2017 National Information Technology Conference (NITC), 13-15 September, Colombo, Sri Lanka

System Abnormality Detection in Stock Market Complex Trading Systems Using Machine Learning Techniques

P. A. Samarakoon
Department of Computer Science
University of Colombo School of Computing
Sri Lanka
prameeshasamarakoon@yahoo.com

D. A. S. Athukorala
Department of Computer Science
University of Colombo School of Computing
Sri Lanka
aja@ucsc.cmb.ac.lk

***Abstract*— Stock market trading systems are real time systems that process thousands of data per minute and are considered to be critical as well as complex. It incorporates the features of complex business processing and sophisticated in-memory processing techniques for speed and throughput. These systems are distributed in nature, and they use a large number of processing nodes incorporating fault tolerance mechanisms. Complex systems also have a large effective number of strongly interdependent variables. Hence detecting faults and failures in stock market systems is a complex and cumbersome task. The study explores machine learning techniques to detect anomalous behavior to provide warnings before a system results in a fault or failure state.**

The study extensively utilizes a supervised learning approach with machine learning algorithms such as C4.5, Naïve Bayes, and ensemble techniques; bagging and Random Forest. The system statistics captured from log files are preprocessed and transformed to eliminate system environment dependencies. For each of the three components the initial feature selection is carried out manually using domain knowledge and expertise. Initial feature selection based on domain expertise was required as the number of features per component is large and does not closely relate to the system state. Feature selection methods (Info Gain algorithm with Ranker search) have been successfully employed to filter out unrelated attributes and to reduce computational complexity. A comparative evaluation is performed under each component status prediction. This study also utilizes oversampling techniques to overcome limitations caused by the class imbalance phenomena.

A range of evaluators are used to analyze the results and effectiveness of the models. The highest accuracy and Receiver Operating Characteristic (ROC) values are achieved when C4.5 decision tree is applied to the oversampled feature set and when the Random Forest algorithm is applied to the oversampled feature set. However, precision, recall and F-measure values vary. Root cause detection for anomalies and numeric values for system health predictions are future work in the research.

Keywords*— *machine learning, supervised learning, distributed systems

I. INTRODUCTION

The main objective of this study is to explore the possibility of applying machine learning approaches efficiently and effectively to detect component anomalies to derive system health. Due to the complex behavior of the system, a user would also need specific domain knowledge and experience to accurately and efficiently analyze issues in the system. Adding to the complexity, a stock market system handles various kinds of data, hence information observable by the system is unstructured and complex. Due to the heavy processing of data, the amount of data received from observations, is difficult to analyze. Each component plays a critical role for the system to function as expected. The distributed nature states that components correlate, and hence one process termination could lead to a failure of the entire system. However, there is no single system aware of the overall system health or status. When the system terminates, a user needs to manually check all processes, clear error warnings and identify the root cause of the failure, which is cumbersome and time consuming.

With the use of a learning model to detect abnormal behavior, it is possible to create trading systems that are aware of its current state; also defined as 'self-aware' systems. The system is aware of its current state and hence is able to take remedial actions and activate fault tolerance mechanisms. The research works closely with a tier one trading solution provider to obtain the required data and statistics. Trading systems are distributed in nature and contain multiple components.

Machine learning (ML) has been recognized as a type of artificial intelligence which focuses on computer program development that can teach themselves to nurture without being explicitly programmed and change when exposed to new data [2].

The main objective of the research is to create a framework to detect abnormal behaviors of a system and the predict system states.

II. RELATED WORK

Anomaly detection in relation to machine learning is an area that is heavily studied, however the application of machine learning to complex and distributed trading systems

has not been deeply explored. However, related applications of machine learning techniques for distributed systems are as follows:

