Using Data Mining in Combination with Machine Learning to Enhance Crowdsourcing of a Formal Model of Biodiesel Production

Michael Fischer, Derek Riley

University of Wisconsin- Parkside

Kenosha,WI

fisch037@rangers.uwp.edu, rileyd@uwp.edu

Abstract

Formal modeling, simulation, and analysis of complex systems is valuable because it can provide insights into complex systems that are too expensive or difficult to analyze otherwise. In this work, we present an approach for improving simulation trajectory choices in a Monte Carlo framework using a combination of crowdsourcing, machine learning, and data mining. We apply machine learning to analysis of a formal model of biodiesel production as a method of improving the efficiency of the crowd sourced mobile simulation analysis of the model. Data is collected and data mined in a central server where machine learning is applied and recommendations from the machine learning algorithm are fed back to crowd workers via suggestions on the mobile app. Ultimately, we show that this approach can improve efficiency of optimal safe state identification in the biodiesel model analysis.

1 Introduction

Modeling and simulation of complex systems holds great potential because it can provide additional information about a wide variety of complex systems that are otherwise too expensive or difficult to analyze using traditional methods like experimentation. Models, while imperfect by nature, can be improved and refined, and their analysis allows for extracting as much data possible about the system with the intent to improve understanding of the complex dynamics of these systems, and ultimately improve them.

Biodiesel is an alternative fuel source that can be easily made by novices with an inexpensive home-made reactor using waste vegetable oil, but producing high quality fuel using a home-made reactor is difficult due to the complexity of the chemical interactions and the configurations of the processor. A biodiesel processor is a complex system that can be modeled and simulated using formal modeling methods, but accurate modeling can require prohibitively expensive analysis [1].

Formal modeling paradigms often restrict the types of dynamics that can be captured, but Stochastic Hybrid Systems (SHS) allow for the formal incorporation of discrete, continuous, and probabilistic dynamics, which allows for modelling of many realistic,

complex systems in a variety of fields including industrial systems, medical systems, biological systems, and much more [1].

Due to the size and complexity of realistic SHS models and the complexity of their analysis data, it can be difficult to extract meaningful data from simulation or verification of such systems. Complex dynamics of these systems can require exhaustive analysis to generate comprehensive insights into their dynamics, which is too computationally expensive for large systems.

Analysis using simulations of complex models can reduce computational complexity, but choosing the right simulations to test to find desirable properties is usually done by brute force methods or via guessing. Both of which are inefficient methods and can lead to a large amount of wasted computational time.

In this work, we present an approach for improving simulation trajectory choices using a combination of crowdsourcing, machine learning, and data mining. We apply machine learning to analysis of a formal model of biodiesel production as a method of improving the efficiency of the crowd sourced mobile simulation analysis of the model. Machine learning algorithms are used to analyze the simulation data generated, as a part of data mining the simulation results.

To alleviate the computational strain of our approach, the model is distributed on a crowd-sourced mobile platform via mobile application. Data is collected and data mined in a central server where machine learning is applied. Knowledge discovery is provided to the crowd workers in real-time by means of a suggestion generator. The suggestion generator analyses the user's input variables and determines the reactants that limit a successful conversion of biodiesel. Actionable parameter tweaks are then provided to further improve their simulation scores. The intent of this feedback loop is for the crowd workers to formulate intuitive conclusions from the provided example to then use on subsequent simulations runs to better explore the state space.

Using our proposed Machine Learning and Data Mining (MLDM) techniques on the collected simulation data we can provide the user with a suggestion generator, delivering real-time input suggestions based on their previous simulation runs. This will in turn give the user further insight into the simulation and allow them to better understand previous simulations run by any user of the system, thus enhancing their natural pattern recognition capabilities. Ultimately, we hope to show that this approach will greatly improve efficiency of optimal safe state identification in the biodiesel system.

2 Related works

A parametric Machine Learning (ML) method is introduced for rule extraction of Recurrent Neural Networks (RNN) modeling deterministic finite-state automata (DFAs). The symbolic learning algorithm infers the underlying DFA by considering only the input and output behavior of the RNN [2]. As such the internal state and transitions are ignored completely. The learning algorithm has a polynomial time complexity causing scalability issues with complex systems.

Multi-modal symbolic regression (MMSR) is a combination of methods that constructs a data-driven symbolic model of hybrid dynamical systems [3]. This technique uses non-linear symbolic expressions for representing behavior and transitions using unlabeled time-series data. MMSR demonstrated the capability of inferring a hybrid dynamical system up to three discrete modes in a two-dimensional non-linear input-out mapping.

