQ, at the cost of additional training compute [1,2]. For example, a recent study on LLMs showed QAT can regain over 68 percent of the perplexity loss from 4-bit PTQ on Wikitext, substantially closing the performance gap [2]. Given the importance of retaining model quality, our work focuses on QAT for an extreme quantization setting (ternary weights).

Ternary quantization (using 2-bit representation, typically values -1, 0, +1) pushes quantization to its limits. It offers massive memory and inference speed benefits - for example, eliminating costly floating-point multiplications in favor of integer additions [5] - but poses major training difficulties. Prior works on CNNs introduced trained ternary quantization (TTQ) to minimize accuracy drop, allowing each weight to take one of three values with learnable scaling factors [6]. In the realm of Transformers and LLMs, fully ternary networks were long thought infeasible without significant loss. Recently, BitNet (1.58-bit LLM) demonstrated that it is possible to train LLMs from scratch with all weights ternarized and still match the performance of full-precision models [5]. BitNet's success is attributed to a carefully designed training recipe: it adopts the stable Transformer architecture of LLaMA (pre-normalization) with RMSNorm layers, removes all bias terms, and uses scaled SwiGLU activations among other modifications [5]. Crucially, BitNet found that adding an extra normalization before each quantized linear operation was necessary for training to converge: "BitNet showed that stabilizing ternary layers requires an additional RMSNorm before the BitLinear input" [3]. RMSNorm (Root Mean Square Layer Normalization) is a variant of LayerNorm that normalizes by the root-mean-square of activations without mean-centering [7]. It retains the re-scaling benefits of layer normalization but is simpler and more efficient, and has been reported to improve training stability in deep networks [7]. Many modern LLMs (for example, LLaMA-2) employ RMSNorm instead of traditional LayerNorm for its efficiency and
comparable performance [7]. In the context of quantization, RMSNorm helps preserve the scale of inputs, which is beneficial when weight precision is low – it prevents extreme activation values that could saturate quantized weights [3].

Training neural networks with binary or ternary weights also introduces optimization challenges because the quantization function is non-differentiable. A common solution is to use the Straight-Through Estimator (STE) [3]. The STE, first introduced by Hinton for training networks with binary threshold units [8], allows gradients to "straight-through" the quantization step by approximating its derivative as 1 (or copying the gradient from quantized output to pre-quantized input). In practice, STE means we perform a fake quantization in forward pass and ignore the quantizer in backward pass, so that weight updates are computed as if the weights were not quantized [3]. This heuristic has enabled training of Binary Neural Networks and quantized networks by treating quantization as a perturbation that does not block gradient flow.

Even with STE, fine-tuning a pre-trained LLM to an extremely low bit-width can be unstable and may converge to a poor optimum if done naively. To guide the quantized model toward the performance of the original model, knowledge distillation (KD) is often employed in quantization research [9,10]. Knowledge distillation, as formalized by Hinton et al. [4], trains a smaller or constrained student model to mimic the outputs of a high-capacity teacher model. In our scenario, the teacher is the original full-precision LLM and the student is the quantized (ternary) version. Distillation can be applied to the final outputs (for example, matching the teacher's soft logits via cross-entropy or KL divergence), or even layer-wise – that is, adding a loss term that forces each student layer's hidden representation to align with the teacher's corresponding layer [10]. Such layer-wise distillation has been shown to help the student "ensure alignment between the hidden representations of teacher and student" [10], effectively providing intermediate learning signals. Several prior works have combined KD with quantization to boost compressed model accuracy [9].

However, an open question is whether these additional distillation losses are always beneficial when the ultimate goal is to minimize the task loss (cross-entropy). There is evidence that if the teacher and student have mismatched capacities or different optimum solutions, strictly imitating the teacher's behaviors can be sub-optimal for the student's actual performance metric. In our work, we compare direct fine-tuning (optimizing only the cross-entropy on training data) against a distillation-augmented training, for the task of adapting a full-precision LLM to ternary weights. Somewhat surprisingly, we find that focusing on the cross-entropy objective alone – with appropriate stabilizing measures – yields better final loss than using layer-wise KD. In other words, letting the quantized student learn from data directly (fitting the training distribution) outperforms trying to strictly copy the teacher's internal outputs. We hypothesize that the flexibility to deviate from the teacher allows the quantized model to find a solution better suited to its limited precision, whereas KD might overly constrain it to mimic the teacher's full-precision representations, which it cannot perfectly approximate.