C Schneider et al. presents a method of achieving self-healing in systems, detecting and predicting software faults through artificial neural networks (ANN). The approach focuses on evaluating the source of a fault with a system by using restricted Boltzmann machines (RBM) to predict the state of a feature [1]. The data captured by the Boltzmann machine is then classified through the fitness tests as either valid or invalid. Results from these tests will determine the overall state of the system and subsequently categorize the data collected. RBMS use a learning algorithm to evaluate and predict changes in the feature behaviour by periodically analysing the data gathered periodically from the system. The RBM uses a lazy evaluation strategy on features that are determined to have changed from the last good configuration and the faulty configuration data. Features that are determined to behave in an unexpected manner are added to the list of potential faults with a confidence value assigned by how unlikely the behaviour is to have occurred according the RBM. The list sorted in descending order according to its confidence value provides the application a method of determining the root cause of the fault and the ability to prioritize self-healing strategies. The drawbacks stated in this approach include costs of longer initial wait time for results, higher variability within the results and the requirement of larger training sets for accurate results.

N Gormitz et al. state that even though anomaly detection is more of a unsupervised task due to the unlimited number of possibilities of errors/defects, it fails to match the required detection rates in many applications, hence the need to have labeled data to guide the model generation [2]. Further, the use of classical semi supervised anomaly detection originating from classical supervised methods hardly detects new and unknown anomalies. The proposed approach here is to follow an unsupervised learning approach while using labeled information to obtain better accuracy in error detection. The results of the research credit supervised and semi supervised approaches over a purely unsupervised approach.

C Schneider et al. introduce a novel method for autonomously generating investigation leads that help identify system faults [3]. The Hidden Markov Model (HMM) is proposed to heuristically identify the root cause of a fault in an unsupervised manner. As per the approach using Baltzman machines [1] the HMM is to learn the existing model without human intervention and evaluate the changes in feature behavior with the use of a learning algorithm. However, ANN identifies more leads than HMM with a higher level of confidence, however, HMM predicts the fault accurately despite the excess information given to the model. The approach solely focuses on combining fitness functions and HMMs to autonomously detect faults and provide a list of potential root causes.

C Schneider et al. contrasts the different management styles of the system with the learning methodologies used in a survey of self-healing systems [4]. The bottom up management style which has ad hoc interaction between systems widely use semi supervised and unsupervised learning methodologies.

This is more commonly associated with unsupervised learning systems. Anomaly detection can be broadly classified into two categories, namely Anomaly Detection Frameworks (ADF) and Unsupervised Behavioural Learning (UBL) [5].

ADFs sample information, storing changes in vectors used to train primitives to predict future feature changes and behaviours. UBM uses Self Organizing Maps (SOM) to map relationships between features to a two dimensional lattice which are used to predict the behaviour of features [5].

Schneider et al. discussing prevailing technologies in self-learning and defect prediction note that for single point predictions HMM utilizes Baum-Welch, while ANN leverages Naïve Bayes. Multi-point predictions produced by the RBM use Contrastive Divergence Learning (CDL). Single point prediction algorithms use previously observed feature behaviour to predict the current set of features. A set of previously known configurations are classified as good or bad according to a predefined fitness function. Hence considering the conclusions of related studies this study utilizes a supervised learning approach.

III. METHODOLOGY

.

The research primarily focuses on predicting a component level system state from which the overall system status can be derived. The trading system used for this research has over 20 components varying in functionality and complexity. Hence three main components have been selected primarily; namely the sequencing component, distribution component and the matching component. The selected components are key components in the trading system, and they represent an interconnected component structure. Figure 1 models the architecture for the proposed framework.

