Abusing Pattern Detection System for Stock Market

W.P Dinuka

2021

Abusing Pattern Detection System for Stock Market

A dissertation submitted for the Degree of Master of Computer Science

W.P Dinuka
University of Colombo School of Computing
2021

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Student Name: W. P Dinuka

Registration Number: 2018/MCS/015

Index Number: 18440156

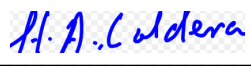
29/11/2021

Signature of the Student & Date

This is to certify that this thesis is based on the work of Mr. /Ms. _____ under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by,

Supervisor Name: Dr. H.A Caldera



29-11-2021

Signature of the Supervisor & Date

I would like to dedicate this thesis to my beloved parents.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. H. A Caldera for the continuous support of my research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my MSc study.

Besides my advisor, I would like to thank Mr. Banusha Lakmal, Mr. Primal Silva, and Miss. Pamuditha Liyanage for their guidance, encouragement, insightful comments, and hard questions.

Last but not the least, I would like to thank my family: my parents, for giving birth to me in the first place and supporting me spiritually throughout my life.

ABSTRACT

Investors participated in wholesale company's trade among market contributors at particular granted prices. The stock market is the place that fulfills those requirements. However, with the rapid dissemination of new information, maintaining efficient markets are hard to achieve and maintain. Anomaly is taken more important place which can be repeatedly happening or persists once and disappears. Some of them are led to taking profit using this strange behavior.

Investors/ need to be well aware of abnormalities in market parameters and they are likely to get into difficulty due to a lack of perceptions of market fluctuations. Detecting manipulations methods need to exist in the stock market and widely used Rule-based patterns as practiced. However, Operators are changing their trading patterns and they are mostly looking for the newest methods to operate stock market behavior. Rule-based or static recognition methods fail to recognize these new maneuver attempts. The main objective of the project is to overcome these challenges by implementing a research methodology to identify these evolving stock fluctuations patterns.

Artificial Immune system (AIS) theories are a class of computationally intelligent, rule-based machine learning systems inspired by the principles and processors of the vertebrate immune system. The algorithms are typically modeled after the immune system characteristics of learning and memory for using problem-solving (Coello, 2005). These applications which are based on AIS theories do not include separate training phases. The novelty of this research is having a training phase using domain expertise knowledge. The system was tested based on transaction data collected from Saudi Stock Exchange that is described in the next chapter.

The system is designed to detect price or volume abnormality detection using 10 years back archive data by various machine learning methods to get maximum accuracy. Price change, Volume, Turnover, and Trades feature are extracted through the more than 20 labels in the dataset that gives the best feature combination as the conclusion of domain expertise and using wrapper method which are described in the next chapters.

Statistical calculations are used to mark abnormality of selected data set with help of domain expertise. The literature review was done to find similar projects which still have limitations and identify a research gap. Some major design issues were identified when the designing phase and getting rid of them by designing solutions for each of them. The final implementation was done by selecting the most suitable techniques to design web applications. The evaluation phase was done based on the conclusion of the domain expertise knowledge.

Table of Content

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Chapter Overview

The introduction chapter gives brief overview of the stock market problem and discuss the selected market "Saudi stock exchange" and how the solution approaches the mentioned problem domain. Introduction also includes the project's aim and objectives. An expected distribution schedule and also the resources need to make the project successful will also enclose in this chapter.

1.2 Project Introduction

Investors participated in wholesale company's trade among market competitors at particular consent prices. Those requirements fulfill through the stock market. It is a great income source for investors where they can hold some percentage of ownership of several companies and get profit when the company earns money and when the stock price goes high.

There can be a single market or many markets and exchanges listed in the stock market where Bond, options, futures, and mutual funds can be traded. However, Stock traders can be directly traded through the stock market and some of the traders registered to third-party stockbrokers and trade through the broker. All the transactions take place through electronic mediums. Therefore, investors need relevant information about the stocks to make investment decisions.

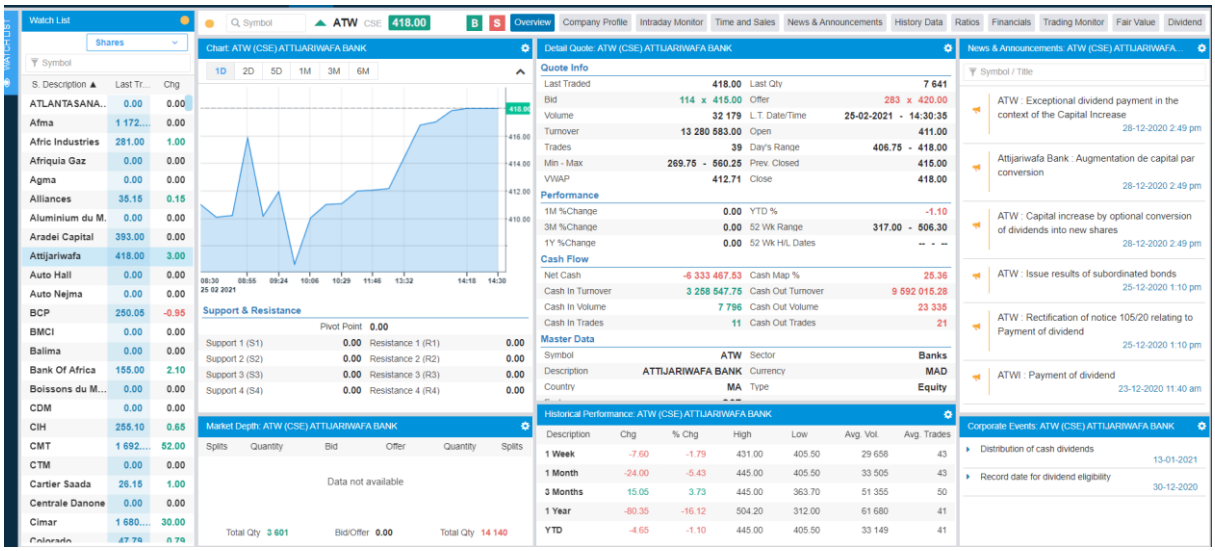


Figure 1: Summary of the trades in a selected company

Figure 1 Looking at this kind of dashboard broker or investor can be able to get an idea about how the stock market behaves in a particular company.

Saudi Stock Exchange

The Saudi Stock Exchange offers a wide range of services to market participants. In the current situation, the Saudi Stock Exchange (TDWL) has a highly advanced trading platform that provides a smooth trading experience through full automation and processing. All trade activities are electronically adjusted, validated, and activated according to the T+2 settlement cycle. The trading engine is designed to deliver several orders that can meet the needs of investors. TDWL offers negotiated deals and negotiated advertisements as special deals.

TDWL is a group of services provided by the Securities and Exchange Center Company in association with its members. It has a new bunch of financial services which aims at market participants, issuers, and traders. It benefits the investment community in the Saudi capital market and provides valuable services to contributors.

There can be all sorts of abnormalities in the stock market that result in new investors losing money, significant people making more profits, market indicators not showing the true picture of the market and stakeholders, and even companies as a whole falling. Fluctuations that evolve to maintain the stock market as a fair and secure place for all parties must be immediately identified and reported.

A market abnormality is an unusual action that violates the expected behavior of the market. Some abnormalities appear only once and are discarded. However, some of the others appear frequently. Traders and investors can use these unusual stock market practices to find opportunities throughout the stock market (Poterba et al., 2015), because of these, investors need to be well aware of abnormalities in market parameters and new investors are likely to get into difficulties due to a lack of awareness of these abnormalities happened in the market. Generally, people are likely to react to these sudden changes (increment & decrement) that happen in the market since they are hoping to make a huge profit. However, there are unseen scenarios that can be happened, some movements can be created to deceive people and get back their investing money.

Markets which are considered to be efficient are very difficult to initiate and difficult to maintain. The behavioral financial theory (Statman, 2008) explains their recurrence and how

traders can take advantage of the extraordinary market. Most of the abnormalities take place considering an unusual market behaves.

Figure 2 illustrates how some special announcements are informed to investors in the Saudi stock exchange.

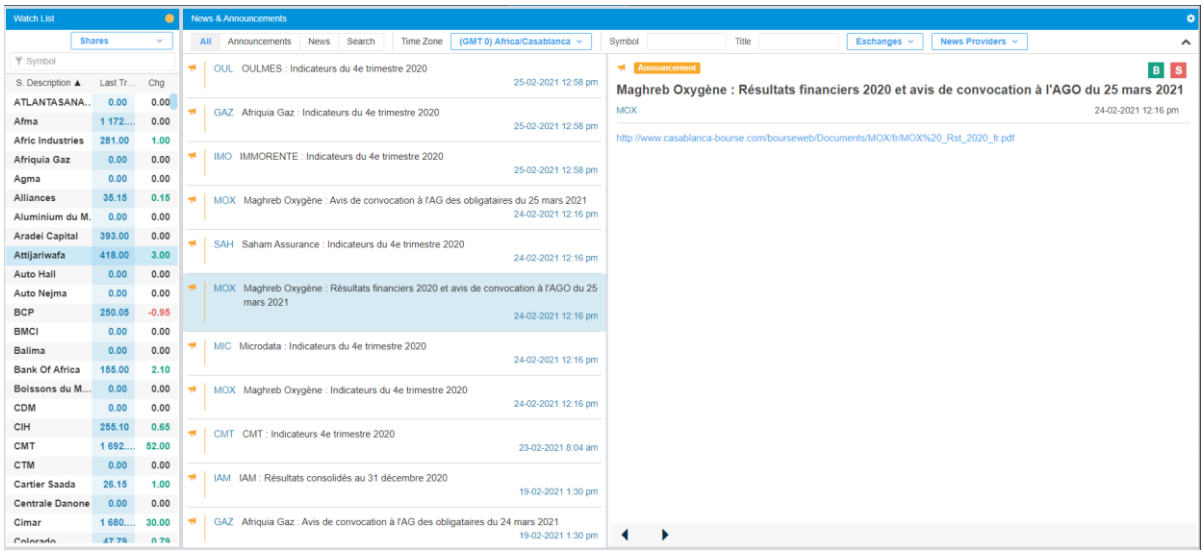


Figure 2: Inform special announcement and news to investor

Stock manipulation is performing transactions differently than usual investor or trader, intending to cheat others and get more profit. Some of those techniques are not illegal but they will cheat other investors or third party dealers. Many manipulations take place in the stock market due to this loss of its standing and led to some countries having laws against unauthorized transactions in the market.

Figure 3 shows the price graph of a company that has suspicious behavior (marked in circles and Red curve indicates normal scenario).

Figure 3: Stock Value Variation - 4300

Some of the abnormalities happen once and are discarded, while others recur. As a country economy can be affected by the stock market's behaves, the market need to be maintained as a trustworthy and steady place for every party to develop disparities as quickly as possible.

This methodology will be capable of identifying some of the stock market anomalies to a greater extend.

Ex: Insider Trading, Front Running, Pump and Dump, Poop and Scoop, Painting the Tape, Wash Sales.

This project will give a better solution to this problem by intelligently predicting anomalies and helping investors to perform transactions and gain profit without a loss.

1.3 Motivation

Identifying abnormalities in certain domains is a very critical task and abnormalities are evolving and detection systems must also compete to detect and defeat them. Abnormalities that happened in the market, identifying and distinguishing those from normal scenarios is not a direct forward task it is a very critical task. Natural immune system theories can be used to solve difficult problems using simple techniques. In the current situation, the stock market is more popular in the trading world since it affects the economy of the country so it is important to have a highly advanced security system to create a market that is more reliable and steady.

The stock market has very critical tasks, one of the important is predicting stock market forecasts. A lot of investors would like to know the future of the market, some of the forecasting systems indirectly assist traders which are giving back up information such as market direction. Data mining is also one of the techniques that provided various algorithms on the data to predict future market direction. Rule-based patterns are widely used in practice. However, according to discussions had with superiors in the domain it is failed to detect anomalies with the rapid dissemination of the stock market by using rule-based detection systems. Impact-driven algorithms also provide only trading strategy rather than anomaly detection and prediction too. Considering impact-driven algorithms it is highly motivated to implement an anomaly detection system using machine learning techniques too.

1.4 Statement of the problem

A Stock market is a place where symbols (companies) can be traded according to an agreement where it occurs many transactions some of them are led to abnormalities. Investors need to have many ideas of these evolving attempts because newcomers are likely to get into difficulties due to ignorance of these fluctuations.

Most of the people who don't have much experience in trading generally respond to sudden changes happening in the market with the hope of making a huge profit. However, some fluctuations can be artificially created to deceive people. Due to this handling, the stock market is one of the major problems in markets, thereby undermining the credibility of the stock market. Some countries have laws against trading in the stock market illegally.

Rule-based patterns and statistical detections are most used as existing maneuver detection methods. Operators are periodically changing their patterns and they are looking for new ways to handle the market. Recognize the volatility of stock market transactions, which is reflected in abnormal fluctuations in price and volume. Therefore, existing methodologies like rule-based or static recognition fail to recognize these new evolving maneuver attempts. Therefore, the problem of identifying abnormalities in the market is still open.

1.5 Aims and Objectives

1.5.1 Aim

Rule-based patterns and statistical methods are widespread in existing maneuver detection methods. However, as invaders rapidly changed their trading patterns and found new ways to attempt the stock market, static methods failed to recognize these maneuvers with rapidly changing markets and strategies.

The aim of this thesis is to gain an understanding of the stock market behavior and manipulation detection methods and design and implement anomaly detection systems, evaluate using real-time transaction by machine learning methods.

The system will be tested based on real transaction data collected from Saudi Stock Exchange. The first stage is supervised learning for price and volume anomalies detection for single company data against single classifier and the second stage is to optimization of the results to detect single company data against multiple classifier. 10 years of real transactions are used for testing in various models. The degree of the anomaly of a transaction is marked based on conclusions of three domain experts and system output will be evaluated based on them.

1.5.2 Objectives

As illustrated in 1.2 Project Introduction, Rule-based patterns are still widely used and have many drawbacks in fluctuation detection.

The main objective is to implement a research methodology to identify these stock fluctuations. This project approaches the problem by analyzing the price change, volume, turnover, and trades values of the transaction.

- To conduct a Literature survey to study the anomalies focusing on data.
- To identify and analyze the gap of existing solutions in detecting stock market anomalies
- To identify a data set that can be used in the context of the problem and means to perform data preprocessing such that it would be usable by the algorithms
- To examine machine learning approaches for the solution
- To understand machine learning approaches concerning a single model and Ensemble model approach
- Implementation of a model which is capable of finding abnormalities of data and calculating the degree of abnormality of a suspected fluctuation
- To implement the prototype fulfilling the identified gap using supervised learning approaches
- To evaluate the system in terms of the quality of the content

1.6 Scope

This anomaly detection system consists of price, volume, turnover, and trades values. The system only detects illegal anomalies. Illegal anomalies are anomalies that are treated as price or volume manipulation without the prior notice of the responsible parties in the market. The scope of the project is restricted to detecting fluctuations in stock market transactions, which

are reflected in abnormal fluctuations in price change and volume. Below anomalies were mainly targeted in the system proposed.

Insider Trading: Although all parties should have the same information about the company's financial Profiles, Company Managers / Directors (Internal) know the financial profile of the company in advance. They operate based on these and make at least huge profits risk

Front Running: When some people are going to buy a large number of shares, people learn about it in advance and buy shares at current prices and create an artificial price and increase those shares and then sell

Pump and Dump/ Poop and Scoop – Group of people attempts to push up/down the price by spreading rumors

The dataset is flown through Directing archive data systems. Feed data contains price change, volume, turnover, and trades that require evaluation.

The system predicts transaction that is normal, or abnormal whether it is price fluctuation or volume fluctuation and it shows the accuracy, false positive, and False-negative values of the system.

1.7 Structure of the thesis

An introductory chapter provides a brief overview of the stock market domain and discusses the selected market “Saudi stock exchange” and how the solution approaches the mentioned problem domain. Introduction also includes the project’s aim and objectives.

In literature review chapter will cover essential findings that are similar research conducted in recently, associated works along with algorithmic review then research gap will be identified as a results and Methodology chapter provides selected research and design methodology in-depth with key aspects of data gathering technique and system components and described how they interact with each other.

In System design chapter covers an important part of the dissertation by designing system components analyzing design problems along with a solution for each of them and discussing algorithmic aspects too. The system implementation chapter provides a detailed explanation of

each module discussed in the System design chapter along with system implementation details with technology/framework used.

Evaluation and Results chapter is also the major chapter that needs to be discussed in the dissertation which includes an overview of findings in the solution proposed along with evaluation methodology by defining evaluation criteria and selected approach feature combination is also discussed in this chapter.

In the end, the conclusion chapter summarizes the implementation and its findings along with limitations of the solution proposed and achievements with their status, and the future direction of the proposed solution.

CHAPTER 2

LITERATURE REVIEW

2.1 Chapter Overview

The Literature review chapter shall give necessary background information about similar research conducted in recent years and some of the existing solutions will be critically evaluated and identified limitations and workflow there. The flow of the literature review will be covering associated work along with algorithmic analysis too. At the end research gap will be identified as the conclusion.

2.2 Anomaly detection of Time Series Data

2.2.1 Introduction

The proposed solution is a pre-determined process of analyzing the incoming dataset and identifying abnormalities. First, the dataset is divided into separate pieces using sliding windows. Then encode each piece and store the values as auto values. Randomly generates threads of values that match the stored auto-values. Finally, store values that do not match self-values. Reliable and effective tool breaking technology is required for instant delivery responding to unexpected tool failure to prevent damage to the work piece and machine tool. Observe the behavior of the cutting force and report possible failures in the proposed solution. This solution can be used to identify variables in the data set that are stable and timely behavior.

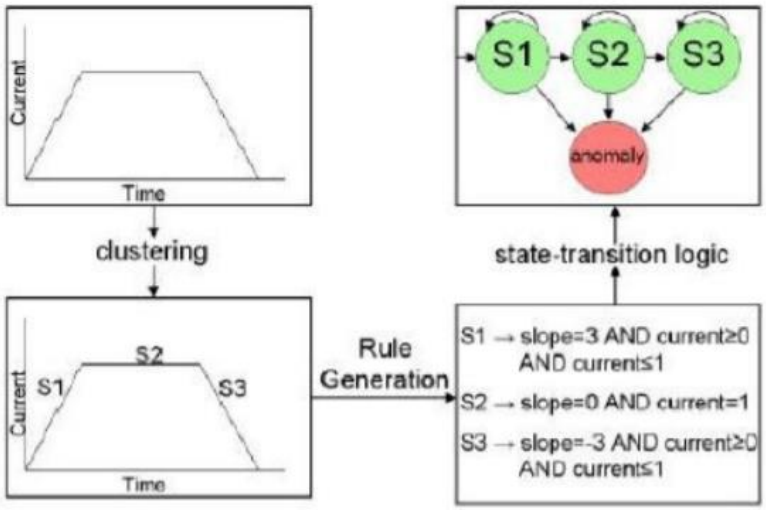


Figure 4: Main steps in time series anomaly detection

Many researchers have introduced different methods for identification abnormalities in time series data. (Salvador and Chan, 2005) introduced a statically activated method. There are three parts to data clusters, rule generation, and anomaly detection. Gecko algorithm used for clustering is shown in Figure 4. The data is clustered to the specified maximum number. Cut and reassemble into a specific number of blocks. To identify the exact number "L" method is used. After the cluster generation, a specific set of rules is generated and passed through its identified data locations. The data flow is expected to conform to the rule generated by this solution setup. Therefore, this solution is not suitable for a highly dynamic data stream behavior.

.