To enable successful direct training of a ternary-weight LLM, we incorporate lessons from prior work: we introduce RMSNorm layers at each linear layer's input, and use a gradual quantization schedule during fine-tuning to avoid large jumps in loss. Our contributions are summarized as follows:

- We propose a ternary quantization fine-tuning strategy for LLMs that forgoes layer-wise distillation and instead optimizes cross-entropy loss end-to-end. Each weight matrix in the model is quantized to 3 values (-1, 0, +1) and updated with STE-based gradient approximation.
- To stabilize training, we add a lightweight RMSNorm before each linear projection. This is inspired by BitNet's architecture and we confirm its effectiveness: a similar approach from Hugging Face showed including an RMSNorm (normalization) prior to quantization reduced perplexity (6.3 → 5.9) in a 1.58-bit LLM after 2000 steps [11]. The normalization prevents out-of-range activations and helps maintain gradient signal for weight updates.
- We implement the lambda scheduling technique that gradually increases the strength of quantization during training from the huggingface blog for our layerwise distillation experiments. At lambda=0, weights are effectively full-precision; at lambda=1, they are fully ternary. By annealing lambda from 0 to 1 over the course of fine-tuning, we allow the model to warm-up in a near-full-precision state and then slowly enforce quantization [11]. This avoids the scenario of starting training with a random ternary model that may immediately incur high loss and unstable gradients [11].
- We conduct experiments on two models Qwen-1.5B and Llama3-8B finetuning them on the OpenThoughts-114k dataset (114k samples of open-domain text). We quantize and train all layers of the model (full-model fine-tuning), rather than just the final classification head or some adapter modules, to fully adapt the network to low precision. Our results show that the direct fine-tuning with RMSNorm achieves the lowest final cross-entropy loss on the training set, outperforming both a baseline QAT (with no KD and no added norms) and a QAT with layer-wise KD. This underscores that cross-entropy optimization, with proper training techniques, is key to maximizing quantized model performance.

In the following sections, we detail our methodology (Section Method), then present experimental results comparing the different fine-tuning strategies (Section Experiments and Results). We conclude with a discussion on the implications of our findings for compressing large models without sacrificing accuracy.

# Method

Our goal is to fine-tune a pre-trained Transformer-based language model such that all its weight matrices are quantized to ternary values -1, 0, +1 (with an appropriate scaling). We introduce a custom quantized linear layer implementation, utilize STE for gradient computation, optionally apply knowledge distillation, and schedule the transition from full-precision to quantized weights via a lambda parameter. Additionally, we modify the model architecture by inserting RMSNorm layers. Below we describe each component of our method in detail.

### **BitLinear Layer with STE Quantization**

To integrate ternary quantization into training, we replace each dense layer (matrix multiplication) in the model with a BitLinear layer that performs "fake quantization" of weights (and optionally activations) during the forward pass. Pseudocode for our BitLinear layer is shown conceptually below (inspired by an open-source implementation [11]):

- Let W be the weight matrix of the linear layer and x be the input activation vector. We first apply a normalization x\_norm = RMSNorm(x). This ensures the scale of x is consistent.
- Define a function weight\_quant(W) that maps the full-precision weights to ternary values. In our implementation, we scale W by a factor proportional to the inverse of its average absolute value, then round to the nearest integer in -1, 0, +1, and finally scale back. Similarly, we can define activation\_quant(x) to quantize activations (for example, to 8-bit integers), though in our experiments we kept activations at higher precision.
- During the forward pass, we compute quantized versions: W\_q = weight\_quant(W) and x\_q = activation\_quant(x\_norm). However, directly using these in backpropagation would block gradients. We therefore apply the Straight-Through Estimator (STE) trick [3,8] by constructing:

 $W_q = W + (weight_quant(W) - W).detach()$ 

so that in the forward pass  $W_q$  equals the quantized W, but in the backward pass the gradient ignores the quantization step. The same technique can be applied to  $x_q$  if we quantize activations.

• Finally, the linear layer multiplies the quantized input with quantized weights: y = W\_q \* x\_q (plus bias if any, though we removed biases following BitNet's design). This produces the output for the layer.

By using STE in this manner, we can train the network parameters via gradient descent even though the forward pass uses quantized values. STE is an approximation – it assumes the quantization function is the identity for the purpose of gradients – which has been empirically effective for training quantized networks [3,8]. Additional measures (described below) are needed to ensure stable convergence.

### Gradual Quantization via Lambda Scheduling

Introducing quantization from the first step of fine-tuning can lead to a large training loss initially, since the pre-trained weights are suddenly constrained to low precision. If the loss is too high, it may destabilize training dynamics. To mitigate this, we implement a gradual quantization schedule controlled by a parameter lambda in the range [0,1]. We modify the quantization step in BitLinear as:

W\_q = W + lambda \* (weight\_quant(W) - W).detach()

and similarly for inputs if needed. When lambda=0, this means no quantization effect (the model is effectively full-precision); when lambda=1, it becomes full ternary. By setting lambda initially to 0 and then increasing it over time, we interpolate between the full-precision model and the quantized model over the training course [11].