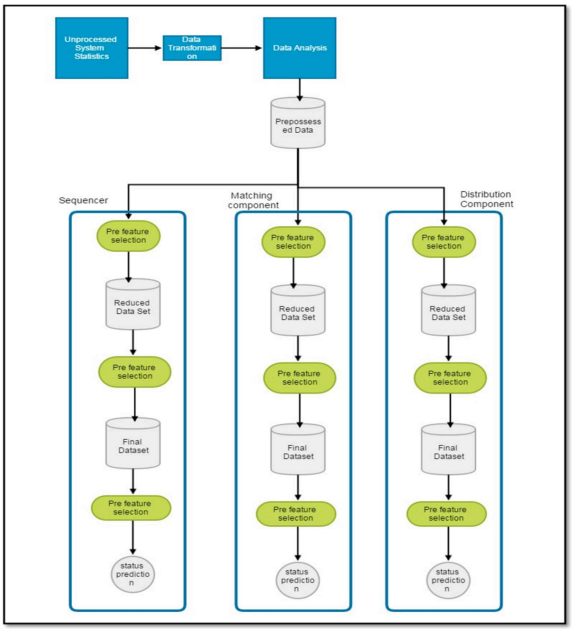

Figure 1 : Overall Architecture

A. Feature Selection

Feature selection is a necessity in many supervised learning tasks as the input is represented by a large number of features, most of which are not needed or related to predict the labels. Also referred to as attribute selection or variable selection, it chooses a smaller related set of features from a larger feature space. The Naïve Bayesian algorithm considers all attributes to be conditionally independent of each other. Adding redundant attributes reduces the efficiency of the classifier. However, as the features added to a classifier determine the outcome of the learning model, it is important to ensure that important features must not be eliminated in the feature selection process, more specifically at a manual feature selection stage. Hence the absorbing and understanding the feature space and acquiring domain expertise are of utmost importance. In relation to the current research the statistics are heavily dependent on the system and the domain it operates on. As all these components are of complex nature, catering to critical functionality and of a very large number, acquiring domain expertise can be time consuming. However, domain expertise was obtained by referring to specifications and discussing with development and support team experts for each component. Witten et al. state that manual selection is the best way to select attributes, understanding the learning problem and the actual representation of the features [16].

Fundamentally there are two types of feature selection algorithms, i.e. Filter and Wrapper methods. Filter methods determine the predictive subsets of feature sets based on simple statistics calculated from data, and they do not depend on the classification algorithm. This leads to a faster learning pipeline[10]. Wrapper methods approach is algorithm specific, but has the advantage of utilizing feature dependencies and taking into account the interdependencies of feature subset search[11]. Due to the expensive computational time taken when wrapper methods are applied and due to the benefits of filter methods discussed before, only filter methods will be used in this study for the feature selection. Two popular filter feature selection methods were used with the Ranker search method which ranks the attributes by their importance. Ranker was used with Info Gain Attribute Eval and Gain Ratio Attribute Eval, selected as the evaluation algorithm.

B. Classification Algorithms

- C4.5 Algorithm: Among decision tree algorithms, C4.5 is the most commonly used algorithm. It is capable of handling pruning, missing values and numeric values. C4.5 algorithm is implemented in Weka as J48 open source implementation.
- Naïve Bayesian: Naïve Bayes is a simple probabilistic classifier, based on the bayes theorem. It calculates posterior probability $P(C_i|X)$ using $P(C_i)$, $P(X)$ and $P(X|C_i)$ where $P(C_i)$ denotes prior probability of class, $P(X)$ denotes prior probability of data and $P(X|C_i)$ denotes the likelihood of the data given class [20].
- Random Forest : Random Forest operates by building a large number of decision trees at training time and outputting the class that is the mode of the classes (for classification prediction) or the mean of the label (for regression prediction) of the individual trees. Random Forests correct the overfitting problem of decision trees. The algorithm for inducing a Random Forest was developed by Breiman and Cutler and "Random Forests" is their trademark [16].

C. Evaluation Measures

It is essential to evaluate the performance of the derived models and methods using a certain evaluation criteria, in order to quantify the predictability and the accuracy of the prediction model. Accuracy is the most commonly used criteria used in evaluation. But in this research, it is highly probable that the datasets contain imbalanced data. Hence relevant measures must be set to evaluate such scenarios. These metrics can be described in three different categories.
Based on the True positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) values taken from the confusion matrix, individual performance measures such as Accuracy, Precision, Recall (Sensitivity) and ROC can be defined.

D. Class Imbalance Problem

The class imbalance problem occurs when the total number of a class of data is relatively far less than the total number of instances in another class of data; the number of instances in a class is not equal.

To mitigate the impact of the class imbalance problem mitigation techniques such as sampling was used. Oversampling and undersampling are the two main techniques used to correct the bias in the original dataset. The model should be capable of predicting the status of the components as ‘Normal’, ‘Not Normal’ or ‘Warn’. The research utilizes the oversampling technique, Synthetic Minority Over-sampling Technique (SMOTE) to generate synthetic minority instances to over-sample the minority class and ensemble based methods such as Bagging, Boosting and Random Forest have been applied on the balanced dataset.