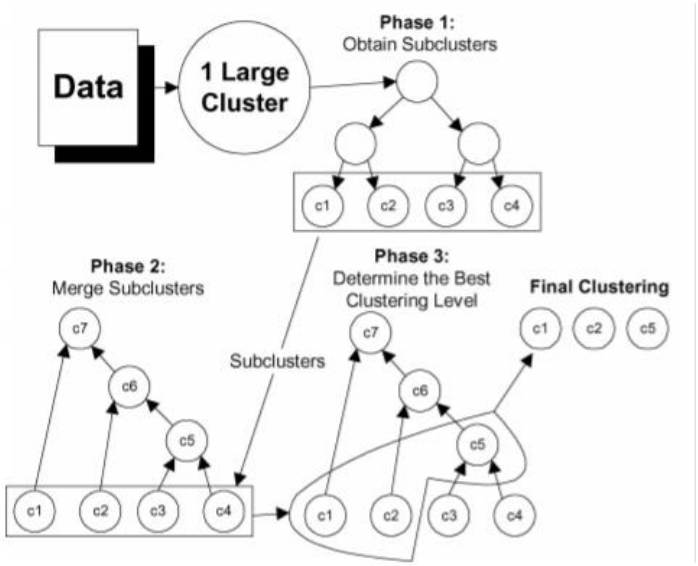


Figure 5: Overview of the Gecko Algorithm

2.3 Anomaly detector for financial in the retail sector

2.3.1 Introduction

Refers to the detection of fraud in the retail sector (Jungwon, Ong and Overill, 2003). With the major intervention of technology in retail, e-commerce has become a major component, although there are many advantages to electronic money transactions and some security issues.

Because the system is based on electronic transactions, different technologies are used to deceive buyers and sellers into not seeing each other.

Other than that, fraudulent transactions or split transactions are included in the system and they will get a higher salary as the payment depends on the number of transactions. (Jungwon, Ong and Overill, 2003) suggest that they identify these discrepancies by observing transaction patterns and identifying unusual transaction patterns. A-Priori algorithm and stored rules are used to detect abnormalities. This paper points out when using positive selection, using the system can handle large amounts of data reducing the workload of negative selection. The methodology behind this approach is the generation of mutants that are away from self-antigens. This introduced a novel AIS which is designed to expand a huge volume of real data.

2.4 Abnormality detection of stock markets

2.4.1 Introduction

Stock market abnormalities can be categorized into two categories as mentioned in (Ferdousi and Maeda, 2006)

- Abnormal fluctuations of price
- Abnormalities in individual behavior

The solution to the discrepancy was focused solely on the detection of anomalies by analyzing the unusual patterns done by the stockbrokers. This is of considerable importance as most are done by stockbrokers. The proposed solution is called peer group analysis (PGA), which identifies stock brokers with the same set of features. It then statically analyzes and recognizes the group. Ultimately, the way people react to the individual is different than what others recognize.

This solution is tested in the stock market other than the Saudi stock exchange and shows specific results.

2.4.2 Drawbacks of the solution

- PGA is not capable of identifying normality as normal since it depends only on peer behavior.
- Consider stock brokers behavior only

- This approach does not consider the number of individuals involved in a single transaction.
- Sudden Change which is increment and decrement of price and volume is not taken into account in this solution.

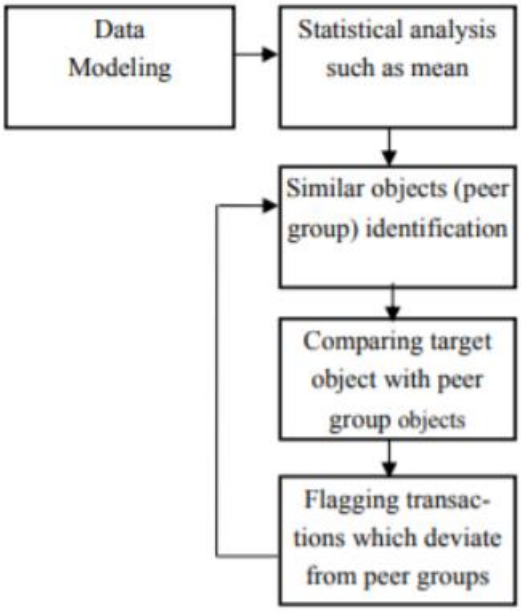


Figure 6: Anomaly detection based on PGA

2.5 Detect Stock Market Manipulation: Supervised Learning Approach

2.5.1 Introduction

System results show that monitoring learning algorithms can be promising to identify market maneuver attempts samples using a case-based labeled database. This highlights the importance of systematically synthesizing maneuverable samples that can be integrated with real-market data to train and test. The main limit of this project (Golmohammadi, Zaiane and Diaz, 2014) is limiting the probability of market shifts on time.

Algorithm	Sensitivity	Specificity	Accuracy	F_2 measure
Naïve Bayes	**0.89**	**0.83**	**0.83**	**0.53**
CART	0.54	0.97	0.94	0.51
Neural Networks	0.68	0.81	0.80	0.40
CTree	0.43	0.95	0.93	0.40
C5.0	0.43	0.92	0.89	0.35
Random Forest	0.32	0.96	0.92	0.30
kNN	0.28	0.96	0.93	0.26

Figure 7: Performance of supervised learning approach

2.5.2 Drawback of the system

Model is trained using a single dataset, the problem of their generalization can be reasonably raised.

2.6 Identifying timeline contextual anomalies (CAD)

2.6.1 Introduction

Detection of time series anomalies is one of the basic problems of data digging and it solves various problems in different domains such as detection of unauthorized access to computer networks, detection of health security sensor data irregularities, and detection of insurance or securities fraud. Although extensive work has been done to identify anomalies, many techniques look for individual objects that are different from ordinary objects but do not take into account the temporal aspect of the data.

2.6.2 Drawbacks of the system

The main shortcoming that makes supervised learning approaches inappropriate for identifying potential behaviors in the stock market is the need for named data since this approach relies upon labeled data, so the results are based on a very restricted set of samples compared to the number and variability of the various industry segments in the stock market (Golmohammadi, Zaiane and Diaz, 2014)

2.7 Stock market prediction using CNN based approach

2.7.1 Introduction

Extracting features from the financial data of the market forecasting domain where many approaches have been proposed is a very important issue. Among other contemporary tools, convolutional neural networks (CNN) have recently been used for automated feature selection and market forecasting. However, have focused less on the correlation between different markets as a potential source of information for feature extraction, and this approach based on CNN can be applied to data collection from a variety of sources, including different markets, to quote features to predict the particular market direction (Hoseinzade and Haratizadeh, 2019).

2.7.2 Drawbacks of the system

However, according to the research paper, Limitations can be identified as noisy and irregularity of prices in the market then prediction becomes complex using this solution.

2.8 Mining illegal internal stock trading

2.8.1 Introduction

Illegal internal trade in stocks is based on the release of unpublished information before the disclosure of information. Illegal internal trading is difficult to identify due to the complicated, irregular, and unstable nature of the stock market is described in (Haft and Haft, 2014) .

In this solution, the author presents an approach to early detection and forecasting of illegal internal trade from large varied sources of both structured and non-structured data using an in-depth learning-based approach with the processing of discrete signals in time series data.

In addition, the author uses a tree-based method that depicts events and actions and helps analysts understand large amounts of non-structural data.

2.8.2 Drawback of the system

This approach is limited to the number of case studies. In general, the more cases that can be studied the more, the outcome does not change.

2.9 Natural Immune System Applications

2.9.1 Negative Selection

Negative selection algorithm is used to classify and pattern recognition problems. In an abnormality detection domain, the algorithm develops a set of patterned detectors that have been trained on general (non-heterogeneous) patterns that can be identified by invisible or invisible patterns. The NSA derives its motivation from the negative selection process of the natural immune system. At the thymus, if an autoimmune cell is detected by a T-cell, it is removed and then immunosuppressed during a T-cell maturation process. It is mainly used to identify anomalies. This algorithm creates a set of detectors that include only automatic threads. Later, this set of detectors is used to detect anomalies.

The NSA algorithm has two main steps. The first step is to block the matching of self-threaded and randomly generated threads. Matching threads are rejected Figure 8. Unmatched threads will be moved to the detector set. In the second step, the security threads are matched to those in the detector kit. Recognize that the matching threads are not auto and re-match the rest (Scholar, 2017).

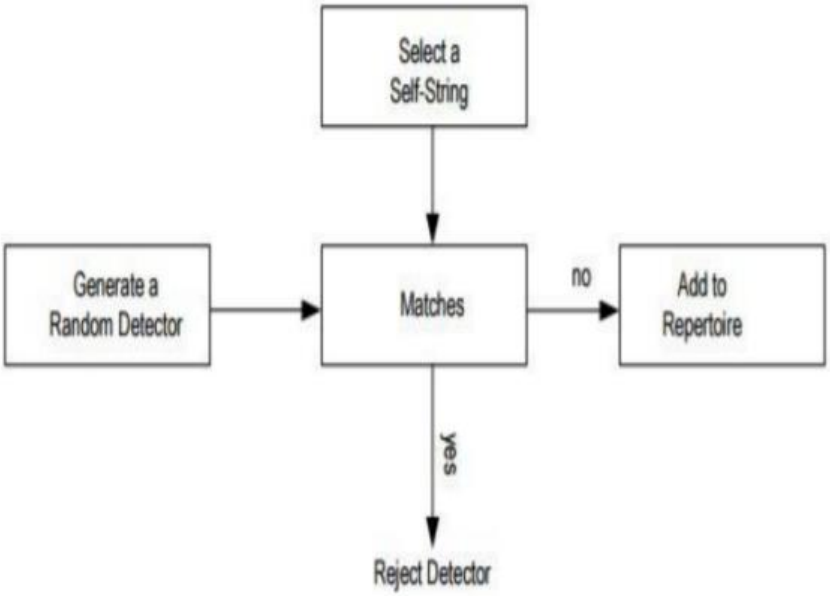


Figure 8: Selection (Scholar, 2017)

As described in the above figure, the immune system generates possible detectors randomly and then those detectors are sent through a mutation process. Generated candidates (detectors)

are matched with sample self-cells and destroyed if matched. Likewise, detectors are generated which might be matched with non-self-cells (Yegnanarayana, 1994)

2.9.2 Clonal Selection

In the Clonal selection process, when a detector identifies an antigen, it is subjected to proliferate process which is diversified detectors generated which are more capable of capturing the same antigen next time. Detectors that are closely related to antigen will eliminate it and finally, it will be stored (Yegnanarayana, 1994). This algorithm was inspired by the clone selection theory of acquired immunity, which explains how B and T lymphocytes improve their response to antibodies over time. Clone selection is the process of identifying antibodies, cell proliferation, and modification into memory cells.

Many AIS algorithms use clonal immunization features. This theory states that the organism has a heterogeneous (uniquely unique) antibody pool that existed before. These antibodies can identify all antibodies that are present at a certain level, when matched with an antigenic antibody, the cells can regenerate and form cells. During the cell proliferation phase, genetic mutations occur in cell clones.

This allows the cells to improve their ability to bind over time by exposure to antibiotics. According to the Darwinian microcosm, the best-suited cells for survival are selected as shown in Figure 9.

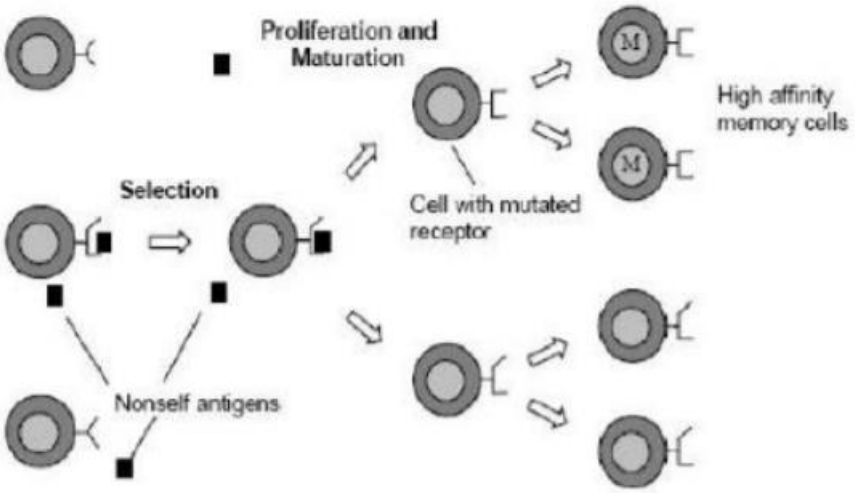


Figure 9: Clonal Selection (Yegnanarayana, 1994)

2.9.3 Fraud detection monitoring: Supervised Learning

Labeled training data are required to identify monitored anomalies. Labeled instances are used to construct a predictive model that categorizes normal and anomalous activities. The unlabeled new data is compared across the created model to determine which class it belongs to. Using this Fraud detection method is that it is very difficult to obtain a specific and representative labeled database and anomalous data characteristics are very low compared to the average data characteristics of the training datasets. This issue of imbalance needs to be addressed (Leangarun, Tangamchit and Thajchayapong, 2019)

Events published in the Turkish Capital Markets Board have labeled market manipulation opportunities. The purpose of this research was to create a predictive model using two data excavation methods, the auxiliary vector machine (SVM) and the artificial neural network (ANN), and compare it with statistical methods (Yegnanarayana, 1994). Experimental results show that SVM and ANN surpass statistical technologies.

2.9.4 Rule Induction

Rule induction is a data mining technique used to track stock market fraud. This system is similar to the existing regulatory rules for market monitoring, which makes it interesting among

auditors and securities market analysts. A fraud detection system is implemented using legal induction.

First, randomly generated rules using the association rules algorithm apparition apply to a set of data containing only legitimate transactions. Any rules that match the data will be ignored. The rest of the rules are used to monitor new transaction data in the system. Any law that detects an anomaly is reproduced by adding a small random distortion. Retains all successful rules for identifying anomalies (Grzymala-busse, 2005).

2.9.5 Regression

Most of the economic models are intended to predict market trends and to identify maneuvers through linear reversals, the Automatic Reactive Movement Average (ARMA), the Automated Combined Movement Average (ARIMA), and other such models.

However, these statistical models can only handle linear data which do not handle very noisy, informal, non-linear, irregular data like stock market data. These approaches often fail to accurately predict markets and maneuvers. Logistic retrograde models were used to identify the market manipulation of the Shanghai and Shenzhen markets (Bhuriya *et al.*, 2017).

Market characteristics are analyzed using basic component analysis to enhance the predictive performance of the model. This model proved to be better than linear regression models with a high success rate for predictions (Bhuriya *et al.*, 2017).

2.10 Research Gap

After reviewing existing solutions two main approaches were identified, the use of statistical approaches, rule-based systems, and non-supervised learning models. Data manipulation can be done by simply changing the anomaly patterns by changing the rules. In statistical or rule-based models, the system must be changed whenever these rules change and also this kind of system can’t be used for similar markets since thousands of real-time transactions occur per minute.

And considering unsupervised solutions, the systems which build using an unsupervised approach are much better since those address the main drawbacks as mentioned above. However, those models generally do not handle noisy data, large amounts of data. Noisy data

can be an anomaly so separate noisy and actual anomalies will be a challengeable task with rapid dissemination and it also needs to be considered as the accuracy of the proposed approach.

Most of the existing applications are capturing only price fluctuations and given priority to them rather than volume changes, which lead significant false positive error rate of results. Some methodologies has been used to apply to detect variation of a dataset that has consistent and periodical behavior (Salvador and Chan, 2005).

Nearly 10 years back archive data were gathered in the Saudi stock market that contains more than 20 independent variables. Feature selection methods have been successfully used to filter out unrelated attributes and to reduce computational complexity. The study utilizes a supervised learning approach with machine learning algorithms to perform with good accuracy to identify abnormality of market transactions. And also using this solution users would be able to get suspected scenarios in minimum fault tolerance with better performance when compared to rule-based models. If this research is a success in filling the identified gap, the system will be the most wanted solution for a very common problem in the domain.

2.11 Chapter Summary

Application	Methods	Separate learning phase	Identify both price and volume	Is machine learning technology involved	Suitable for dynamical behave markets
Learning States and Rules for Detecting Anomalies in Time Series	Gecko Algo, L method Rule based patterns	x	x	x	x
Anomaly detector for financial fraud in retail sector	A priory algorithm, stored rules, positive selection	✓	x	x	x
Anomaly detection of stock markets	PGA	✓	x	x	x
Detect Stock Market Manipulation: Supervised Learning Approach	Case based single database	✓	x	x	x
Stock market prediction using CNN based approach	CNN	✓	x	x	✓

Figure 10: Summary of conclusion in the selected findings

The Literature review chapter summarizes the essential background information about similar research papers, journals conducted in past few years to up to date. Existing applications were

critically evaluated and identified drawbacks and addresses limitations thereby discussing workflow associated along with algorithmic analysis. At last research, a gap was identified. The next chapter discuss methodology of the solution proposed along with data gathering methodology, feature extraction and at last class imbalance problem discuss in detail. The next chapter provides methodology of the system in detail along with feature extraction, class imbalance problem and data gathering methodology.

CHAPTER 3

METHODOLOGY

3.1 Chapter Overview

The methodology chapter discuss selected methodology in detail and design methodology with design assumptions related to the scope of the concept, research approach with system workflow. A key aspect of the data-gathering technique will discuss at end of the chapter.

3.2 Research methodology

There are several types of research related to artificial immune systems and their applications which are currently used for stock market abusing pattern detection. Artificial immune system domain, stock market anomalies, and targeted artificial immune applications were analyzed and conducted few researchers explained in the literature review chapter.

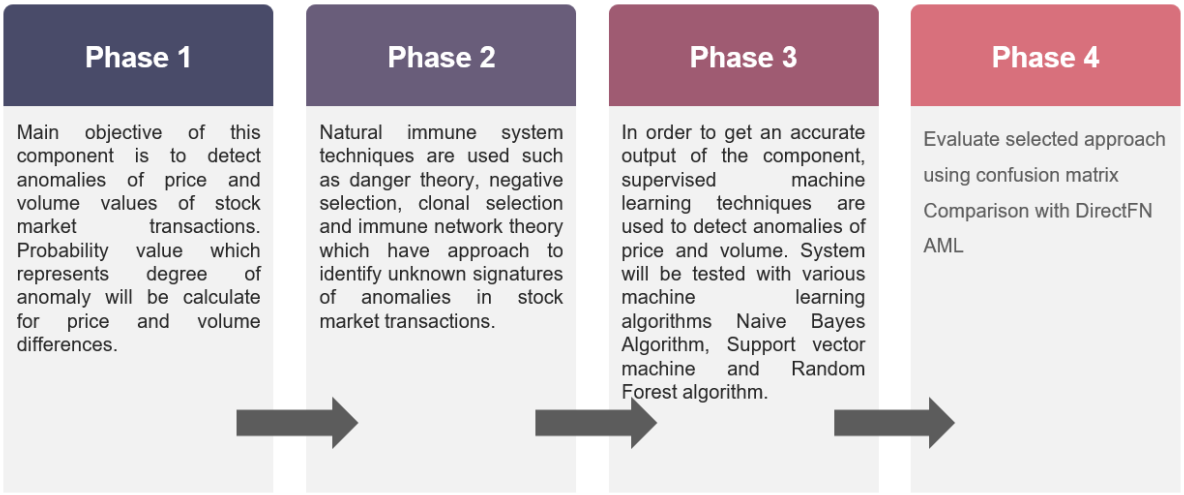


Figure 11: Overview of methodology

3.3 Data Gathering methodology

The literature review was done at the beginning to identify the research gap of the problem domain. Then handled interviews with three domain experts and developers helped to gather

major usages and requirements. It was a critical period to select which market should be more effective for this approach after analyzing the information given by expertise. The "Investing.com" platform was also used to gain an understanding of various markets. Riyadh bank (1010) company has been selected for a single company with a single classifier when considering the highest volatility.