When compared to a variety of neural networks MMSR achieved higher numerical accuracy with less free parameters [3]. The individual modes of the hybrid system underlying the unlabeled time-series data is determined by Clustered Symbolic Regression (CSR) which can be regarded as a generalized solution to learning piecewise functions by applying an Expectation-Maximization (EM) framework to Symbolic Regression (SR) [3].

Through CSR the data is separated into clusters and then assigned mathematical expressions for each sub function. Transition conditions from one mode to the next are found by Transition Modeling (TM) which can be regarded as a generalized solution to classification with symbolic expressions [3]. TM searches for the classification boundaries and represents them as a symbolic inequality.

Previous approaches share the same general goal of reducing simulation costs for complex model analysis. The combination of stochastic simulation and model-checking is pursued to further understand biochemical systems [4]. Simulation is used to identify upper and lower bounds for specific species, molecular populations. The bounded species are used with PRISM model-checker to overcome the well-known state-space explosion problem while allowing temporal properties to be expressed through PRISM's reward structure [4].

The combination of simulation and model-checking are used collectively to investigate steady-state behavior along with transient properties of the modeled system to gain a more comprehensive understanding. Transient properties (i.e dependent on time) are of great interest to biologists because they can be compared with available experimental data and examined for correctness [4]. However, the PRISM framework cannot be used with models that that capture both discrete and continuous dynamics, which are often found in realistic, complicated biochemical systems [5].

A machine learning approach for generating temporal logic classifications of complex model behaviors presents a systems biology workflow that applies principal component analysis (PCA) prior to Density-Based Clustering (DBC) of time-series data generated by the simulation of Colored Petri Nets (CPN) [6]. Statements in probabilistic linear-time are generated automatically for each cluster to discriminate between clusters. DBC adjust well to varying cluster shape, but handles cluster density poorly. This is due to the use of a single density parameter that holds globally causing clusters to be missed if set too high and clusters merged if set too low [7].

Monte Carlo simulations are performed on models of synapse configurations of varying structural and physiological characteristics to generate a time-series representation of the synapses' behavior. Each synapse was run numerous times and the resulting simulation data was averaged and fit by Nonlinear Least Squares (NLS) to a variety of mathematical

functions to determine their coefficients and ranking measurements corresponding to the fit of the function to the simulation results obtained [8].

A linear regression model was applied to the considered functions to define a relationship between their coefficients and the modeled synapse parameters, but a linear correlation couldn't be derived. Machine learning regression algorithms are used to learn the relationship between AMPA behavior series (as the training data) and the open AMPA percentage (as the target value).

The predicted values from the machine learning methods where used in the final curve fitting stage through NLS to determine the coefficients of the candidate functions in the first step. The final prediction model of receptor activation was capable of predicting the "average percentage of open AMPA receptor for a given synapse configuration" [8].

A prediction model can estimate the unseen behavior of synapses with different characteristics instead of requiring additional computationally expensive Monte Carlo simulations. The reduction in computational costs (CPU hours) for the prediction model was $1/400^{th}$ that of Monte Carlo simulations. The limitation to this approach is the initial Monte Carlo simulations needed to produce the comprehensive dataset required for the final prediction model. In all 3,500 CPU hours were needed to produce more than ten million data points. The receptor prediction model depends greatly on the quality of the initial training data to be able to learn and extract useful synapse behavior patterns [8].

3 Data Representation and Preprocessing Techniques

Dimensionality reduction techniques are crucial in ML to reduce the time and space complexity along with the required number of training examples needed to accurately produce a classifier or regressor [9]. However, care must be taken in the way the reduction is performed so essential information is not lost by the altered feature space. Additionally, for the suggestion generator to provide feedback based on tweaks to the users input the 5- dimensional input space must be maintained to access input parameters that are limiting the reaction.

Clustering methods can intuitively be thought of as organizing a large number of "items" into a smaller number of groups to represent them in a more concise manner. Clustering methods divide the data set into individual groups of data points that are regarded as similar to each other by a distance or similarity measure. Once the groups are comprised of the individual data points each cluster can be understood by its collective behavior providing a summary of the data points that represent each cluster [10]. The individual data points can be are assigned to their corresponding cluster number by an added feature column providing a new feature space that can be used by a classifier or regressor to further determine key properties defining each cluster.