IV. EXPERIMENTAL SETUP

The study identifies three critical components of a trading system; Sequencer, Matching component and the Distribution component to detect anomalies and predict the status of the components. The overall system health is determined based on the component status derived from the machine learning model. The study was based on statistical data obtained on a test trading system. The data preprocessing steps included handling missing values and data transformations.

Missing values can occur during generating test data sets when the component is offline. When the component is killed or offline no statistics will be generated from the component. Hence to avoid this situation the last recorded statistic will be recorded. These values need to be handled carefully as it may create a bias towards certain system status.

Data transformation is concerned with creating new features and attributes, and it also ensures that the features are not coupled to the system. Further, it creates more meaningful variables that are relevant and provides improved predictions. In this study most numeric data values, if used as they are, would tightly couple the machine learning model to the system. Hence to prevent this all numeric values have been captured as a ratio by dividing it from the previous value obtained.

The final feature set selected for each component is as depicted by Table I and contains statistics captured at each component of their respective connected components.

Table I : Feature set

Sequencer	Matching Component	Distribution Component
ME_SEQ_LastReceived	ME_DS_Connection	ME_DS_LastSent
ME_SEQ_Pending	DS_ME_PendingBA	ME_DS_Connection
SEQ_ME_Pending	ME_TransReceived	DS_ME_PendingBA
SEQ_ME_Size	ME_SEQ_LastReceived	DS_ME_LastRecived
SEQ_ME_BatchAck	ME_DS_LastSent	DS_DiskLatency
SEQ_DiskLatency	DS_ME_LastReceived	DS_ME_LastBA
ME_SEQ_Connection	ME_SEQ_Pending	
	ME_InternalLatency	
	SEQ_ME_Pending	
	SEQ_ME_Size	
	DS_ME_LastBA	
	SEQ_ME_BatchAck	
	ME_SEQ_Connection	

This study uses a 70%, 30% training data, testing data model as well as the 10 fold cross validation methods. The reason for using both methods is as the number of entries in the data set is limited. However, the results may be similar in both evaluations.

V. EVALUATION OF MODELS

Three approaches have been considered for each component. First is the initial approach of applying a single classifier algorithm. The algorithms used here are Naïve Bayesian algorithm and C4.5 Decision Tree algorithm. The second approach is to use sampling techniques to overcome the data imbalance problem using data oversampling. The third approach is to use ensemble based methods on the balanced dataset.

The dataset is categorized into three different states, namely 'Normal', 'Not Normal' and 'Warn'. The 'Normal' state represents an instance where no errors or warnings exist in the statistics of the component while 'Not normal' represents a state where a component shows visible failure. The 'Warn' state is when a connected component is in a fault or failure state.The dispersion for the original data for the sequencer module is as depicted in Figure 2. The total number of component statistic entries were 6500 records. The matching component and the distribution component follow the same dispersion.

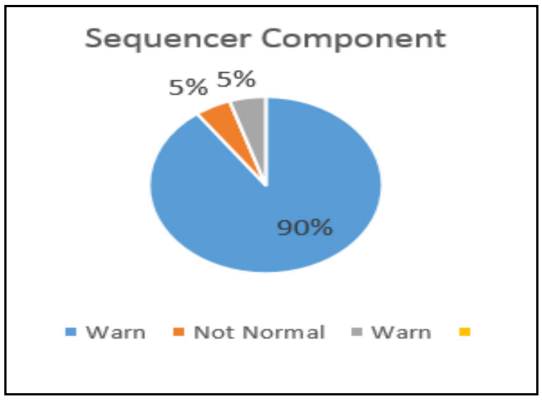

Figure 2 : Sequencer component data dispersion

The results of the Sequencer component with the Naïve Bayes algorithm post oversampling did not show significant results. The precision values for the 'Normal' state was 0.5663 which is significantly low. The results for the sequencer component with pre and post sampling, applying C4.5 algorithm have been analyzed in Table II. The results for matching component and the distribution component are similar to the results of the sequencer component.

