Northumbria University
NEWCASTLE

ANOMALY DETECTION APPROACHES FOR STOCK PRICE MANIPULATION DETECTION

BAQAR ABBAS RIZVI

FACULTY OF ENGINEERING AND ENVIRONMENT

NORTHUMBRIA UNIVERSITY

THIS DISSERTATION IS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

September 2021

DECLARATION

I hereby declare that the work contained in this thesis has not been submitted for any other award and that it is all my own work. I also confirm that this work fully acknowledges opinions, ideas and contributions from the work of others.

Any ethical clearance for the research presented in this thesis has been approved. Approval has been sought and granted by the University Ethics Committee in July 2017.

I declare that the Word Count of this Thesis is 44077 words.

Baqar Abbas Rizvi

30.09.2021

Table of Contents

List of Abbreviations

AE	Autoencoder
AHMMAS	Adaptive Hidden Markov Model
AIS	Artificial Immune System
AMD	Advanced Micro Devices Stock
AMISE	Asymptotically Mean Integrated Square Error
ANN	Artificial Neural Networks
AUC	Area Under the ROC Curve
BGM	Bayesian Gaussian Mixture
BPS	Basis Points
V-BOMM	Voting Based Outlier Mining on Multiple Time Series
CEA	Commodity Exchange Act
CEC	Congress on Evolutionary Computation
CFTC	Commodity Futures Trading Commission
CME	Chicago Mercantile Exchange
COF	Connectivity based Outlier Factor
CPAD	Collective Probabilistic Anomaly Detection
CSM	Co-Stimulatory Molecules
CSRC	China Securities and Regulatory Commission
DC	Dendritic Cell
DCA	Dendritic Cell Algorithm
DPGMM	Dirichlet Process Gaussian Mixture Model
DS	Danger Signals

DWT	Discrete Wavelet Transform
EBAY	Ebay stock
ED	Encoder-Decoder
EMD	Empirical Model Decomposition
EU	European Union
FA	Finance and Aviation
FAR	False Alarm Rates
FBI	Federal Bureau of Investigation
FCA	Financial Conduct Authority
FN	False Negatives
FP	False Positives
FPR	False Positives Rates
FSA	Financial Services Act
FSMA	Financial Services and Markets Act
GAN	Generative Adversarial Networks
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GMM	Gaussian Mixture Models
HFT	High Frequency Trading
HIS	Human Immune System
HMM	Hidden Markov Model
IMF	International Monetary Fund
INTC	Intel Corp
IS	Immune System
ISE	Istanbul Stock Exchange

KDE	Kernel Density Estimate
KL	Karhunen–Loève
KPCA	Kernel Principal Component Analysis
LDOF	Local Distance Based Outlier Factor
LLC	Limited Liability Company
LOBSTER	Limit Order Book Reconstruction System
LOF	Local Outlier Factors
LSE	London Stock Exchange
LSTM	Long Short Term Memory
LU	Linear Unit
MAR	Market Abuse Regulation
MCAV	Mature Context Antigen Value
MI	Mutual Information
MIDAS	Market Information Data Analytics System
MKDE	Multidimensional KDE
MSFT	Microsoft Stock
NASDAQ	National Association of Securities Dealers Automated Quotations.
NBS	Newcastle Business School
NEAT	National Exam Analytics Tool
NFLEX	Netflix Stock
NN	Neural Networks
NSA	Negative Selection Algorithms
NYSE	New York Stock Exchange

OCSVM	One Class Support Vector Machine
OTC	Over the Counter
PAMP	Pathogen Associated Molecular Patterns
PCA	Principal Component Analysis
PCC	Principal Component Classifier
PDF	Probability Density Function
PGA	Peer Group Analysis
PNN	Probabilistic Neural Networks
QUALCOM	Qualcomm Stock
QUEST	Quaternion Estimation algorithm
RBF	Radial Basis Function
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristics
RST	Rough Set Theory
SEC	Securities and Exchange Commission
SET	Stock Exchange of Thailand
SFO	Serious Fraud Office
SGMM	Summarization Based Gaussian Mixture Model
SIRIUS	Sirius US Stock
SS	Safe Signals
SVM	Support Vector Machines
TCN	Temporal Convolutional Network
TN	True Negative
TP	True Positive

TPR	True Positive Rates
UCI	University of California
UK	United Kingdom
US	United States
USC	United States Code
USD	United States Dollar
VOMM	Variance Outlier Based Mixture Model
WAB	Westinghouse Air Base

Acknowledgement

Firstly, I would like to thank Almighty for its blessings throughout the course of my PhD. I would like to extend my gratitude for giving me wonderful supervisors and my family to stand beside me while going through this journey over the last few years.

It is hard to express gratitude and thanks in words to my supervisor Dr Ammar Belatreche for the tremendous amount of help and support he has offered me during the course of PhD. Indeed, he is more than a supervisor, a friend and a guide and I am really lucky to have him. I would also like to thank my co-supervisor Prof Ahmed Bouridane for his support that throughout during the past few years.

I would like to thank my mother, my wife and kids for their support and to act as my strength when I was in desperate need of it. They have been extremely patient with me and sacrificed their fun for many a times I asked.

In the end I would like to thank the Department of Computer & Information Sciences, Northumbria University for the Research and development fund scholarship and providing me the relevant resources required. I would ever miss this throughout my life, it’s been a great experience.

Abstract

Stock price manipulation uses illegitimate means to artificially influence market prices of several stocks. It causes massive losses and undermines investors' confidence and the integrity of the stock market. It is evident from the literature that most existing research focused on detecting a specific manipulation scheme using supervised learning but lacks the adaptive capability to capture different manipulative strategies. This begets the assumption of model parameter values specific to the underlying manipulation scheme. In addition, supervised learning requires the use of labelled data which is difficult to acquire due to confidentiality and the proprietary nature of trading data. This thesis presents novel manipulation detection models that can generally detect all of the targeted manipulative schemes independent to the need of varying parameters for specific schemes.

This thesis contributes five different detection algorithms for stock price manipulation in unsupervised domain that are categorised into three major models: decomposition based, artificial immune inspired and deep learning based. Decomposition based models transform stock price trades into orthogonal and principal components whilst preserving the original information of the input data. The transformed components are then subjected to a proposed multi-dimensional binary clustering techniques for manipulation detection. Two decomposition based algorithms have been proposed in this category that efficiently improved detection rates with reduced computational complexity. Immune inspired detection model translates the natural immune system approach into market manipulation treating a manipulative instance as a pathogen. The proposed approach is adapted for scaling down the dimension of the input data set to a set of only three outputs that are then clustered using KDE clustering. This avoids the need for assigning different threshold parameters as in a conventional DCA, hence automating the detection process. One of the main advantages of using this approach is the significant reduction in false positive rates while further improving the detection rates from the decomposition models. Deep learning based models can further simplify the problem by

providing a set of features that can be used for training a model avoiding the need of designing features using an expert. Two deep learning algorithms are presented in this category: one model exploits the relationship among trading instances in the form an affinity matrix and later train an autoencoder based upon it. The second model presents a novel idea to reduce the false positives by detecting the overlap among normal and abnormal trades using a defined context. It proposes to jointly train a temporal convolutional network (TCN) and a generative adversarial network (GAN) together under the context extracted from the input data. Additionally, an updated similarity metric is explored using the feature representations learned by the GAN's discriminator as the basis for reconstruction.

All of the proposed research models are comprehensively assessed on multiple datasets of some highly traded stocks and outperforms some of the selected state-of-the-art models in anomaly detection. The robustness of the proposed models is further evaluated by comparing the results with selected benchmark models in stock price manipulation detection. Further a series of experiments on multiple datasets are also performed including when two or more manipulative activities occur within a short duration of each other and by varying the window length of the dataset fed to the model to evaluate the effectiveness of the models. The results show a significant performance enhancement in terms of the AUC, F-measure values while a significant reduction in false alarm rate (FAR) has been achieved.

Chapter 1: Introduction

1.1 Rationale of the Thesis

Market manipulation is an action performed by fraudsters to create a delusional image of an established commodity's price, volume etc. through some illegitimate means [1], pertaining to its own personal profit. Such an illusion often leads to a false impression of a product's stocks and even the market. This ruins the investor's interest not just in a given stock but also in the market. It also mutilates public confidence in the market integrity and undermines market efficiency, which will prevent the development and mitigate the economic power engine of a country.

Over the years, markets have evolved with diversification in trading practices, globalisation, sheer competition with more modern businesses being added every day. Since the markets are an integral part of the modern business world, they play a significant role in the economy of their respective countries. Indeed, most of the stock exchanges are being under constant monitoring by several regulatory authorities, market analysts and researchers in a bid to detect and identify market manipulation.

However, it is computationally expensive both in terms of manpower and time. For example, nearly five years after the flash crash of 2010, US department of justice arrested the man responsible of the trillion-dollar crash in 2015 [2]. Stock price manipulation, as part of the trade-based manipulation [3] is related to influencing the trading price of financial security using abusive schemes. It consequently effects the faith and the predicted gross return from a stock. The process of manipulation detection and further conviction used was later described as 'bicycles to try and catch Ferraris' by Bloomberg [4]. Intelligent computational techniques such as data mining, machine learning/deep learning including bio-inspired techniques can be extremely helpful in reducing the amount of input effort both in time and labour [5,6,7]. Allen and Gale [3] classified market manipulation into three main types: action based, information based, and trade based manipulation. Action based manipulation is an action rather than trading, performed by the company managers or executives who hold

the supply of a well-established product by increasing its demand and hence the stock price. Information based manipulation intends to spread a rumour or release some inside information about a company or its stock with an intention to influence the price. The third type is trade based manipulation that specifically deals with the manipulation within the stock exchange. This thesis focusses only trade based manipulation given the impacts of it on market sentiments.

1.2 Problem Formulation

1.2.1 Trade-based Manipulations: Stock Price Manipulation

One of the major types of trade-based manipulation is price manipulation in which the trader targets to influence the buy/sell prices of any company stock. The other types of trade-based manipulation include volume-based manipulation and cross-market manipulation. However, the detection for these types is beyond the scope of this thesis. Moreover, stock price manipulation is excessively used and hence presents has the largest impact on stock markets [8, 9]. It implements a variety of strategies like quote stuffing, pump and dump [8], ramping or gouging also known as Spoof Trading [10] etc. Several researchers, mentioned in the review section made few attempts in this field using both labelled and unlabelled datasets [5-7] but failed to acknowledge the dataset, which is rare and expensive when labelled, along with individual detection models for different manipulation schemes and the heuristically assumed values for the model parameters involved in the decision-making process.

1.3 Challenges, Aim and Objectives

As is evident from the above-mentioned sections, stock price manipulation detection suffers from a wide variety of challenges including lack of labelled data due to privacy and confidentiality issues in most markets. Also, the evolving manipulation strategies where no substantial features are defined for a manipulative pattern of any scheme. Formulating multiple manipulative behaviours using different modelling techniques for a financial instrument exhibiting different volatility rates is the major challenge that exist within this field. The abundance of normal trading data for a few manipulative instances makes the task even more complicated for two reasons (1) making the whole dataset unbalanced and the detection model biased (2) overlapping normal and

abnormal data patterns that increases the number of false positives. Furthermore, the unavailability of additional features such as news, global trends, and an investor's trading behaviour on a stock over several days, real-time detection of manipulation detection is avoided.

The main aim of this thesis is the development of generic, robust and efficient manipulation detection models that can widen the gap between normal and abnormal trade instances by using efficient feature extraction techniques and density-based clustering. To achieve such aim, the following objectives have been defined,

1. Explore various manipulation schemes along with their real-life examples to study the effects they have on both price and volume.
2. Design efficient feature extraction methods that are able to capture the effects of different manipulation schemes.
3. To compare proposed approaches with existing benchmark methods in stock price manipulation by experimentally evaluating on the given datasets.
4. Develop a clustering algorithm for manipulation detection based on kernel density estimation where the number of clusters is not required *a priori* and validate it through experimental evaluations.
5. Develop an effective detection that can avoid dependencies on data annotations and test it on large and multiple datasets targeting different objects including price, volume, spread etc of varying size and volatility levels.

1.4 Thesis Contributions

This thesis presents novel approaches to detect stock price manipulation using efficient intelligent approaches after thoroughly studying market microstructure and the behaviour of various manipulative patterns. The research works mentioned in this thesis have been peer reviewed for one journal [29] with one journal under preparation and four reputed international conferences [30-33]. The major contributions of this thesis include:

- After studying multiple stock price manipulation behaviours, two novel manipulation detection methods have been proposed in Chapter 3 based on the decomposition of the input stock price and volume data into instantaneous frequencies using empirical mode decomposition (EMD) and principal components based on varying order of variance in higher dimensions using kernel based principal component analysis (KPCA). The idea is to observe the behaviour of such decomposed components, assumed to be active indicators of stock price changes and apply proposed clustering techniques on them to judge normal and abnormal patterns.

- A fully unsupervised model based on the novel idea of learning the relationship among stock price instances, in the form of an affinity matrix is proposed in chapter 5. It is used to train an under-fitting autoencoder in order to learn an efficient representation of the normal stock prices using its inherent density estimate as the reconstruction error. The model envisaged the relationship among the normal trading instances and evaluate the performance while testing it with the relationship among normal and abnormal trades.

- An immune-inspired manipulation detection technique that translates the process of detecting a pathogen or any foreign agent in human bodies to stock price manipulation detection treating the manipulative instance as a pathogen. In this chapter a small set of extracted features were categorized into PAMP, Danger and Safe signal based on mutual information, calculated with the output class. The outputs so obtained are then subjected to KDE clustering that assigns data instances that form a cluster of unit size as manipulative instances.

- An experimental analysis of detecting stock price manipulation under a context for activities otherwise treated as normal trades is proposed in chapter 5. The research proposed a tempGAN model jointly training a temporal convolutional network (TCN) and a generative adversarial network (GAN) together under the context extracted from the input data. The idea is to study anomalous information defined in an optimally

generated context using a convolutional neural network over temporal domain. Additionally, an updated similarity metric is explored using the feature representations learned by the GAN's discriminator as the basis for reconstruction.

- A thorough literature review of existing market manipulation detection techniques including a review of anomaly detection techniques in general and those used specifically for time series as presented in chapter 2. Furthermore, an experimental evaluation of a selection of those benchmark approaches both in market manipulation and anomaly detection has been performed in chapter 3.

1.5 Scope of the Thesis

The scope of this thesis is limited by the detection of stock price manipulation rather than identification of manipulative scheme using unsupervised machine learning. Each proposed approach is validated on standard datasets free from any manipulative instance and the results are compared with existing state-of-the-art approaches using quantitative evaluation metrics. This thesis also discusses the regulatory aspects of market manipulation and explores case studies associated with different types of manipulative schemes both in US and UK. However, the impacts of market manipulation on trading across markets, privacy and training of auditors are treated as beyond the scope of the thesis.

The dataset considered for validating every proposed approach is level – 1 intraday tick data which shows the best bid and ask price for a security including the trades avoiding level – 2 and level – 3 data which shows more complex information about best bid and offers from multiple investors at different depths of trade. In other words, the scope of this work is limited to finding manipulative behaviours in the intraday time series considering only one best bid and offer as the reference. The proposed approaches also focus only on three different types of manipulative schemes that includes pump and dump, spoof trades or spoofing and quote stuffing. It is also limited in its capability in detecting specific volume based manipulation schemes, wash trades and cross market manipulation.

1.6 Publications

This section provides the details of the published papers or are ready to be submitted in both peer-reviewed journals and reputable conferences based on the research work mentioned in this thesis.