In practice, we experimented with different schedules for lambda. A naive linear schedule that reaches lambda=1 only at the very end of training leaves minimal time in the fully quantized regime. We found that a better strategy was to ramp lambda to 1 by around the midpoint of training and then keep it at 1 thereafter (a "two-phase" schedule). For example, lambda(t) = min(2\*(t/T), 1) linearly increases lambda to 1 by halfway through training. This gave the model ample time in the fully quantized state to adjust, improving final performance over purely linear slow ramp or abrupt jumps [11].

### **Optional Layer-wise Knowledge Distillation**

As an additional technique, we implemented layer-wise knowledge distillation (KD) during fine-tuning, which can be enabled or disabled to compare its impact. In the KD-enabled setup, we assume we have access to the pre-trained teacher model (the original LLM in full precision, kept frozen). For each transformer block in the student (quantized model), we add a loss term that measures the discrepancy between the student's activations and the teacher's activations at that layer. We use mean-squared error (MSE) for this: if h\_student and h\_teacher are the hidden states, we add L\_KD =  $||h_student - h_teacher||^2$  to the loss. The overall objective becomes:

 $L_{total} = L_{CE} + alpha * (sum of L_{KD} across layers)$ 

where L\_CE is the usual cross-entropy loss on language modeling, and alpha is a hyperparameter controlling the strength of distillation. We set alpha so that initially KD loss and CE loss are of comparable scale. The intention is to encourage the quantized model to closely mimic the full model's internal representations [9,10].

It is important to note that KD can also be applied only at the final output level (for example, training the student with a softened softmax target), but for this paper we focus on layer-wise alignment. While distillation may help keep the student from diverging, it also constrains the student to follow the teacher's representation, which might not be optimal for a quantized model's capacity. Our experiments compare KD vs. no KD to see which yields better final cross-entropy performance.

#### **RMSNorm Insertion in Each Layer**

Perhaps the most critical modification enabling successful ternary fine-tuning is the insertion of RMSNorm layers at the input of each linear layer in the transformer. Standard Transformer architectures include LayerNorm or RMSNorm once per sub-layer, but following BitNet [5], we add an additional normalization right before every weight projection that is quantized. Concretely, if a transformer block has y = W2 \* sigma(W1 x), we place RMSNorm on x before multiplying by W1, and another RMSNorm on sigma(W1 x) before W2, and so on for attention projections.

The motivation is to maintain a consistent input distribution for each quantized weight matrix. Without normalization, some layers might see inputs with drifting scale, leading

to erratic outputs when multiplied by ternary weights. RMSNorm ensures that, regardless of upstream variation, the input norm is controlled. Prior research has noted the importance of normalization in low-bit networks [3,5,11]. RMSNorm is simpler than standard LayerNorm since it does not subtract the mean; we found this more stable when zero is one of the quantized weight values. We observed that removing these extra norms often led to training divergence or much higher final loss. With RMSNorm, training was stable and reached lower losses.

#### **Model and Dataset Details**

We validate our approach on two Transformer-based LLMs of different scales: Qwen-1.5B and Llama3-8B. Qwen-1.5B is a 1.5 billion-parameter model released by Alibaba as part of the Qwen family. Llama3-8B is an 8 billion-parameter variant similar to Meta's LLaMA, used in QAT experiments [2]. Both are decoder architectures for causal language modeling.

We fine-tune on the OpenThoughts-114k dataset, which contains 114,000 text samples from diverse open-domain sources. The task is next-token prediction. We use this as a stand-in for a relatively small-scale scenario, testing whether the model can maintain performance without vast data. We apply full-model updates: all transformer weights (attention, feed-forward, embeddings, etc.) are quantized and trained. This is more aggressive than freezing certain layers or using adapters, but ensures the entire model can adapt to low precision.

We use the AdamW optimizer with learning rates in the range 1e-4 to 5e-4 (tuned per model), linear warmup for 5 percent of steps followed by cosine decay, and a batch size that yields about 1000 parameter updates per epoch. We train Qwen-1.5B for 3 epochs (roughly 10k steps) and Llama3-8B for 2 epochs (roughly 20k steps). For the runs with KD, we keep the same schedule and the teacher is the original FP checkpoint. We set alpha=1.0 for KD loss. The lambda schedule linearly ramps from 0 to 1 by 50 percent of training steps, then remains at 1. We measure final cross-entropy loss on both training and validation sets.

# **Experiments and Results**

We first examine the training dynamics to understand the effect of knowledge distillation and our quantization approach, and then compare the final model performance across different fine-tuning strategies.