Table III : C 4.5 algorithm sampling

	Pre sampling				Post Sampling			
Class	Precision	F-Measure	ROC Area	Recall	Precision	F-Measure	ROC Area	Reca
Normal	0.929	0.957	0.729	0.988	0.952	0.537	0.741	0.37
Not Normal	0	0	0.659	0	0.455	0.622	0.707	0.98
Warn	0.656	0.519	0.773	0.429	0.951	0.59	0.768	0.42

The usage of the ensemble technique random forest yielded results as depicted in Table III. The Precision, F-Measure and ROC Area values show significant improvements.

Table III: C 4.5 algorithm random forest

Class	Precision	F-Measure	ROC Area	Recall
Normal	0.982	0.564	0.758	
Not Normal	0.461	0.628	0.719	
Warn	0.954	0.596	0.775	

Figure 3 shows a comparison between the ROC values of the three instances; namely C4.5 before sampling and C4.5 after oversampling, and with the application of random forest. It is evident that the ROC values of all the states of the sequencer have increased significantly with each respective method. The implementation of C4.5 algorithm confirms this

phenomenon. The increase in the ROC value of the 'Normal' state is not significant in the scenario, however the ROC value of the minority classes have notably increased. The precision and F- Measure values for each method follow the same trend.

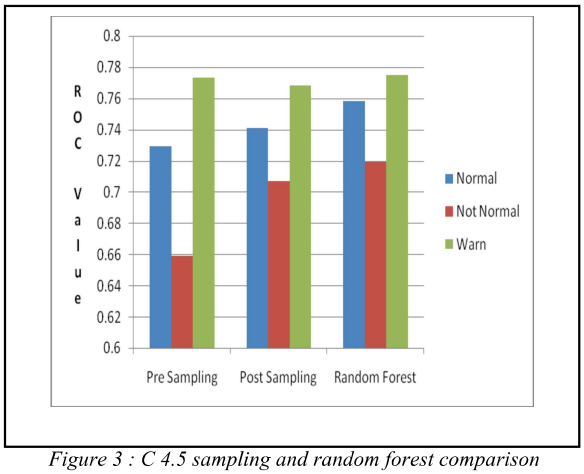

Figure 3 : C 4.5 sampling and random forest comparison

Table IV : Sequence component

Class	Precision	Recall	F-Measure	ROC Area
Normal	0.982	0.396	0.564	0.758
Not Normal	0.461	0.989	0.628	0.719
Warn	0.954	0.433	0.596	0.775

Table V : Matching component

Class	Precision	Recall	F-Measure	ROC Area
Normal	0.961	0.387	0.552	0.814
Not Normal	0.469	0.983	0.635	0.817
Warn	0.96	0.423	0.587	0.832

Table VI : Distribution component

Class	Precision	Recall	F-Measure	ROC Area
Normal	0.941	0.359	0.52	0.751
Not Normal	0.452	0.985	0.619	0.734
Warn	0.935	0.472	0.627	0.788

VI. CONCLUSION AND FUTURE WORK

Stock market trading systems are complex, critical and distributed systems which cannot afford system failure or faults. Hence it is imperative to ensure mechanisms to monitor and prevent faults. Abnormality detection is the key step to achieving the self-healing concept. With this study we have explored the application of supervised machine learning techniques to detect abnormal behavior in systems. The study focuses on three critical components which represent the complete system. This research demonstrates that it is plausible to detect a system state, given that the absolute essential feature set required is captured and the test data generation mechanisms are accurate. The research notes that class imbalance problems in generating and obtaining test data sets is inevitable, however it is important to ensure that it is mitigated, hence results are not affected.

Another major objective of this research was to explore the ability to identify the minimum relevant subsequent feature set to detect an abnormality of a component. Since the trading system it is important to capture the statistics of components that are connected to the principle component and hence should be present in the feature set.

The research also notes that the highest level of accuracy was obtained through the application of the Random Forest algorithm with oversampling techniques. Table V summarizes each of the evaluation models for the component. All three components hold similar values in terms of precision, recall, f-measure and ROC area. The low recall values for the 'Normal' class suggests false positives, hence needs further improvements before being applied to the production system. Future works on the study initially focuses on reducing false positives in the model.