Journal papers:

1. **B. Rizvi**, A. Belatreche, A. Bouridane and I. Watson, "Detection of Stock Price Manipulation Using Kernel Based Principal Component Analysis and Multivariate Density Estimation," in IEEE Access, vol. 8, pp. 135989-136003, 2020. (**Contributes to Chapter 3, Price Manipulation using Decomposition Techniques**)

2. **B. Rizvi**, A. Belatreche and A. Bouridane, "Manipulation Detection using Contextually Learned Similarity Metric for Anomaly," IEEE Transactions on neural networks and learning systems (under review), 2021. (**Contributes to Chapter 5, Price manipulation using Deep Features**)

Conference papers:

1. **B. Rizvi**, A. Belatreche, A. Bouridane and K. Mistry," Stock Price Manipulation Detection based on Autoencoder Learning of Stock Trades Affinity," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, pp. 1-8, 2020. (**Contributes to Chapter 5, Price Manipulation using Deep Features**)

2. **B. Rizvi**, A. Belatreche and A. Bouridane, "A Dendritic Cell Immune System Inspired Approach for Stock Market Manipulation Detection," 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, pp. 3325-3332, 2019. (**Contributes to Chapter 4, Price Manipulation using Bio-inspired Artificial Immune Systems**)

3. **Baqar Rizvi**, Ammar Belatreche and Ahmed Bouridane, "Immune Inspired Dendritic Cell Algorithm for Stock Price Manipulation Detection," in Conference proceedings Intelligent Systems and Applications, Intellisys, Advances in Intelligent Systems and

Computing, vol 1037. Springer, Cham, pp. 352-361, 2019. (**Contributes to Chapter 4, Price Manipulation using Bio-inspired Artificial Immune Systems**)

4. **Baqar Abbas**, Ammar Belatreche and Ahmed Bouridane, "Stock Price Manipulation Detection Using Empirical Mode Decomposition Based Kernel Density Estimation Clustering Method," in Conference proceedings Intelligent Systems and Applications, Intellisys, Advances in Intelligent Systems and Computing, vol 869. Springer, Cham, pp. 851-866, 2018. (**Contributes to Chapter 3, Price Manipulation using Decomposition Techniques**)

1.7 Structure of the Thesis

This thesis is designed in a way to first introduce the stock market manipulation and its types in the first chapter. Along with problem formulation and the challenges, existing regulations in US and UK are also introduced in the first chapter. As always imperative the aim and objectives of the thesis along with published research in the form of contributions made so far are also mentioned here.

A detailed study about the literature is explained as part of the second chapter: Literature review. Both empirical and anomaly detection models using machine/deep learning within stock price trades are discussed here. In addition, anomaly detection models within time series are also discussed in this chapter for a better understanding of how other models have progressed upon time series data and what makes them vulnerable to false positives if employed for stock price manipulation detection.

The rest of the thesis is organised as follows: Chapter 3 discusses the decomposition methods including empirical mode decomposition and principal component analysis along with their experimental results and a detailed discussion about the approach followed by conclusion. Chapter 4 briefly explains Dendritic Cell Algorithm under the abstract of artificial immune system. It further elaborates the implementation of dendritic cell algorithm

approach to stock price manipulation detection. It also discusses the experimental evaluation of the results followed by discussion and conclusion. Chapter 5 discusses the role of deep learning in detecting price manipulation by introduce some basic approaches first and how they are adapted towards detecting manipulative instances. One of the major challenges faced within market manipulation is the contextual evaluation of a manipulative behaviour to reduce the number of false positives in detection, is also discussed and a novel solution using deep learning techniques is proposed within this chapter, the experimental evaluation discussed and concluded. The thesis is finally concluded in Chapter 6 including future aspects of the manipulation detection that can be implemented.

Chapter 2: Literature review

2.1 Introduction

2.1.1 Background

There is an increasing demand of analysing stock price data at most of the stock exchanges around the world. One of the key objectives in doing so is the establishment of a detection model that can identify manipulative instances caused by the market manipulators or market abusers. Allen and Gale [3] classified market manipulation into three main types: action based, information based, and trade based manipulation. Action based manipulation is an action rather than trading, performed by the company managers or executives who hold the supply of a well-established product by increasing its demand and hence the stock price. Information based manipulation intends to spread a rumour or release some inside information about a company or its stock with an intention to influence the price. Four Kaupthing Bank executives were caught financing their own share purchases in large and hefty amounts arousing the interest of others [34].

Trade based manipulation on the other hand has everything to do inside a stock exchange where traders, investors, or brokers buy/sell stocks at different prices for different volumes (number of shares or bonds etc. for any security, traded during a period of time) [3], [5]. Unlike action and information based, in trade based manipulation, market manipulators use fraudulent strategies by following up a series of actions imposed on the order-book to influence the equity price of a commodity. The Securities and Exchange Commission (SEC), in a press release, 2015 charged Costa-Rica based MoneyLine Brokers Firm and its founder for engaging in "Pump & Dump" schemes to artificially inflate a stock's price of Warrior Girl, a former shell company and then sell their own shares [34]. According to the report, MoneyLine and its subordinates made illegal profits estimated at a total of $2.3 million. One of the major types of trade-based manipulation is price manipulation in which the trader targets to influence the buy/sell prices of a financial security.

It is important to explore some important definitions used throughout this thesis. Table 2.1 describes some of the important terminology along with this explanation relevant to stock market manipulation.

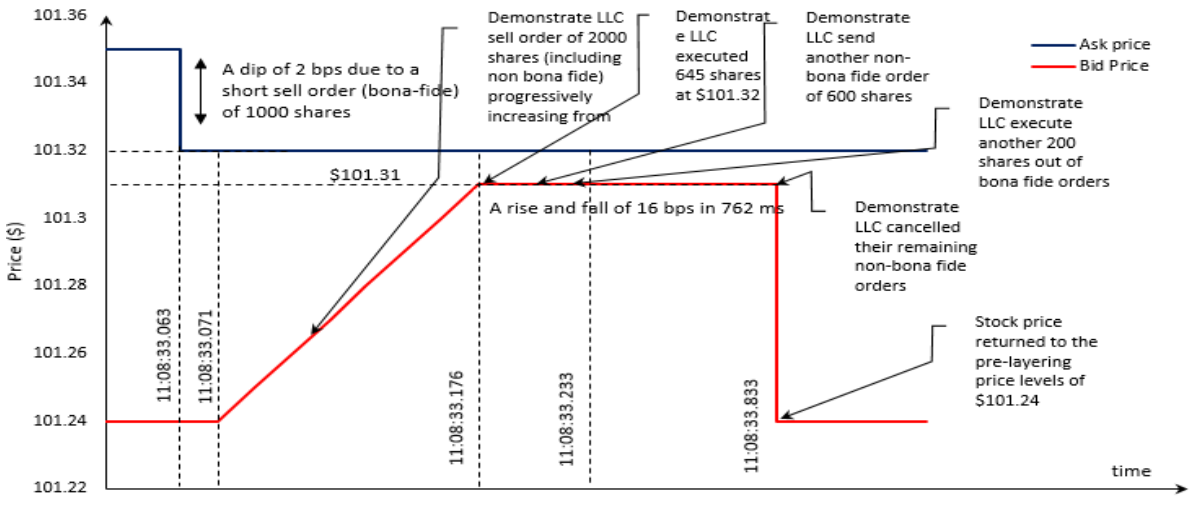
Table 2.1: List of terms commonly related to stock markets

Security	A financial instrument that can be traded on a market e.g., stocks, bonds, forex, shares, etc.
Trade	A transaction resulting from the matching of two orders (buy and sell sides)
Regulatory Market	A market that is regulated through a Directive 2014/65/EU [1].
Regulator	A competent authority responsible for the investigation and prosecution of market manipulation/abuse. More details provided later in this chapter
Order	A buy or a sell order in association to a financial instrument submitted on any trading platform by an investor/trader/broker.
Order type	A buy/sell order with a particular set of features i.e., limit order, auction order etc.
Order-book	A record of orders made by trading members with time-stamps, order ID, price, and volume for a given security including order cancellation.
Trade book	A record of trades by trading members with time-stamps, trade ID, price and volume including the trade and cancellation information.

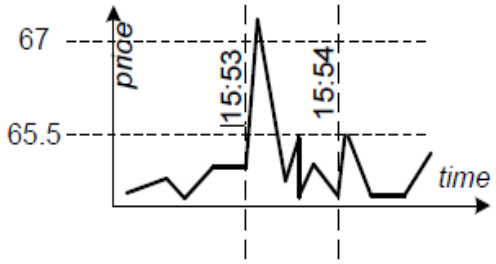
Allen and Gale, 1992 [3] first proposed the theory and principle behind trade based manipulation. It explained that the possibility of trade based manipulation is increasingly higher when it is uncertain that a market buyer has potentially

sound knowledge about the firm's prospects. Aggarwal and Wu [35] further studied this idea and made several important conclusions in their work. They studied nearly 192 SEC cases from 1992 to 2006 and found that stocks that were of low value and illiquid were commonly targeted. This meant that stocks that were less traded and remained in low volumes were the focus of manipulated schemes. It also explored market efficiency through the trades made by normal investors which indicates the true information about a traded financial instrument but warned that it also favours a market abuser as the number of normal investors increases. Because as the normal trader competes for shares, the profitability of the manipulator also increases thereby making the manipulation more possible.

A vast number of empirical and theoretical studies have been conducted in stock price manipulation cases as compared to the detection of trade based stock price manipulation. However, most of them claimed significant improvements in the detection results either only based on certain assumptions in their applied research or using labelled datasets, which makes it easier for the model to learn the anomalous patterns and provide better detection accuracy on the test data. This chapter aims to study the effects of different manipulation detection algorithms proposed in the past upon different schemes. Furthermore, for a manipulation case within a stock, it is always observed that there is an abundance of normal stock price and volume data compared to the manipulative instances. In addition, the scarcity of a manipulated dataset makes anomaly detection within unsupervised domain very challenging. It is to this effect that a comprehensive evaluation of the anomaly detection techniques suitably applied over time series are explored and described as part of this research project (included in this chapter). Besides, it is also relevant to explore some of the general concepts associated with market manipulation including the regulations, regulatory agencies, types of manipulation schemes with real life cases along with the need to impose machine learning on market manipulation detection due to the regulatory challenges faced.

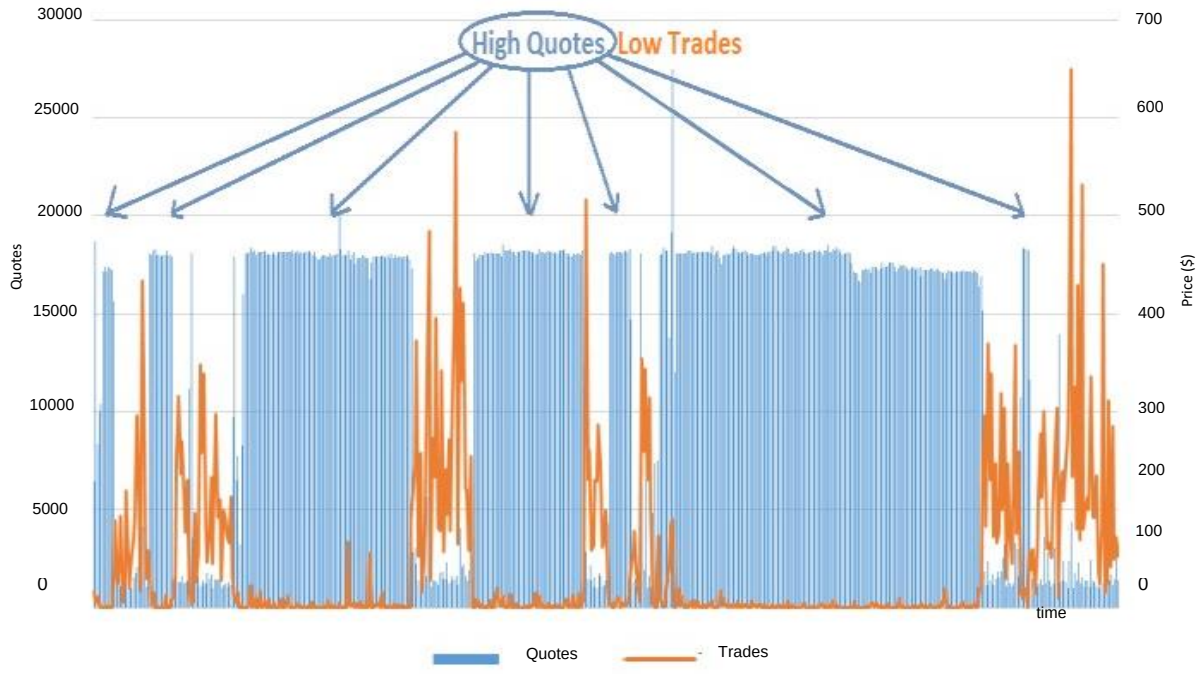


(a)



snapshot from Dec 14, 2011, WAB price: rising 8% in 1s and reverting back 3s later.

(b)



(c)

Figure 2-1: (a) Illustration of the spoofing activity (Saw-tooth waveform) on Sept 25, 2012 and (b) Snapshot of the pump and dump (Spike waveform) manipulation activity from Dec 14, 2011 shows an 8 % rise of Westinghouse Air Brake (WAB) Tech. price within 1 sec and return to previous level 3 secs later. (c) Snapshot of Quote stuffing activity from Nov 01, 2012 shows thousands of quotes been sent to flood the market from 12:26:50 to 12:39:42 pm as is observed the number of trades fell to a lowest level during this interval.

2.2 Manipulation Schemes

It should also be noted that market abusers trying to influence the prices of a stock by fraudulent activities like spoof trading, pump and dump, etc., follow a sequence of well-defined and more importantly 'evolving' actions or strategies to control the equity price of a stock. Few of such schemes covered in this approach are traditional pump and dump and emerging high-tech schemes like spoof trading/ramping, quote stuffing. They are selected because of their impact on the market and the increasing number of cases SEC put to trial [2].

Spoof Trading: One of the most prominent type of price manipulation tactic is Spoof Trading [10] also known as ramping. As an example, a manipulator wants to sell a stock at a higher price than the current ask price. The manipulator will enter spoofed buy order in a larger volume at a higher price than the current bid making other investors believe that this increased price is genuine thus expecting other legitimate investors to join. Once the order is matched, the manipulator will withdraw the large spoofing buy order then issue a sell order of large volume of shares at this manipulated price as shown in Fig 2.1 (a). A manipulative spoofing order stays in the grey zone until disclosed, as the orders mentioned in the order book cannot guarantee which of them is real or fake.

Pump and Dump: In the case of pump and dump, the manipulator begins by creating a high demand of a stock using false information [9] leading to its price rises (pumped) and the manipulator sells it (dumped) when sufficient number of orders are added or when the desired bid price is achieved shown in Fig 2.1(b) [16].

Quote Stuffing: Quote stuffing is associated with high frequency trading (HFT), where the manipulator uses high frequency trading algorithms to flood the market by quickly entering and withdrawing a large number of non bona-fide buy and sell orders [11]. This hereby creates a confusion among the traders about the amount of trading activity. This further affects the normal investors in delaying their trades especially the participants that do not use HFT algorithms and consumes a lot of exchange resources [12]. One of such a case study has been presented in Figure 2.1(c) [15]. It can easily be comprehended that the number of trades fell to a lowest level during the time interval (651

seconds ~ 11 mins) when abnormally large amount of quotes/sec (~ 10000 plus) were made [13].

As illustrated in Figures 2.1 (a-c), most price manipulation activities follow a trend of increasing the price of a stock by submitting non-bona fide orders, executing the sell at the manipulated price and then a rapid withdrawal of the buy order leads to a sudden drop in prices as well. As stated before, the implication of manipulation schemes like spoofing trading, ramping and pump and dump can be critical on the market [14]. A detailed representation on Spoofing shown in Figure 2.1 frames up the rise and fall of prices for Demonstrate holdings LLC listed on NYSE in a total span of 1.3 secs [15]. The sale was executed at $101.32, which is around 8 bps (basis points: unit of change, 1bps = 0.01%) up than the current bid price as shown in Figure 2.1(a). Another manipulation case of pump and dump is illustrated by a spike pattern on Westinghouse Air Brake (WAB) Technologies Corp. where the manipulated bid price is moved 8% and reverted to its prior level in tiny time interval of 3 secs as shown in Figure 2.1(b) [16]. A detailed survey report presented in [17] provides an insight into the modelling techniques used in financial data. Along with prediction, a vast number of research studies have been carried out on stock market manipulation detection. Since the financial crisis of 2008, Volatility Index reaching record levels, the flash crash of 2010 [18, 19] and because of the abusive activities, markets have been highly monitored by market analysts, regulatory organisations, and researchers. Due to our focussed unsupervised learning model, much of these schemes were recreated following several cases of manipulation [11, 13, 16, 20-22] before injecting them into the original stock prices.