Figure 1. Layer-wise Distillation Loss vs. Training Steps (Qwen-1.5B).

When layer-wise KD is used, we can track the mean-squared error (MSE) between student and teacher representations over steps. The MSE starts high (ternary-weight activations deviate from the teacher) but steadily decreases, indicating successful knowledge transfer. By the end of training, the student's hidden states are much closer to the teacher's, suggesting strong alignment.



Figure 2. Layer-wise Distillation Loss vs. Training Steps (Llama3-8B).

For the Llama3-8B model, we see a similar trend: the KD loss decreases throughout finetuning. Although the final MSE is lower than the Qwen run, reflecting possible architectural differences, it still shows effective teacher-student alignment.

Despite the success of distillation in reducing the student-teacher mismatch, we must check whether it actually leads to better language modeling performance.



Figure 3. Final Cross-Entropy Comparison (Lower is Better).

We compare three strategies on each model: (1) Baseline QAT (ternary fine-tuning without KD or extra norms), (2) QAT with Layer-wise KD (but no extra norms), and (3) Direct QAT with RMSNorm (our proposed method, no KD). All three use the same lambda scheduling. We plot the final training cross-entropy loss after fine-tuning:

- The Baseline QAT model finishes with the highest loss, indicating it struggles with ternary weights without additional stability measures.
- The QAT + KD model converges to a better solution than the baseline, helped by the teacher's guidance.
- Our Direct QAT + RMSNorm achieves the lowest cross-entropy, outperforming the KD approach.

On Qwen-1.5B, for instance, the final cross-entropy is around 1.85 for Direct QAT + RMSNorm, versus approximately 1.90 for KD and 2.10 for the baseline. A similar pattern holds for Llama3-8B, where Direct QAT + RMSNorm again attains the best final loss. This finding suggests that directly optimizing cross-entropy, combined with careful normalization and a gradual quantization schedule, can outperform layer-wise distillation for ternary fine-tuning of LLMs.

An ablation confirms that removing the additional RMSNorm layers causes unstable training or higher final loss, consistent with prior work on ultra-low precision LLMs [3,5]. Interestingly, if we keep RMSNorm and add KD, performance does not substantially exceed our direct method. Thus, for simplicity and reduced overhead, we favor direct cross-entropy fine-tuning with RMSNorm.

# Conclusion

We presented a study on fine-tuning large language models with ternary weight quantization, introducing an approach that prioritizes direct optimization of the crossentropy objective. Our key finding is that with the proper training enhancements – chiefly, the integration of RMSNorm layers at each quantized layer input and a gradual quantization schedule – the quantized model can be trained end-to-end to achieve better final performance than if we had used auxiliary distillation losses. The direct QAT with RMSNorm outperformed a more complex strategy involving layer-wise knowledge distillation, achieving the lowest cross-entropy optimization is key: ultimately, the model needs to do well on the task at hand (next-token prediction), and sometimes the most effective route is to let the model learn from data directly rather than forcing it to mimic a full-precision teacher's internal representations.

Our work builds on and corroborates recent advances in training ultra-low precision networks, such as BitNet [5], by demonstrating that even a pre-trained model can be adapted to 1.58-bit weights with minimal loss of accuracy, given the right techniques. We have shown this on up to 8B-parameter models; in future work, we aim to scale to larger LLMs and more diverse tasks to verify that the advantage of direct fine-tuning with RMSNorm holds broadly. Another interesting direction is to explore activation quantization in addition to weight quantization, further reducing memory footprints and potentially introducing additional training considerations.

Ultimately, our findings show that extremely compressed models (ternary weights) are not only feasible but can be effectively fine-tuned from existing LLMs. By emphasizing the correct optimization objective and employing stability measures, we can avoid the complexity of per-layer distillation and still achieve strong results. This furthers the goal of making LLMs more efficient and accessible, without significant sacrifices in performance.

# References

[1] Introduction to Quantization, Hugging Face Blog (https://huggingface.co/blog/merve/quantization)

[2] Quantization-Aware Training for Large Language Models with PyTorch, PyTorch Blog (https://pytorch.org/blog/quantization-aware-training)

[3] Scalable MatMul-free Language Modeling (<u>https://arxiv.org/abs/2406.02528</u>)

[4] Back to Basics: Understanding the Foundational Paper on "Distilling the Knowledge in a Neural Network," Medium (<u>https://medium.com/@nayounghoon0223/back-to-the-basic-a-foundational-knowledge-distillation-paper-of-distilling-the-knowledge-in-a-10fba70bab3c</u>)

[5] The Era of 1-bit LLMs: All Large Language Models are in 1.58 Bits (<u>https://arxiv.org/abs/2402.17764</u>)