Figure 12: 1010 technical chart

Prev. Close	**26.05**	Day's Range	**25.95 - 26.15**	Revenue	**6.56B**
Open	**26.15**	52 wk Range	**16.54 - 27.2**	EPS	**1.53**
Volume	**881,313**	Market Cap	**78B**	Dividend (Yield)	**0.50 (0.00%)**
Average Vol. (3m)	**1,890,540**	P/E Ratio	**17.02**	Beta	**1.25**
1-Year Change	**44.28%**	Shares Outstanding	**3,000,000,000**	Next Earnings Date	**Jul 21, 2021**

Figure 13: overview of 1010 basic features

3.4 Transaction Data

Dataset has been collected from real market transaction archive data collected for several companies listed in Saudi Stock Exchange nearly 10 years back for a single company with a single classifier module and collect 5 years back data for all companies against multiple classifier model. The system training and evaluation phase is performed effectively with real transaction data. A set of data has been analyzed by three different domain experts and marked

real manipulation scenarios, which is effectively used to train the system and other stock archive data is selected to input to the system as test data.

3.5 Feature selection

The input dataset is represented by a large number of features (more than 20) as below.

Date, Open, High, Low, Close, Volume, Turnover, Trades, VWAP, CIT, CIV, CITR, COT, COV, COTR, CHG, PCHG, PCLS, ANN, SACT, PRV, LTP, MCAP, BBP, BAP

Most of them are not related to predicting the labels. It needs to be selected related set of features through large feature space. Domain expertise knowledge is used to the selected limited number of features from feature space when considering redundant attributes that reduce the efficiency of a selected classifier. It is needed to ensure that important features should contain as labels after the feature selection process. After the basic feature selection process it is needed to identify rectified feature set to proceed with this. The wrapper method's forward selection is used to perform that which follows a greedy approach by evaluating all possible combinations using the Naïve Bayes approach and measuring accuracy, precision, and recall values as well.

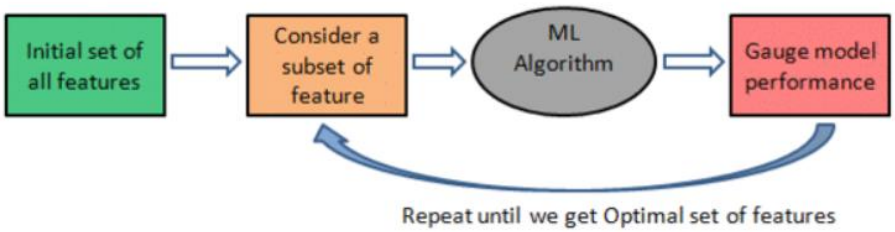


Figure 14: Flowchart: wrapper methods (analytics vidya)

	Close price	Volume	Date	Turnover	%Chg	Best bid	Trades	Accuracy(%)	Precision(%)	Recall(%)
Feature set 1	X	X	X	X	X		X	45	31	23
Feature set 2	X	X	X		X			65.45	38.38	23.55
Feature set 3	X	X	X			X	X	64.33	37.25	25.85
Feature set 4		X	X	X	X	X	X	70.34	89.35	22.38
Feature set 5		X	X	X				66.55	88.22	26.90
Feature set 6		X	X		X	X	X	41.23	33.24	12.35
Feature set 7	X	X		X		X		40.25	34.55	11.53
Feature set 8	X	X		X	X		X	42.44	38.22	12.29
Feature set 9	X	X		X		X		63.76	37.55	22.43
Feature set 10	X		X	X	X	X	X	64.55	38.63	24.36
Feature set 11	X		X	X	X			48.98	66.70	33.54
Feature set 12	X		X	X	X	X	X	66.45	67.26	36.76
Feature set 13		**X**		**X**	**X**		**X**	**99.98**	**100**	**100**

Figure 15: Feature combination results analysis

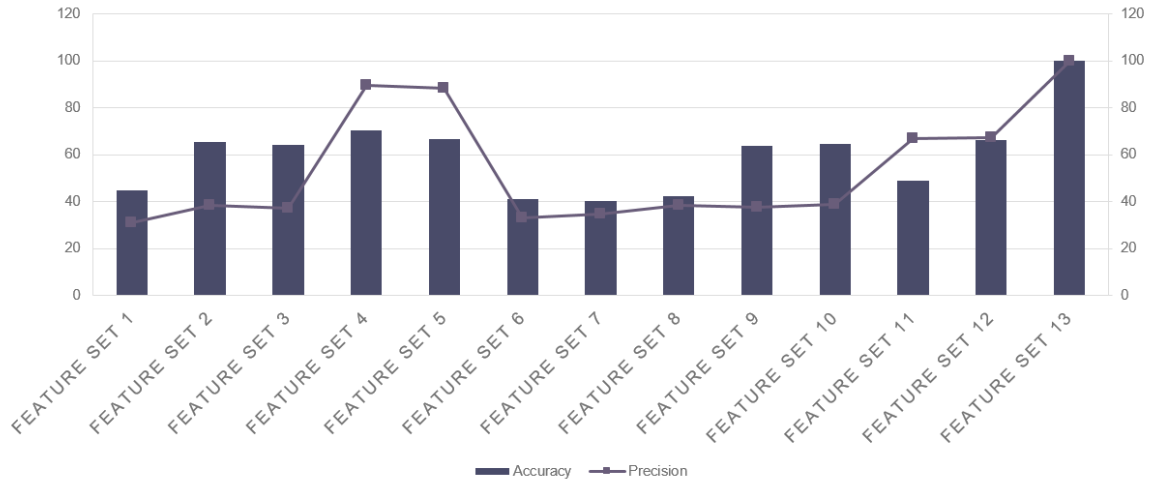


Figure 16: Optimize best combination set identification

Conclusion of the feature selection method, “Price change, Volume, Turnover, and Trades” feature combination was selected as an optimized feature set. Price change denoted percentage change in a selected day which includes OHLC (open, high, low, and close) values, volume is the total number of shares in a selected day, turnover is a measure of liquidity, calculated by dividing the total number of shares traded during the day and trades means the transfer of stock from a seller to a buyer.

3.6 Class Imbalance Problem

The Class imbalance problem occurs when the class labels are relatively different from other class labels. To overcome the imbalance of the problem sampling method was used. This system

should be capable to identify the status of the transaction as “Normal”, “Abnormal: price”, “Abnormal: volume”. The random oversampling technique was used to utilize the research. It balances the data by replicating the minatory class which does not cause loss of the information but the dataset is prone to over fitting.

3.7 Chapter Summary

The conclusion of the Methodology chapter was to cover the selected approach methodology in detail along with design methodology with design assumptions. Scope of the concept research approach with data gathering techniques is also discussed in the chapter. The next chapter discuss design issues and solutions implemented for proposed solution along with system components with their major functionalities.

CHAPTER 4

System Design

4.1 Chapter Overview

The system Design chapter discusses mainly system design techniques. Design problems with solutions for each of them are also included. Each system component's responsibility along with its functionalities also discuss in this chapter.

4.2 Design Problems and Solution Analysis

Difficulties faced when designing the solution for the main problem are listed below along with the solutions for each design problem.

1. It is not possible to have a global threshold value for data streams and individual behaviors for all Symbols. Abnormality cannot be identified for price and volume by a universal value. If a company stock price is 500, then a price change of 10 may not be an abnormal change. But if another company has a stock price of 10, there is a more probability to be an abnormal change if their price changes by 10. The solution for the above design problem is to calculate dynamic threshold values. Those threshold values will be valid for the particular company and particular period only. They will not be reusable even for the same company because price or volume change is not consistent.

2. Since defining boundary values to detect abnormal or normal cases is not a straightforward task, the possibility of growing false-negative and the false-negative error rate is high. Even the experts cannot define exact threshold values directly. One of the main considerations of the main solution is to minimize false-negative error which is considering particular abnormal scenarios as normal. The solution for this design problem is to facilitate change sensitivity in the system by allowing external change of sensitivity variables of the system, which leads to a reduced error rate for the particular data set.

3. There is a problem that even though the system captured certain scenarios as stock manipulation, there are other factors that need to be considered when concluding a

scenario as stock manipulation. Those factors are hard to capture but the scenario depends on them. The solution is to that particular problem is to get expert knowledge on these kinds of scenarios. Otherwise same kind of errors can be reoccurred by the system. Since experts carry investigations on suspected scenarios, feedback can be fed to the system.

4. After the initial learning phase of the system, new suspected scenarios should be able to identify. System anyway keeps on learning with feedback continuously.

4.3 Design methodology

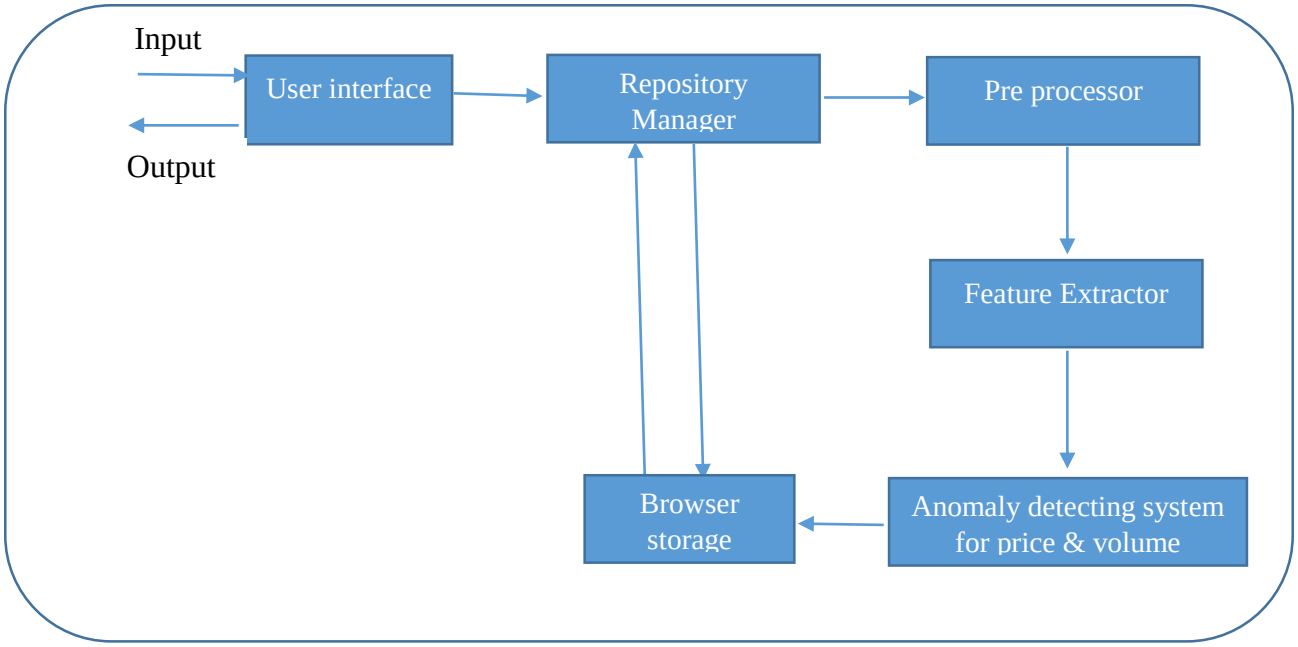


Figure 17: System Components

The system component diagram illustrates the component interaction in the proposed solution. In the beginning, selected companies were picked out. Then, collect historical stock price change, volume, turnover, and trades data from the Saudi stock exchange nearly 10 years back. In preprocessing stage, need to identify invalid transaction-related details to remove noise data in the data set, using domain expert knowledge in this domain can be identified those unnecessary details and success in the preprocessing stage mentioned in the feature selection process. Data feed into the anomaly detection system. The responsibility of each component is discussed below.

4.4 System Components

- User interface
- Repository Manager
- Preprocessor
- Feature Extractor
- Anomaly detecting system for price & volume
- Browser storage

4.4.1 User Interface

This component represents an abstract system to the external user. User inputs are taken to the system and system outputs are given to the user via User Interface. Below is the list of functionalities of Web Interface Component

• Showing Results of system output

• Set Initial parameters to the system

• Get user feedback

This is a critical design decision to use a browser web as a system container because there are more limitations of a web application than a desktop application such as limitations of accessing computer storage etc. But current usages of web applications are growing due to many reasons such as easily accessible, use with many devices which having Internet facility, lightweight, etc. Therefore solution will be closer to end-users but very challenging for the researcher. Challenges and the way they have overcome will be explained under each topic of the rest of the chapters.

4.4.2 Repository Manager

This component is responsible for performing three tasks during the system run.

1. Format input datasets and make them ready to pre-process.

2. Improving training data set which gives better results

3. Executing normalizing process to improve suspicious scenarios

This functionality is based on Immune network theory concepts. This is executed depending on expert feedback given for the system output. This will ensure the quality of the detectors in a way that the feedback from the environment is received and controls the memory accordingly. But the immune network theory component to demoting and killing detectors which are not used recently is not used here because anomalies that can be popped up very rarely should be identified as well.

4.4.3 Browser Storage

Browser Storage is responsible for holding the detector set for the system. Confirmed cases with price/volume fluctuations, normal transactions store in browser storage. Stored detectors are used in identifying newly suspected cases.

This component gives stored cases with a degree of the anomaly as the output. This aligns with the detector set concept of the negative selection algorithm whereas the repository manager optimizes the quality of the dataset.

This also is the main repository to store all transaction details fed to the system. The information included symbol information, price change, and volume information for a certain period, turnover, etc.

4.4.4 Data Preprocessor

This component is responsible for processing input data and preparing them for the feature extraction phase. Corresponding steps of data processing are stated below

1. Eliminate invalid transactions.
2. Eliminate unwanted columns.
3. Data Categorization

4.4.5 Feature Extractor

This component is responsible for calculating the necessary feature values required for the learning and testing phases. Price change and volume anomaly detector is based on machine learning techniques and required features are calculated and data models are prepared according to input models of the classifier

4.4.6 Anomaly Detector for price and volume

This component's functionality is to detect abnormal fluctuations of price and volume for a particular period. Knowledge of the expertise in the domain will be used to label abnormal fluctuations in the dataset. Machine Learning techniques (Naive Bayes Algorithm, Support vector machine, Decision tree, and Random Forest algorithm) are used in identifying abnormality. Since there are no global threshold values and those vary even for one company, the probability of being an anomaly will be received. The methodology is based on supervised learning techniques some of which are introduced below.

	1	2	3	4	5	6	7
1	DATE	VOL	PCHG	ABNORMA	TYPE	NOTR	TOVR
2	20100102	20164	0	0	Normal	96	490381.2
3	20100103	125181	-0.49	0	Normal	90	3041441
4	20100104	149824	-0.25	0	Normal	115	3632195
5	20100105	124600	2.72	0	Normal	147	3078438
6	20100106	121924	0.24	0	Normal	105	3025296
7	20100109	102680	0.96	0	Normal	142	2580051
8	20100110	87833	-0.24	0	Normal	158	2185512
9	20100111	525061	-3.58	0	Normal	92	1.27E+07
10	20100112	232248	-0.99	0	Normal	241	5587105
11	20100113	100571	1	0	Normal	134	2425558
12	20100116	492639	2.72	0	Normal	721	1.21E+07
13	20100117	807828	3.37	0	Normal	776	2.06E+07
14	20100118	318419	0.23	0	Normal	447	8190359
15	20100119	171167	0.7	0	Normal	375	4420913
16	20100120	145292	-0.69	0	Normal	308	3745964
17	20100123	369088	-3.72	0	Normal	226	9225963
18	20100124	476484	-0.97	0	Normal	221	1.18E+07
19	20100125	199000	0.98	0	Normal	206	4909106
20	20100126	173907	-0.97	0	Normal	174	4324381
21	20100127	224392	1.46	0	Normal	133	5525110
22	20100130	195300	-1.44	0	Normal	204	4774956
23	20100131	72127	2.44	1	price	114	1799438
24	20100201	62833	-1.19	0	Normal	136	1579562
25	20100202	28565	1.2	0	Normal	178	718109.6
26	20100203	139445	0.95	0	Normal	173	3519763
27	20100206	213759	0	0	Normal	140	5402384
28	20100207	54331	-0.47	0	Normal	87	1373160
29	20100208	30537	-0.24	0	Normal	78	772310.2
30	20100209	72572	-0.24	0	Normal	387	1819131

Figure 18: sample labeled data

The main objective of this component is to detect anomalies of price and volume values of stock market transactions. The probability value which represents the degree of anomaly will be calculated for price and volume differences.

Several tests have been carried out which give the most promising results for Price and Volume Anomaly Detector.

Supervised machine learning methods are used to identify price and volume abnormalities to obtain an accurate output of the component. The system is tested by various machine learning algorithms such as the Naive Base Algorithm, Support Vector Machine, Random Forest Algorithm, and Decision Tree Algorithm. After testing and evaluating various classifiers, best classifier was selected to evaluate other models when considering training speed, cost to the system and performance. Optimum and minimum dataset was selected as mentioned in selection of best feature combination section.

These algorithms implemented using NodeJS are used to train and test the price and volume anomalies. Features produced from the Feature Extractor component are used as the input, and the output of the component is the probability of being an abnormal transaction.

Below are the functional steps of the system.

1. Initialize Classifier with required parameters
2. Input Training data array consists of features extracted for price and volume
3. Train the classifier with provided data
4. Testing data are fed to the classifier
5. Obtain results for test data

4.4.6.1 Naïve Bayes Classifier

Naive Bayes Classifier implemented using NodeJS is used to train and test the price and volume anomalies. Features produced from Feature Extractor components are used as the input, and output is the component is the probability of being an abnormal transaction.

1. Initialize Classifier with required parameters

2. Input Training data array consists of features extracted for price and volume –Training data headers are Price Change, Volume, turnover, trades and Anomaly Index

3. Train the classifier with provided data

4. Testing data are fed to classifier - Testing data headers are Price Change and Volume

5. Obtain results for test data

Symbol ‘1010’ (Riyadh Bank) which is listed in the Saudi stock exchange has been selected for a single company data module for testing. After pre-processing data which is divided into a 2:1 ratio for training and testing. After features extracting then training data will be input to the classifier and trained. The degree of abnormality of transactions is marked based on domain expertise conclusion. System output will be calculated based on domain expertise knowledge.

4.4.6.2 Support Vector Model Classifier

Support Vector Machines (so-called SVMs) are also generally used as supervised learning techniques and since this solution also uses supervised learning techniques, tested the system and obtained results with an SVM classifier.

Support Vector Machines (so-called SVMs) are also generally used as supervised learning techniques and since this solution also uses supervised learning techniques, tested the system and obtained results with an SVM classifier.

The support vector machine is used for classification by building single or multiple high planes using dimensions high or infinite, which can be used for classification. Extracted features are arranged suitable for SVM classifier inputs and initialize the classifier. After that arranged input data of the particular company is fed to the system and evaluated. One significant difference of input data of SVM compared to Naive Bayes is abnormal indexes of training data provided separately as an array.

Symbol ‘1010’ (Riyadh Bank) is used to train the system for the single company against SVM. Price change, Volume, Turnover, and Trades values features are extracted after preprocessing.

And these column headers were used to train the system then marked system output against domain expertise conclusion.