2.3 Experimental Dataset

The dataset used in this research comprises of thirteen different stocks including Apple, Amazon, Google, Intel Corp and Microsoft for 21st June 2012 and others including Apple, Amazon, Microsoft, Google, Intel Corp, EBAY, Cisco, Netflix, Nvidia, Facebook, SIRI US, QUALCOM and AMD from 12th November 2018. It consists of level 1 tick data of stock price information along with its derivative for 21st June 2012 on NASDAQ Stock Exchange, USA taken

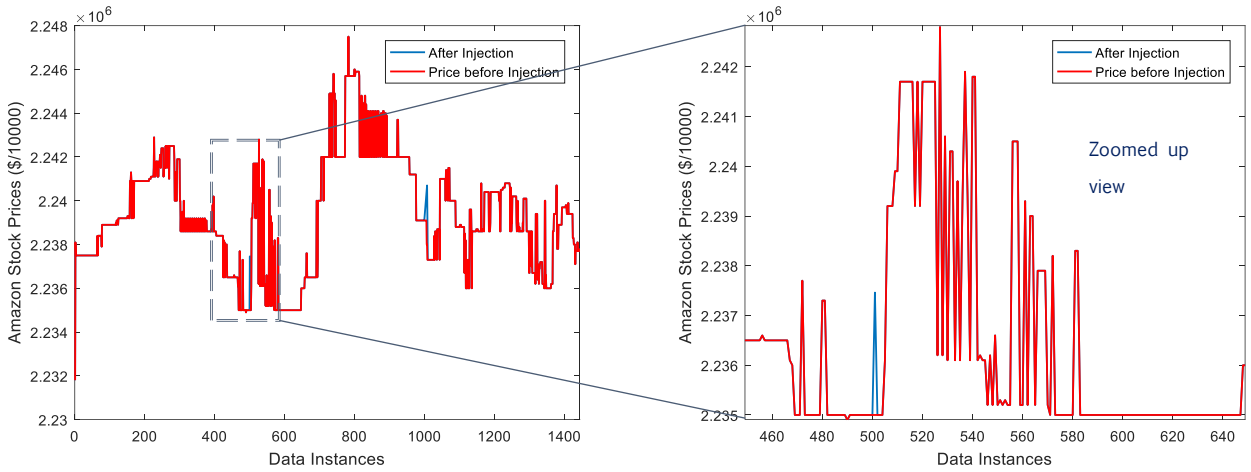


Figure 2-2 Synthetic dataset generated on Amazon stock prices by injecting manipulative instances at random locations.

from the LOBSTER project [48], and the stocks from 12th November 2018 taken from Bloomberg trading platform, Newcastle business school (NBS), Northumbria University, Newcastle, UK. Figure 3.8 shows the variation of the bid price for different stocks beginning from 9:30:00 to 9:30:52 from 21st June 2012. Such a data is selected for its high volatility, high trading frequency and the total number of trades per day (~1 million per stock) that makes it prone to manipulation as aforementioned. The dataset from the LOBSTER project, employed in this research is free from any manipulation activity [9, 27]. Hence a synthetic dataset is prepared by injecting artificially generated anomalies similar to the ones shown in Figure 2.1 onto randomly generated locations making it a combination of normal and manipulative trades. Since the dataset collected from NBS has not been reported to have price manipulation yet; the results calculated from such are not compared with the existing research in stock price manipulation detection. Figure 2.2 shows the normal input Amazon stock prices from LOBSTER dataset with manipulative instances injected into it.

3.3 Regulations

This section describes the market manipulation regulations in two major active markets: United States and United Kingdom. Stock market act as a medium of not just a trading facility associated with the movement of stocks, futures, etc. they also act a medium of communication over one's subjective analysis about the price of a financial instrument. The US Supreme court illustrates this as "Well developed markets are efficient processors of public information. In such

markets, the ‘market price of shares’ will ‘reflect all publicly available information’” [23]. Alternatively, it means that “The idea of a free and open public market is built upon the theory that competing judgments of buyers and sellers as to the fair price of a security brings about a situation where the market price reflects as nearly as possible a just price” [24]. In this context, market manipulation can be understood as an act that hinders such fair operation based on,

- Fictitious financial transactions to manoeuvre the price at an intentional level.
- A sequence of orders/contracts that aims to create a misleading impression.
- Sharing and spreading false information that misleads investors.

a. US Federal Regulations on Market Manipulation

After the market crash of 1929, several provisions related to market manipulation were described and included under US federal law in 1933 Securities Act, 1934 Exchange Act, 1936 Commodity Exchange Act (CEA) observed at different levels within the hierarchy of US federal market regulators shown below. Specifically, sections 9(a) and 10(b) of 1934 Securities Act included key statutes about market manipulation under which different schemes like pump and dump, marking the close and wash trades were covered.

US department of Justice

Jurisdiction: global

Investigatory Arm: Federal Bureau of Investigation (FBI)

US Securities and Exchange Commission

Jurisdiction: OTC and exchange traded securities

Investigatory Arm: Division of Enforcement

Commodity Futures Trading Commission

Jurisdiction: OTC and exchange traded securities

Investigatory Arm: Division of Enforcement

a.1 Section 9(a) (15 USC § 78i(a)) – Prohibition Against Manipulation of Securities Prices

US Congress designed section 9(a) stating "to prevent rigging of the market and to permit operation of the natural law of supply and demand" [25]. Several objectives were defined within this law making it illegal to

- Establish "a false or misleading appearance of active trading in any security other than a government security, or a false or misleading appearance with respect to the market for any such security".
- Engage in a series of transactions that creates "actual or apparent active trading" or raises or depresses prices "for the purpose of inducing the purchase or sale" of a security by others; or
- Knowingly spread false information about a security in order raise or depress its price and thereby induce the purchase or sale of a security by another.

a.2 Layering and Spoofing under Exchange Act '34 and CEA

Spoofing and layering described above, are made unlawful both under the sections 10(b) and Rule 10b-5 of exchange act and also under section 4(c)(1)-(4) (7 USC § 6c(a)(1)-(4)) of CEA for prohibited transactions. Originally proposed in 1936 as prohibiting any trade that cause any financial instrument price to be reported, registered, recorded that is not a true and bona-fide price [26]. Apart from spoofing and layering the act also covered major manipulation schemes such as wash trades, cross trades, and accommodation trades. Congress further expanded section 4(c) following the flash crash of 2010 and more importantly based on the financial crisis of 2008/9. The new section (7 USC §

6c(a)(5)) states that it "shall be unlawful for any person to engage in any trading, practice, or conduct on or subject to the rules of a registered entity that (A) violates bids or offers; (B) demonstrates intentional or reckless disregard for the orderly execution of transactions during the closing period; (C) is, is of the character of, or is commonly known to the trade as, 'spoofing' (bidding or offering with the intent to cancel the bid or offer before execution) [27].

b. UK Regulations on Market Manipulation

Market regulations about manipulation were incorporated into the UK domestic law in 2014 through EU's Market Abuse Regulation (MAR) No 596/2014 that came into effect in 2016. It remained unaffected by the Brexit transition due to the amendments made to Financial Services and Markets Act (FSMA) 2000 and Financial Services Act (FSA) 2012. There are three major regulatory authorities in UK that includes Financial Conduct Authority (FCA) mainly convicting using civil enforcements, Serious Fraud Office (SFO) offers criminal convictions and Office of Gas and Electricity Markets (Ofgem) regulating energy markets.

Financial Conduct Authority

Jurisdiction: OTC and exchange traded instruments (Global)

Investigatory Arm: Enforcement Division

Serious Frauds Office

Jurisdiction: OTC and exchange traded instruments

Investigatory Arm: In-house investigators, Metropolitan Police, National Crime Agency

Office of Gas and Electricity Markets

Jurisdiction: Exchange instruments in UK energy markets

Investigatory Arm: In-house investigators

b.1 EU's Market Abuse Regulation

The regulations embedded in FCA stems from the EU' MAR article 15 makes it unlawful to involve or attempt to engage in market manipulation activities defined in article 12 of the same that includes [28]:

- Entering a transaction which gives, or is likely to give, false or misleading signals as to the supply of, demand for, or price of, a financial instrument, a related spot commodity contract or an auctions product based on emission allowance.
- The placing of orders to a trading venue, including any cancellation or modification thereof, by any available means of trading, including by electronic means, such as algorithmic and high-frequency trading strategies, and which has a manipulative effect.
    - Disrupting or delaying the functioning of the trading system of the trading venue or being likely to do so.
    - Making it more difficult for other persons to identify genuine orders on the trading system of the trading venue or being likely to do so, including by entering orders which result in the overloading or destabilisation of the order book.

According to the rules defined in FSMA § 123, FCA can only impose civil penalties on any person/firm in violation of the MAR article 15. This is due to UK declining to adopt EU's regulation specifying criminal penalties for market manipulation making it restricted to dues to be imposed under civil court.

2.3.1 Regulatory Challenges

The regulatory policies enacted by the statute often face challenges in the face of ever changing technological advancements within the market. Increasing

high frequency trading (HFT) and continuous investments by private firms to thrive in the modern marketplace make the process of detection manipulation extremely slow as the old rules and law need to be updated with sharp and modern financial realities. Although, several initiatives taken by SEC including MIDAS and NEAT [36] tends to tighten the grip of regulation but still lags behind the technological capabilities of private firms/individuals. Regulatory challenges exist in three forms including resources, detection and enforcement. It was reported in 2017, that CFTC lacked enough resources required to regulate the data that they were getting from CME group, a leading commodity and future trading exchange. In terms of detection, marketplaces have become extremely balkanised due to the high frequency transactions and data deluge with the use of some strategies like quote stuffing. Recently, HFT account for nearly 35% of all European equity trading and about 60% of the US equity trading. It is one of the reasons why most of the regulators have now focused on *ex poste* investigation rather than *ex ante* detection. In addition, other marketplaces private stock exchanges also known as *dark pools* facilitate more financial arrangements in terms of liquid instruments when it comes to market manipulation. In the sense that such dark pools, in the current dynamics, are more prone to manipulative or fraudulent behaviour. Due to the above mentioned issues, it becomes imperative for the need of data mining techniques in market manipulation detection as it can help with decision making by learning the normal trends of different financial instruments including market features.

2.3.2 Case Studies: Traditional and New Market Manipulation Schemes

Market manipulation has existed since its very inception along with the regulations to monitor it. However, markets are always populated by both upstanding participants and the market abusers. Following the depression of 2008 and the flash crash of 2010, markets have evolved towards more technological advancements but so have the ‘modes of manipulation’ or the ‘manipulative schemes’ in general terms. This section presents a selection of case studies covering some of the traditional manipulative schemes like pump

and dump and new market manipulation schemes like spoofing and quote stuffing.

- Pump and Dump:

  SEC vs Diversified Corp, 2004: In this landmark case, majority stakeholder (Joseph Radcliffe from Diversified Corp.) in one of thinly traded OTC stock collected millions of facially unrestricted shares and distributed it to several people involved within the scheme. He then pumped up the buy price by bidding against himself at a rate higher than the current market price even when there was no demand for the stock. More upstanding investors followed this stock, when the co-conspirators issued several press releases indicating that this is a good stock to buy. Once the price of the given stock raised by nearly 2000%, the defendants then dumped all the shares. Interestingly the case was perpetrated by the defendants between 1996 and 1999, however bought to justice in 2004 by SEC under section 10 article 10b-5 [37].

  SEC vs Whittemore, 2011: In May 2005 SEC prosecuted David Whittemore and associates on account being involved in a pump and dump scheme under section 10b-5 (previously explained in Chapter 1). As per the complaint, in July 2004, Peter Cahill, one of the perpetrators who acquired a substantial position in an energy company named Triton contacted Whittemore, who is the owner and sole employee of a voicemail service company. On account of a fake service bought from Whittemore, Cahill offered him 594,0000 shares of Triton which he bought back at $142,000. Furthermore, Whittemore broadcasted several false messages about Triton stock misleading the impression about Triton. Prior to this, Triton's last trade was at 32 cents for about 10,000 shares which later shoot up to 97 cents per share. Towards the end, Whittemore and associates made a total profit of $508,085 [38].

  United States vs Delgado, 2017: In May 2017, Damian Delgado was sentenced by a United States Attorney for his alleged involvement in a pump and dump scheme for a total of 84 months. According to the court

statements, from 2009 to 2016 Delgado contrived legitimate investors in a fraudulent scheme by using his associates to hold several positions over a thinly traded OTC stock at pumped up prices. He and his associates further made false statements, press releases and through calls for the same stock to falsely inflate its prices. The companies making those false statements were more or else shell companies run by his team members. Once, the desired price for a given stock is matched, the team dumped all the shares for the same which eventually led to the fall of the stock [39].

- Spoof Trading:

  SEC vs Lek Securities Corp., 2019: In 2019, SEC gained significant progress in a spoofing case against Avalon FA Ltd. in which a US broker Lek Securities conspired with the Avalon to pursue spoofing scheme in US equity markets. As per the complaint, Avalon then purportedly entered non bona-fide orders on one side of the markets, never meant to be executed, to gain support from the normal investors which eventually raised the market prices of the targeted stocks and once the orders were matched, they executed the sale and withdraw the non bona-fide orders. SEC alleged that Avalon used this method nearly thousand times from 2010 and 2016 with a net profit of $28 million USD. Following the trial that finally ended in 2020, Lek was penalised with a total fine of $10 million USD with $5 million in penalty and $4.48 million in disgorgement [20].

  CFTC vs Mirae Asset Daewoo Co. Ltd, 2020: CFTC enforced a civil penalty of $700,000 on the traders at Mirae Asset Daewoo Co. Ltd. which allegedly were involved in the entering large spoofing orders from December 2014 to April 2016 with an intention to subsequently raise the price of a stock on the other side of the order-book. The large orders, intended to be cancelled as soon as the selling price of the stock is matched, influenced many normal investors as a misleading opinion about the actual price of the stock were created. Smaller sell orders were made on the opposite side of the order-book and within seconds of their trade, large orders were cancelled.

2.4 Price Manipulation Detection

Stock price manipulation refers to artificially influence market prices of stocks using illegitimate means. The intention is to affect the current market price of a stock manually, though illegally for potential benefits. Market manipulators tend to influence stock prices using a variety of manipulation schemes, out of which three schemes are the main focus of a generic detection in this thesis. However, to achieve price manipulation detection in a plethora of challenges, already mentioned in Chapter 1, is far from unique. A vast number of empirical studies upon price manipulation have been conducted compared to its detection. However, many of them claimed significant improvements in the detection results either based on certain assumptions in their applied research or using labelled datasets. This makes it easier for the model to learn the anomalous patterns and provide better detection accuracy on the test data. This section provides detailed reviews of some state of the art works in stock price manipulation detection beginning from bio-inspired models to models that combine multiple algorithms.

2.4.1 Bio-Inspired Based Techniques

Most of the bio-inspired based detection methods try to mimic the natural immune system by proposing artificial immune system for price manipulation detection. Artificial Immune Systems (AIS) are computational intelligence techniques inspired by the biological immune system. An AIS trains a set of pattern detectors based on normal data [40]. It assumes or defines an inductive bias (a set of patterns) only for normal data, which also evolves over time (non-stationary data). In [41, 43] Lee and Yang proposed an abnormal transaction detection system in real time. The research proposed to design and develop variables that can act as antibodies by first learning the structure of an invading pathogen i.e., learn the data pattern for manipulative instance in a real time data stream. However, the authors did not mention the targeted anomalous scheme and few parameters involved, making it difficult to replicate the models for comparative analysis. The proposed approach was also not evaluated under an evaluation metric, making it really difficult to assess the performance of the model.

In a similar approach by Wu et al [42], a negative selection algorithm, a basic form of AIS is trained upon abnormal or manipulative data pattern. The authors describe their approach as an adaptation of the original negative selection as they incorporated a measure to avoid manipulation altogether. The proposed algorithm described in the paper forms a string of abnormal feature patterns taken from the manipulative portion of the intra-day tick data. Unlike natural immune system, if the chromosome strings match the incoming protein it is considered as a normal trade or manipulative otherwise, it is the opposite with negative selection. However, the approach lacked experimental evaluations, choice of manipulative trades focussed or even briefly introduced, dataset description and rationale over the comparative significance of negative selection over other AIS models.

Both research works [42, 43] focussed on detecting manipulative trading patterns with stock price data but failed to improve the significance of detection in terms of experimental evaluations, manipulative schemes and even description of the approach. No following research is conducted on market manipulation detection using immune-inspired approaches to the best of our knowledge.

2.4.2 Decomposition Based Mixture Models

A variance outlier based mixture model (VOMM) was proposed by Qi and Wang [44] to detect abnormal patterns in the stock price time series. According to which, a VOMM approach decomposes stock price time series into normal and abnormal components. A residual variance function is defined as the objective which is maximised over a set of conditions with the premise of being an outlier. The research was evaluated over tick price data and claimed significant improvements over models that decomposes stock price employing Gaussian and Generalized Auto-Regressive Conditional Heteroscedasticity (GARCH) distributions. The authors argued that any heuristic selection of stock price distributions e.g., Gaussian, Rayleigh etc. will lead to more false positives in the detection. The research claimed to achieve significant results in detecting abnormal patterns but failed to evaluate the model over manipulative trading. In addition, the approach only considered one variable as the input to the model

with the risk of increasing the complexity of a multi-dimensional distribution. However, other important feature variables including moving average, volatility, rate of change are key factors that can improve the detection capability of a model.