K-means is a representative-based algorithm that selects one partitioning representative as the centroid (mean point) of the cluster to assign the remaining data points to one of the k closest representative [7]. The main factor in finding a good clustering is the determination of good partitioning representatives [7].

K-means requires a pre-defined number of clusters to be set and is often performed iteratively to find the appropriate number of clusters. Each k cluster then represents simulations of similar behavior that can be used to summarize descriptive qualities of the group. The locations of independent clusters can be used comparatively to find interesting behavior and transition properties from one cluster to another.

Representative-based clustering methods are known to favor spherical shaped clusters while adjusting better to varying clustering density [7]. While agglomerative and density-based clustering methods adjust better to differing cluster shapes but perform poorly on varying density of the clusters [7].

Feature extraction methods like Principal Component Analysis (PCA) are used as a preprocessing technique to simplify and describe the key factors in the data. However, unlike clustering methods that derive a new feature space in PCA the features are combined to form a lower dimensionality while preserving the greatest amount of information from the original dimensions [11].

PCA has been used prior to clustering methods in the hopes of better representing each cluster while taking advantage of an additional leveling of description properties that clustering allows. PCA has been applied as a preprocessing stage before three clustering methods, namely the hierarchical average-link algorithm, the k-means algorithm, and the Cluster Affinity Search Technique (CAST) do not foster improved cluster quality and often low overall cluster quality [11].

4 Biodiesel Model

We present our updated model analysis approach designed to `crowdsource' simulations of a formal Stochastic Hybrid System (SHS) model of biodiesel production. We use the SHS modeling paradigm because it is a flexible, efficient modeling paradigm that can capture discrete, continuous, and stochastic modeling components [1]. In this section, we present abbreviated details of the formal modeling and simulation approach and our model of biodiesel production.

SHS have been used to model and analyze complex, interesting systems with discrete, continuous, and probabilistic dynamics. SHS contain a set of discrete states, invariants associated with the discrete states, and continuous dynamics associated with the discrete states. Discrete transitions between the states occur either because the continuous state x satisfies the transition guard or based on an exponential firing rate (probabilistic transition). A reset measure R is associated with any transition. We use an existing, validated model of biodiesel production presented in [1].

The biodiesel chemical reactions involve six chemical species and six highly-coupled reactions [1]. Vegetable oil in its purest form is made up of triglycerides (TG); however, it breaks down into diglycerides (DG) and monoglycerides (MG) as it is heated. An alcohol, methanol (M), is combined with the TGs, DGs, and MGs to convert them into biodiesel esters (E) and glycerine (GL).

Since temperature significantly affects the rates at which reactions occur, it is important to use accurate models of the kinetic coefficients of the chemical reactions. Our kinetic

rate equations were derived using the Arrhenius equation and known dynamics of the reactions and were presented previously [1]. Since chemical dynamics are inherently stochastic, SDEs are an ideal modeling paradigm for the chemical concentrations.

It is critical to determine whether or not a biodiesel processor will be able to produce high quality biodiesel, which will pass the American Society for Testing and Materials (ASTM) biofuels tests. Studies have shown that only very small quantities of oil (TG+DG+MG) in the final biodiesel will allow the resulting fuel to meet ASTM specifications [1]. Challenges exist in designing biodiesel processors because different feed stocks have different concentrations of TGs, DGs, and MGs, and these concentrations have a direct impact on the catalyst used and the amount of M required to make high quality fuel.

A score for the reaction can be calculated to provide further information about the reaction trajectory. We use the fuel cost because the goal of the simulation game we have created is to minimize the fuel cost (while still producing quality fuel). The fuel cost is calculated assuming the initial oil is free (as is the case with many waste oil reactors).

The fuel cost takes into account the cost of methanol, catalyst, and the electricity to heat the tank. All of these variables change depending on how the user configures their simulation. The equation below shows the way we calculate the cost where *T* is the temperature, *M* is the amount of methanol, *TG* is the concentration of triglycerides, *DG* is the concentration of diglycerides, and *MG* is the concentration of monoglycerides. The initial oil quantity is translated into *TG*, *DG*, and *MG* by dividing the initial oil amount into the percentages of *TG*, *DG*, and *MG* found in canola oil. Overall reaction time is also incorporated into the equation since the time affects the amount of energy (heat) used to create the fuel.

$$cost = \frac{\left(\frac{T}{1050} - 0.012\right) * ReactionTime + \frac{M}{5}}{TG + DG + MG}$$

5 Classification

Classification seeks to determine the separation of different groups (classes) by determining the relationships of attributes (features) to the individual class. The outcome of the analysis technique can be described by a set of rules, which assign new unseen instances to their corresponding classes. These “rules” (discriminates) partition the variable space into localized regions that correspond to each class. The rules can then provide guidelines for determining interesting simulation input space.