4.4.6.3 Random Forest Classifier

Price and volume data streams are fed as the input and the degree of abnormality of the input values are provided as the output. The functionality of this component is aligned with the Danger Signal theory concepts. Basic steps of Random Forest classifier.

1. Initiate RF classifier
2. Train classifier with the training dataset
3. Train the model using the training sets as parameters using the Fit function
4. Perform predictions on the test dataset
5. Find accuracy or error using matrices

To get accurate predictions some important considerations need to be addressed.

1. Need to have an actual signal to build models to get maximum accuracy
2. Predictions using individual models need to have minimum correlations between them

4.4.6.4 Decision Tree Classifier

The decision tree classifier is also a supervised learning approach. Price change, volume, turnover and trades, and anomaly type are fed to train the classifier, and data is continuously split according to the anomaly type parameter. There are certainly two types of approaches in design trees classification trees and regression trees. A classification approach will be used in this approach.

Price change, volume, turnover, trades, and anomaly type values are input to the classifier to train the classifier, and predicted anomaly types against system output ratios are calculated.

4.5 Flow of activities

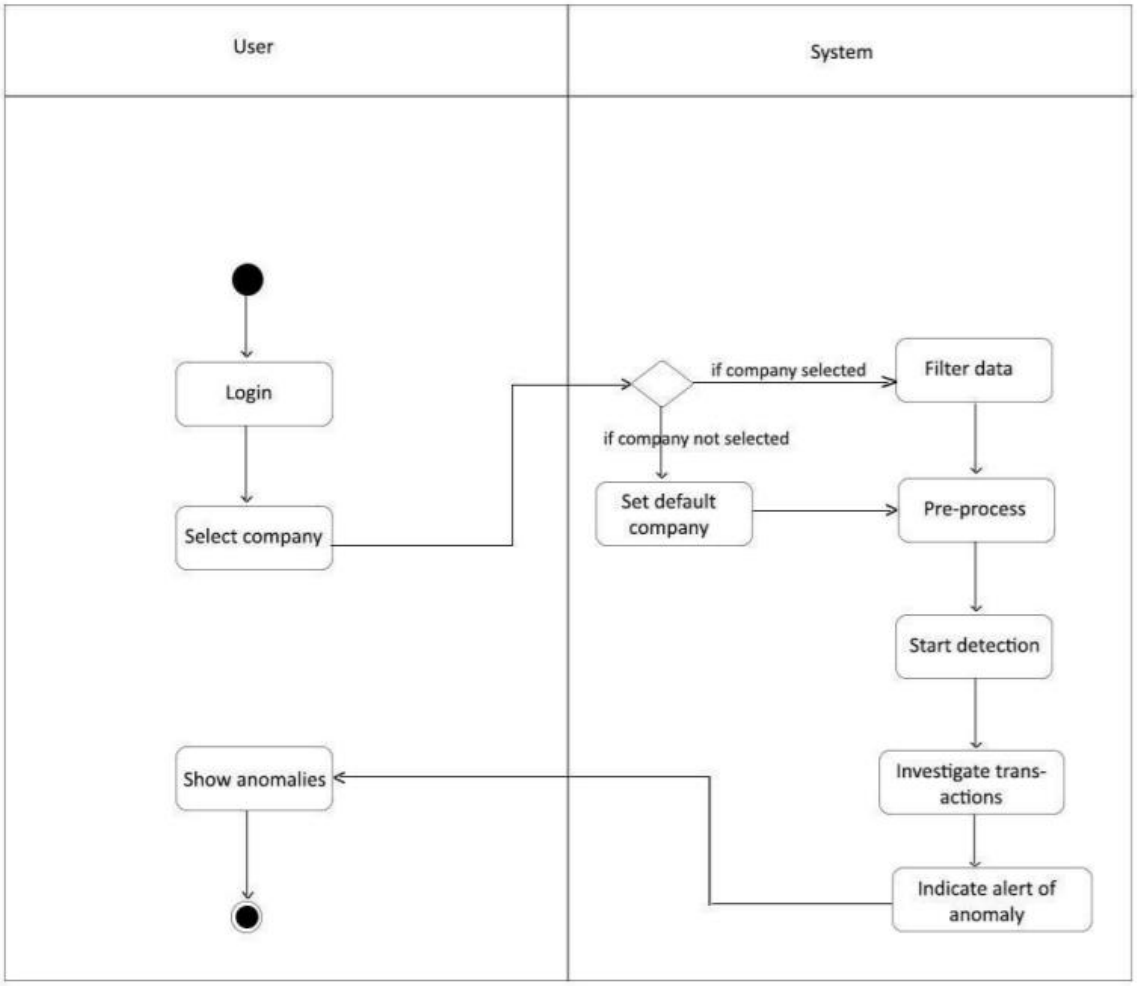


Figure 19: Flow of activities

This diagram indicates the flow of activities that associated with proposed solution

4.6 Chapter Summary

This chapter discussed design problems faced during the designing the solution and proposed solution analysis for each problem and analyze system components for proposed approach in depth along with system workflow. At the end introduced Naïve Bayes, Support Vector Machine, Random Forest Classifier, and Decision Tree supervised learning techniques and how they are arranged when designing the system. The next chapter discuss how technologies, framework were decided and what was the result.

CHAPTER 5

Implementation

5.1 Chapter overview

This Chapter provides a detailed explanation of each module mentioned in the previous chapter and discusses related techniques used in each of the problems mentioned in the previous system design chapter. In the end, selected techniques for this approach will discuss along with an overview of system implementation.

5.2 Technology/ Framework selection

The system is based on web application JavaScript is a more popular web development technology since JavaScript is supported by cross browsers and multiple devices and platforms proposed solution was implemented based on JavaScript.

This system is a web application and JavaScript is one of the most popular programming languages for web development, the author chose JavaScript as the programming language. It also supports JavaScript on various platforms. Many JavaScript frameworks can be used to create applications to reduce the time and effort required for the development. The author also found machine learning libraries that support JavaScript are required and can be extended the functionality required for this system.

Ember is a browser-built application framework made with HTML, CSS, and JavaScript. Instead of using pure JavaScript, it was decided to go with a JavaScript framework when considering high-performance. It is easy to use in a web application as a framework when used according to the module. It has richly formatted templates that update automatically with less code.

5.3 Library selection

JavaScript/Node JS supported machine learning libraries were selected to implement the basic functionalities. The confusion matrix that is based on Node JS was used to calculate accuracy, precision, and recall values.

```
const CM2 = ConfusionMatrix.fromLabels(trueArray, predictArray);

let accuracy = this.utils.formatters.formatNumberPercentage(CM2.getAccuracy() * 100, 2);
```

Figure 20: Confusion matrix implementation

5.4 User interface

This system has a user interface that requires the user to handle the system. Implementation is done using a JavaScript framework called Ember. There were doubts about the implementation of architecture as to whether the use of the web could be restricted the interface, however, it has useful features with the latest web technologies such as HTML5, Ember, and Hyper grid.

Functionalities of user interface

- View suspected scenarios using table hyper grid
- Show price change and volume of input data
- View system output

The user interface module consists of several components as listed below.

1. Data input page: Training and testing data has been input into the system

2. Data preprocessing page: Remove noisy data and divide data into two separate chunks (Training & Testing)

3. The single model with a single company: Show suspected scenarios with results in a selected single company

4. Multiple classifiers with single-company data: show suspected scenarios with results

against multiple classifiers for a single company

5. Multiple classifiers with all company data: show suspected scenarios with results against multiple classifiers for multiple companies.

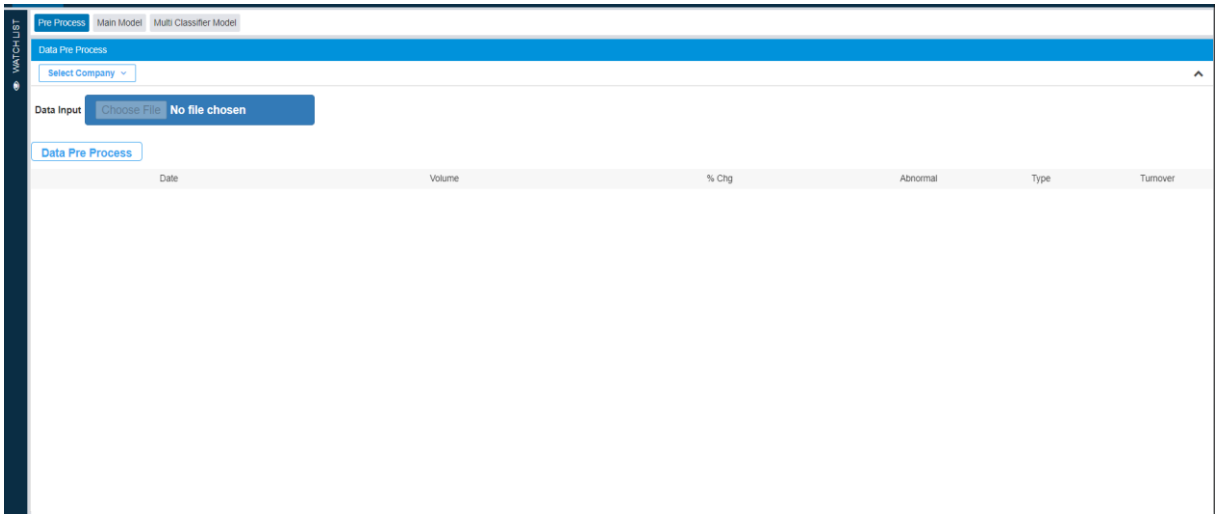


Figure 21: Data import and preprocessing user interface

5.5 Repository Manager

Repository manager is used to handle input data and preprocessed data of some of the components to provide accurate output.

Data is read and the header of the file is recognized to create transaction objects. Transaction objected are created as a map and pushed into a specified array for preprocessing.

```
loaded: function (evt) {
    // Obtain the read file data
    let fileFormat = this.fileName.split('.')[1];
    let noOfColumns;

    if (fileFormat.toLowerCase() === 'csv') {
        var fileString = evt.target.result;

        var count = 0;
        noOfColumns = this.get('selectedCompany').code === '1' ? 8 : 7;

        fileString = fileString.replace(/(\n)+/g, ",");

        if (noOfColumns <= noOfColumns) {
            var fileArray = fileString.split(',');
            var headerArray = fileArray.splice(0, noOfColumns);
            var dataArray = Ember.A();

            while (fileArray.length > noOfColumns - 1 && count < 400000) {
                var rowArray = fileArray.splice(0, noOfColumns);
                var rowObject = {};

                Ember.$.each(headerArray, function (key, header) {
                    rowObject[header] = rowArray[key];
                });

                dataArray.pushObject(rowObject);
                count = count + 1;
            }

            this.set('masterContent', dataArray);
            this.filterCompany();
            this.set('processSuccessText', 'Successfully Pre Processed Data');
            this.set('processCss', 'up-fore-color');
            console.timeEnd("Time consumed for pre-processing");
```

Figure 22: read the dataset and create a transaction object

5.6 Browser Storage

1. Train data - Used in training various models of price change, volume abnormality detectors

2. Test data - Used in testing various models of price change, volume abnormality detectors

After importing the data set to the system, the system will be capable to store datasets in local storage. User should be able to view the last imported data details using this feature until the user update system with the new dataset.

This module will be considered as one of the major repositories which contain multiple features like local storage, session storage, and cache this can be used to store mater data.

```
filterCompany: function () {
    var that = this;
    var filterCompany = this.get('selectedCompany');
    var filteredStocks = this.get('masterContent');

    filteredStocks = this.get('masterContent').filter((function (that) {
        return function (stock) {
            return that.checkFilterMatch(stock, filterCompany);
        };
    })(this));

    this.set('filteredContent', filteredStocks);

    if (!Ember.AllDataMeta) {
        Ember.AllDataMeta = {};
    }

    Ember.AllDataMeta = filteredStocks;

    var saveString = utils.jsonHelper.convertToJson(Ember.AllDataMeta);
    localStorage.setItem('AllDataMeta', saveString);

    filteredStocks = this.removeNoise(filteredStocks);
    filteredStocks = this.checkAnomaly(filteredStocks);

    // have conflicts whether this os showing train or test data
    this.set('content', filteredStocks);
    this.set('dataArray', filteredStocks);

    this.storeFilteredData(filteredStocks, filterCompany);
},
```

Figure 23: Store data imported data in storage

5.7 Data Pre-Processing

This module is responsible for processing input data and preparing data set for the feature extraction phase.

Functional steps for data preprocessing module

1. Remove noisy, redundant data

   The transaction which is considered to be invalid by exchange, due to price mismatch when trading time, quantity mismatch, invalid trading symbol, or bypass buying power is not enough unless the customer is not entitled to bypass buying power or quantity in hands is not sufficient. This kind of transaction becomes invalid this is not affected by the market change. Because of these those transactions need to be identified.

2. Remove unnecessary columns

Need to filter out extracting needed details through preprocessed transaction object. Price change, volume, turnover, and trades will be extracted in this phase.

```
var saveString = utils.jsonHelper.convertToJson(Ember.AllDataMeta);
localStorage.setItem('AllDataMeta', saveString);

filteredStocks = this.removeNoise(filteredStocks);
filteredStocks = this.checkAnomaly(filteredStocks);

// have conflicts whether this os showing train or test data
this.set('content', filteredStocks);
this.set('dataArray', filteredStocks);
```

Figure 24: Remove noisy any unnecessary data

5.8 Abnormality detection

The abnormality detection module main purpose is to detect price and volume abnormality in the given dataset. The probability value represents the degree of abnormality.

There were several feature combinations were carried out to select the best promising result for the price and volume detector.

Supervised machine learning algorithms used to detect abnormalities of price and volume. The system was tested again with various machine learning algorithms.

Functionalities of this module

1. Initialize selected Classifier with required parameters
2. Training data headers are Price Change, Volume, Turnover, and Trades
3. Train selected classifier with input content
4. Testing data headers are Price Change, Volume, Turnover, and trades
5. Obtain results for test data

```
prepareBayesDataSet: function () {
    let trainContent = this.get('trainContent');
    let arrayTraining = [];
    let trainingColumns = ['PCHG', 'VOL', 'abnormal?'];

    Ember.$.each(trainContent, function (key, stock) {
        arrayTraining[arrayTraining.length] = [stock.PCHG, stock.VOL, stock.anomalyIndex];
    });

    let classifier = new bayes.NaiveBayes({
        columns: trainingColumns,
        data: arrayTraining,
        verbose: true
    });

    classifier.train();

    this.testDataBayes(classifier);
},
```

Figure 25: Prepare dataset for Naive Bayes approach

```
testDataBayes: function (classifier) {
    let that = this;
    let testContent = this.get('testContent');

    let totalTestRec = testContent.length;
    let successRec = 0;
    let predictArray = Ember.A([]);
    let trueArray = Ember.A([]);
    let testRec, predict;
    let companyId = localStorage.getItem('ADSCompany');

    Ember.$.each(testContent, function (key, transaction) {
        testRec = [transaction.PCHG, transaction.VOL];
        predict = classifier.predict(testRec);

        transaction.sysAnomality = Number(predict.answer) === 1 ? '1' : '0';
        predictArray.push(Number(predict.answer));
        trueArray.push(transaction.anomalyIndex);
    });

    this.set('predictArray', predictArray);
    this.set('trueArray', trueArray);

    const CM2 = ConfusionMatrix.fromLabels(trueArray, predictArray);
```

Figure 26: Test dataset in Naïve Bayes approach

Trained classifier as input to test the classifier which predicts transaction is normal or abnormal and gives the accuracy of the prediction.

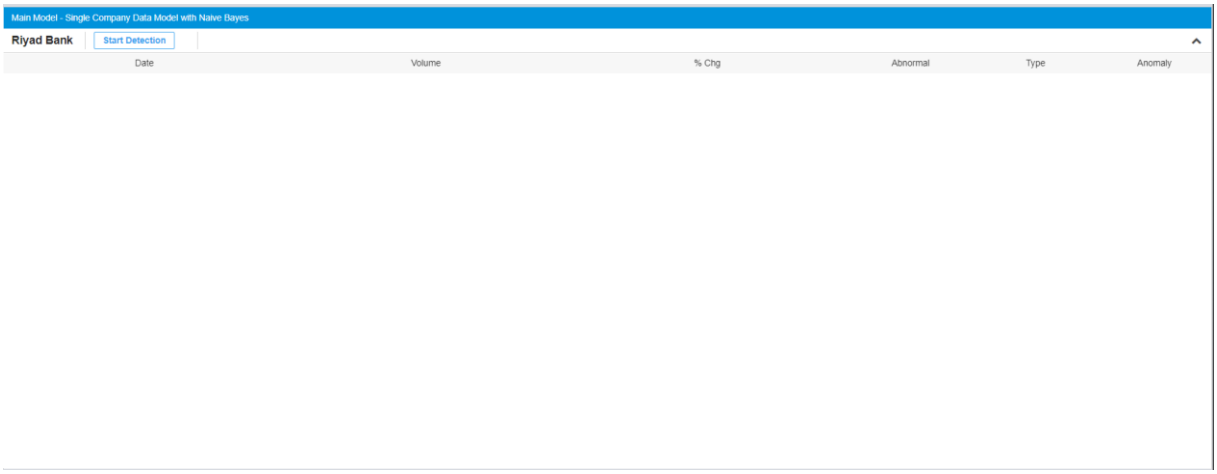


Figure 27: Single classifier against single company detection

5.9 Sidebar details

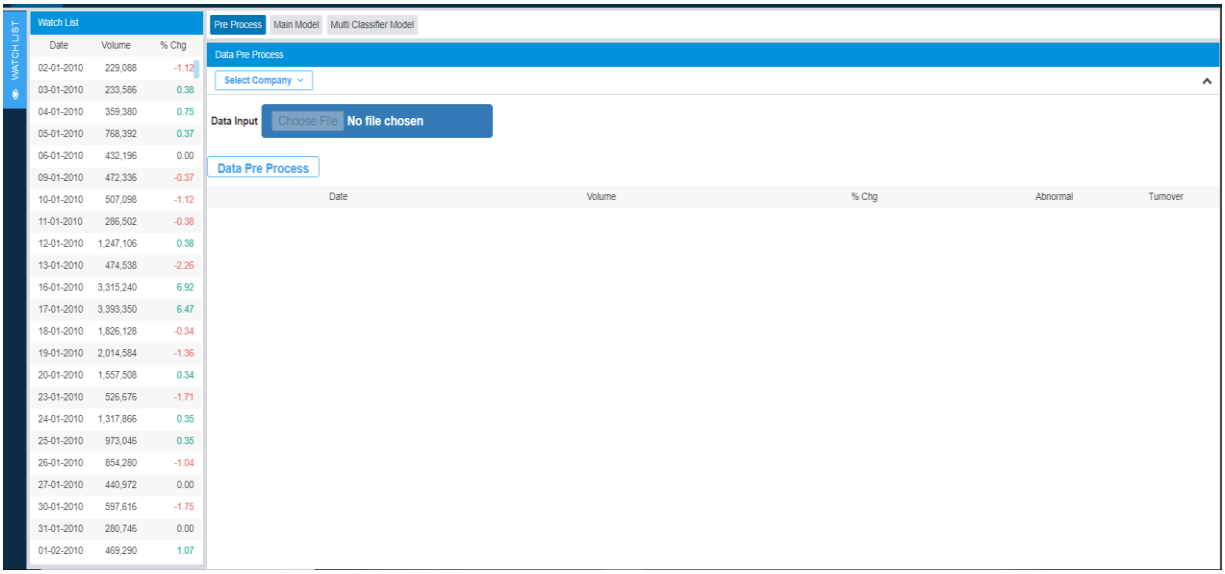


Figure 28: Sidebar details

Sidebar is the view that displays all imported data than can be used to view previous imported data.