Luo et al [45] leveraged the idea proposed in [44] by considering multi-dimensional features. The authors proposed an approach that takes into account the outliers generated by principle curve algorithm described in VOMM based approach for three individual conditions described. The second step of the approach combines all three groups of outliers using a voting based method and a probability based method. In summary, the approach proposed an outlier detection method in stock price data based on voting based outlier mining on multiple time series (V-BOMM) and a probability based outlier mining on multiple time series (P-BOMM). For a V-BOMM method, outliers were predicted based on the majority voting among the three measures defined whereas for P-BOMM method, a probability based quantitative approach was used, highest probability associated with the outlier is chosen as the final label. The results were finally combined in the next step based on a ranking scheme. Along with the stock prices, daily price range and daily trading amount was also included for the inter-day trading data considered for validation. Although the research claimed significant improvements over the compared ones, but it focussed only on the point based anomaly and did not consider the sequence based anomalies. As is the case with spoofing orders, a manipulator breakdown their orders at different prices ranges at different volumes.

Yang et al. [46] constructed a prediction model for the detection of stock price manipulation activities using logistic regression followed by a factor analysis and primary component analysis (PCA) to reduce the dimensionality of the input data. The primary components computed after PCA were input to logistic regression model and trained against abnormal return as the defined label. The stock price dataset considered in this approach is taken from China Securities and Regulatory Commission (CSRC). However, the evaluation metrics used for results do not justify the detection capability of the approach. More importantly, use of labelled dataset makes the detection model biased towards the considered

dataset that could suffer in terms of accuracy when considered over other datasets.

Cao et al. [47] proposed a novel approach for stock price manipulation including ramping and pump and dump using Adaptive Hidden Markov Model with hidden states as anomalies (AHMMAS). The method claims an improved performance in terms of the area under the ROC curve and the F-measure, for the four features proposed over other classification techniques like One Class SVM (OCSVM) and kNN. This approach is validated on a dataset from the LOBSTER project [48]. Although, this research aimed to provide better detection capability for an anomaly in the financial data, it relied on the assumption that data is generated from a particular distribution and used semi-supervised training for the Hidden Markov Model (HMM) by calling normal and abnormal instances from the GMM distribution. However, this assumption often does not hold true, especially for high dimensional real data sets but could have been justified by using several hypothesis tests which could have been added in the research. Again, the set of derivative features used in this research are not calculated as per the definition rather just as the differential of the variable with time but did not consider the time gap between any two consecutive samples. The approach focused on decomposing the data using Dirichlet Process Gaussian Mixture Model (DPGMM) into different components defining normal and abnormal components and then trained a Markov model upon those components. Furthermore, the research specified the number of decomposition components, which is misleading as the distribution of the normal-abnormal patterns, might overlap with each other, if the specified number is changed.

2.4.3 Clustering Based Algorithms

Palshikar et al. [49] proposed a graph clustering algorithms based method to detect stock price manipulation using collusion sets. It states that many manipulative cases in the stock market involved collusion sets. A collusion set is a group of traders who trade heavily among themselves. This research generated a synthetic database based on probability distributions (rather than using real-world datasets) and collusion sets of different characteristics and sizes

were injected. Furthermore, it also considered the whole dataset rather than dividing it into smaller timestamps which will make the clustering process more robust. Islam et al. [50] tried to improve this work by considering purely circular collusion sets using Markov clustering algorithm but did not address the similar problem of detection under timestamp.

Ferdousi and Maeda [51] applied an unsupervised learning approach called peer group analysis (PGA) to the stock manipulation and claimed to detect cases of manipulation. PGA is a technique to compare similarity among various features as they progress over time and in lieu detect changes in their normal behaviour that leads to market manipulation. However, they did not consider the change of peer groups over time, which decreases the detection probability when some members in the same peer group may gradually exhibit distinct behaviour from that of other members. Kim and Sohn [52] extended this concept and tried to improve the PGA approach by updating the size of the group with time and achieved acceptable detection accuracy (AUC ~ 0.845) but failed to identify the exact location in time of the suspicious activity. Although they tried to generalise the concept of anomaly in financial data rather than detecting individual schemes, a subsequent step should have been added to identify the type of the manipulation activity. Most of the manipulation schemes follow a sequence of patterns rather than a single event that can be identified as an anomalous behaviour, an aspect that is also missing from this approach.

2.4.4 Models With Multiple Algorithms

Diaz et al. [53] analysed and compared the knowledge discovery techniques of data mining such as linear and logistic regression for stock price manipulation. They modelled the returns, liquidity, and volatility as well as the news and events related to the stocks using logistic regression. Although, the authors claim to detect stock price manipulation (inclusive to any specific scheme) using unsupervised learning over market moves like trading volume effects, liquidity and returns as part of a quantitative analysis, no account of specific unsupervised techniques used were mentioned. The authors, however, used intra-day stock data but considered average returns, average volume, and average volatility rather than tick features that again make it difficult to specifically locate

anomalous data. This knowledge gap between the statistical features and detection techniques leads to irregularities in the manipulation models developed and hence is prone to suffer from a higher error rate even for the legitimate trading activity. The authors also trained several supervised classifiers like C5, QUEST and CR&T for the same feature set and achieved higher detection results (Accuracy ~ 93%) but used no proper labelling in terms of the timing instances for manipulative data, as the time frame for manipulation from SEC proceedings was highly vague. Also, a subsequent analysis of the manipulation results was also missing from the work.

Ögüt et al. [54] compared the performance of Probabilistic Neural Networks (PNN) and Support Vector Machines (SVM) with statistical multivariate methods like Discriminant Analysis and Logistic Regression. The dataset from Istanbul stock exchange (ISE) used in this research was labelled for normal and manipulative content making it suitable to employ supervised learning techniques. Results proved that popularly used machine learning techniques like artificial neural network (ANN) and SVM performed better as compared to the statistical multivariate analysis in terms of classification accuracy. In order to further improve the performance of a neural network, Leangarun et al. [55] implemented a two-step method for the calculation of the feature set and then used a feed-forward neural network model for detecting pump and dump and spoofing manipulations. The dataset from the LOBSTER project [48] used by the model is a combination of level 1 and 2 at the depth of the order book consisting of labelled data, normal trades from level 1 and manipulative ones from level 2. The model achieved 88.28% accuracy in the detection of pump and dump case but failed to identify the spoof trading case effectively.

2.4.5 Deep Learning Models

Recently, Leangurun et al. [55] proposed a GAN model for learning the normal trading behaviour of stocks from Stock exchange of Thailand (SET) with the aim of detecting manipulative behaviours as easily differentiated during test phase. The authors implemented LSTM layers for generative modelling while taking random noise as the input and then further passing it along the discriminative model for classification. The authors work can be appreciated in

the sense of including temporal aspects while using LSTM layers, but it simultaneously makes the model unnecessarily computationally expensive when generating normal trading samples from random noise input. This makes the model pretentious as it assumes the normal trading behaviour being gaussian in nature.

Wang et al. [56] claimed significant improvement over the existing approaches using recurrent neural networks (RNN) by leveraging the ensemble model using trade based features along with characteristic features towards price manipulation detection. Based on traditional methods such as feature selection, modelling and prediction upon labelled dataset, the authors trained an RNN model using ensemble learning for detecting manipulation instances. The research is validated upon Shanghai stock exchange, China. Apart from using annotated data, the research also fails to mention the manipulation schemes focused. As mentioned before, using supervised approach makes any model biased towards given stocks and becomes prone to fail given a contemporary model is present.

2.5 Anomaly Detection Models in Time Series

This section presents a detailed review of the application and analysis of anomaly detection methods developed for time series in various industrial issues. It also presents the rationale behind consideration of such methods for market manipulation detection and a comparative analysis with the proposed approaches in the later chapters of this thesis.

It is clearly fair to assume in a scenario where there are sufficient number of normal data instances and rare abnormal instances, anomaly detection rather than classification is an apt choice. Moreover, the existing challenges in market manipulation including the pinpointing of anomalous instances and privacy concerns both by the regulation and trading platforms narrows the anomaly detection to be construed in an unsupervised environment. In addition, the abnormal patterns varying with time and applications create an undefined stretch of features about anomalous data. Anomaly detection aims to identify behaviours that are not consistent with the normal data features [7]. The idea is to construct a model using features captured from normal data and detect

abnormal instances that fall outside the normal decision boundary during the assessment of the model. It is also referred to as novelty detection [57], outlier detection or one-class classification [58].

Several research conducted in the past have accomplished considerable achievements towards stock price manipulation using supervised and unsupervised learning techniques. However, based on the above mentioned issues anomaly detection in unsupervised domain is a seemingly fair choice. Among the massive quantities of data generated in the stock market, only a fraction of its percentage is associated with market manipulation. Out of which many manipulation patterns defined within a manipulative scheme also evolves over time. The research conducted over past cases may not significantly capture the features of the evolved manipulation types and may further lead to multiple false positives if applied. Such reasons effectively contribute to the rationale that market manipulation detection can be treated as anomaly detection that aims to identify manipulative instances that the substantially differs from the ones the model has been exposed during training.

To summarise, this thesis avoids manipulation detection using supervised learning for several reasons; as it makes any model biased towards the data, over-fitting if not enough data instances, large computational time while optimising etc with the obvious challenge of using a labelled manipulative dataset which is difficult to obtain due to confidentiality concerns, being expensive and the possible discrepancy in labelled anomalous instances (no precise information about the time stamps). Following subsections explains the application of anomaly detection techniques in various time series.

2.5.1 One Class Support Vector Machines (OCSVM) Based Approaches.

OCSVM aims to identify data from a class amongst all other classes based on learning from a training dataset that contains objects only from that class. Conventionally, it creates a decision boundary during training and anything falling outside the boundary is considered anomalous. However, many variations of OCSVM includes the application of a kernel parameter that transforms the data onto a higher dimensional space while the anomaly detection works appropriately.

OCSVM finds a lot of application in detecting abnormalities. A time adaptive OCSVM model has been proposed for fault detection in a non-stationary setting [59] given that the variable under investigation decays very slowly. Although, the method claimed to achieve better results but without the substantial definition of slow decay, it is difficult to replicate the model in other applications. In financial environments, OCSVM had been used for credit-risk modelling including risk assessments problems. A variant of OCSVM model was implemented in [9] to detect whether certain financial news is relevant to a given stock. From the dataset used for training, significance in terms of similarities for a given set of stocks were computed. The model once trained, called as model of critical news can determine if a selected news is relevant to a certain stock. A credit risk assessment model was developed [60] to detect the default cases from a set of cases among credit-worthy and manipulated ones. The model was originally trained on a balanced dataset of both up-sampled manipulated cases with normal cases or down-sampled normal cases with manipulated ones. Finally, a rule-based algorithm was used to ensemble the decision boundaries specified by models trained on both the datasets.

2.5.2 K-Nearest Neighbour (K-NN) Based Approaches.

k-NN is one of the most commonly used approach for anomaly detection in supervised domain, however it can be used as part of an unsupervised system. The general idea for anomaly detection in *k*-NN is that data instances close to the neighbours are normal while the ones that goes beyond a certain threshold are treated as anomalies. *k*-NN and its variants have been applied in various applications and claim several advantages either in computational efficiency such as top-n *k*-NN [61] and *k*-NN-local outlier factor (LOF) [62] or using a Euclidean distance metric such as local distance based outlier factor (LDOF) [63].

k-NN finds its applications in different fields including fault detection in semiconductor manufacturing processes [64]. A diffusion map based *k*-NN approach developed for fault detection that tries to reduce the dimensionality of the input dataset in order to improve the data storage efficiency. The approach proposed to infer the intrinsic dimensionality of the dataset using a correlation

dimension technique and then impose a diffusion map analysis to reduce it. Finally, the results were compared to the existing models and claimed to outperform them.

A similar technique based on *k*-NN method was developed to identify the transient anomalies in electromechanical equipment industries. The input dataset consisted of variables necessary in manufacturing. Similarities were computed between every two variables and were treated as anomaly indexes by the model which were learned by the proposed model. The research claimed significant improvement in terms of the detection accuracy. Later, it was also applied to industrial gas processing plants and proved its sustainability in detection and distinguishing transients from noise, oscillations, ramps etc. However, the robustness of the model cannot be guaranteed as the model was not compared with the existing state-of-the-art models.

2.5.3 Clustering Based Approaches

K-means is a process of grouping input data set into clusters with the nearest mean and variance [65] where K is the number of clusters selected. A relevant application of K-means for anomaly detection has been applied in [66] for network intrusion detection. The idea is to first transform the input data flow records into feature datasets. Then generate separate clusters of normal and abnormal samples using a Euclidean distance measure.

Here, the data is partitioned into blocks or cells with its mean calculated using an iterative refinement like the expectation maximization approach in mixture models. The mean of a cluster so formed is also the centroid of the space defined for a given cluster. For the input data, a window of *n* observations is considered and passed on to the K-means clustering. In order to detect anomalous data, once the clustering for the dataset is done, the intra cluster Euclidean distance between each data instance and its centroid is calculated for every cluster using the Mahalanobis distance method. Mahalanobis distance is used because of its utility in calculating the distance as per the transformation along the principal component axis in the cluster space [67]. Along with the intra-cluster distance so calculated, Mahalanobis distance between each cluster centroid and the points that are not clustered is also calculated. A threshold is applied on the distances

so calculated and the data sample exceeding the threshold value is marked as an anomaly. The threshold used here is decided to be as the 90% of the maximum distance calculated within each cluster. A major downside to such an approach is the choice of the value, K which has to be heuristically selected and makes any model biased. However, there are methods that can help in the determining the value of K but were not implemented.

Liu et al [170] proposed a genetic clustering algorithm for anomaly detection in network traffic. It is proposed to detect abnormal data in two-folds of its implementation. First, implement nearest neighbour algorithm to cluster the incoming data and then to implement genetic algorithm to optimise the clustering centres in an attempt to combine the normal clusters and label the non-clustered data as anomalous. The approach was validated over a range of unidimensional datasets but small in size. Given the smaller size, the approach still utilises the associated labels for optimisation which may provide larger false positives when tested over financial tick-data. Recently, Li et al [] implemented a similar scheme to detect anomalies in multi-variate time series data by using particle swarm algorithm to optimise the clustering centres. However, the approach proposes to use fuzzy c-means clustering to group input features of larger size and reduced computations. The approach is also validated on multiple datasets and claimed to achieve the detection accuracy above 98% on all of them.

2.5.4 Mixture Model Based Approaches.

The simplest approach to any anomaly detection problem can be modelling the density distribution of the input dataset and applying a threshold on either the probability density or variance of it. However, it is always challenging to determine the inherent data distribution. Many approaches proposed in the past heuristically assume the underlying distribution to be Gaussian in nature which can lead to several false positives [65]. There has been updates where the dataset is assumed to be a combination of multiple distributions such as Gaussian mixture models which is a mixture of weighted and linear Gaussian distributions.

Bhat and Kumar [68] proposed an option pricing model based on a combination of several distributions, all Gaussian in nature. The model was used for option

pricing for European call options using a Markov tree model and claimed that the log returns density curve can be more accurately estimated by using a mixture of normal distributions. The prediction results were compared with the Black-Scholes model and outperformed it.

Bigdeli et al [69] proposed a noise resilient anomaly detection model using summarization based Gaussian mixture model (SGMM) for clustering incoming data and then implement a collective probabilistic anomaly detection (CPAD) method to distinguish normal and abnormal clusters. The idea is to calculate similarities between test data and the established clusters subject to thresholding. The similarity metric used is Kullback – Leibler divergence. The model was also compared with the *k*-NN-LOF based and SVM based anomaly detection techniques to prove its significance.

Li et al [70] presents a flight safety model with the aim to detect abnormalities during the whole operation of the flight. The approach is validated on flight data that consists of various complex variables along with the flight path, engine configuration, pressure altitude, density altitude etc. It proposes to implement a clustering model using GMM based on normal flight data with a few abnormalities that can be avoided, later compute the Euclidean distance of the test data with the established clusters. Although, GMM and its variants claimed to be effective in estimating the distributions, but the parameters required to compute the distribution such as the number of decomposition components needs to manually update which can lead to false positives.

2.6 Remaining Challenges

It is evident to state that many of the past research using data sets having manipulated samples prosecuted by SEC or synthetically created, false detection rates for many of the proposed approaches have not been evaluated, which also challenges the appropriateness of the features used. Moreover, the success of the existing models was based on specific data sets and the lack of adaptive capability to capture the different manipulative strategies. We summarise the major challenges in designing a detection algorithm for trade-based manipulation.

- Use of labelled datasets, which is difficult to acquire as it is rare and expensive. It further makes the detection model biased towards the given dataset.

- Focus on specific manipulation scheme and the choice of specific parameter values necessary for the detection of the chosen manipulation scheme. This makes the proposed model biased towards a particular manipulation pattern rather diverse and lacks the adaptability towards other manipulation schemes.