Decision trees are a classification algorithm that assumes no prior form of class densities (successful simulations or unsuccessful) or model parameters. Decision trees are discriminant–based techniques that aim to determine the boundaries that separate the classes [9].

The tree structure consists of decision nodes and leaf nodes that are constructed by a number of recursive splits based on the provided training data. The splits in a classification tree are preformed to minimize an impurity measure. The decision nodes

resulting from the recursive splits each define a discriminant function that regionalizes the input feature space. The leaf nodes correspond to a localized region that represents all simulations with the same class label that followed the recursive splits down the tree structure. With the classification tree the target class does not need a single description to which all instances follow, but has the capability to represent all localized regions.

Traversing from the root to any combination of decision nodes further subdivides the input space into more localized regions by the combination of each nodes discriminate function. The combination of decision nodes can be interpreted as a "rule" that defines a particular class. In this regard the decision tree can be seen as various alternatives of user inputs. See Figure 1 for an example of a decision tree that can be used for classification of our model.

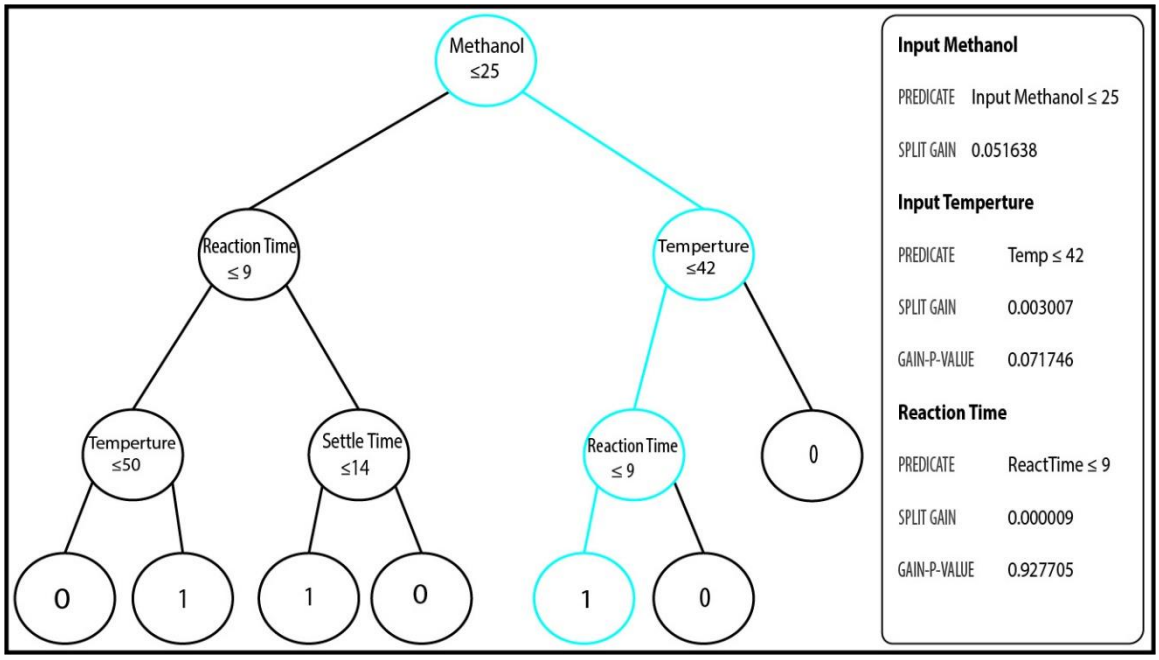

Figure 1: Decision tree for our MLDM approach showing a potential regionalization of the simulation input space.

Ensemble learning methods seek to construct a collection of base learners (classifiers), which are used collectively in a combination scheme to provide a more accurate classifier or regressor [12]. Boosting is an ensemble learning method, which trains base learners sequentially on the training error of the previous learner [12]. With each iteration, the base learning algorithm produces a weak prediction rule that the boosting algorithm combines by a weighted voting scheme to form a more robust prediction rule [12].