[6] Zhu, C., et al. Trained Ternary Quantization. OpenReview (<u>https://openreview.net/pdf?id=S1\_pAu9x1</u>)

[7] Zhang, B., et al. Root Mean Square Layer Normalization. arXiv:1910.07467 (https://arxiv.org/abs/1910.07467)

[8] Straight-through estimator for the argmax function? Reddit (https://www.reddit.com/r/MachineLearning/comments/53uln9/straightthrough\_estimator for the argmax\_function)

[9] Layer-wise Convolutional Neural Network Distillation, OpenReview (<u>https://par.nsf.gov/servlets/purl/10171699</u>)

[10] Model compression through distillation with cross-layer integrated..., ScienceDirect (https://www.sciencedirect.com/science/article/abs/pii/S0925231224019337)

[11] Fine-tuning LLMs to 1.58bit: extreme quantization made easy, Hugging Face Blog (<u>https://huggingface.co/blog/1\_58\_llm\_extreme\_quantization</u>)

# Adaptive vs Traditional Q-Learning Methodology

Saif Khan Computer Science Department Minot State University Minot, ND 58707 Saif.khan@ndus.edu Kaif Khan Computer Science Department Minot State University Minot, ND 58707 Kaif.khan@ndus.edu

Cayden White Computer Science Department Minot State University Minot, ND 58707 Cayden.white@ndus.edu

### Abstract

This research investigates if an adaptive exploration methodology in Q-learning can enhance the learning speed and performance of an agent in the Frozen Lake environment compared to the traditional decaying epsilon methodology. The study tests an adaptive approach where the exploration rate is dynamically adjusted based on the agent's learning progress to try to balance exploration and exploitation. Using the Frozen Lake environment from OpenAI Gym, a stochastic grid-world with slippery surfaces, the goal is for an agent to travel from a start point to an end point using reinforcement learning. Key metrics such as cumulative reward, moving average, and success rates are analyzed for both methodologies. The results try to determine if adaptive exploration enables the agent to navigate the environment more effectively. This work offers insights to test different exploration methodologies in reinforcement learning for discrete environments.

# **1** Introduction

*Reinforcement learning* (RL) is a branch of machine learning where an agent learns to make decisions by interacting with some environment that it has been put in. The agent takes actions, receives feedback in the form of rewards or penalties, and tries to maximize its cumulative reward over time. Unlike supervised learning, RL does not rely on labeled datasets; instead, the agent learns through trial and error over time, making it well-suited for tasks in dynamic and uncertain domains such as game playing, robotics, and autonomous systems [14].

For testing of RL algorithms, we utilized *OpenAI Gym*, which is an open-source toolkit designed to provide a standardized platform for RL research. The goal of OpenAI Gym is to offer a diverse set of environments, ranging from simple grid-world problems to very complex simulations. Researchers can implement, test, and compare RL algorithms. For this experiment we used the *Frozen Lake environment*, provided by OpenAI Gym. Frozen Lake is a grid-world environment, the agent must navigate from a starting point to a goal across a frozen lake, avoiding holes in the ice. The environment is stochastic, actions may not always produce the intended outcome due to slippery surfaces which adds complexity to the learning task. This setting is very important for studying exploration strategies because it requires the agent to carefully balance exploration and exploitation to find an optimal policy. [3][7].

In this study, we focus on *Q*-learning, a foundational RL algorithm that lets an agent learn optimal actions without requiring a model of the environment. Q-learning is a value-based, model-free method that maintains a Q-table, a data structure that stores the expected future rewards for each state-action pair. The agent updates the Q-table based on its experiences using the Bellman equation, slowly learning the value of taking specific actions in given states. A critical challenge in Q-learning is the exploration-exploitation trade-off: the agent must balance exploring new actions to discover better strategies with exploiting its current knowledge to maximize rewards. The traditional approach to this trade-off is the *decaying epsilon method*, where the probability of selecting a random action (exploration) starts high and decreases over time according to a fixed schedule. This allows the agent to explore extensively early on and shift toward exploiting learned values as training progresses. An adaptive strategy is when the exploration rate would dynamically adjust based on the agent's performance or learning progress. While the decaying epsilon method is effective in many cases, we hypothesized that: Can an adaptive exploration strategy in Q-learning significantly improve the learning speed and/ or performance of an agent in the Frozen Lake environment compared to the traditional decaying epsilon approach? [1][2][4].