- Most of the approaches have focussed on a limited number of stocks listed on a local exchange rather than platforms like NASDAQ and LSE where stock prices are affected on a global scale.

- Overlap among normal and manipulative trades that leads to large amount of false positives i.e., a manipulative price data pattern can look similar to a normal trade unless observed with necessary parameters/contexts.

2.7 Summary

Market manipulation refers to artificially influencing market prices of stocks using illegitimate means. To pursue such aim different manipulation schemes are used by the abusers. As is evident from the literature, there has been only a handful number of article that explored price manipulation in unsupervised domain and only few of them that investigate manipulation under contextual cues. In addition, most of the attempts to generalise model over different manipulation schemes have failed, in the sense that they specifically focus on one of manipulation scheme. This makes any research model biased and adds to the complexity of any regulatory organisation curbing manipulation.

The problem of stock price manipulation detection becomes challenging due to the following issues.

1. Extremely rare annotated real datasets that can help detail the sequence of actions in time series. Besides regulations prohibiting the disclosure

of such datasets, if available, the cost of purchasing such datasets is extremely high.

2. However, the behaviour for various manipulative schemes remains the same individually, the time dependent features differ from each other based on the volatility of every stock.

3. Overlap among normal and manipulative samples that leads to a number of false positives.

4. Evolving nature of manipulative schemes over time.

The above mentioned issues motivate to consider manipulation detection as anomaly detection problem. Evidently, this section covered an exhaustive review of the state-of-the-art manipulation detection models along with anomaly detection approaches used in time series with a rationale about their use over manipulated stock price data.

Chapter 3 presents two novel approaches for stock price manipulation detection by transforming input features into instantaneous frequency components and principal components using Gaussian kernels.

Chapter 3: Manipulation Detection using Decomposition Techniques.

3.1 Introduction

Stock market manipulation creates a false impression of stock prices through some illegitimate means [1]. It not only affects investor's interest in the manipulated stocks but also undermines their confidence in the integrity of the entire market. Several research mentioned in the review section made few attempts in this field using both labelled and unlabelled datasets [6, 7, 71, 72] but failed to acknowledge the rare and expensive labelled dataset, diverse detection model for multiple manipulation schemes and the heuristically assumed values for the model parameters involved in the decision-making process. This chapter focuses on decomposition of input feature set into orthogonal and independent components. The idea is to define a non-linear boundary among independent components whilst preserving the information of the original data. The two major contributions of this work are as follows:

The contribution in the first model is the combination of Empirical Mode Decomposition (EMD) followed by Kernel density estimation (KDE) based clustering for anomaly detection using a selective set of features while detecting two types of manipulation patterns. The rationale behind using EMD is that it is a data-driven approach that does not require a priori the level of decomposition. Moreover, the basis function needed is extracted from analysing the dataset compared to other decomposition methods where it has to be specified [6]. Further application of KDE clustering upon the decomposed components (instantaneous frequencies in this case) helps in grouping them into clusters while fitting a Gaussian distribution without specifying the number of clusters up front [73]. This makes it easier to analyse the data within a cluster because of its small size and better detection of price manipulation can be achieved.

The major advantage of using this approach is its decision-making capability based on analysing the patterns of instantaneous frequency behaviour subjected being an anomaly. Moreover, it also outperforms existing benchmark approaches (unsupervised learning) when comparing the Receiver Operating

Characteristics (ROC) curve and the Area Under the Curve [74] as demonstrated by the experimental results. Another merit of applying this model is that it is not trained for a specific type of price manipulation scheme. That is, the detection is performed without any prior knowledge about the anomalies injected, be it their location in the time series or magnitude.

The second model is based on decomposition of input features using kernel based principal component analysis. It proposes the combination of a distribution modelling approach using kernel techniques and a non-linear transformations technique onto higher dimensions in order to create linear manifolds among data points. For non-linear data analysis, KPCA is used to project the original dataset onto higher dimensions, sorted as per their variances. Once the KPCA forms a non-linear boundary among the transformed data in higher dimensions, the first step of the detection model is implemented.

Although, one of the conventional approaches for calculating such transformed feature vectors aim to compute the reconstruction error and forms isopotential curves as the decision boundaries which is limited by the highly computational complexity [75, 76]. A rather simpler approach is to limit the number of extracted feature vectors (principal components) and to subject them onto the proposed multidimensional kernel density estimation (MKDE) based clustering algorithm for further evaluation in the second step.

The proposed MKDE clustering helps in grouping the data into clusters (only normal trades) without asking the number of clusters up front [76]. The major advantage of using this approach is its decision making capability based on analysing the patterns that are subjected being an anomaly without prior information about the location or the nature of the manipulation and also helps in reducing the total amount of computations. This can be achieved by clustering the data, without asking for the number of clusters upfront using the proposed clustering algorithm, which is now linearly separable due to KPCA transformation and marking the data points left unclustered as anomalies.

A dataset involving thirteen different stocks intraday price information from multiple resources (both UK and US stock exchanges) and three distinct

manipulation schemes are considered for an exhaustive evaluation of the proposed approach. A distinctive comparison of the proposed approach with the existing benchmark approaches and conventional anomaly detection techniques indicates a significant improvement in terms of detection accuracy, F-measure and a substantial fall in the false alarm rates. In order to check the validity of the proposed approach in terms of non-stationarity, stock price data from both UK and US leading stock exchanges are considered.

3.2 Manipulation Detection Models using Decomposition Methods

This section details the stock price manipulation detection using two decomposition based techniques. The first model is based on a technique that decomposes the input feature set into instantaneous frequencies which are further imposed on KDE clustering for univariable data whereas the second model decomposes the input into orthogonal components based on their variance and then clustered using multi-dimensional KDE technique for anomaly detection. Besides the proposed decomposition models, this section also presents a comprehensive assessment of manipulation detection using other decomposition methods,

3.2.1 Dirichlet Process Gaussian Mixture Model Based Manipulation Detection

A mixture model for generating probability distribution is a mixture of multiple distributions. It is a weighted sum of multiple Gaussian distribution functions assigned to different subsets of data whose means, and variances are calculated using expectation maximization technique [77]. For a given multi-dimensional data set, it is assumed that it is drawn from a model having multiple Gaussian distributions. The data is grouped into K clusters for every K component specified and every data instance is assigned to a cluster that maximise components posterior probability. The Dirichlet Process is then employed followed by Gibb's sampling to calculate prior probability for each cluster's component parameters [77] assumed as the likelihood of a given data instance belonging to that cluster.

For a set of input features, the data is first windowed with no overlapping between windows and then grouped into different clusters, their corresponding probability distribution function (PDF) is learned. A threshold value can be usually set that separates the normal and the anomalous regions in the PDFs of each component according to minimum data likelihood value adopted from the industry reference detection algorithm from Smart Group [5]: which has the 99% cumulative distribution cut-off. This means that data values falling in the region above 99.5% and below 0.5% value of cumulative probability are anomalies [78].

3.2.2 Principal Component Analysis Based Manipulation Detection

Principal Component Analysis is usually applied to reduce the number of dimensions of the input data set. It involves the transformation of a highly correlated input data into a set of components, orthogonal to each other. Among these, the first component having maximum variance or latent is the projection of the input data, having multiple dimensions, onto a single dimension. The data points are then further projected onto a new orthogonal dimension but having a lesser variance than the first component and the process is repeated until the stopping criteria is matched. An important property of principal components is that they are uncorrelated i.e., orthogonal to each other and the principal components are arranged in the order of decreasing variances [79]. Here the components having large variance are called major components and the ones smaller are called minor. This categorization is explained below:

Once the data set is projected onto several components: major and minor, anomaly detection approach is implemented. According to which, normalized major components, PC_i (i = 1, 2…p) and minor components PC_j (j = 1, 2…. q) are thresholded and categorized as follows:

$$\text{Manipulative if,} \quad \begin{cases} \sum_{i=1}^{p} \frac{PC_i^2}{\lambda_i} > c_1 \in major\ components \\ \sum_{j=1}^{q} \frac{PC_i^2}{\lambda_j} > c_2 \in minor\ components \end{cases}$$

Normal instance if, $\begin{cases} \sum_{i=1}^{p} \frac{PC_i^2}{\lambda_i} \leq c_1 \in major\ components \\ \sum_{j=1}^{q} \frac{PC_i^2}{\lambda_j} \leq c_2 \in minor\ components \end{cases}$

Where, p are the number of major components, q are number of minor components and λ_i are eigen values.

Considering a heuristically selected window of the normalized components (major and minor) at a time (30 samples), the anomaly detection is performed. The divide between the number of major and minor components is that the top 50% of the variance for the original data set has the major components and the remaining 50% comprises of the minor components [80, 81]. The value of c_1 and c_2 are decided heuristically in the approach [82], but 95% of the maximum value in each principal component is considered as a threshold here.

3.2.3 K–Means Clustering Based Manipulation Detection.

K-means is a process of grouping input data set into clusters with the nearest mean and centroid [65]. Here, the data is partitioned into blocks or cells with its mean calculated using an iterative refinement similar to the expectation maximization approach in mixture models. The mean of a cluster so formed is also the centroid of the space defined for a given cluster. For the input data, a window of 100 observations is considered and passed on to the K means clustering. In order to detect anomalous data, once the clustering for the dataset is done, the intra cluster distance between each data point and its centroid is calculated for every cluster using the Mahalanobis distance method. Mahalanobis distance method is used because of its utility in calculating the distance as per the transformation along the principal component axis in the cluster space [67]. Along with the intra-cluster distance so calculated, Mahalanobis distance between each cluster centroid and the points that are not clustered is also calculated. A threshold is applied on the distances so calculated and the data sample exceeding the threshold value is marked as an anomaly. The threshold used here is decided to be as the 90 % of the maximum distance calculated within each cluster [76].

3.3 Manipulation Detection Model based on Empirical Mode Decomposition (EMD)

The flow of the methodology used in this report is as follows, for a manipulated input time series containing stock prices, some artefacts restrain the detection of an anomaly. In addition, as the time series is non-stationary in nature, its statistical properties like mean and variance for the high frequency components violently evolve with time and the distribution of prices deviates from normality. As the high frequency components of the time series are more prone to the anomalies, wavelet transform is employed to filter out the low frequency components in the signal i.e., only the high frequency components are considered and is used as a feature, $\hat{x}(t)$ [83] where $x(t)$ is the input time series (stock prices). Such a feature is calculated using discrete wavelet transform (DWT) where the input signal is decomposed up to single level into approximate and detail coefficients using a level-4 Symlet wavelet. Approximate coefficients represent low frequency components and detail coefficients represents high frequency components,

$$X_{a,b} = \begin{cases} X_{a,b}, & X_{a,b} \geq \lambda \\ 0, & X_{a,b} < \lambda \end{cases} \tag{3.1}$$

A hard thresholding algorithm is then applied inversely on the detail coefficient, $X_{a,b}$ where a and b are shifting and scaling parameters for the given coefficient and λ is the threshold, so that the detail coefficients outside the threshold are set to zero. This threshold value in this case is calculated using universal threshold estimation method [84]. These filtered components are then reconstructed using Inverse DWT.

The two anomalous patterns that describe the price manipulation are saw tooth and spike patterns as illustrated in figure 1.1 (a) and figure 1.1 (b). The effect of such patterns needs to be captured in the features used. In order to do so, the stock price values, $x(t)$ and a new feature vector $w(t)$, that extracts only the change between two consecutive samples and then amplifies that difference if it exceeds a given threshold are selected as the feature values. Further, gradient of

the price i.e., the rate of change of prices, the gradient of the new feature, $\frac{\partial(w(t))}{\partial t}$ that further magnifies the change are used as a feature set.

A feature set consisting of five individual feature signals as follows:

1. Input time series (Stock Prices), $x(t)$
2. Gradient of the price time series, $\frac{\partial(x(t))}{\partial t}$
3. A new univariate variable feature $w(t)$ explained below,
4. The gradient of the new variable $\frac{\partial(w(t))}{\partial t}$and
5. A signal containing only high frequency components, $\hat{x}(t)$ were considered.

The feature set $w(t)$ is described as follows,

$$s(t) = x(t) - x(t-1) \tag{3.2}$$

$$w(t) = \begin{cases} 3 * s(t), & s(t) > threshold \\ s(t), & s(t) \leq threshold \end{cases} \tag{3.3}$$

Where $x(t)$ is the input time series and $s(t)$ is the difference between two consecutive samples. Typically, a threshold value of 3 bps and a multiplication factor of 3 is selected from literature [84, 134] as the intention here is to amplify a certain abrupt change in price value over two consecutive instances.

3.3.1 Empirical Mode Decomposition (EMD)

EMD is a process of decomposing a time series into components that preserves the characteristics of the varying frequency as that of the original signal and are called intrinsic mode functions (IMFs). These decomposed components are orthogonal to each other and to the original signal, are of the same length as that of the original signal and remains in the time-domain [85]. Since the decomposition is based on the analysis of local time scale of the data and since the obtained components (IMFs) provides instantaneous frequencies as functions of time, it can be applied to non-linear and non-stationary process. An

IMF has same number of maxima and minima throughout the duration of the signal and the mean value within an envelope having maxima and minima will be zero i.e., it will have equal number of positive and negative values within a localized envelope. The process of calculating an IMF is called the sifting process. According to which, first the mean (m_1) of the upper and lower envelope of the original signal is calculated using cubic-spline interpolation method [86]. The difference between x(t) and m_1 is the first component (4a), which should ideally satisfy the conditions for IMF.

$$x(t) - m_1 = s_1 \qquad (3.4a)$$

However, if it does not, the process is repeated now considering the difference as the new signal and further calculations of upper and lower envelope's mean unless the new difference satisfies the condition of being an IMF.

$$s_1 - m_{11} = s_2 \qquad (3.4b)$$

An IMF, so calculated will be the first IMF component, $r(t)$ of the original time series,

$$s_k - m_{1k} = r(t) \qquad (3.4c)$$

Then, the first obtained IMF is separated from the original signal,

$$x(t) - r(t) = x_1(t) \qquad (3.4d)$$

This process is again repeated for $x_j(t)$, until the number of zero-crossings and the number of extrema is the same or almost differ by one. A situation in which the resulting signal becomes mono-component i.e., it has no negative frequency component [85]. The first IMF contains most of the high frequency components, which can be considered as random noise, but is the most interesting feature while tracking down anomaly-effected portions of the signal (high frequency) [87]. So, for all of the proposed five features, one dimensional empirical mode decomposition is applied to each feature and the first IMF of each feature so calculated is preserved and the rest are forsaken. The IMF values will now act as an input to the clustering algorithm via Kernel Density Estimation (KDE)

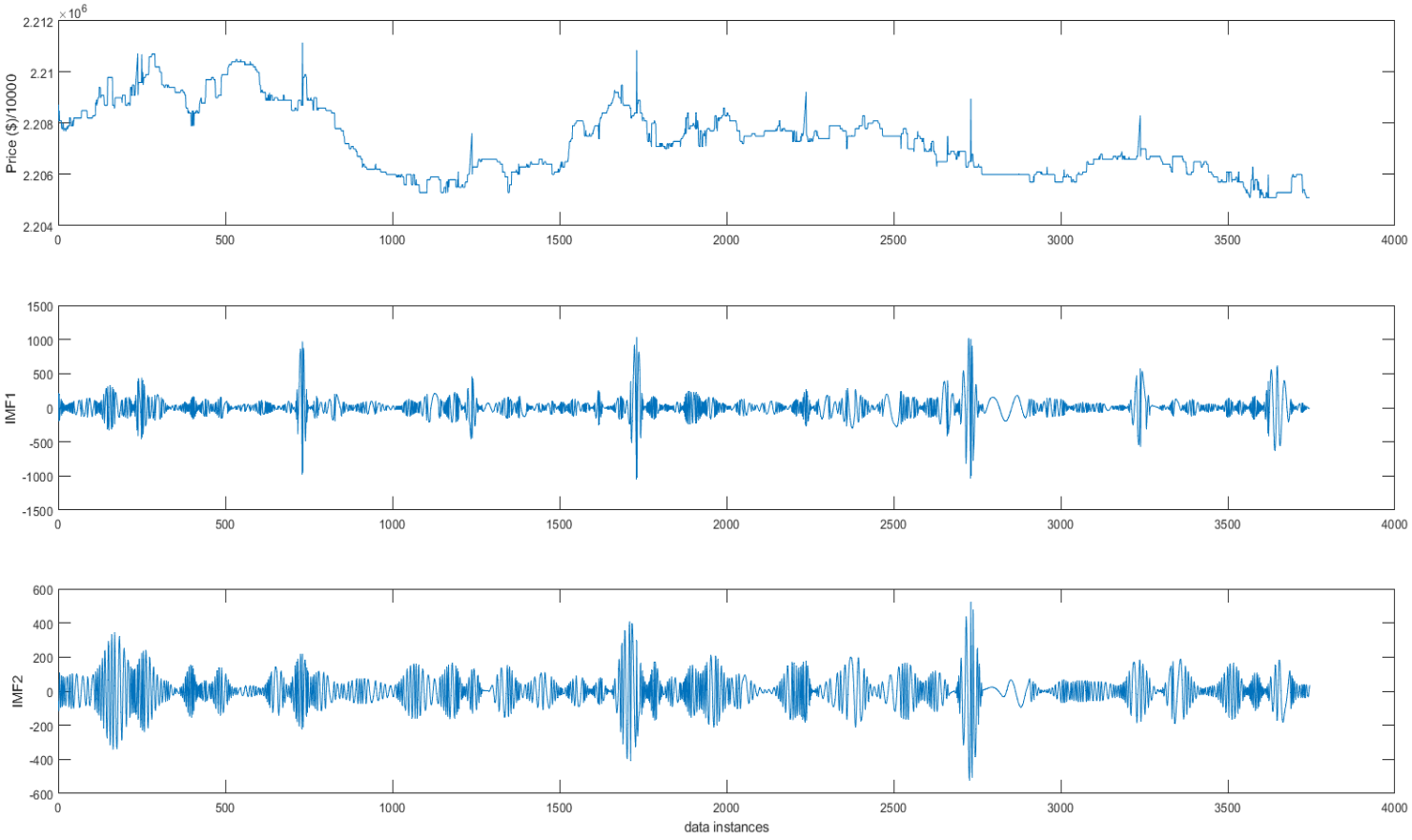


Figure 3-1: Decomposed first and second IMFs along for the input stock price (Amazon Stock)

approach. Figure 3.1 displays the original input data (stock prices) along with first and second IMFs for same.