Boosting can be applied to any classifier and in this work decision trees are the base learner considered. A decision tree is constructed on a portion of the training data. The classification error for this decision tree is given a higher weight for the next decision tree to try and create an accurate prediction rule to correctly classify the instance. Sequentially this processes builds trees on the hardest instances to label.

6 Architecture

The overall architecture of our MLDM system is shown in Figure 2. Users from the general public use our iPad or iPhone application to run biodiesel simulations as part of a game. These simulations send the results to a central database that is used for the MLDM approach. The MLDM results create recommendations that are fed back to the users to improve and encourage further simulations. A web service is used to update and improve the MLDM approach.

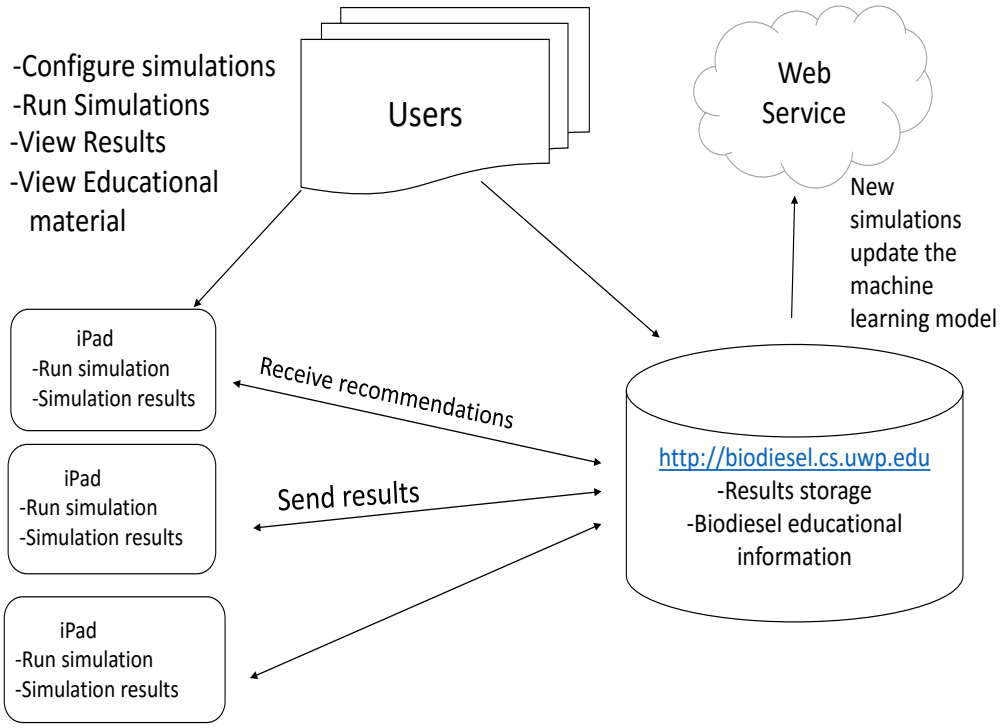

Figure 2: Depiction of the system architecture.

This architecture allows for the combination of these simulations as Monte Carlo simulations to perform validation of the system. Users are notified and presented with immediate results if they choose a simulation that is already in the database to reduce wasted computation.

6.1 MLDM Approach

We apply preprocessing techniques to the stored data in our database and once the data is in the right format, it is split into training data and a validation set. Machine learning algorithms are then applied to the training data in order to obtain a model. The model is then validated using the validation set. The prediction model is then deployed so it can be accessed by crowd workers' mobile applications. Figure 3 depicts the architecture of the MLDM approach.