5.10 Chapter summary

This chapter covers the technology used to implement the proposed solution along with the framework. System implementation and discussed each module's responsibilities with its functionalities. At the end system interfaces are attached. The next chapter discuss how the evaluation has done and results of the solution proposed in detail.

CHAPTER 6

Evaluation and Result

6.1 Chapter Overview

The evaluation and Result chapter is an overview of the findings and evaluation of the research. This chapter provides an overview of an evaluation methodology and defines evaluation criteria for the selected approach and discusses the selected machine learning approach by including evaluation protocol, selected most suitable feature combination, and results obtained in this research.

6.2 Evaluation methodology

To get the expert opinions on the project, the author will share a document with the selected evaluators describing the system, main features of the system, implementation in brief, and screen-shots of the system following the main flow of the system. Moreover, a demonstration will be done to the evaluators explaining each step of the system using Skype. Then the feedback sent via emails of the evaluators will be collected and analyzed by the author.

6.3 Evaluation criteria

Evaluation criteria consist of the following steps to find anomalies in stock market transactions. Below are the various models and combinations tested in this phase.

- Evaluation of system with single-company data against single classifier
- Evaluation models of different classifiers
- Anomaly detection of Price and Volume data
- Evaluation of system with multiple company data against multiple classifiers

6.4 Transaction Data

Testing has been carried out with real market data collected for several companies listed on Saudi Stock Exchange. The system training and evaluation phase is performed effectively with real transaction data. Source of the dataset of transactions flown through DirectFN (DirectFN Technologies Pvt Ltd.) Order Management Systems. Feed data contains price and volume data

of companies for transactions taking place which requires evaluation which guarantees credibility of the dataset.

A set of data has been analyzed by three different domain experts and marked real manipulation scenarios, which is effectively used to train the system. Other transaction data is used to test the system. Testing has been carried out with data sets of different companies in various models stated earlier.

6.5 Training & Testing models

Training and testing have been carried out in the following models and accuracy, precision, recall are calculated which is required for system evaluation.

System models with different classifiers

In this case, the system is trained and tested for single company data for different classifier models. Below are the classifiers used for evaluation?

1. Naïve Bayes Classifier
2. Support Vector machine
3. Random Forest Classifier
4. Decision Tree

6.5.1 Naïve Bayes Classifier

Extracted features are arranged suitable for Naive Bayes classifier inputs and initialize the classifier. After that arranged input data of the company is fed to the system

The below table denotes the classifier output results of the Naive Bayes Classifier. Please note that results are not normalized with customer data. The degree of the anomaly of a transaction is marked based on conclusions of domain experts and system output is evaluated based on them. All the percentages are calculated for system output to conclusions of domain experts

```
prepareBayesDataSet: function (trainContent, anomalyKey) {
    this.set('isShowTable', false);

    let arrayTraining = [];
    const trainingColumns = ['PCHG', 'VOL', 'TOVR', 'NOTR','abnormal?'];

    Ember.$.each(trainContent, function (key, stock) {
        arrayTraining[arrayTraining.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR, stock[anomalyKey]];
    });

    const classifier = new bayes.NaiveBayes({
        columns: trainingColumns,
        data: arrayTraining,
        verbose: true
    });

    classifier.train();

    return classifier;
},
```

Figure 29: Bayes classifier implementation

Measurement	Value
Accuracy	91.58%
Precision	85.95%
Recall	99.58%

Table 1: Results of Naïve Bayes classifier

6.5.2 Support Vector Machine

Support Vector Machines (so-called SVMs) are also generally used as supervised learning techniques and since this solution learning using supervised learning techniques tested the system and obtained results with SVM classifier

The support vector machine is used for classification by building single or multiple high planes using dimensions high or infinite, which can be used for classification. Extracted features are arranged suitable for SVM classifier inputs and initialize the classifier. After that arranged input data of the company is fed to the system and evaluated. One significant difference of input data of SVM compared to Naive Bayes is, abnormal indexes of training data provided separately as an array

The below table denotes the classifier output results of the SVM Classifier. Please note that results are not normalized with customer data. The degree of the anomaly of a transaction is

marked based on conclusions of domain experts and system output is evaluated based on them. All the percentages are calculated for system output to conclusions of domain experts.

```
prepareSVMDataSet: function () {
    var trainContent = this.get('trainContent');
    var arrayTrainingIn = [];
    var arrayTrainingOut = [];

    Ember.$.each(trainContent, function (key, stock) {
        arrayTrainingIn[arrayTrainingIn.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR];
        arrayTrainingOut[arrayTrainingOut.length] = stock.anomalyIndexSVM;
    });

    var svm = new ml.SVM({
        x: arrayTrainingIn,
        y: arrayTrainingOut
    });

    svm.train({});

    return svm;
},
```

Figure 30: Support vector machine implementation

Measurement	Value
Accuracy	50.00%
Precision	0.00%
Recall	0.00%

Table 2: Results of SVM classifier

The above table shows that the SVM model is not capable of identifying any of the positive anomalies. Therefore Recall and Precision are zero. And also this classifier took a long time to classify compared to the other two classifiers. High accuracy does not make this classifier interested because anyway suspicious cases are not very common as normal cases in stock market transactions, but need to identify those suspicious cases.

6.5.3 Random Forest Classifier

Random forest is a taxonomic algorithm consisting of multiple decision trees. When building each model it uses packaging and feature randomness and creates a non-interconnected model that is more accurate than any single model predicted by the individual model.

To have accurate class predictions these considerations need to be addressed in random forest classification (Elagamy, Stanier and Sharp, 2018).

1. Models built using those features should have some authentic signal in our features so that they work better.

2. Predictions made by every single model need to have low correlations with each other

The below table denotes the classifier output results of the Random Forest Classifier. Please note that results are not normalized with customer data. The degree of an anomaly of a transaction is marked based on conclusions of domain experts and system output is evaluated based on them. All the percentages are calculated for system output to conclusions of domain experts

```
prepareRegDataSet: function (trainContent, anomKey) {
    let arrayTrainingOut = [],
        trainingSet = [];
    let that= this;

    Ember.$.each(trainContent, function (key, stock) {
        trainingSet[trainingSet.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR];
        arrayTrainingOut[arrayTrainingOut.length] = anomKey === 'anomalyTypeIndex' ? stock[anomKey] === 'high risk' ? 4 : stock[anomKey] === 'price' ? 3 : stock[anomKey]
    ));

    const randClassifier = new RandomForestClassifier.RandomForestClassifier();
    randClassifier.train(trainingSet, arrayTrainingOut);

    return randClassifier;
},
```

Figure 31: Random forest implementation

Measurement	Value
Accuracy	99.68%
Precision	99.36%
Recall	100%

Table 3: Random forest classifier results

6.5.4 Decision Tree

Decision Tree can be used for both classification and regression problems, however, it is often preferred for solving classification problems. It is a tree-structured classification that represents

the characteristics of an internal node dataset, the branching decision rules, and each leaf node represents the result.

Below table denotes the classifier output results of Decision Trees Classifier. Please note that results are not normalized with customer data. The degree of an anomaly of a transaction is marked based on conclusions of domain experts and system output is evaluated based on them. All the percentages are calculated for system output with to conclusions of domain experts

```
prepareDecisionTreeDataSet: function () {
    var trainContent = this.get('trainContent');
    var arrayTrainingIn = [];
    var arrayTrainingOut = [];

    Ember.$.each(trainContent, function (key, stock) {
        arrayTrainingIn[arrayTrainingIn.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR];
        arrayTrainingOut[arrayTrainingOut.length] = stock.TYPE;
    });

    var classifier = new ml.DecisionTree({
        data : arrayTrainingIn,
        result : arrayTrainingOut
    });

    classifier.build();

    return classifier;
},
```

Figure 32: Decision Tree implementation

Measurement	Value
Accuracy	99.05%
Precision	85.95%
Recall	99.58%

Table 4: Decision Tress results

6.5.5 Ensemble Approach

Random forest is a taxonomic algorithm consisting of multiple decision trees. When building each model it uses packaging and feature randomness and creates a non-interconnected model that is more accurate than any single model predicted by the individual model. The ensemble

approach is developed by combining multiple models. The transaction is suspected as fraud manipulation transactions in other models combined and used as content to the ensemble input.

Below table denotes other feature combinations tested and their results. The rationale behind this was mentioned in feature selection process.

Close Price	Volume	Date	Turnover	Best Offer	Turnover	%Chg.	Best Bid	Trades	Accuracy	Precision	Recall
X	X	X	X		X	X		X	45%	31%	23%
X	X	X		X		X			65.45%	38.38%	23.55%
X	X	X			X		X	X	64.33%	37.25%	25.85%
	X	X	X	X		X	X	X	70.34%	89.35%	22.38%
	X	X	X	X	X				66.55%	88.22%	26.90%
	X	X		X		X	X	X	41.23%	33.24%	12.35%
X	X		X	X			X		40.25%	34.55%	11.53%
X	X		X	X	X	X		X	42.44%	38.22%	12.29%
X	X		X				X		63.76%	37.55%	22.43%
X		X	X	X	X	X	X	X	64.55%	38.63%	24.36%
X		X	X	X		X			48.98%	66.70%	33.54%
X		X	X			X	X	X	66.45%	67.26%	36.76%
	X				X	X		X	99.98%	100.00%	100.00%

Table 5: feature combinations tested and their results

Some of the possible reasons for getting results of less accuracy have been listed for tested features.

- Close price values themselves cannot be considered as features because they fluctuate with time even in normal cases

- Turnover linearly depends on Price and Volume and useless features to be used along with Price and Volume

Calculate the IQR of % Chg., and calculate the upper and lower zone for outsiders. Filter the % Chg. values that fall outside the upper and lower limits and mark them as external. And volume upper bound is calculated as the average value of dataset volume. The results of the classifier prediction are stated below. The capability of identifying anomaly scenarios is optimum in this case. The degree of the anomaly of a transaction is marked based on conclusions of domain experts and system output is evaluated based on them. All the percentages are calculated for system output to conclusions of domain experts.

Measurement	Value
Accuracy	91.58%
Precision	85.95%
Recall	99.58%

Table 6: Result with %Chg. and volume difference

After analyzing these results of different feature sets, IQR of %Chg. and volume are considered as features for other models discussed in this chapter

6.7 Single Company Data Model

In this model, the system is trained and tested for single company data only.

Company symbol ‘1010’ which is listed on Saudi Stock Exchange has been selected for testing. After cleaning and pre-processing data, 2583 valid transaction records were there in the data model, which is divided into a 2: 1 ratio for training and testing data.

	A	B	C	D	E	F
	DATE	VOL	PCHG	ABNORMA	TYPE	TOVR
	20100102	229088	-1.12	1	volume	3049042
	20100103	233586	0.38	0	Normal	3096840
	20100104	359380	0.75	0	Normal	4789781
	20100105	768392	0.37	0	Normal	1.04E+07
	20100106	432196	0	0	Normal	5827307
	20100109	472336	-0.37	0	Normal	6374803
	20100110	507098	-1.12	1	volume	6775180
	20100111	286502	-0.38	0	Normal	3819986
	20100112	1247106	0.38	0	Normal	1.65E+07
	20100113	474538	-2.26	1	volume	6259433
	20100116	3315240	6.92	1	high risk	4.58E+07
	20100117	3393350	6.47	1	high risk	4.83E+07
	20100118	1826128	-0.34	0	Normal	2.73E+07
	20100119	2014584	-1.36	1	volume	2.94E+07
	20100120	1557508	0.34	0	Normal	2.25E+07
	20100123	526676	-1.71	1	volume	7554824
	20100124	1317866	0.35	0	Normal	1.91E+07
	20100125	973046	0.35	0	Normal	1.41E+07
	20100126	854280	-1.04	1	volume	1.23E+07
	20100127	440972	0	0	Normal	6282226
	20100130	597616	-1.75	1	volume	8426639
	20100131	280746	0	0	Normal	3961483
	20100201	469290	1.07	1	volume	6657362
	20100202	558540	-0.7	0	Normal	7937389
	20100203	394666	0	0	Normal	5585957
	20100206	393436	-1.06	1	volume	5478737
	20100207	192208	-0.72	0	Normal	2663386
	20100208	256894	0.36	0	Normal	3571936
	20100209	1539014	0	0	Normal	2.10E+07

Figure 33: Sample dataset of '1010' company

6.8 Parallel Classifier Model

In this model, the system is trained and tested for single company data. Each classifier is trained with corresponding company data and multiple classifiers are activated and test company data parallel.

After the classification by price, volume detector, accuracy, precision, and recall have been calculated. All the percentages are calculated for system output to conclusions of domain experts.

Measurement	Value
Accuracy	91.58%
Precision	85.95%
Recall	99.58%

Table 7: Single company data using naïve Bayes

6.9 Multiple company data model with multiple classifiers

In this model, the system is trained and tested for multiple company data. Each classifier is trained with corresponding company data and multiple classifiers are activated and test company data parallel.

One classifier testing and evaluation steps are the same as those described in the previous subsection. Apart from company symbol ’1010’ (company Name is ’Riyadh Bank), company symbol ’1050’ (company Name is Banque Saudi Fransi） has been selected for testing for the single model with a single classifier.

After the classification by price, volume detector, accuracy, precision, and recall have been calculated. All the percentages are calculated for system output to conclusions of domain expert

6.10 All Company Data Model

In this model, the system is trained and tested for all companies with a single classifier. This scenario is more towards practical situations as the solution is expected to detect transaction anomalies. Training and testing data are obtained for all companies and provided as inputs to the system.

After the classification by price, volume detector accuracy, precision and recall have been calculated. The degree of the anomaly of a transaction is marked based on the conclusions of domain experts.

Some of the possible reasons for getting results of less accuracy have been listed for tested features.

- Price and Volume value averages are not much meaningful when many companies contribute to the calculation
- Training data abnormalities are defined initially, are not related to the other companies traded in the same period
- Abnormality of a transaction changes from company to company, as one transaction is abnormal for a particular company but not for another company

	A	B	C	D	E	F	G	H	I
234	20171206	1010	237644	0	0	Normal	93	2835151	
235	20171207	1010	473588	0.42	0	Normal	100	5663772	
236	20171210	1010	890683	1.08	1	price	228	1.08E+07	
237	20171211	1010	755106	-0.41	0	Normal	109	9135902	
238	20171212	1010	953659	0.75	0	Normal	169	1.16E+07	
239	20171213	1010	419675	0	0	Normal	104	5097451	
240	20171214	1010	556720	0.49	0	Normal	150	6797895	
241	20171217	1010	862900	0.08	0	Normal	173	1.05E+07	
242	20171218	1010	961842	2.21	1	price	237	1.19E+07	
243	20171219	1010	715407	1.04	1	price	199	9023743	
244	20171220	1010	285595	0.4	0	Normal	106	3620380	
245	20171221	1010	382835	0.16	0	Normal	85	4859477	
246	20171224	1010	239742	0	0	Normal	74	3034655	
247	20171225	1010	320453	-0.31	0	Normal	80	4063209	
248	20171226	1010	403986	-0.47	0	Normal	99	5090939	
249	20171227	1010	438420	0	0	Normal	77	5532296	
250	20171228	1010	657808	0.08	0	Normal	182	8289010	
251	20171231	1010	991705	-0.87	0	Normal	211	1.25E+07	
252	20170101	1030	21509	-0.07	0	Normal	30	306279	
253	20170102	1030	70497	0	0	Normal	37	1008575	
254	20170103	1030	90048	0.99	1	price	62	1282608	
255	20170104	1030	140375	-0.65	0	Normal	109	2004079	
256	20170105	1030	33949	0.13	0	Normal	29	483637.7	
257	20170108	1030	63819	-1.83	0	Normal	64	896102.8	
553	20170313	1050	488656	-1.27	0	Normal	427	1.14E+07	
554	20170314	1050	415493	-1.93	0	Normal	601	9528834	
555	20170315	1050	424824	2.8	1	price	248	9860441	
556	20170316	1050	290968	3.24	1	price	203	7019914	
557	20170319	1050	82369	1.65	1	price	81	2014870	
558	20170320	1050	433313	0.2	0	Normal	400	1.07E+07	

Figure 34: all company sample dataset

6.11 Comparison with DirectFN Rule-Based Anomaly detector

DirectFN is one of the largest ICT companies in Sri Lanka and the Middle East, which offers stock market-related solutions to trade and view price data, and also provides a rule-based anomaly detection system called AML

In AML, we can filter suspected transactions for a particular period. For the same period and for a particular company (we have chosen company ’2250’), we have collected predictions from DirectFN AML, predictions from our solution, and predictions of a domain expert who analyzed transactions of that particular period for company ’2250’. Evaluation is executed with the assumption of domain experts’ predictions and results of the other two predictions are analyzed compared to it.

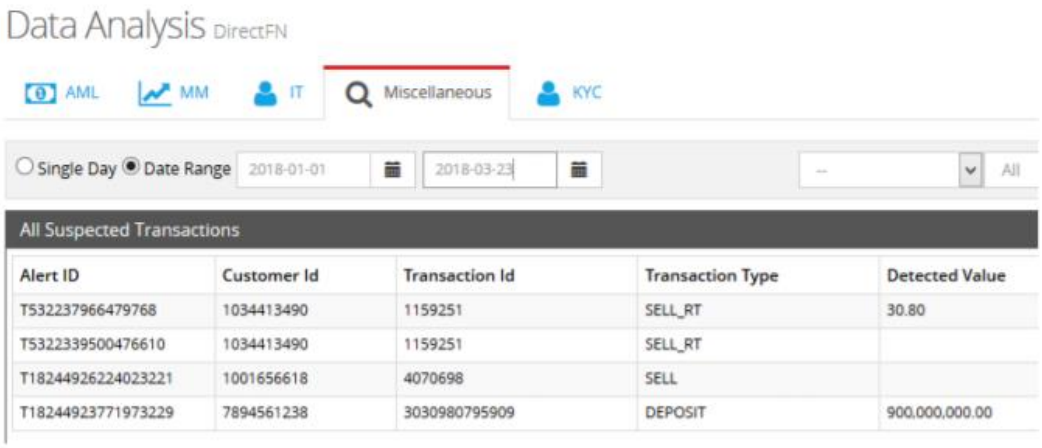


Figure 35: DirectFN AML

Initial Detail	Value
Total No of records	304
Expert’s positive count	21

Table 8: Initial Data parameters

The below tables are showing predictions of this system and DirectFN AML, and accuracy, precision, recall, and F1 score has been calculated compared to domain expert predictions.

The below table shows obtained results from DirectFN AML.

Measurement	Value
Predicted True positive count	5
Accuracy	91.12%

Recall	23.81%
Precision	31.25%
F1 Score	27.02

Table 9: Results of DirectFN AML

The below table shows obtained results from the ‘Abusing pattern detection system for stock market’ solution for the same input data.

Measurement	Value
Predicted True positive count	96.00%
Accuracy	96.00%
Recall	32.00%

Table 10: Results of abusing pattern detection system for stock market

Some of the possible reasons for getting erroneous results for DirectFN AML have been listed

1. The accuracy of AML is fair but relatively low, but that measurement is not that valid for systems that have a low no of positive scenarios.
2. Since rules are common for all companies, higher margins for parameters have applied for AML. Therefore true positive recognition ability is low and it is indicated with low precision.
3. Some of the highly traded or rarely traded company transactions are captured as anomalies even if they are normal for the company. That leads to a decreased recall value and also an F1 score.