3.3.2 KDE Clustering Based Anomaly Detection

KDE clustering-based anomaly detection is a modified approach for anomaly detection via non-parametric density estimation for clustering. It has the advantage that it does not require a prior knowledge of the number of clusters. The method suggests calculating a kernel-based density estimation for a set of data samples and cluster them based on the following algorithm [88]. For an input data sample x,

$$x = \{x_1, x_2, x_3 \dots x_n\} \text{ for } x \in IMF_k, k = 5$$

The kernel density estimator used to calculate the probability density $\hat{f}(x)$ is given by,

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{T} K\left(\frac{x - X_i}{h}\right) \tag{3.5}$$

Where, n is the length of the data to be clustered, X_i is the mean of the data x, h is the bandwidth for every cluster, T is the total duration of the data and the kernel function, K that is Gaussian here,

$$K(x) = \frac{1}{2\pi} exp\left(-\frac{x^2}{2}\right) \tag{3.6}$$

Given: $x = \{x_1, x_2, x_3 \ldots x_n\}$ be a univariate vector that is to be clustered and $\alpha \in R$; the bandwidth h is defined as follows,

$$h = 1.06\sigma n^{-1}\alpha \;\; ; for\ Gaussian\ Kernel \tag{3.7}$$

Considering the length of the cluster C to be zero at first, the algorithm suggests that given the original input data set x, bandwidth parameter h is calculated (3.7). Based on which, if the difference between the mean of the sample points x calculated and the sample points is less than h are grouped into one cluster, C_1. Now, the length of the input vector is reduced by the number of data points already clustered. For the remaining data, bandwidth parameter (h) and mean X_i' are again calculated and the same process continues until all the points in the input vector are clustered into j clusters. α is a parameter calculated for the kernel density estimation and whose value is set to 5 as proposed by Silverman [86]. σ is the standard deviation of x, n is the length of x that keeps on changing at every iteration. Now, within each cluster so formed, each cluster has a different density distribution as shown in figure 3.2 (a) and figure 3.2 (b). The values on the horizontal axis are the random values taken over by the feature set $w(t)$ and on the vertical axis, probability density. For each cluster, feature samples having a probability density, calculated in (3.5), less than 0.5% of the maximum is marked as an anomaly. Given the size of the dataset being small, only focussing on uni-dimensional implementation of clustering algorithm, the threshold set on probability density is chosen heuristically. Once every IMF is separately clustered, the common anomalies out of all of the clusters so formed for every IMF are treated as manipulative instances. In this way EMD based

KDE clustering approach suitably identifies the exact location in time, when the manipulation occurred and provides better performance compared with the literature.

A comparison of the results so calculated with some of the existing approaches for unsupervised learning in anomaly detection like Dirichlet process Gaussian Mixture Model, K-Means, Principal Component Analysis is shown in the next

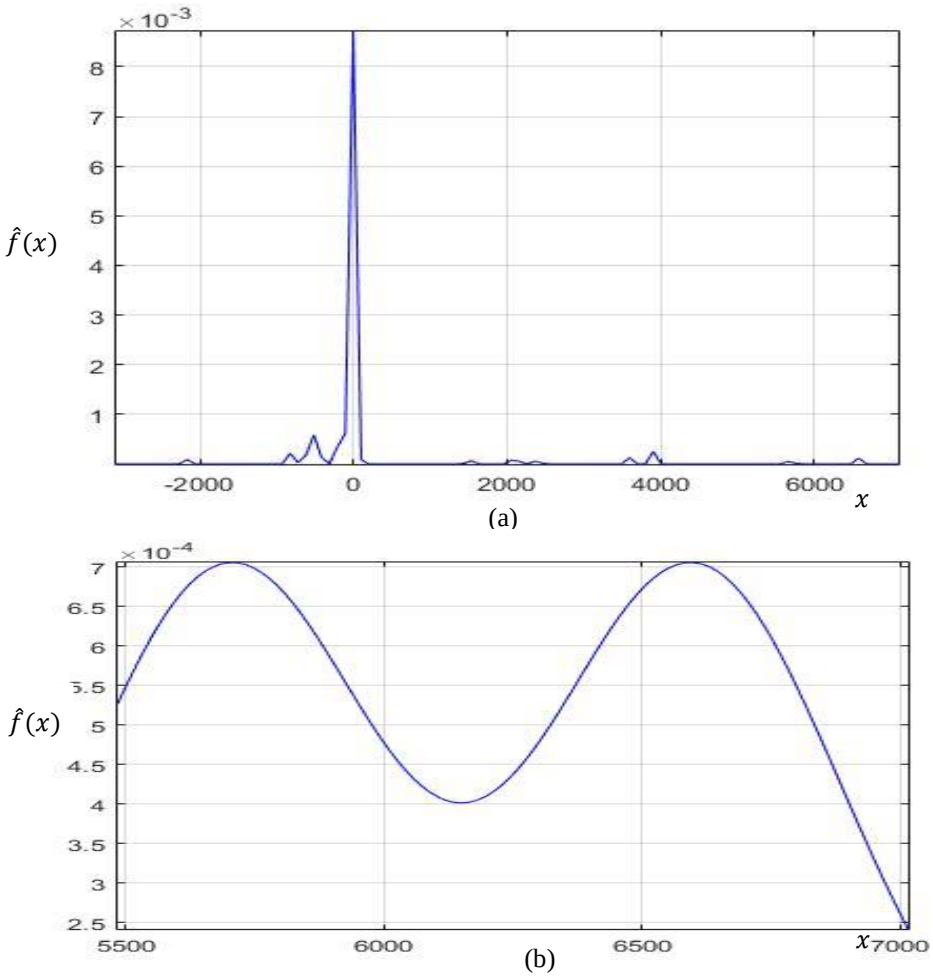
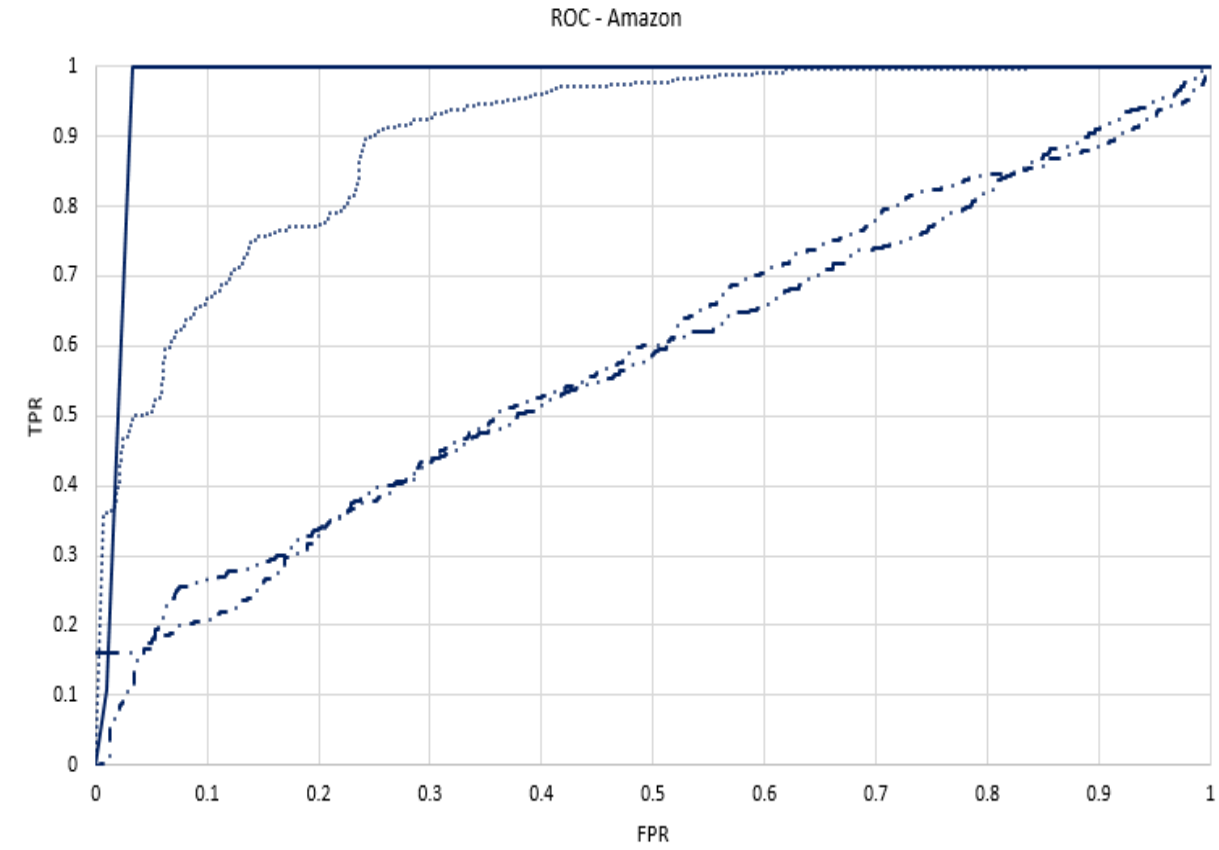
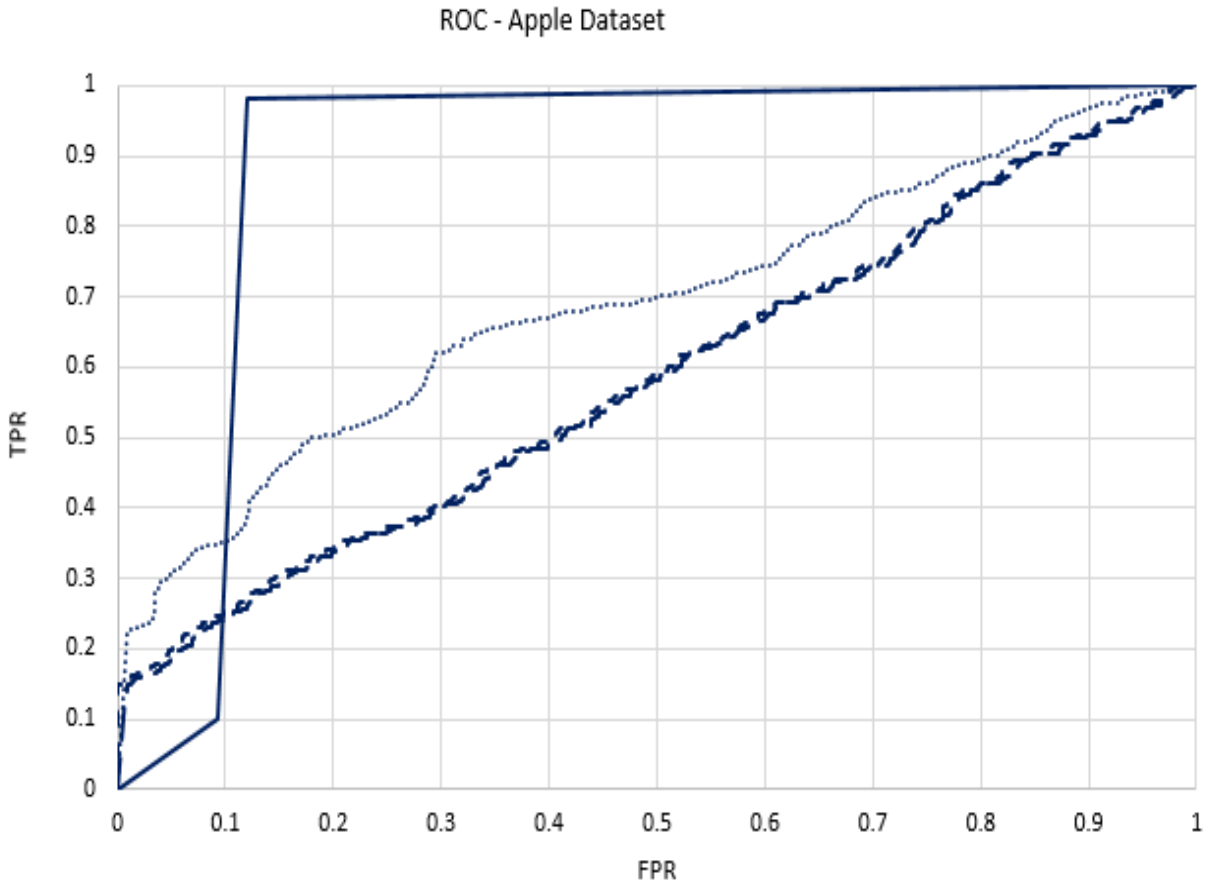
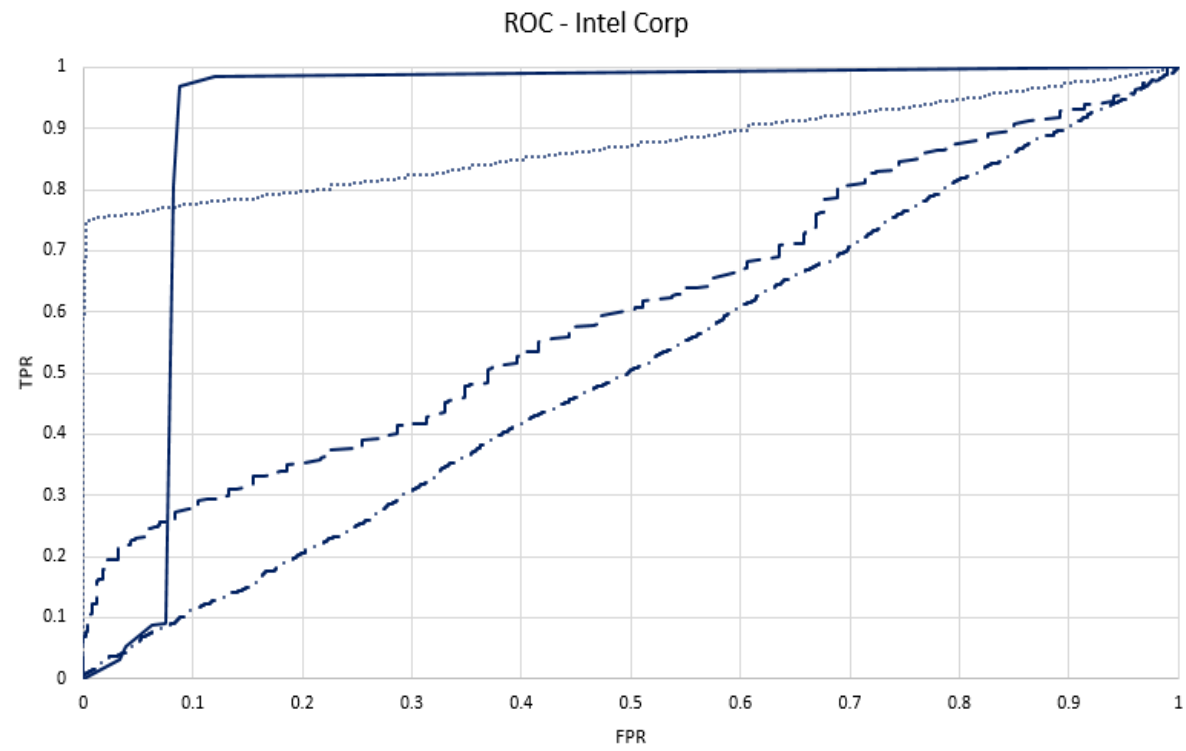