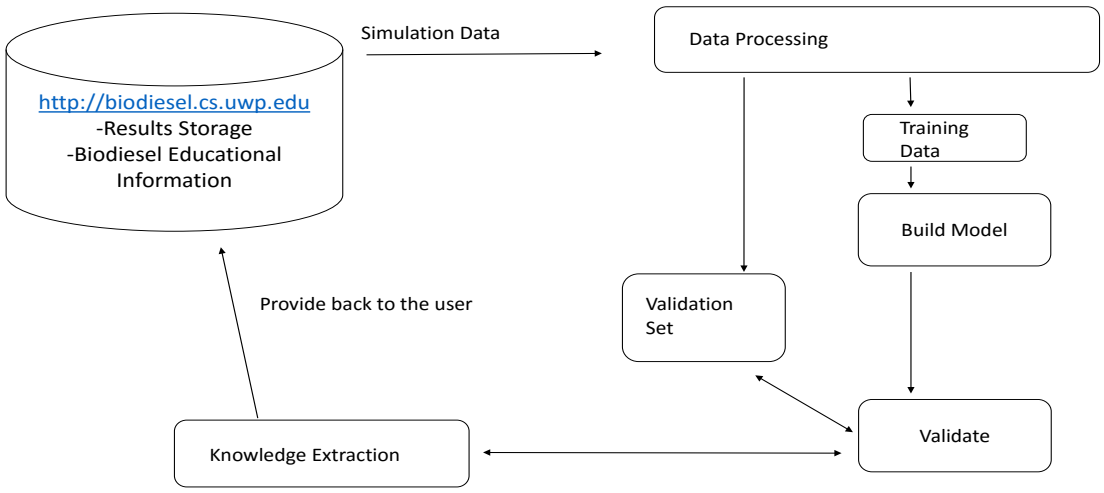

Figure 3: MLDM architecture

6.2 Mobile Application

The mobile application allows a user to play a game-like application that runs the scientific simulation. The user is motivated to play the game to improve their score and contribute to the scientific computing goal of the app. Screenshots of the application can be seen in Figure 4.

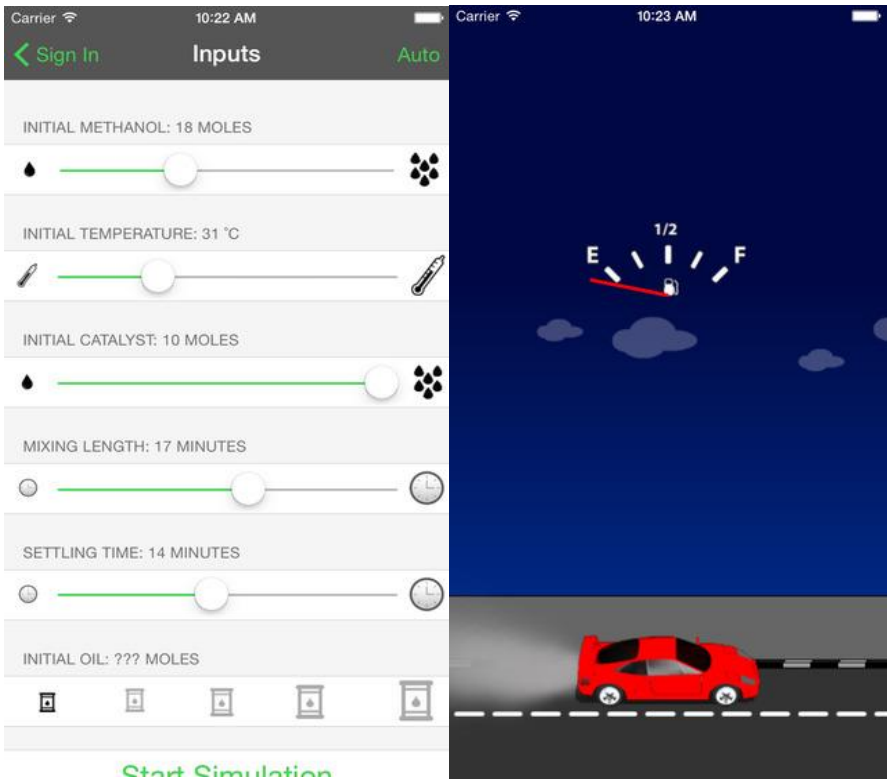

Figure 4: Screenshots of the mobile application.

Users start by selecting the input variables and initial configuration for the simulation, and then they are presented with a car animation that shows the results of how much fuel is produced, what quality it is (percentage conversion), and what the cost of the fuel produced is.

The goal of the game is to find various simulation inputs that influence the final conversion rate and overall cost of successful biodiesel conversions that meet American Society for Testing and Materials (ASTM) standards for quality biodiesel fuel mentioned previously.

The application architecture allows for any amount of mobile users to simultaneously configure and run simulations. The simulation results are displayed to the user in the form of the ending chemical concentrations and the cost of the fuel made. The results are also stored on database and can be accessed through a companion website. More information about the companion website can be found in [1].

The knowledge gained from these methods is validated in two ways. First the approach is validated with the convention standards meet within the MLDM community (charts, graphs). Second, the knowledge gained is applied practically as a suggestion generator to provide the crowd workers a feedback response system to improve their biodiesel production scores by tweaking their inputs values that are limiting the successful conversion of biodiesel.