### **2** Literature Review

Another research done by Watkins and Dayan (1992) shows new ways how Q-learning, a modelfree reinforcement learning algorithm that can incrementally approximate the optimal actionvalue function using temporal differences, therefore enabling agents to learn optimal policies in controlled Markovian environments without the need of explicit models of state transitions or rewards. The authors prove that, provided every state–action pair is visited infinitely often, the Q-values converge to the optimal values with probability one under bounded rewards and appropriate learning rates, basically encouraging better and faster learning. They have also done their analysis of non-discounted Markov processes with absorbing states and scenarios where multiple Q-values are updated in parallel, demonstrating the method's strength and practical applicability. This work establishes a theoretical foundation for Q-learning but also has significantly influenced other types of research in reinforcement learning by ensuring that optimal decision-making can be achieved through iterative, experience-driven updates (Watkins & Dayan, 1992). [10]

Reinforcement learning (RL) is the leading approach for training agents in complex environments, with Q-learning is known for its model-free, trial-and-error learning of optimal policies (Sutton & Barto, 1998). One challenge in Q-learning is balancing exploration, selecting actions known to yield high rewards. The traditional epsilon-greedy method addresses this by having an agent choose a random action with probability epsilon ( $\epsilon$ ) and the best-known action with probability  $1-\varepsilon$ , decaying  $\varepsilon$  over time. However, fixed decay schedules often fail to adapt to the agent's evolving knowledge, sometimes leading to premature convergence or excessive exploration. In response to that, adaptive exploration strategies have been developed that dynamically adjust  $\varepsilon$  based on performance feedback, such as improvements in cumulative reward or changes in Q-values, thereby helping the agent escape local optima and accelerate convergence (Hiraoka et al., 2015). The research done by Alahmari, Yuan, and Kudenko (n.d.) demonstrates these challenges and potential solutions in a different domain-abstract argumentation—where O-learning was applied using rewards derived from grounded extensions (Dung, 1995). Although their study focuses on argument games and not the typical navigation tasks, the knowledge regarding state representation and exploration strategies stays largely the same. In benchmarks like the Frozen Lake environment-a discrete grid world with stochastic transitions—the limitations of a fixed decaying epsilon approach become even more noticeable, as a context-aware, adaptive exploration strategy can allow an agent to first explore as much as it wants and then exploit learned paths while still adapting even if trapped in suboptimal routes (Mnih et al., 2015). Overall, while the traditional decaying epsilon approach has been the most common in Q-learning, its inability in dynamic environments have led researchers to explore adaptive methods that promise faster learning and improved performance by continuously tuning exploration in response to real-time feedback. [11]

Zhang et al. (2021) proposed a self-adaptive reinforcement-exploration Q-learning (SARE-Q) algorithm designed to overcome the limitations of traditional Q-learning, mainly heavy repetition and an imbalance in exploration and exploitation. This was done by incorporating a behavior utility trace that dynamically adjusts action selection probabilities, a two-phase attenuation process for the exploration factor ( $\epsilon$ ) that initially emphasizes broad exploration and then transitions to exploitation based on success rates, and an adaptive mechanism that modulates  $\epsilon$ 

2

according to state access frequencies to reduce redundant exploration in well-visited areas while enhancing it in underexplored regions. The algorithm is evaluated through simulation experiments in symmetric grid map environments on the OpenAI Gym platform, where it outperforms both conventional Q-learning and a self-adaptive Q-learning variant (SA-Q) in various performance metrics such as earlier target acquisition, reduced number of turning times in optimal routes, and higher overall success rates, demonstrating its potential to improve path planning tasks and offering a significant contribution to the reinforcement learning literature by providing a strong framework for promoting more exploration efficiency in dynamic and complex environments (Zhang et al.2021) [13]

### **3** Methodology

#### **3.1 Experiment Overview**

This study investigates the performance of two reinforcement learning strategies using the Frozen Lake environment from the OpenAI Gym library: a traditional Q-learning algorithm with a decaying epsilon-greedy approach and an adaptive exploration strategy that adjusts the exploration rate dynamically based on how the agent learns. The Frozen Lake environment, as explained in the introduction, is a 4x4 grid-world problem where an agent tries to navigate from a starting point to a goal while avoiding holes, and it gets a reward of 1 for reaching the goal and 0 for everything else in the middle. The experiment has two phases. First, we implemented and tested the traditional Q-learning algorithm with a fixed epsilon decay schedule to establish a baseline, balancing exploration (random actions) and exploitation (optimal actions) over time. Second, we developed and evaluated an adaptive Q-learning strategy that adjusts the exploration rate based on recent performance. We tried to increase exploration with poor performance and increased exploitation when the performance was good. Both strategies were trained on 1,000, and, 5,000 episodes, and their performance was compared using metrics such as cumulative rewards, success rates, etc.