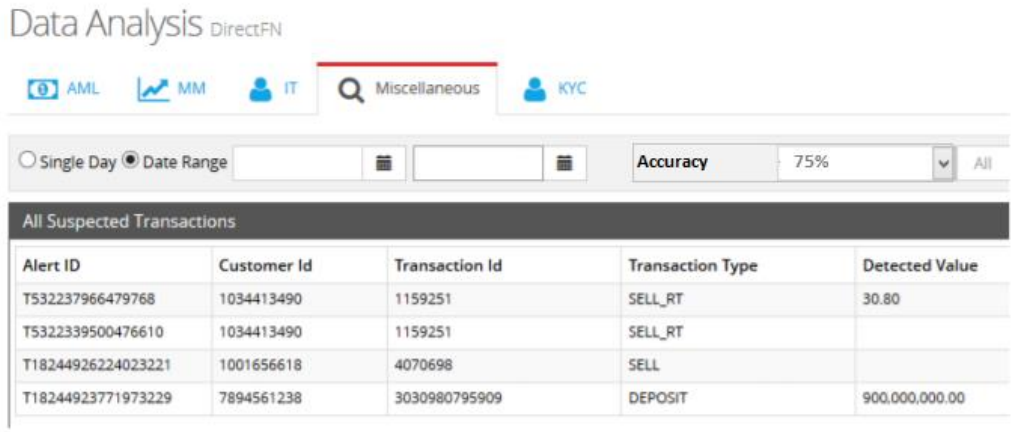


Figure 36: AML system's accuracy for selected transactions

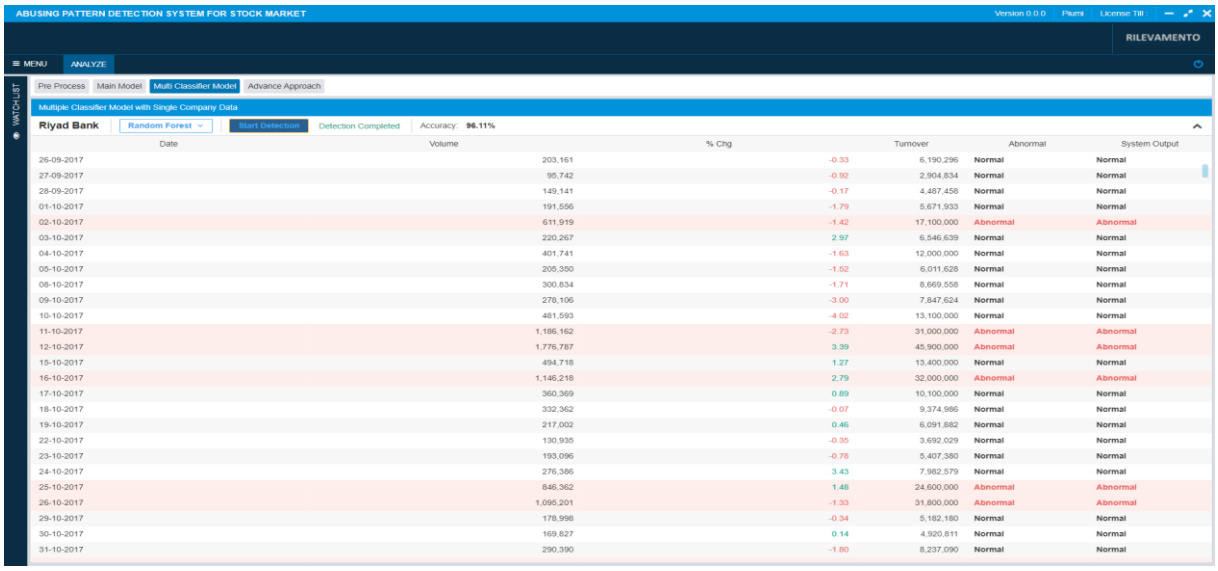


Figure 37: System accuracy for same transactions

6.12 Chapter Summary

This chapter discussed evaluation methodology, evaluation criteria, and machine learning approach along with their results. At the last it discussed evaluation of the system module in detail and comparison results with DirectFN AML system. The next chapter discuss limitations of the system, conclusion of solution proposed and future direction of this system.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Chapter Overview

This chapter summarizes the implementation of its findings and its contributions and discusses limitations of the current work and achievement of project objectives described in chapter 1. Future work discusses the outline of the research for the direction of the future.

7.2 Summary of the Achievement

The main purpose of this solution was to implement a research methodology to identify these evolving patterns of abuse.

The first objective is to detect abnormalities in the stock market. Targeted companies were identified and studied about selected companies which are listed in Saudi stock exchange in order to select best company. Find optimum feature set was next milestone in this solution, achieved that conclusion of the domain expertise and using supervised learning wrapper method. Statistical calculations are used initially in existing solutions that has higher fault tolerance and time consuming for predicting abnormalities. Therefore, next objective was to select supervised learning methods. After testing and evaluating various classifiers, the best classifier was selected to testing other models.

Then feature set is also optimized testing various sets of features and obtaining results and analyzing what are the reasons for those results in a selected individual company. The optimum and minimum feature set was price change compared with the interquartile price range, volume compared to average volume, turnover value compared to average turnover, and trades value compared to average trades.

The second sub-objective is achieved by analyzing all companies listed in the Saudi stock exchange and adding its biased value to results of price, volume detector, and taking the final output of the system. Next milestone was to test this model using real time markets when have highly dynamic behavior since existing solutions are not capable to detect abnormalities in

volatile markets with minimum fault. Finally System prediction output and DirectFN AML solution output were compared against domain expertise knowledge.

7.3 Limitations of the system

- The system only detects illegal anomalies
- The system is only capable to identify illegal transaction only and it is not be able to find suspected user
- Limited accessing computer storage when designing a web application
- The system marks anomaly using domain expertise knowledge only trained dataset is not included real manipulations.

7.4 Critical Appraisal of Objective

The objective of the research is successfully achieved by implementing a system abusing pattern detection system for the stock market by analyzing price change, volume, turnover, and trades using archive dataset with machine learning techniques within the given time duration. Evaluation is based conclusion of domain expertise.

Objective	Status
To conduct a Literature survey to study the anomalies focusing on data	Completed
To identify and analyze the gap of existing solutions in detecting stock market anomalies	Completed
To identify a data set that can be used in the context of the problem and means to perform data preprocessing such that it would be usable by the algorithms	Completed
To examine machine learning approaches for the solution	Completed
To understand machine learning approaches to a single model and Ensemble model approach.	Completed
Implementation of the model which is capable of finding abnormalities of a data and calculating the degree of abnormality of a suspected fluctuation	Completed
To implement the prototype fulfilling the identified gap using supervised learning approaches	Completed

To evaluate the system in terms of the quality of the content	Completed

Table 11: Status of the research objectives

7.5 Future Work

In this solution, the initial identification of frauds with Price and Volume is moved from statistical techniques to supervised learning techniques. This solution can be further optimized by adding domain-specific scenario effects to transactions more and more. One such example is Stock Splitting. When the stock value becomes to a saturated level, the price is set to half of the initial price and make no of stocks twice. We have to neglect those price, volume fluctuations as they are not anomaly behavior.

Only suspected behavior identification is performed in this solution and has to proceed with a manual check for each suspected scenario produced. That manual check should be done without letting the customer know because if it is not a fraud, the customer will be disappointed in suspecting him.

This solution is not tested with a time series model with time series theories. Taking IQR of %Chg and the average value of volume value of particular scenario can be further fine-tuned to a small period maybe lead to better results. And also new feature sets can be tested obtained from time-series data.

Domain expert knowledge is used to pre identification of suspicious scenarios and it will be better if the user can train the system with real confirmed scenarios, then the system will be more towards identifying real transaction anomalies. Real frauds are rare compared to the normal transaction and rare anomalies may have to be subject to better classification techniques apart from techniques used in this solution. Since dealers are focusing news and announcement data more and more when trading it should be more advantage the system can be able to integrate real time news and announcement data to ensure abnormalities in the market.

REFERENCES

Bhuriya, D. *et al.* (2017) 'Stock market predication using a linear regression', *Proceedings of the International Conference on Electronics, Communication and Aerospace Technology, ICECA 2017*, 2017-Janua, pp. 510–513. doi: 10.1109/ICECA.2017.8212716.

Coello, C. A. C. (2005) 'Solving Multiobjective Optimization Problems Using an Artificial Immune System', pp. 163–190.

Elagamy, M. N., Stanier, C. and Sharp, B. (2018) 'Stock market random forest-text mining system mining critical indicators of stock market movements', *2nd International Conference on Natural Language and Speech Processing, ICNLSP 2018*, pp. 1–8. doi: 10.1109/ICNLSP.2018.8374370.

Ferdousi, Z. and Maeda, A. (2006) 'Anomaly detection using unsupervised profiling method in time series data', *CEUR Workshop Proceedings*, 215.

Golmohammadi, K., Zaiane, O. R. and Diaz, D. (2014) 'Detecting stock market manipulation using supervised learning algorithms', *DSAA 2014 - Proceedings of the 2014 IEEE International Conference on Data Science and Advanced Analytics*, pp. 435–441. doi: 10.1109/DSAA.2014.7058109.

Grzymala-busse, J. W. (2005) 'Chapter 13 RULE INDUCTION', in *Data Mining and Knowledge Discovery Handbook*, pp. XXXVI, 1383.

Haft, R. J. and Haft, R. J. (2014) 'The Effect of Insider Trading Rules on the Internal Efficiency of the Large Corporation THE EFFECT OF INSIDER TRADING RULES ON THE INTERNAL EFFICIENCY OF THE LARGE CORPORATIONt Professor Kenneth Scott recently summarized the three primary justifications f', 80(5), pp. 1051–1071.

Hoseinzade, E. and Haratizadeh, S. (2019) 'CNNpred: CNN-based stock market

prediction using a diverse set of variables', *Expert Systems with Applications*, 129, pp. 273–285. doi: 10.1016/j.eswa.2019.03.029.

Jungwon, K., Ong, A. and Overill, R. E. (2003) 'Design of an artificial immune system as a novel anomaly detector for combating financial fraud in the retail sector', *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings*, 1, pp. 405–412. doi: 10.1109/CEC.2003.1299604.

Leangarun, T., Tangamchit, P. and Thajchayapong, S. (2019) 'Stock Price Manipulation Detection using Generative Adversarial Networks', *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, (1), pp. 2104–2111. doi: 10.1109/SSCI.2018.8628777.

Poterba, J. M. *et al.* (2015) 'Stock Ownership Patterns , Stock Market Fluctuations , and Consumption', 1995(2), pp. 295–372.

Salvador, S. and Chan, P. (2005) 'Learning States and Rules for Detecting Anomalies in Time Series', pp. 241–255.

Scholar, M. (2017) 'An AIS Based Anomaly Detection System', (Iccmc), pp. 708–711.

Statman, M. (2008) 'What Is Behavioral Finance?', *Handbook of Finance*, pp. 1–9. doi: 10.1002/9780470404324.hof002009.

Yegnanarayana, B. (1994) 'Artificial neural networks for pattern recognition', *Sadhana*, 19(2), pp. 189–238. doi: 10.1007/BF02811896.

APPENDICES

Appendix A: Immune System

Immune system techniques used in this project are described in detail in this section.

A.1 Natural Immune System

Many systems are capable of executing amazing functionalities to keep the human body in a normal state. They are complex in structure but use simple techniques to fulfill tasks. The immune system is also very important to humans since it is responsible for protecting the human body from viruses and bacteria, etc.

The immune system uses negative selection techniques to detect abnormal patterns. After detection, the immune response is mounted on foreign invaders. This process is called clonal selection. The immune system maintains a valuable detector repository by eliminating redundant detectors. Apart from this, dangerous alerts are identified using the Danger theory.

A.2 Negative Selection

The main purpose of this process is to produce detectors that can detect foreign invaders. It is based on the main technology of the Times, which produces a set of mature T cells that can bind only those that are not autoimmune

The first step in this algorithm is to generate a set of auto-thread (P) that is considered normal to the system. The main purpose is to generate a set of detectors (M) that detect only opposite threads in the S that can be applied to new data to classify these detectors as auto or non-auto.

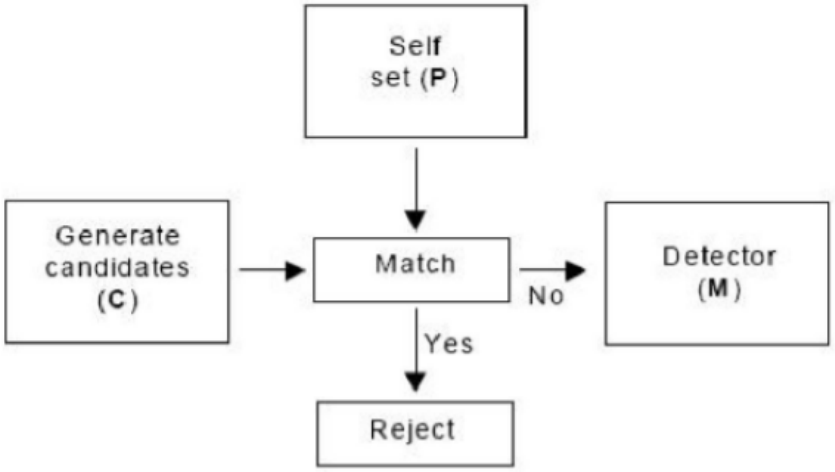


Figure 38: Negative Selection

As described in the above figure, the immune system generates possible detectors randomly and then detectors are sent through a mutation process. Generated candidates (detectors) are matched with sample self-cells and destroyed if matched. Likewise, detectors are generated which might be matched with non-self-cells.

A.3 Clonal Selection

In the Clonal selection process, when a detector identifies an antigen, it is subjected to proliferate process which is diversified detectors generated which are more capable of capturing the same antigen next time. Detectors that are closely related to antigen will eliminate it and finally, it will be stored.

Antibodies enter the body and attack the body, activating the immune cells (B cells) and producing and responding to specific antibodies to attack the antibodies. Antibodies are molecules that bind to B cells that target the detection and capture of antibodies. A process called proliferation is carried out, in which the attacking antibodies are successfully detected and the cells produce new two types of cells.

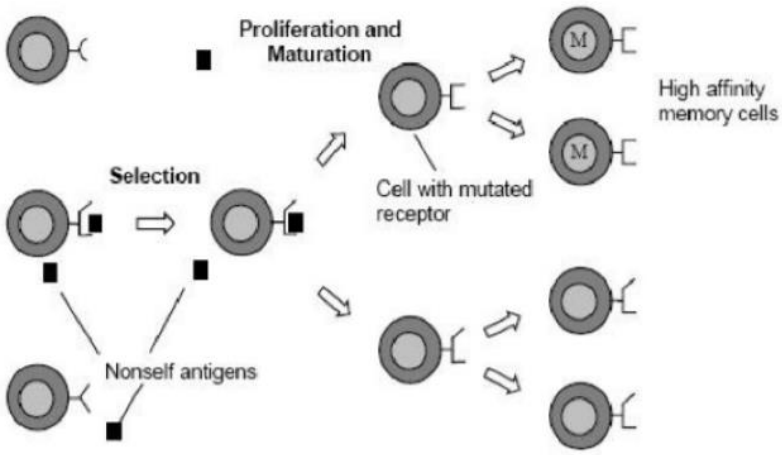


Figure 39: Clonal Selection

1. Attacking Cells

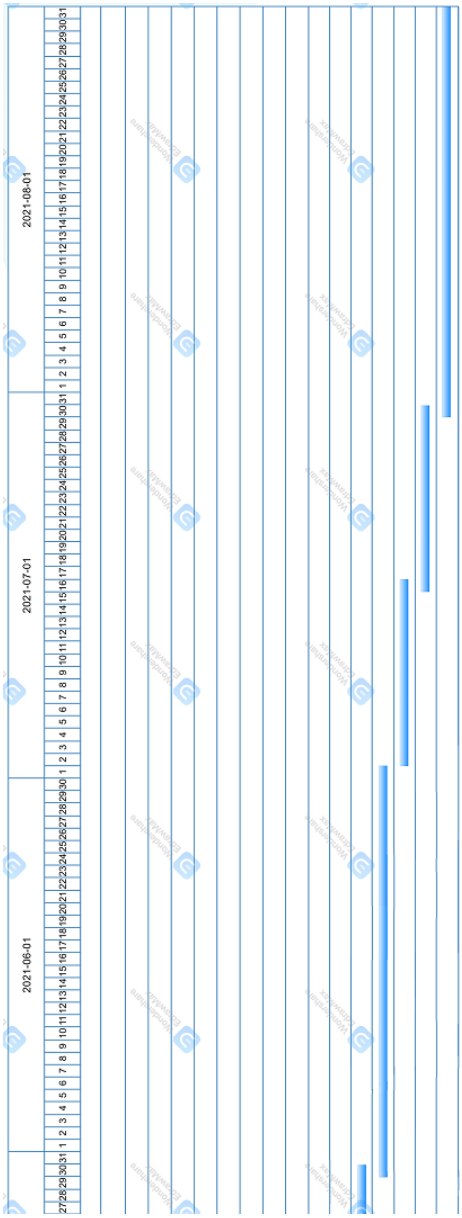
2. Memory Cells

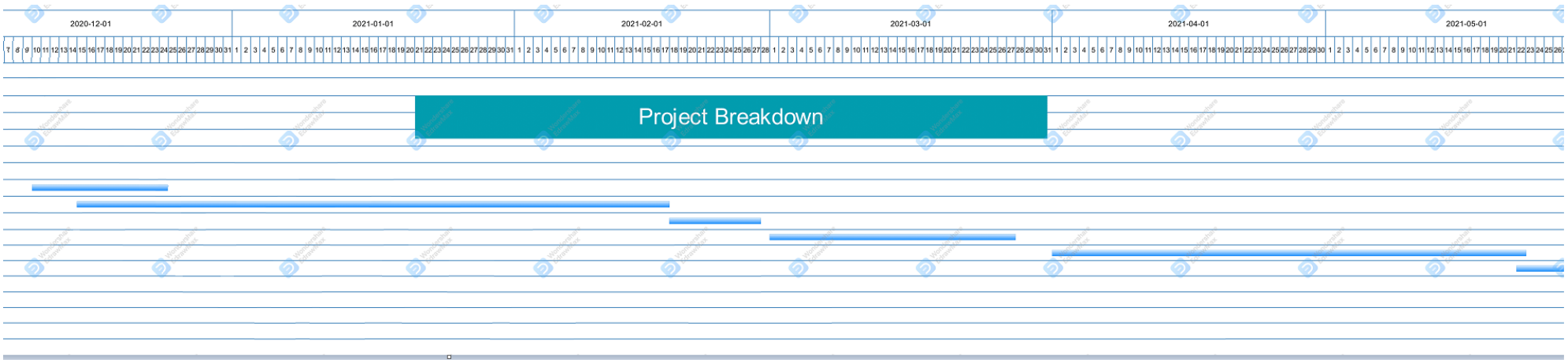
Attacking cells act as an effective antibody to immediately defeat the attacking antibodies. Memory cells have a long lifespan and their function is to attack the same or similar antibody exposure in the future.

A.4 Danger Theory

This is introduced to overcome some drawbacks of the negative selection process. Danger theory is well ahead of negative selection in scalability, fault rate, and evolution of detectors. In this process, a danger signal is sent as a confirmation of detecting dangerous cells, therefore all the non-self-cells will not be considered as foreign invaders.