7 Results

The feature columns of the database are methanol, temperature, oil, mixing length, settling time, overall run time, final conversion rate, and the class label. Each simulation is recorded as an individual row in the database, collectively comprising the training data provided to the MLDM techniques. For classification, successful simulations with a conversion rate of 95% and higher are marked as 1 for the class label while unsuccessful simulation with a conversion rate below 95% are marked with 0. Normalization is applied to the data before applying k-means to avoid features from dominating one another; however, this normalization is not used with the two-class boosted decision tree.

K-means clustering algorithm is applied in a semi-supervised mode by incorporating the target label to guide centroid selection. The drawbacks known to k-means are diminished when preformed in a semi-supervised mode, which has been demonstrated in [13]. For these reason k-means was determined as the most suitable clustering algorithm compared with the alternatives.

Our data set formed seven clusters. These clusters where then analyzed (by hand) and profiled individually and then used comparatively to determine the transition properties between each cluster. Clusters 6 and 7 in figure 5 represent a grouping of successful simulations. The percentage in the clusters represents the purity of the target class to the undesired class (successful to unsuccessful).

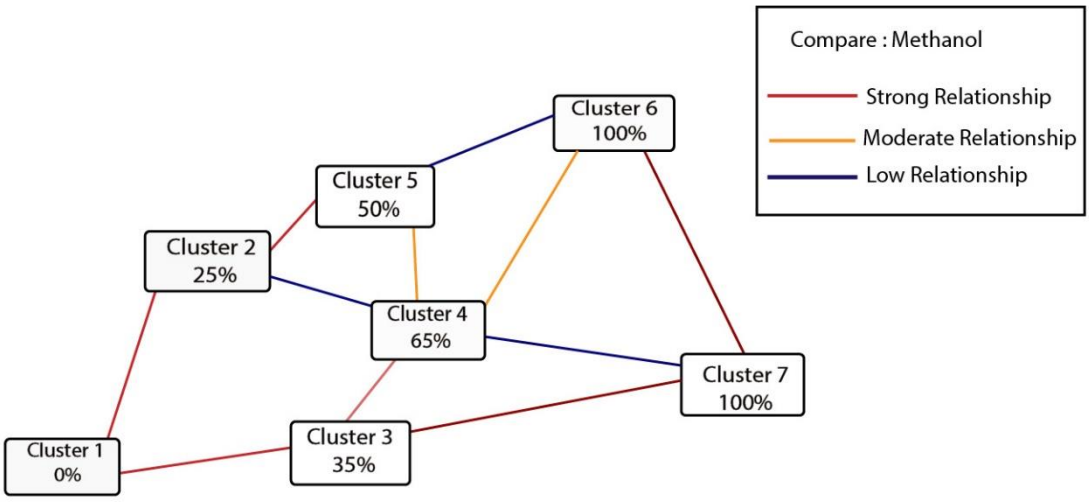

Figure 5: Cluster representation for the biodiesel system. The relationships of the clusters, denoted in Figure X by colored lines, are determined by the transition properties of the variable being compared, in this case, methanol.

7.1 Suggestion generator

Nearly two thirds of the simulation trajectories in the database result in biodiesel that does not meet American Society for Testing and Materials (ASTM) standards. These simulations do not help obtain potential regions of preferred behavior. The suggestion generator compares the users' input to the known clusters properties. If the input simulation values can be determined with a high level of confidence to a particular cluster that cluster is then used to compare transition qualities to other clusters with high conversion rates.

The data point itself is also weighted to the individual clusters properties as to not rely on the cluster identification process solely. The scores from both the cluster identification and individual data point are weighted and the suggestion generator determines the least number of changes needed to be made to the simulation input to produce a successful biodiesel meeting ASTM standards.

The suggestion generator is solely the end result of the more encompassing MLDM process, which has successful partition the simulation input-space into regions corresponding to successful simulations and non-successful simulations. This information is then produced to the crowd workers as previously described.

7.2 Machine Learning Measures

Different metrics are appropriate in different settings making the judgment of supervised learning algorithms dependent on the application [14]. The final performance measures we used are found in Table 1. They are achieved by 10-fold cross-validation (CV). An additional CV was performed to provide a reliable estimate of the final classifier's performance, which provided similar results.