### **3.2 Variables and Parameters**

The experiment employed the following key variables and parameters for both Q-learning implementations. Starting with *Environment (env)*: The Frozen Lake environment, featuring 16 states (4x4 grid) and 4 actions (left, down, right, up), with a discrete state-action space accessed via env.observation\_space.n and env.action\_space.n. *Episodes (episodes=1000 or 5000)*: The total number of training episodes or iterations for policy learning. Learning Rate (alpha=0.1):

<sup>3</sup> 

Controls the influence of new information on Q-value updates, set to 0.1 for a balanced learning pace. Discount Factor (gamma=0.99): Emphasizes long-term rewards, prioritizing future outcomes with a value near 1. *Exploration Rate (epsilon=1.0)*: Initial probability of random action selection, starting at 1.0 (fully exploratory). Minimum Epsilon (min epsilon=0.01): Lower bound for epsilon in both approaches, ensuring minimal exploration persists. Decay Rate (decay rate=0.999): Used in traditional Q-learning to reduce epsilon after each episode, gradually shifting toward exploitation. *Maximum Epsilon (max epsilon=1.0)*: Upper bound for epsilon in the adaptive approach, capping exploration increases. Increase Threshold *(increase threshold=50)*: Frequency (every 50 episodes) for adjusting epsilon in the adaptive approach based on performance. Rolling Window (rolling window=100): Number of recent episodes used to compute average rewards for epsilon adjustments in the adaptive approach. Here are the metrics we tracked: *Rewards (rewards)*: Cumulative reward per episode, reflecting task performance. Successes (successes): Binary indicator (1 if reward > 0, 0 otherwise), denoting goal attainment. Epsilons (epsilons): Exploration rate recorded per episode. O-Differences (q diffs): Average absolute change in Q-values per episode, indicating learning stability.

#### **3.3 Implementation Details**

The experiment was implemented in Python, with the traditional and adaptive Q-learning strategies coded as traditional\_q\_learning() and adaptive\_q\_learning(), respectively. Both functions initialize a Q-table with zero (env.observation\_space.n × env.action\_space.n) and update it using the Q-learning rule which is the bellman formula:  $q_table[state, action] = (1 - alpha) * q_table[state, action] + alpha * (reward + gamma * max(q table[next state])).$ 

#### **3.3.1 Traditional Q-Learning Method**

The agent follows an epsilon-greedy policy, it selects random actions with probability epsilon or the highest Q-value action otherwise. Epsilon decays after each episode via *epsilon* =  $max(min\_epsilon, epsilon * decay\_rate)$ , reducing exploration over time. Metrics (rewards, successes, epsilons, q\_diffs) are tracked per episode, with q\_diffs calculated as the mean absolute difference between the Q-table before and after updates.

#### 3.3.2 Adaptive Q-Learning Method

4

Adaptive method is like the traditional approach but defers in dynamically adjusting epsilon every 50 episodes (increase\_threshold) based on the average reward over the last 100 episodes (rolling\_window). If the average reward is below 0.2, epsilon increases by 0.05 (up to max\_epsilon); otherwise, it decreases by a factor of 0.95 (down to min\_epsilon). The same metrics are recorded for comparison.

### **4 Results**

The performance of both strategies was evaluated over 1,000 and 5,000 episodes, with our results showing that the traditional Q-learning approach outperformed the adaptive strategy.

#### 4.1 1,000 Episodes Results

The results for the first test using 1,000 episodes are graphed in Figure 1. Figure 1 represents the Moving Average Rewards which means over a period of 100 episodes how many rewards each method received. A reward indicates that the agent is learning and is reaching its goal. In Figure 1 the blue line represents the traditional method vs the orange line represents the adaptive method. It clearly shows how well the traditional method is performing every 100 episodes.



Figure 1: 1,000 episode Moving Average Rewards.

Table 1 shows results for 1,000 episodes. The important metrics are Overall Average Reward which is 0.04 for traditional method and 0.00170 for adaptive and Average Reward (Last 100 episodes) which is 0.06 for traditional and 0.04 for adaptive. Traditional method outperformed on all metrics.

			Ũ
0	Overall Average Reward	0.0400	0.0170
1	Std Dev of Rewards (Overall)	0.1960	0.1293
2	Avg Reward (Last 100 Episodes)	0.0600	0.0400
3	Final Epsilon	0.3681	1.0000

Metric Traditional Q-Learning Adaptive Q-Learning

Table 1: 1,000 episodes Metrics.

#### 4.2 5,000 Episodes Results

The results for the second test using 5,000 episodes are graphed in Figure 2 which represents the Moving Average Rewards. The blue line represents the traditional method vs the orange line represents the adaptive method. It clearly shows how well the traditional method is performing every 100 episodes. Over time the success rate keeps on increasing for the traditional method while the adaptive method stays has little learning over time.



Figure 2: 5,000 episode Moving Average Rewards.