Appendix B: Gantt chart




2020-12-01
2021-01-01
2021-02-01
2021-03-01
2021-04-01
2021-05-01
Project Breakdown

ID
Task Name
2020-07-08
2020-08-01
2020-09-01
2020-10-01
2020-11-01
1 Research on project background
2 Research on project domain
3 Research on existing systems
4 Specify project scope
5 Identify resource requirments
6 Prepare Project proposal draft
7 Prepare project proposal
8 Prepare revised proposal
9 Collect Dataset and prepare
10 Introduction draft submission
11 Interim report draft submission
12 Interim report submission
13 Design Prototype
14 Implementation of prototype
15 Prototype testing
16 Prototype evaluation
17 Thesis submission and evaluation

Appendix C: Sample Dataset

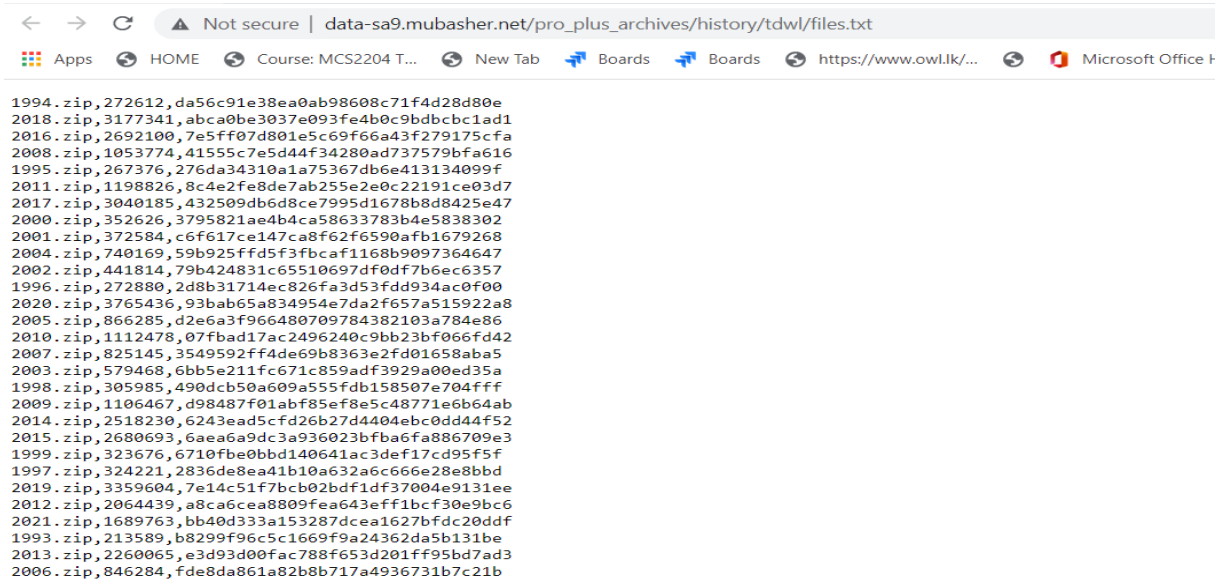


Figure 40: archive dataset format

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
INS	DATE	OP	HIG	LOW	CLS	VOL	TOVR	NOTR	VWAP	CIT	CIV	CITR	COT	COV	COTR	CHG	PCHG	PCLS	ANN	SACT	PRV	LTP
0	20100102	13.45	13.45	13.25	13.3	229088	3049042	96	13.31	0	0	0	0	0	0	-0.15	-1.12	13.45	0	null	null	26.6
0	20100103	13.3	13.35	13.15	13.35	233586	3096840	90	13.26	0	0	0	0	0	0	0.05	0.38	13.3	0	null	null	26.7
0	20100104	13.25	13.45	13.15	13.45	359380	4789781	115	13.33	0	0	0	0	0	0	0.1	0.75	13.35	0	null	null	26.9
0	20100105	13.45	13.55	13.4	13.5	768392	1.04E+07	147	13.5	0	0	0	0	0	0	0.05	0.37	13.45	0	null	null	27
0	20100106	13.45	13.5	13.4	13.5	432196	5827307	105	13.49	0	0	0	0	0	0	0	0	13.5	0	null	null	27
0	20100109	13.5	13.55	13.4	13.45	472336	6374803	142	13.5	0	0	0	0	0	0	-0.05	-0.37	13.5	0	null	null	26.9
0	20100110	13.5	13.5	13.15	13.3	507098	6775180	158	13.36	0	0	0	0	0	0	-0.15	-1.12	13.45	0	null	null	26.6
0	20100111	13.3	13.35	13.25	13.25	286502	3819986	92	13.34	0	0	0	0	0	0	-0.05	-0.38	13.3	0	null	null	26.5
0	20100112	13.3	13.5	13.05	13.3	1247106	1.65E+07	241	13.21	0	0	0	0	0	0	0.05	0.38	13.25	0	null	null	26.5
0	20100113	13.3	13.3	13	13	474538	6259433	134	13.19	0	0	0	0	0	0	-0.3	-2.26	13.3	0	null	null	26
0	20100116	13.5	14	13.5	13.9	3315240	4.58E+07	721	13.81	0	0	0	0	0	0	0.9	6.92	13	0	null	null	27.8
0	20100117	14	14.95	13.85	14.8	3393350	4.83E+07	776	14.23	0	0	0	0	0	0	0.9	6.47	13.9	0	null	null	29.6
0	20100118	14.95	15.15	14.75	14.75	1826128	2.73E+07	447	14.95	0	0	0	0	0	0	-0.05	-0.34	14.8	0	null	null	29.6
0	20100119	14.75	14.75	14.45	14.55	2014584	2.94E+07	375	14.61	0	0	0	0	0	0	-0.2	-1.36	14.75	0	null	null	29.1
0	20100120	14.55	14.6	14.3	14.6	1557508	2.25E+07	308	14.47	0	0	0	0	0	0	0.05	0.34	14.55	0	null	null	29.2
0	20100123	14.35	14.45	14.25	14.35	526676	7554824	226	14.35	0	0	0	0	0	0	-0.25	-1.71	14.6	0	null	null	28.7
0	20100124	14.3	14.55	14.3	14.4	1317866	1.91E+07	221	14.46	0	0	0	0	0	0	0.05	0.35	14.35	0	null	null	28.9
0	20100125	14.4	14.65	14.4	14.45	973046	1.41E+07	206	14.48	0	0	0	0	0	0	0.05	0.35	14.4	0	null	null	29
0	20100126	14.45	14.5	14.25	14.3	854280	1.23E+07	174	14.38	0	0	0	0	0	0	-0.15	-1.04	14.45	0	null	null	28.6
0	20100127	14.3	14.3	14.2	14.3	440972	6282226	133	14.25	0	0	0	0	0	0	0	0	14.3	0	null	null	28.6
0	20100130	14.3	14.3	14	14.05	597616	8426639	204	14.1	0	0	0	0	0	0	-0.25	-1.75	14.3	0	null	null	28.1
0	20100131	14.1	14.15	14.05	14.05	280746	3961483	114	14.11	0	0	0	0	0	0	0	0	14.05	0	null	null	28.1
0	20100201	14.05	14.25	14	14.2	469290	6657362	136	14.19	0	0	0	0	0	0	0.15	1.07	14.05	0	null	null	28.4
0	20100202	14.3	14.3	14.1	14.1	558540	7937389	178	14.21	0	0	0	0	0	0	-0.1	-0.7	14.2	0	null	null	28.3
0	20100203	14.1	14.2	14.1	14.1	394666	5585957	173	14.16	0	0	0	0	0	0	0	0	14.1	0	null	null	28.2
0	20100206	13.9	14	13.8	13.95	393436	5478737	140	13.93	0	0	0	0	0	0	-0.15	-1.06	14.1	0	null	null	27.9
0	20100207	13.85	13.95	13.8	13.85	192208	2663386	87	13.86	0	0	0	0	0	0	-0.1	-0.72	13.95	0	null	null	27.7
0	20100208	13.85	13.95	13.85	13.9	256894	3571936	78	13.91	0	0	0	0	0	0	0.05	0.36	13.85	0	null	null	27.8
0	20100209	13.95	14.05	13.2	13.9	1539014	2.10E+07	387	13.63	0	0	0	0	0	0	0	0	13.9	0	null	null	27.9

Figure 41: format of single datasheet

DATE	VOL	PCHG	ABNORMAL	TYPE	NOTR	TOVR
20100102	20164	0	0	Normal	96	490381
20100103	125181	-0.49	0	Normal	90	3041441
20100104	149824	-0.25	0	Normal	115	3632195
20100105	124600	2.72	0	Normal	147	3078438
20100106	121924	0.24	0	Normal	105	3025296
20100109	102680	0.96	0	Normal	142	2580051
20100110	87833	-0.24	0	Normal	158	2185512
20100111	525061	-3.58	0	Normal	92	12700000
20100112	232248	-0.99	0	Normal	241	5587105
20100113	100571	1	0	Normal	134	2425558
20100116	492639	2.72	0	Normal	721	12100000
20100117	807828	3.37	0	Normal	776	20600000
20100118	318419	0.23	0	Normal	447	8190359
20100119	171167	0.7	0	Normal	375	4420913
20100120	145292	-0.69	0	Normal	308	3745964
20100123	369088	-3.72	0	Normal	226	9225963
20100124	476484	-0.97	0	Normal	221	11800000
20100125	199000	0.98	0	Normal	206	4909106
20100126	173907	-0.97	0	Normal	174	4324381
20100127	224392	1.46	0	Normal	133	5525110
20100130	195300	-1.44	0	Normal	204	4774956
20100131	72127	2.44	1	price	114	1799438
20100201	62833	-1.19	0	Normal	136	1579562
20100202	28565	1.2	0	Normal	178	718110

Figure 42: final dataset format

Appendix D: Source codes

```
loaded: function (evt) {
    // Obtain the read file data
    let fileFormat = this.fileName.split('.')[1];
    let noOfColumns;

    if (fileFormat.toLowerCase() === 'csv') {
        var fileString = evt.target.result;

        var count = 0;
        noOfColumns = this.get('selectedCompany').code === '1' ? 8 : 7;

        fileString = fileString.replace(/(\n)+/g, ",");

        if (noOfColumns <= noOfColumns) {
            var fileArray = fileString.split(',');
            var headerArray = fileArray.splice(0, noOfColumns);
            var dataArray = Ember.A();

            while (fileArray.length > noOfColumns - 1 && count < 400000) {
                var rowArray = fileArray.splice(0, noOfColumns);
                var rowObject = {};

                Ember.$.each(headerArray, function (key, header) {
                    rowObject[header] = rowArray[key];
                });

                dataArray.pushObject(rowObject);
                count = count + 1;
            }

            this.set('masterContent', dataArray);
            this.filterCompany();
            this.set('processSuccessText', 'Successfully Pre Processed Data');
            this.set('processCss', 'up-fore-color');
            console.timeEnd("Time consumed for pre-processing");
```

Figure 43: Data loading through the csv

```
storeFilteredData: function (filteredContent, filterCompany) {
    var companyId = filterCompany ? filterCompany.code : priceWidgetConfig.transactions.filterCompany;
    var content = filteredContent;

    if (!Ember.ADSMeta) {
        Ember.ADSMeta = {};
    }

    // Test data having 1/3 of total data, and 2/3 for training
    var dataDivideIndex = Math.round(content.length / 3);
    var testData = content.splice(2 * (dataDivideIndex), content.length - 1);


    var trainData = content;

    if (!Ember.ADSMeta[companyId]) {
        Ember.ADSMeta[companyId] = {};
    }

    Ember.ADSMeta[companyId].trainData = trainData;
    Ember.ADSMeta[companyId].testData = testData;

    var saveString = utils.jsonHelper.convertToJson(Ember.ADSMeta[companyId]);
    localStorage.setItem('ADSMeta-' + companyId, saveString);
    localStorage.setItem('ADSCompany', companyId);
},
```

Figure 44: divide and store filtered data UN repository manager

```
prepareBayesDataSet: function (trainContent, anomalyKey) {
    this.set('isShowTable', false);

    let arrayTraining = [];
    const trainingColumns = ['PCHG', 'VOL', 'TOVR', 'NOTR','abnormal?'];

    Ember.$.each(trainContent, function (key, stock) {
        arrayTraining[arrayTraining.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR, stock[anomalyKey]];
    });

    const classifier = new bayes.NaiveBayes({
        columns: trainingColumns,
        data: arrayTraining,
        verbose: true
    });

    classifier.train();

    return classifier;
},
```

Figure 45: Train naive Bayes classifier

```
prepareRegDataSet: function (trainContent, anomKey) {
    this.set('isShowTable', false);
    // this.get('anomalyArray').clear();

    let arrayTrainingOut = [],
        trainingSet = [];

    Ember.$.each(trainContent, function (key, stock) {
        trainingSet[trainingSet.length] = [stock.priceChg, stock.volumeChg];
        arrayTrainingOut[arrayTrainingOut.length] = stock[anomKey];
    });

    const classifier2 = new RandomForestClassifier.RandomForestClassifier();
    classifier2.train(trainingSet, arrayTrainingOut);

    return classifier2;
},
```

Figure 46: Train random forest classifier

```
prepareSVMDataSet: function () {
    var trainContent = this.get('trainContent');
    var arrayTrainingIn = [];
    var arrayTrainingOut = [];

    Ember.$.each(trainContent, function (key, stock) {
        arrayTrainingIn[arrayTrainingIn.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR];
        arrayTrainingOut[arrayTrainingOut.length] = stock.anomalyIndexSVM;
    });

    var svm = new ml.SVM({
        x: arrayTrainingIn,
        y: arrayTrainingOut
    });

    svm.train({});

    return svm;
},
```

Figure 47: Train SVM classifier

```
prepareDecisionTreeDataSet: function () {
    var trainContent = this.get('trainContent');
    var arrayTrainingIn = [];
    var arrayTrainingOut = [];

    Ember.$.each(trainContent, function (key, stock) {
        arrayTrainingIn[arrayTrainingIn.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR];
        arrayTrainingOut[arrayTrainingOut.length] = stock.TYPE;
    });

    var classifier = new ml.DecisionTree({
        data : arrayTrainingIn,
        result : arrayTrainingOut
    });

    classifier.build();

    return classifier;
},
```

Figure 48: Train decision tree classifier

```
prepareEnsemble: function () {
    this.getNBAnomalies(true);
    this.getLRAnomalies(true);

    const content = (this.get('anomBayesArray').concat(this.get('anomalyArrayLR'))),
        predictions = (this.get('predictBaseArray').concat(this.get('predictRegArray')));

    let arrayTest = [], originalResultTest = [], arrayTrain = [], originalResultTrain = [] ;
    let dataDivideIndex = Math.round(content.length / 3);

    let testData = content.splice(2*(dataDivideIndex), content.length-1);
    let trainData = content;

    Ember.$.each(testData, function (key, stock) {
        arrayTest[arrayTest.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR];
        originalResultTest[originalResultTest.length] = stock.anomalyTypeIndex === 'high risk' ?4 : stock.anomalyTypeIndex === 'price' ? 3 : stock.anomalyTypeIndex ==
    });

    Ember.$.each(trainData, function (key, stock) {
        arrayTrain[arrayTrain.length] = [stock.PCHG, stock.VOL, stock.TOVR, stock.NOTR];
        originalResultTrain[originalResultTrain.length] = stock.anomalyTypeIndex === 'high risk' ?4 : stock.anomalyTypeIndex === 'price' ? 3 : stock.anomalyTypeIndex
    });


    const classifier = this.prepareRF(arrayTrain, originalResultTrain);
    this.testRF(classifier, arrayTest, originalResultTest, testData);
},
```

Figure 49: Prepare ensemble classifier

```
getCustomCellColor: function (dataRow) {
    let abnormalArr = [3, 2, 1];
    let abnormalTypeArr = [4, 3, 2];
    let bkGroundColor = utils.nativeHelper.getStyleOfElement('anomaly-color', 'background-color');

    if (this.columnIds.includes('sysAnomality')) {
        if (abnormalArr.includes(dataRow.sysAnomality)) {
            return bkGroundColor;
        }
    } else {
        if (abnormalTypeArr.includes(dataRow.anomaly)) {
            return bkGroundColor;
        }
    }
},
```

Figure 50: applying custom cell colors for indicating suspected transactions

```
paint: function (gc, config) {
    var value = config.formatValue(config.value, config),
        bounds = config.bounds,
        x = bounds.x,
        y = bounds.y,
        cellWidth = bounds.width,
        cellHeight = bounds.height,
        colors = [],
        leftPadding, rightPadding, isCellNotModified,
        isExpandAvailable = config.isExpandAvailable,
        iconWidth = config.iconWidth || 15,
        isActive = this.isActive(config);

    const isAbnormal = Number(config.dataRow.ABNORMAL);

    if (config.snapshot) {
        isCellNotModified = true;

        isCellNotModified = value === config.snapshot.value && isCellNotModified;
        isCellNotModified = isAbnormal === config.snapshot.isAbnormal && isCellNotModified;
        isCellNotModified = isActive === config.snapshot.isActive && isCellNotModified;
    }

    isCellNotModified = this.getHoverOrSelectedCellColors(colors, gc, config, isCellNotModified);

    if (isCellNotModified) {
        return;
    }

    gc.clearRect(x, y, cellWidth, cellHeight); // clear rectangle before drawing
    gc.fillStyle = config.backgroundColor;
    gc.fillRect(x, y, cellWidth, cellHeight);
```

Figure 51: icon cell rendering implementation in hyper grid

```
export default TableComponent.extend({

    onDataContentChanged: Ember.observer('dtContent.length', function () {
        this.set('dataArray', this.get('dtContent') ? this.get('dtContent') : []);
    }),

    onLoadWidget: function () {
        this.set('tblElementId', '#' + 'sideBarCanvas-' + this.get('wkey'));
    },

    onPrepareData: function () {
        let str = localStorage.getItem('AllDataMeta');

        this.set('dtContent', this.utils.jsonHelper.convertFromJson(str ? str : {}));
        this.set('dataArray', this.get('dtContent') ? this.get('dtContent') : []);
    },

    initColumnConfiguration: function () {
        let defaultColumnIds = priceWidgetConfig.transactions.defaultWatchlistIds.slice();
        let columnIds = this.get('columnIds') || defaultColumnIds;

        this.setProperties({
            columnConfiguration: priceWidgetConfig.transactions.columnDefinition,
            moreColumnIds: defaultColumnIds,
            columnIds: columnIds
        });
    }
});
```

Figure 52: side bar content implementation

```
import ...

export default MultiClassifierModel.extend({
    widgetTitle: 'Multiple Classifier Model with Multiple Company Data',

    onLoadWidget: function () {
        this._super();

        let classifierArray = [
            {code: '1', des: 'Random Forest'},
            {code: '3', des: 'Naive Bayes'},
            {code: '4', des: 'Ensemble Model'},
            {code: '5', des: 'Decision Tree'}
        ];

        this.set('classifierArray', classifierArray);
        this.set('currentClassifier', this.get('currentClassifier') ? this.get('currentClassifier') : classifierArray[0]);
    },

    initColumnConfiguration: function () {
        let defaultColumnIds = priceWidgetConfig.anomalyMain.allCompanyColumnIds.slice();
        let columnIds = this.get('columnIds') || defaultColumnIds;

        this.setProperties({
            columnConfiguration: priceWidgetConfig.anomalyMain.columnDefinition,
            moreColumnIds: defaultColumnIds,
            columnIds: columnIds
        });
    },

    getCurrentCompany: function () {
        return 'All companies';
    },

    actions: {
        setClassifier: function (item) {
            this.set('currentClassifier', item);
```