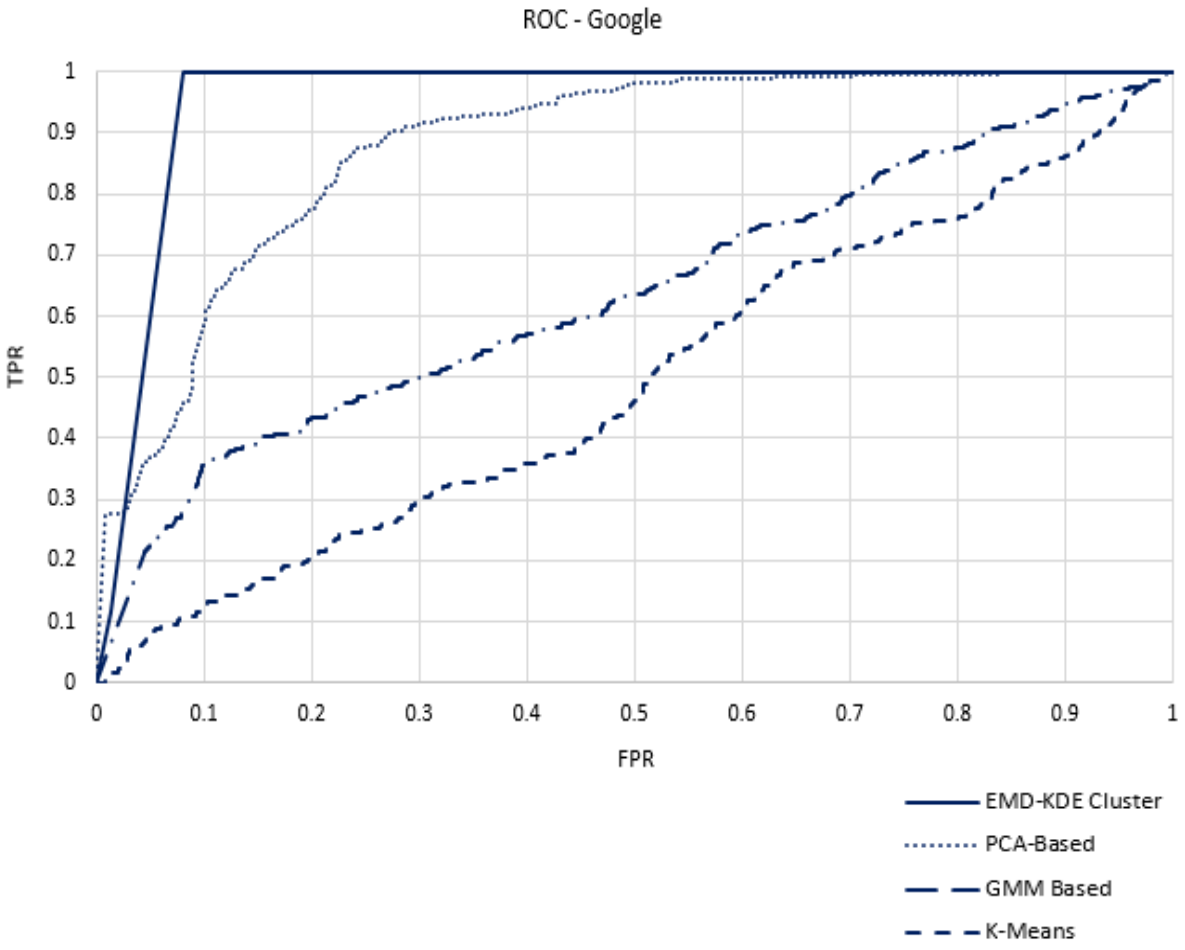
Figure 3-1 (a)&(b): Probability density distribution $\hat{f}(x)$ for instances in different clusters formed using KDE clustering

sections. It should be noted that none of these approaches had been used for stock market manipulation in the literature. The next section explains the dataset used and details of how these existing approaches are implemented considering the above-mentioned features as the input data set for them.


ROC - Amazon
TPR
FPR
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC - Apple Dataset
TPR
FPR
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC - Intel Corp
TPR
FPR
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC - Google
TPR
FPR
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
EMD-KDE Cluster
PCA-Based
GMM Based
K-Means

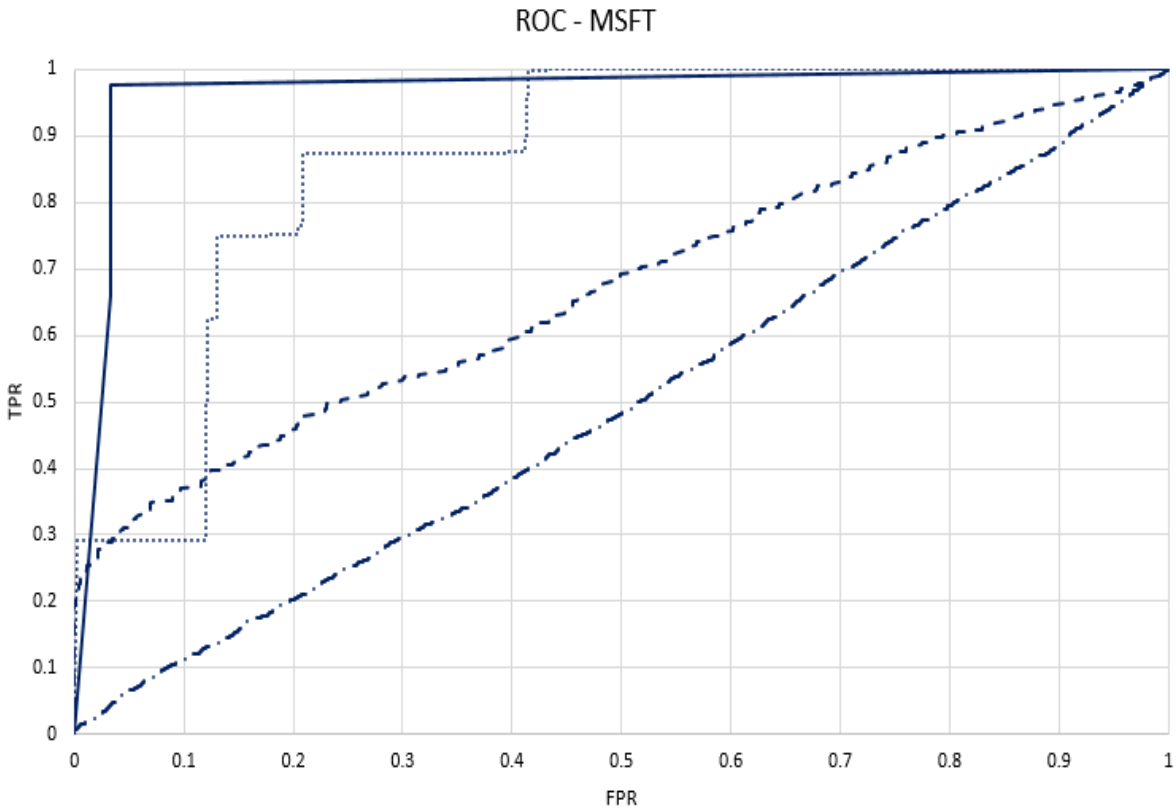


Figure 3-2: ROCs of the five stocks for five different models

3.3.3 Experimental Evaluation

The data set used in this paper is taken from an open source LOBSTER database [48] consisting of Apple, Amazon, Google, Intel Corp, and Microsoft stocks on 12th June 2012. Each stock has a level 1-tick data, and the number of samples varies with different stocks. Based on it, the data set is divided into two groups (Group I & II). Each stock data is reportedly having no manipulation of any sort [89].

An artificial anomalous database is generated in order to test the validity and robustness of the proposed approach. As explained in section I, there are two types of anomalies injected in the original data samples. Type 1 is a synthetic anomalous waveform having a saw-tooth like fall of 7 bps in 95 ms and Type 2 have a rise and then sudden fall of 8% in a time span of 0.1 sec as shown in figure 2.1 (a) & (b). These anomalies are then injected into the corresponding original time series making it a mixture of both normal and anomalous waveforms. To ensure comprehensive assessment of the approach, group I is

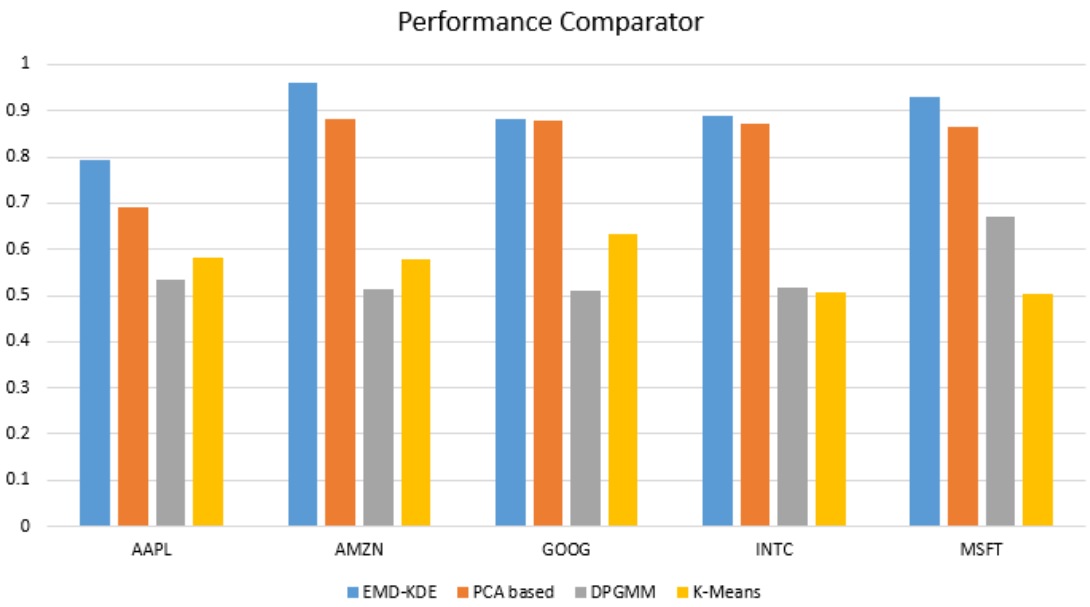


Figure 3-3: Performance comparison based on AUC

injected with 50 anomalies of each type, making a total of 100 anomalies in them and group II's stocks are injected with 200 anomalies of each type making a total of 400 anomalies in it. The place of injection for an anomaly in a time series is performed without taking into account of the time and preceding and succeeding information of the price to make the anomaly detection more challenging. It is even possible that it may be directly followed by a similar waveform, but any prior knowledge of the data set is totally avoided.

For all of the proposed and existing approaches, their performance is evaluated using the Receiver Operating Characteristics (ROC) curve. In order to calculate the ROC, some of its parameters need to be explained first. (i) True Positives represent the total number of normal instances correctly detected as normal, (ii) True Negatives represent the total number of anomalous samples correctly detected as anomalies (iii) False Positives represent the total number of anomalies incorrectly detected as normal instance and (iv) False Negatives represent the total number of normal samples incorrectly detected as anomalies [74]. In this paper, the ROC curve is plotted between True Positive Rate (TPR) and False Positive Rate (FPR), where TPR = TP/ (TP+FN) and FPR = FP/ (FP+TN) are calculated while varying the discriminating threshold for the results obtained from the KDE clustering approach.

The ROC curves for five stocks with 200 and 400 number of anomalies injected are shown in figure 3.2. The tweaking factor in the calculation of a varying TPRs and FPRs is the threshold applied on the output score for each approach for anomaly detection ranging (0, 1). The output score for different approaches are as follows; cumulative probabilities for DPGMM based approach, normalised principal components values for PCA based approach, normalised Euclidean distance measure for K-Means based approach and cumulative probabilities obtained from kernel density estimates. A comparative summary of the AUC values for different techniques is shown in figure 3.3.

The AUC values for EMD based approach and its dominance over other existing approaches clearly indicates the better performance for all the five stocks. As from figure 3.3, it can be shown that the proposed approach retains their advantage in terms of anomaly detection over the existing approaches and can achieve relatively higher values for AUC.

3.3.4 Results and Discussion

The experimental results obtained by the EMD – KDE based approach have shown a significant development in achieving a higher rate of detection of manipulation of two types (Saw tooth & Spike pattern) in the price. These manipulative actions relating to pump & dump, ramping and quote stuffing are carefully selected as they seem to provide similar impact on price of a stock as the ones depicted by these added anomalies. The results also outperformed some of the existing approaches using unsupervised learning for anomaly detection. The robustness of the proposed approach can be explained from the decomposition of the feature sets for a given length of the samples in a window using EMD. The fact that, while considering the cases of price manipulation for pump & dump and quote stuffing, a sudden flip in the prices happens after a long held position of incremental rise in prices. So, the window size for the decomposition of the dataset should be carefully selected taking into account only two components for a given window, one that explains the constant positive or negative slope and the other that represents the sudden drop. Another reason that contributes to its robustness is the threshold value that needs to be set up on

such components or the margin that can probably divide the normal and anomalous boundary.

The EMD based approach followed by KDE clustering for manipulation detection achieved the highest AUC on all of the five stocks and outperformed all of the existing models for unsupervised learning towards anomaly detection. The best AUC is achieved for the Amazon stocks (0.9623) which is about 6.7% higher than the PCA based approach, 65% higher than the K-Means based approach and almost double than under approaches using Dirichlet process GMM and using only raw features. The second best detection is for Microsoft stocks (0.9308) which is 7.5%, 84% and 38.6% higher than using the PCA based approach, the K-Means based detection and the DPGMM based detection, respectively. The lowest performance for the proposed approach is observed with Apple stocks (0.7946) which is still higher by 15%, than the PCA based approach and 36% higher than the K-Means based approach and 48% higher than the DPGMM based approach.

The EMD – KDE clustering based approach for manipulation detection performed variably for some of the data sets and did not attain very high values of AUC as it did for Amazon and Google stocks. This can be attributed to the high variability of the data and the mixing of the anomalies with similar waveforms that created large False Positives (FPs) but still it managed to get AUC values higher than the rest of the existing approaches.

3.4 Manipulation Detection based on Kernel Principal Component Analysis

The idea for anomaly detection is to generate an adept model of the data distribution that can establish a clear manifold between normal and abnormal data instances. The concept of manipulation detection in financial data revolves around the fact that since in a time series, several attributes of anomalous trading transactions overlaps with normal ones [57], proper characterization of manipulation used is required. It makes sense to state that since the stock price data is non-stationary in nature where elementary properties including mean, variance and correlation varies over time. Such variations can be related with

the economics of the market microstructures [90]. Hence, the intention of the proposed model is to derive a set of features linearly independent or uncorrelated from each other when transformed in orthogonal dimensions. It should be kept in mind that since financial data is not sparse in nature [91], a large computational complexity is involved with the conventional approach of orthogonal transformation by calculating the iso-potential curves or surfaces of the reconstruction error. To avoid this, input data is divided into a series of a particular length windows, followed by the proper selection and adaptation of the transformed orthogonal features. A second step of clustering based technique is then applied on to such orthogonal dimensions to identify the abnormal samples. Hence, the methodology of this research follows a two-step approach: Firstly, the input feature set is extracted based on the concept of capturing manipulated patterns and projected onto higher dimensions. Secondly, focus is laid upon to carefully select and adapt the features from the transformed domain. Finally, anomalous stock prices/trades will be detected by using multi-dimensional clustering techniques to cluster normal and abnormal trades.

3.4.1 Feature Characterisation

As for any dataset, the amount of redundancy can be reduced only if relevant information is extracted from it. The dataset used in this research are the stock prices of thirteen different companies operating at NASDAQ and London stock exchange and is gathered from LOBSTER project and from Bloomberg trading platform at Newcastle Business School, Northumbria University, Newcastle upon Tyne. As high frequency components in financial data are more prone to manipulation activities, focus is laid upon extracting relevant features that can capture the effect of high frequencies along with other attributes like derivatives [92] and differences. For time series (stock prices) that consists of synthetically added manipulated samples, denoising techniques [78] is applied using wavelet transform. This is done to filter out the low frequency components in the data and the filtered output is used as a feature, $\hat{x}(t)$ where $x(t)$ is the input time series (stock prices). This is calculated by applying discrete wavelet transform (DWT) on the input data decomposing it up to first level [83] into detail and approximate coefficients. Detail coefficients represent high frequency

components and approximate coefficients represent low frequency components. After, employing (1), Inverse DWT is then applied on the detail coefficients so obtained and approximate coefficients to reconstruct the time series $\hat{x}(t)$. In addition, the volume information along with the features used in the first model based on decomposition completes the feature set used for this model, $F = \{f_1, f_2, f_3, f_4, f_5\}$ where,

1. Input price series, $f_1 = x(t)$.
2. High frequency component, $f_2 = \hat{x}(t)$.
3. Wilson's amplitude [30], $f_3 = w(t)$,

$$s(t) = x(t) - x(t-1)$$

$$w(t) = \begin{cases} 3 * s(t), & s(t) > threshold \\ s(t), & s(t) \leq threshold \end{cases}$$

Here, $s(t)$is the difference between two consecutive samples. Typically, a threshold value of 3 bps is selected.