Table 2 shows results for 5,000 episodes. The important metrics here are Overall Average Reward which is 0.3876 for traditional method and 0.0144 for adaptive and Average Reward (Last 100 episodes) which is 0.5500 for traditional and 0.0200 for adaptive. Traditional method outperformed on all metrics.

	Metric	Traditional Q-Learning	Adaptive Q-Learning
0	Overall Average Reward	0.3876	0.0144
1	Overall Success Rate	0.3876	0.0144
2	Std Dev of Rewards (Overall)	0.4872	0.1191
3	Avg Reward (Last 100 Episodes)	0.5500	0.0200
4	Avg Success Rate (Last 100 Episodes)	0.5500	0.0200
5	Std Dev of Rewards (Last 100 Episodes)	0.4975	0.1400
6	Avg Q-Difference (Last 100 Episodes)	0.0013	0.0008
7	Final Epsilon	0.0100	1.0000

Table 2: 5,000 episodes Metrics.

# **5** Conclusion

In this study, we tested the effectiveness of an adaptive exploration strategy compared to the traditional decaying epsilon-greedy approach in Q-learning, using the Frozen Lake environment as our test environment. Our findings showed that the traditional method, with its decreasing exploration rate, outperformed the adaptive strategy. The first test with 1,000 episodes resulted in the average reward of 0.04 for traditional and 0.00170 for the adaptive method. The traditional approach achieved an average reward of 0.3876 and a success rate of 0.3876 with 5,000 episodes, compared to the adaptive method's 0.0144. These results show that the adaptive approach struggled to leverage the knowledge effectively.

However, the study's outcomes do not mark the end of the investigation but rather point to exciting possibilities for future work. There are many aspects of the experiment that can be changed to get different results. For instance, changing the parameters of the adaptive strategy such as the size of the rolling window, the threshold for increasing epsilon, or the criteria for evaluating performance might improve its effectiveness. Additionally, more experiments could be conducted with entirely different approaches, like basing exploration adjustments on Q-value stability or integrating more advanced performance indicators. Testing these variations in other environments or with different reinforcement learning algorithms could also show us new insights.

### References

- [1] Labonne, M. (2025, January 29). *Q-learning for beginners*. Towards Data Science. <u>https://towardsdatascience.com/q-learning-for-beginners-2837b777741/</u>
- [2] GeeksforGeeks. (2025, February 24). *Bellman equation*. https://www.geeksforgeeks.org/bellman-equation/
- [3] Malin. (2024, May 24). Frozen Lake with RL, my journey. Medium. https://medium.com/%40MaLiN2223/frozen-lake-with-rl-my-journey-9b048e396ac3
- [4] GeeksforGeeks. (2025b, February 25). *Q-learning in reinforcement learning*. https://www.geeksforgeeks.org/q-learning-in-python/
- [5] Sarjit07. (2022, March 16). *Reinforcement learning using Q-table frozenlake*. Kaggle. https://www.kaggle.com/code/sarjit07/reinforcement-learning-using-q-table-frozenlake
- [6] Openai Gym and python for Q-learning reinforcement learning code project. deeplizard. (n.d.). <u>https://deeplizard.com/learn/video/QK\_PP\_2KgGE</u>
- [7] *Gymnasium documentation*. Gymnasium Documentation. (n.d.). <u>https://gymnasium.farama.org/</u>
- [8] OpenAI. (n.d.). <u>https://openai.com/</u>
- [9] Baeldung, W. by: (2025, February 28). Epsilon-Greedy Q-Learning. Baeldung on Computer Science. <u>https://www.baeldung.com/cs/epsilon-greedy-q-</u> learning#:~:text=In%20Q%2Dlearning%2C%20we%20select,return%20A
- [10] Watkins, C. J. C. H., & Dayan, P. (n.d.). *Q-learning machine learning*. SpringerLink. <u>https://link.springer.com/article/10.1007/bf00992698</u>
- [11] University of galway. (n.d.-b). https://ala2017.cs.universityofgalway.ie/papers/ALA2017\_Alahmari.pdf
- [12] Zhang, L., Tang, L., Zhang, S., Wang, Z., Shen, X., & Zhang, Z. (2021, June 11). A selfadaptive reinforcement-exploration Q-learning algorithm. MDPI. <u>https://www.mdpi.com/2073-8994/13/6/1057</u>
- [13] (PDF) a self-adaptive reinforcement-exploration Q-learning algorithm. (n.d.-b). <u>https://www.researchgate.net/publication/352403913\_A\_Self-Adaptive\_Reinforcement-Exploration\_Q-Learning\_Algorithm</u>

8

[14] What is reinforcement learning? - reinforcement learning explained - AWS. (n.d.-d). <u>https://aws.amazon.com/what-is/reinforcement-learning/</u>