Figure 53: Advance model approach implementation

Appendix E: Implemented system UI with results

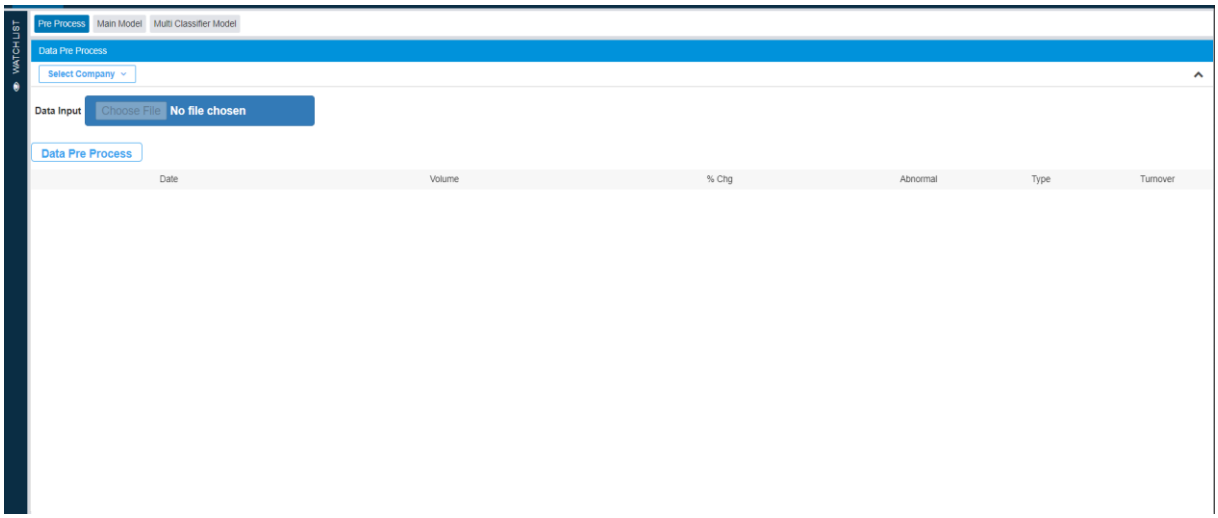


Figure 54: data preprocessing sample UI

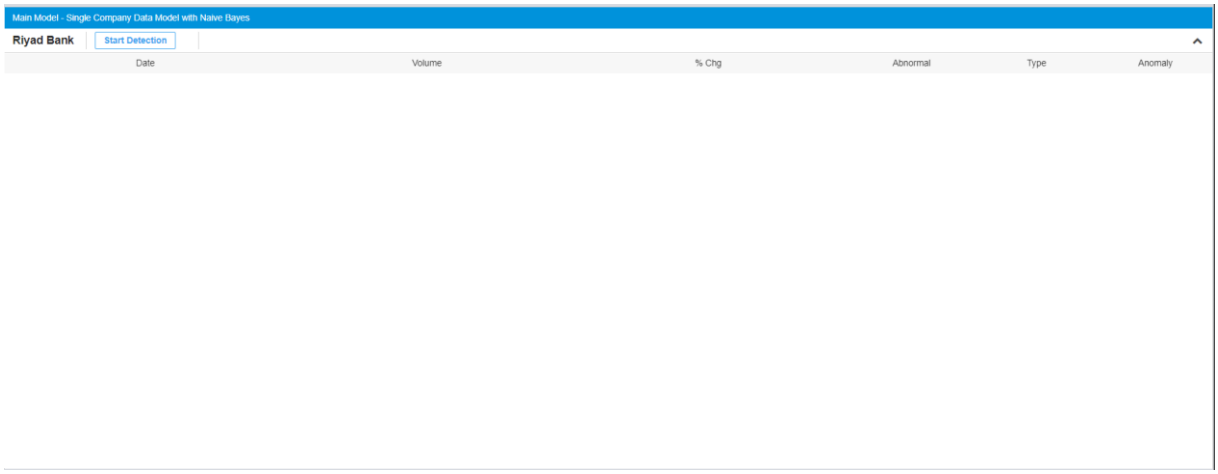


Figure 55: system main model

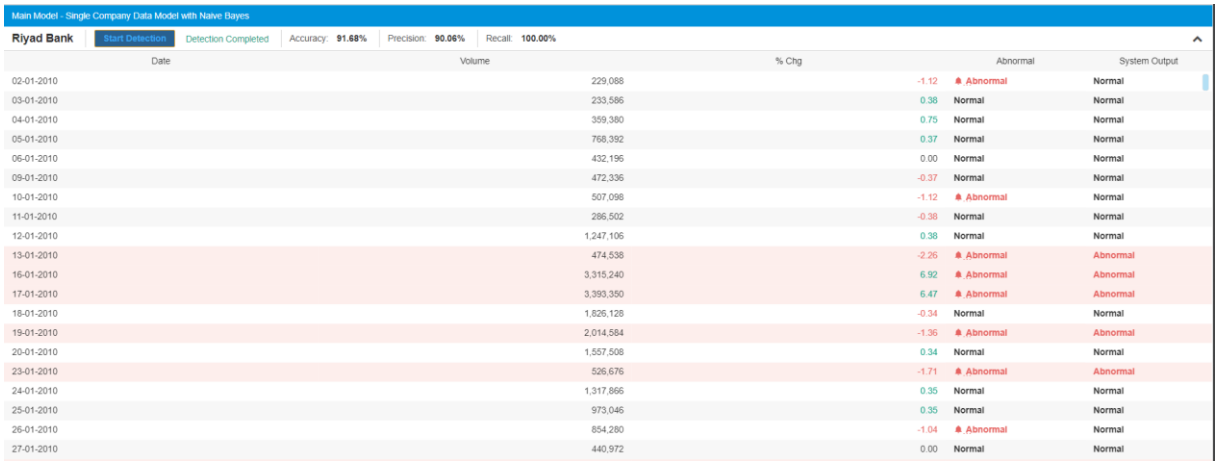


Figure 56: system main model with results

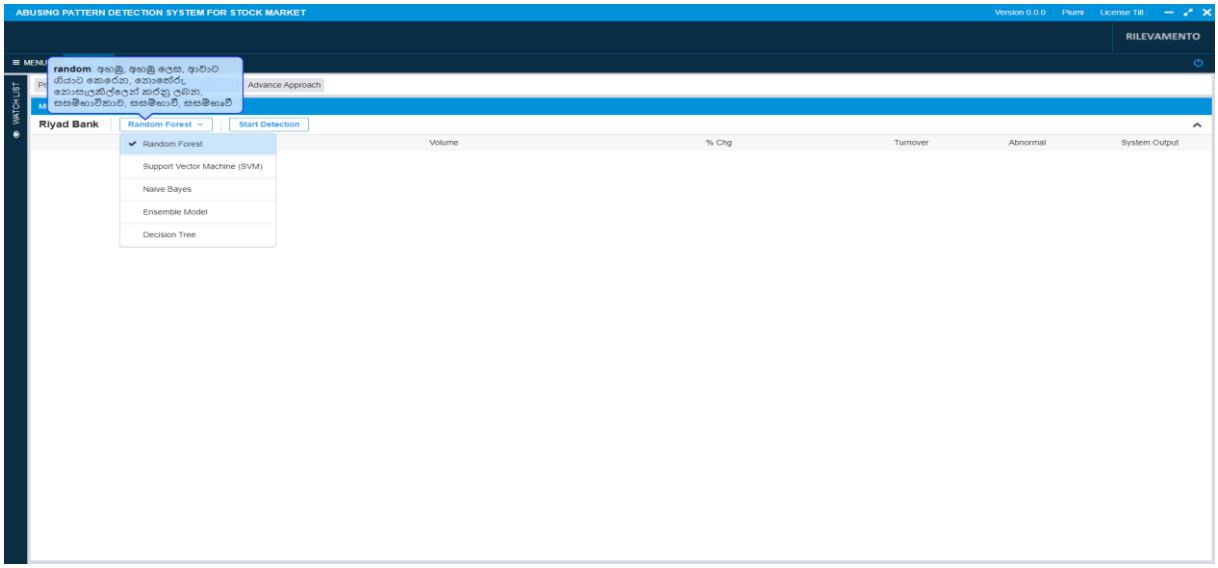


Figure 57: multi classifier model

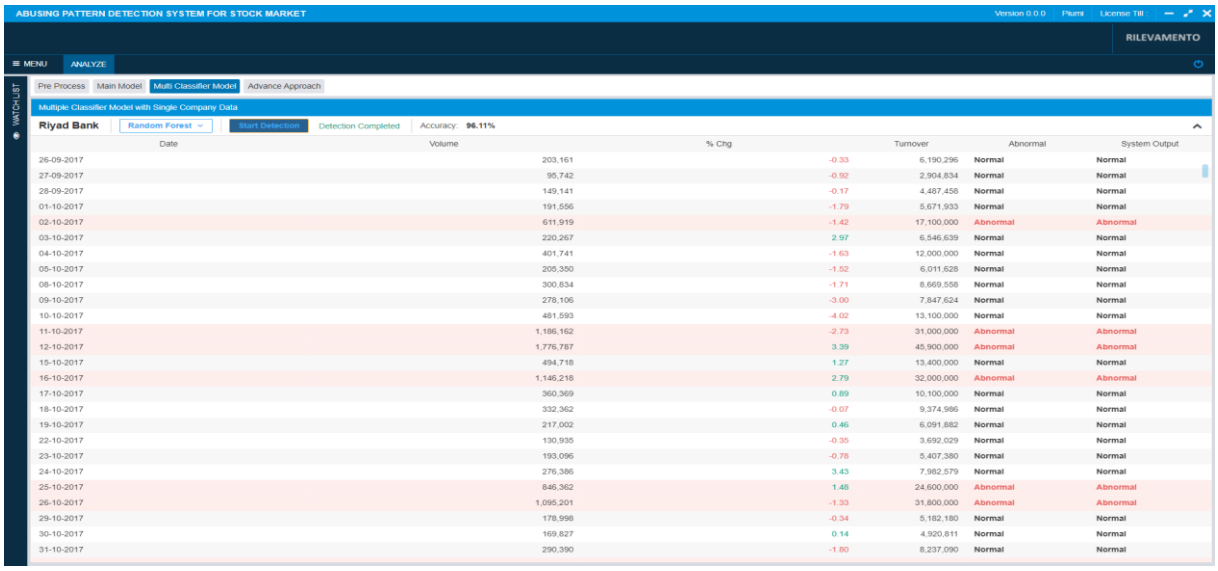
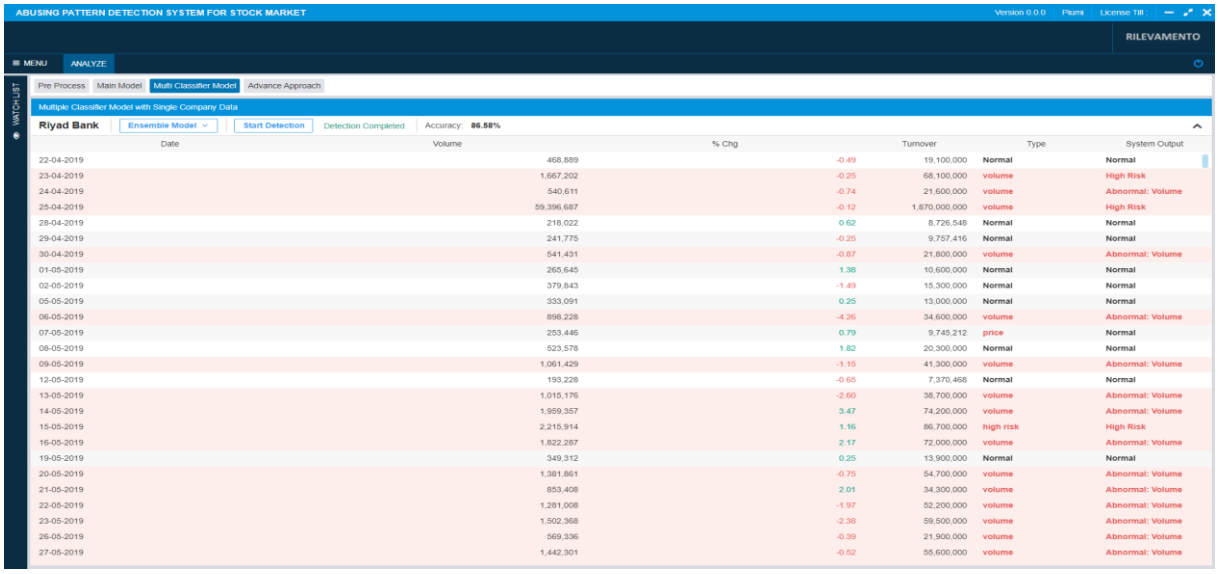
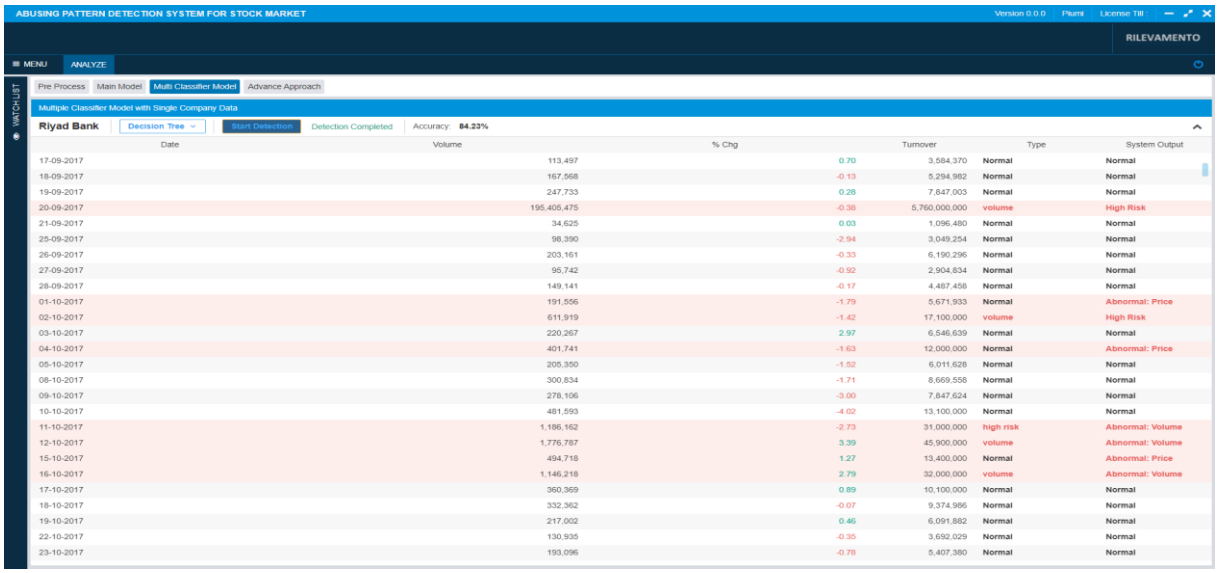


Figure 58: multi classifier results against random forest



Date	Volume	% Chg	Turnover	Type	System Output
22-04-2019	468,889	-0.49	19,100,000	Normal	Normal
23-04-2019	1,667,202	-0.25	68,100,000	volume	High Risk
24-04-2019	540,611	-0.74	21,600,000	volume	Abnormal: Volume
25-04-2019	59,396,687	-0.12	1,870,000,000	volume	High Risk
28-04-2019	218,022	0.62	8,726,548	Normal	Normal
29-04-2019	241,775	-0.25	9,757,416	Normal	Normal
30-04-2019	541,431	-0.87	21,800,000	volume	Abnormal: Volume
01-05-2019	265,645	1.38	10,600,000	Normal	Normal
02-05-2019	379,843	-1.49	15,300,000	Normal	Normal
05-05-2019	333,091	0.25	13,000,000	Normal	Normal
06-05-2019	898,228	-4.26	34,600,000	volume	Abnormal: Volume
07-05-2019	253,446	0.79	9,745,212	price	Normal
08-05-2019	523,576	1.82	20,300,000	Normal	Normal
09-05-2019	1,061,429	-1.15	41,300,000	volume	Abnormal: Volume
12-05-2019	193,228	-0.65	7,370,468	Normal	Normal
13-05-2019	1,015,176	-2.60	38,700,000	volume	Abnormal: Volume
14-05-2019	1,959,357	3.47	74,200,000	volume	Abnormal: Volume
15-05-2019	2,215,914	1.16	86,700,000	high risk	High Risk
16-05-2019	1,822,287	2.17	72,000,000	volume	Abnormal: Volume
19-05-2019	349,312	0.25	13,900,000	Normal	Normal
20-05-2019	1,381,861	-0.75	54,700,000	volume	Abnormal: Volume
21-05-2019	853,408	2.01	34,300,000	volume	Abnormal: Volume
22-05-2019	1,281,008	-1.97	52,200,000	volume	Abnormal: Volume
23-05-2019	1,502,368	-2.38	59,500,000	volume	Abnormal: Volume
26-05-2019	569,336	-0.39	21,900,000	volume	Abnormal: Volume
27-05-2019	1,442,301	-0.52	55,600,000	volume	Abnormal: Volume

Figure 59: multi classifier results against ensemble approach



Date	Volume	% Chg	Turnover	Type	System Output
17-09-2017	113,497	0.70	3,584,370	Normal	Normal
18-09-2017	167,568	-0.13	5,294,982	Normal	Normal
19-09-2017	247,733	0.28	7,847,003	Normal	Normal
20-09-2017	195,405,475	-0.38	5,760,000,000	volume	High Risk
21-09-2017	34,625	0.03	1,096,480	Normal	Normal
25-09-2017	98,390	-2.94	3,049,254	Normal	Normal
26-09-2017	203,161	-0.33	6,190,296	Normal	Normal
27-09-2017	95,742	-0.92	2,904,834	Normal	Normal
28-09-2017	149,141	-0.17	4,487,458	Normal	Normal
01-10-2017	191,556	-1.79	5,671,933	Normal	Abnormal: Price
02-10-2017	611,919	-1.42	17,100,000	volume	High Risk
03-10-2017	220,267	2.97	6,546,639	Normal	Normal
04-10-2017	401,741	-1.63	12,000,000	Normal	Abnormal: Price
05-10-2017	205,350	-1.52	6,011,628	Normal	Normal
08-10-2017	300,834	-1.71	8,669,558	Normal	Normal
09-10-2017	278,106	-3.00	7,847,624	Normal	Normal
10-10-2017	481,593	-4.02	13,100,000	Normal	Normal
11-10-2017	1,186,162	-2.73	31,000,000	high risk	Abnormal: Volume
12-10-2017	1,776,787	3.39	45,900,000	volume	Abnormal: Volume
15-10-2017	494,718	1.27	13,400,000	Normal	Abnormal: Price
16-10-2017	1,146,218	2.79	32,000,000	volume	Abnormal: Volume
17-10-2017	360,369	0.89	10,100,000	Normal	Normal
18-10-2017	332,362	-0.07	9,374,986	Normal	Normal
19-10-2017	217,002	0.46	6,091,882	Normal	Normal
22-10-2017	130,935	-0.35	3,692,029	Normal	Normal
23-10-2017	193,096	-0.78	5,407,380	Normal	Normal

Figure 60: multi classifier results against decision tree approach

Appendix F: Performance Testing Results

	Attempt (1)	Attempt (2)	Attempt (3)	Average Time
Time consumed (1010)	21.296	19.101	14.891	18.429
Time consumed (1050)	16.687	15.832	22.229	18.249
Time consumed (all)	19.604	15.401	14.806	16.603

Appendix G: Model Accuracy Testing Results

	Naïve Bayes	Support Vector Machine	Decision Tree	Random Forest
Accuracy	91.58%	50%	99.05%	99.68%
Precision	85.95%	0%	85.95%	99.36%
Recall	99.58%	0%	99.58%	100%