Accuracy %	Precision	Recall	F- Score	AUC	Average Log Loss	Training Log Loss
0.957	0.961	0.955	0.958	0.992	0.131	0.81

Table 1: Results for the MLDM performance measures

Accuracy measures the proportion of true results to the total cases. Accuracy shouldn't be used alone to determine the efficiency of the classifier as it assumes constant and balanced class distribution for the data set [14]. Additionally, accuracy assumes "equal error costs" meaning a false positive is equivalent to a false negative [14]. To take class distribution and error cost out of the classification measure, rank metrics such as area under the ROC curve (ROC) are performed. This metric is considered as a summary of the model performance across all possible thresholds [14]. F-score takes both false positives and false negatives into account by the weighted average of precision and recall. Average Log Loss can be considered the penalty of a wrong result. Training log loss can be interpreted as the advantage the classifier has over randomly guessing.

Survey data from the Australian central Great Barrier Reef was used to classify soft coral taxa [15]. A variety of categorical attributes were considered along with spatial and physical attributes for four particular species. Classification and Regression Trees were used to identify each species by their corresponding environmental characteristics. The misclassification rate was 9.1% with 34 out of 373 cases being misclassified, which is an accuracy rating of 90.9%.

In [16] different decision tree classifiers where trained on medical data with the intent to diagnose various ailments. The data sets contained varying numbers of attributes with both discrete and continuous features. An Iterative Dichotomiser 3, Classification And Regression Trees, and C4.5 classifier algorithms were used on Diabetes and Heart StatLog data. Comparative from the results obtained in this work, these measures performed poorly with relatively similar data set, ranging from an accuracy of 57% to 78%. The difference in accuracy shows the flexibility and quality of our approach.

8 Conclusions

This work is an example to show how the integration of MLDM into formal methods can provide a way to analyze formal models of complex systems. In this work, our proposed MLDM technique is applied to the simulation results to improve crowd worker simulation quality. While this is a trivial objective for the capabilities of this integration, it shows the flexibility of providing actionable feedback to users to based on 850 simulation runs.

References

[1] Riley, D., Zhang X, and Xenofon Koutsoukos. “Biodiesel Sim: crowdsourcing simulations for complex model analysis” *Proceedings of the ANSS* (2013).

[2] Vahed, A., and Christian W. Omlin. "A machine learning method for extracting symbolic knowledge from recurrent neural networks." *Neural computation* 16.1 (2004): 59-71.

[3] Ly, Daniel L., and Hod Lipson. "Learning symbolic representations of hybrid dynamical systems." *The Journal of Machine Learning Research* 13.1 (2012): 3585-3618.

[4] Ciocchetta, Federica, et al. "Integrated simulation and model-checking for the analysis of biochemical systems." *Electronic Notes in Theoretical Computer Science* 232 (2009): 17-38.

[5] Riley, D. D., and Xenofon Koutsoukos. "Probabilistic verification of a biodiesel production system using statistical model checking." *Mathematical and Computer Modelling of Dynamical Systems* 20.5 (2014): 452-469.

[6] Maccagnola, Daniele, et al. "A machine learning approach for generating temporal logic classifications of complex model behaviors." *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, (2012).

[7] Aggarwal, Charu C. *Data mining: The textbook*. Springer, 2015.

[8] Montes, Jesus, et al. "A machine learning method for the prediction of receptor activation in the simulation of synapses." *PloS one* 8.7 (2013): e68888.

[9] Alpaydin, Ethem. *Introduction to machine learning*. MIT press, (2014).

[10] Jain, Anil K. "Data clustering: 50 years beyond K-means." *Pattern recognition letters* 31.8 (2010): 651-666.

[11] Yeung, Ka Yee, and Walter L. Ruzzo. "Principal component analysis for clustering gene expression data." *Bioinformatics* 17.9 (2001): 763-774.

[12] Schapire, Robert E. "The boosting approach to machine learning: An overview." *Nonlinear estimation and classification*. Springer New York, (2003). 149-171.

[13] Basu, Sugato, Arindam Banerjee, and Raymond Mooney. "Semi-supervised clustering by seeding." *In Proceedings of 19th International Conference on Machine Learning* (2002).

[14] Provost, Foster J., and Tom Fawcett. "Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions." *KDD*. Vol. 97. (1997).

[15] De'ath, Glenn, and Katharina E. Fabricius. "Classification and regression trees: a powerful yet simple technique for ecological data analysis." *Ecology* 81.11 (2000): 3178-3192.

[16] Lavanya, D., and K. Usha Rani. "Performance evaluation of decision tree classifiers on medical datasets." IJCA) International Journal of Computer Applications 26.4 (2011).