The current data set generation mechanism ideally focuses on a particular environment. To avoid being tightly coupled to the environment the research performs data transformations. This functionality integrated to the system creates an additional computational overhead and prevents real time abnormality detection. Thus integrating this to a system could be done by monitoring the statistic logs externally and running the abnormality detection machine learning algorithms separately from the actual trading system.

A further improvement to the current module is to include system errors and warning log files to be processed along with the statistic files and the health status to be depicted as a numeric value. As the ultimate goal of the research once the abnormality is detected, the system should be able to mitigate and rectify the abnormality to ensure a stable state, and upon being unable to self-heal, it should provide a comprehensive analysis on the root cause.

REFERENCES

[1] C. Schneider, A Barker, S. Dobson, "Autonomous Fault Detection in Self-healing Systems using Restricted Boltzman Machines," University of St Andrews, Scotland.

[2] N. Gornitz, M. Kloft, K. Riek, U. Brefeld, "Toward Supervised Anomaly Detection," *J. Artif. Intell. Res.(JAIR)*, 2013, vol. 46, pp. 235-262.

[3] C. Schneider, A. Barker, S. Dobson, "Autonomous Fault Detection in Self-healing Systems: Comparing Hidden Markov Models and Artificial Neural Networks," *Proceedings of International Workshop on Adaptive Self-tuning Computing Systems* (p. 24). ACM, 2014.

[4] C. Schneider, A Barker, S Dobson, "A survey of self-healing systems frameworks," *Software: Practice and Experience*, 2015, vol 45, pp. 1375-1398.

[5] C. Schneider, A. Barker, S. Dobson, "Evaluating Unsupervised Fault Detection in Self-healing Systems Using Stochastic Primitives," University of St Andrews, Scotland.

[6] S. P. Kavulya, K. Joshi, F. Giandomenico, P. Narasimhan, “Failure Diagnosis of Complex Systems,” AT&t Labs Research, NJ, USA.

[7] M. Chen, A. Zheng, J. Lloyd, M. Jordan, E. Brewer, “Failure Diagnosis Using Decision Trees,” University of California at Berkeley and eBay Inc.

[8] S. L. Ting, W.H. Ip, A. H.C. Tsang, “Is Naïve Bayes a Good Classifier for Document Classification?” Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hum, Kowloon, Hong Kong .

[9] M. A. Sapaat, A. Mustapha, J. Ahmad, & K. Chamili, “A Data Mining Approach to Construct Graduates Employability Model in Malaysia,” *International Journal on New Computer Architectures and Their Applications*, *1*(4), 2011,), pp. 1111–1124.

[10] C. F. Aliferis, A. Statnikov, & I. Tsamardinos, “Challenges in the Analysis of Mass-Throughput Data : A Technical Commentary from the Statistical Machine Learning Perspective,” *Cancer Informatics*, *2*006.

[11] R. S. J. D. Baker, & K. Yacef, “The State of Educational Data Mining in 2009 : A Review and Future Visions,” *Journal of Educational Data Mining, 2009.*

[12] Y. Freund, R. E. Schapire, & M. Hill, “Experiments with a New Boosting Algorithm,” *International Conference on Machine Learning* , 1996, pp. 148–156.

[13] B. Hssina, A. Merbouha, H. Ezzikouri, & M. Erritali, “A comparative study of decision tree ID3 and C4.5,” *International Journal of Advanced Computer Science and Applications*, 2014, *vol. 4*, pp. 13–19, http://doi.org/10.14569/SpecialIssue.2014.040203.

[14] L. Breiman, “Bagging Predictors,” *International Conference on Machine Learning*, 1996, vol. *24*, pp. 123–140.

[15] R. Longadge, S. S. Dongre, & L. Malik, “ Class Imbalance Problem in Data Mining,“ *International Journal of Computer Science and Network (IJCSN)*, vol. 2, 2013.

[16] I. H. Witten, E. Frank,. “Data Mining: Practical machine learning tools and techniques”.