4. Derivative of the input stock price [92], $f_4 = \frac{\partial x(t)}{\partial t}$.
5. Gradient of the feature set containing high frequency components, $f_5 = \frac{\partial \hat{x}(t)}{\partial t}$.

3.4.2 Kernel Principal Component Analysis (KPCA)

Principal component analysis (PCA) is the orthogonal projection of data into lower dimension linear space such that the variance of the data in each projected dimension is maximized [79]. Despite PCA's ability to project the data onto lower dimensions, its interpretability remains confined by the fact that components generated by standard PCA have added noise and exhibit no meaningful pattern that can be either well represented or visually observed in a linear subspace [81, 93, 94]. We propose the use of kernel PCA in financial data as it is essential here to uncover localised stock price microstructure patterns using non-linear transformations in higher dimensions that could account for main variability in the temporal data. The role of KPCA also becomes crucial

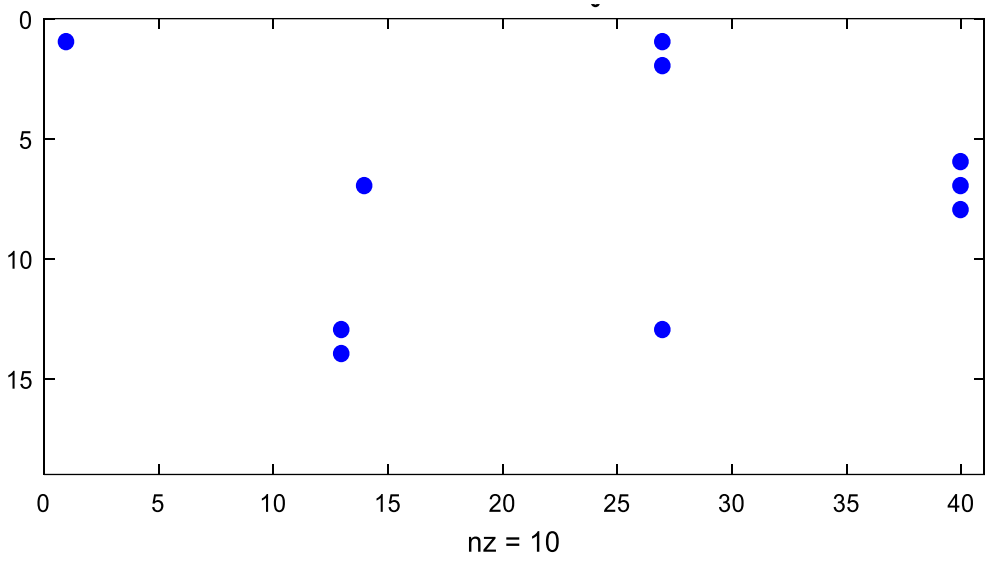


Figure 3-4: Sparse adjacency matrix representation for Apple Stock feature set F_0 from 12:20:05 PM to 12: 21:19 PM on 21st June 2012. 'nz' represents the number of non-zero elements present for that duration. Total number of data instances included = 720

in avoiding the vulnerability of the stock prices during the long held position of stocks that sometimes introduces sparsity in the feature set. Figure 3.4 shows the sparse adjacency matrix representation [90] for such a situation with Apple stock from 12:20:05 PM to 12:21:19 PM on 21st June 2012.

Kernel PCA uses a non-linear transformation of the input data having d dimensions to the m dimensional space ($d << m$) using kernel methods [95]. It does so by mapping the input data points to a higher dimension feature space using kernel trick, forming a linear/non-linear hyperplane, and then reconstructing the data set in the decreasing order of their variances using standard PCA.

An input feature vector, $\boldsymbol{x_i} \in \mathbb{R}^d$ (d=5 and i = 1,2,….N) having N number of input data instances in the feature set $F_o^5 = \{f_o^1, f_o^2, f_o^3, f_o^4, f_o^5\}$ is first transformed to a higher dimension feature space F_t^m (for m dimensions in the mapped space) using a non-linear transformation, $x \rightarrow \varphi(x)$ where φ is a non-linear function. The kernel trick, herein suggests the calculation of extracted features (principal components), covariance matrix (3.8), and subsequently Eigenvectors and Eigenvalues in the transformed domain F_t^m, is possible

without calculating the intractable transformation or mapped data point $\varphi(x_i)$ of a given input data instant, x_i [96].

$$C^{F_t^m} = E_x[(\varphi(x) - E_x[\varphi(x)])(\varphi(x) - E_x[\varphi(x)])']$$

Or
$$C^{F_t^m} = E_x\left[\tilde{\varphi}(x)\left(\tilde{\varphi}(x)\right)'\right] \tag{3.8}$$

Where, $\tilde{\varphi}(x)$ is centred at the origin or a zero mean vector of the transformed/mapped data points. In the feature space F_t^m, Eigenvector V of the

$$V = \sum_{i}^{N} \alpha_i \, \tilde{\varphi}(x_i) \tag{3.9}$$

covariance matrix, $C^{F_t^m}$ can be defined and there are coefficients α_i such that,

Recalling the Eigenvalue and Eigenvector relationship from a standard PCA, we can write,

$$\lambda V = C^{F_t^m}.V \tag{3.10}$$

Note that $C^{F_t^m}.V$ is a dot product and λV is a scalar product where λ being the Eigen value of $C^{F_t^m}$. The length of α can be calculated from the normalisation of Eigenvectors, $V.V^T = 1$ or $\|V\|^2 = 1$. Using (3.8), (3.9) and (3.10), $\|\alpha_i\|^2 = {}^1/_{\lambda_i}$. Now, the Eigenvector V can be calculated by defining a Kernel matrix as the dot product of two feature points in the mapped space F_t^m,

$$K_{i,j} = \tilde{\varphi}(x_i).\tilde{\varphi}\left(x_j\right)' = \tilde{k}(x_i, x_j) \tag{3.11}$$

$\tilde{k}$, can be further defined as the kernel function to calculate the inner product and can be substituted with the most commonly used radial basis function (RBF),

$$\tilde{k}\left(x_i, x_j\right) = exp\left(\frac{-\left\|x_i - x_j\right\|^2}{2\varepsilon^2}\right) \tag{3.12}$$

For $\tilde{\varphi}(x_i), \tilde{\varphi}\left(x_j\right) \in F_t^m$ and ε being the kernel bandwidth parameter, kernel components are amplified given their density estimate falls below 10% of its maximum value (3.13). This is done in order to increase the spread between the

normal and abnormal trading prices in the kernel space, the effect of which can be seen in the transformed feature space (figure 3.5).

$$\tilde{k}(x_i, x_j) = \begin{cases} 3 * K_{i,j} & \mathcal{P}^{K_{i,j}} < 0.1 * \max(\mathcal{P}^{K_{i,j}}) \\ K_{i,j} & \text{otherwise} \end{cases} \quad (3.13)$$

Where $\mathcal{P}^{K_{i,j}}$ is the density estimate of the data points in kernel space. By substituting (3.12) and (3.8) in (3.9), the Eigenvectors and values can be calculated. The projection of new data points onto the mapped Eigenvectors or the principal components in F_t^m is given by, $t_i = \varphi(x_i).V$ which can be further simplified by solving (3.9), (3.10) and (3.11). The objective is to visualize the data points in the kernel space, increase the spread among data points and forward this effect onto the transformed space.

Some of the major constraints in the implementation of KPCA are the choice of RBF kernel parameter and the number of principal components to be used. It is well documented that for anomaly detection using PCA, the number of components extracted l should be such that the cumulative variance must be greater than or equal to 90% of the total variance in the mapped feature set F_t^m [82] that settles down to l = 7. This helps in reducing the uncertainty over the optimal size of the components used and will efficiently reduce the computational complexity of the overall approach. To deal with another constraint about the selection of the kernel parameter, an efficient method is to keep the value of ε fixed for a given input data [97]. The choice of ε is carried out in such a way as it maximizes the amount of variance for the considered number of principal components and minimising the reconstruction error for the projected feature space as proposed in [98]. Figure 3.5 shows the components extracted from KPCA applied to a set of five features for Apple data after normalization (for clear observation only first three components have been shown out of d = 5 in this case). The dataset used, enclosed both normal and anomalous stock prices.

3.4.3 Multi-Dimensional Kernel Density Estimation

Multi-dimensional Kernel Density Estimation (MKDE) clustering based anomaly detection is a modified approach for anomaly detection via non-parametric density estimation for clustering [76]. It has the advantage that it

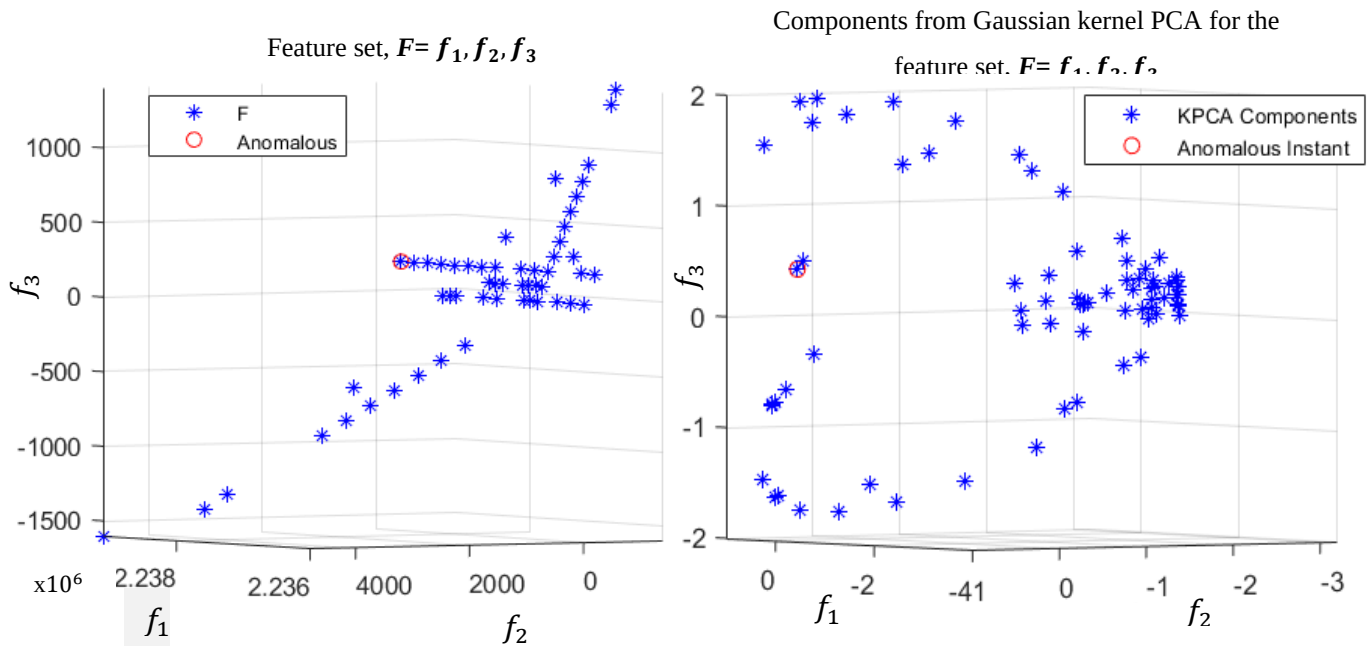


Figure 3-5: Components extracted from input feature space in R^3 using KPCA including normal and anomalous data instances.

does not require a priori knowledge of the number of clusters. The method suggests calculating a kernel based probability density estimation for a set of data samples and cluster them based on the following algorithm [76]. For an input data sample 'F^m',

$$F^m = \{f^1, f^2, \dots f^m\}^T$$

The kernel density estimator used to calculate the probability density $\hat{P}(f)$ is given by

$$\hat{P}(F;h) = \frac{1}{nh}\sum_i K\left(\frac{F - \bar{F}_i}{h}\right) \tag{3.14}$$

Where 'm' is the number of dimensions of the data to be clustered, $\bar{F}_i$ is the mean of i^{th} data sample for a total of n instances, $F_i^m = \{f_i^1, f_i^2, f_i^3 \dots f_i^m\}^T$ and 'h' is the smoothing parameter or bandwidth for m-dimensional input data. The selection of such a smoothing parameter forms an important entity in MKDE estimation. It is seen that for the same dataset, different bandwidth can have serious effects on the results [99]. The kernel function $K(x)$ is calculated via a linear diffusion process [99] leveraging a Gaussian kernel density estimator (3.15) as it lacks the local adaptive behaviour towards outliers [100], resulting

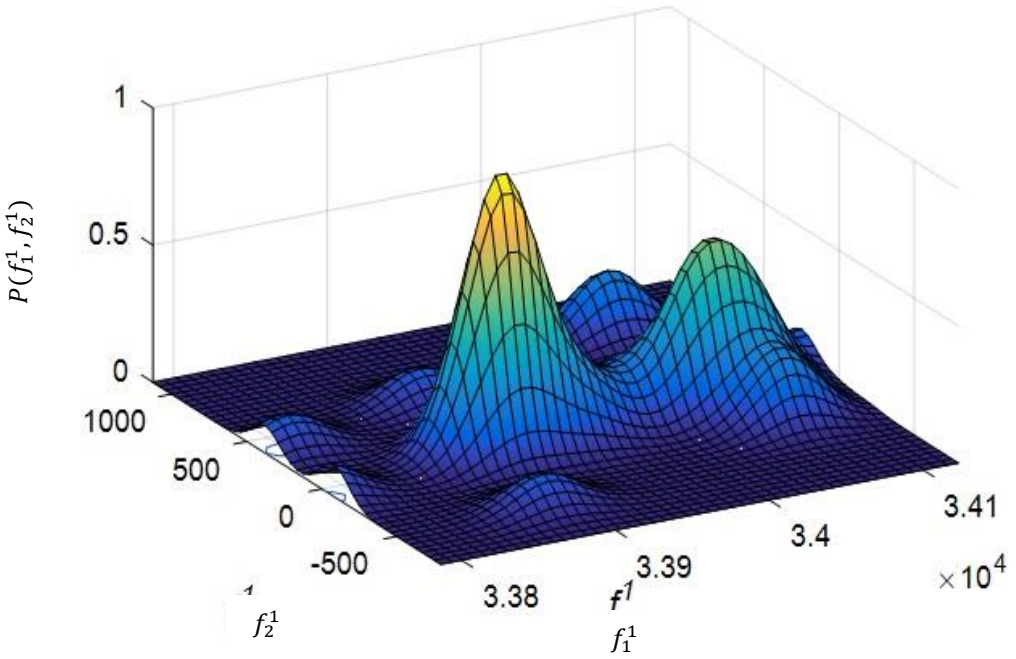


Figure 3-6: Bi-Modal PDF i.e., having two means shown for a subset of features $\{f_1^1, f_2^1\}$ of Apple stock for considering 100 samples from 9:30:01 AM to 9:30:01.05 AM

in misleading bumps and hence flatten peaks and boundary bias. Although such problems can be solved by using high order Gaussian kernels [101] they are unable to provide proper non-negative density estimates [102].

$$K(x) = \left(\frac{1}{2\pi}\right)^{-d/2} exp\left(-\frac{x^T x}{2}\right) \tag{3.15}$$

Given: $x = \{x_1, x_2, \ldots x_q\}^T$ be a $q * m$ size dataset that is to be clustered after using KPCA upon five input original features and $x_i \in \mathbb{R}^m$; the parameterisation of the bandwidth matrix h as a diagonal matrix [103] is optimised again via diffusion estimator in [99] and evaluated using Asymptotically Mean Integrated Square Error (AMISE) [104].

3.4.4 Detection Algorithm

The algorithm for MKDE clustering works by first calculating the kernel density estimate for a given dataset using an adaptive smoothing parameter (h), defined in the previous section also known as bandwidth. For a given set of data instances, if the difference between the mean of the estimate and the data values is less than the bandwidth, the given sample points are grouped into a cluster.

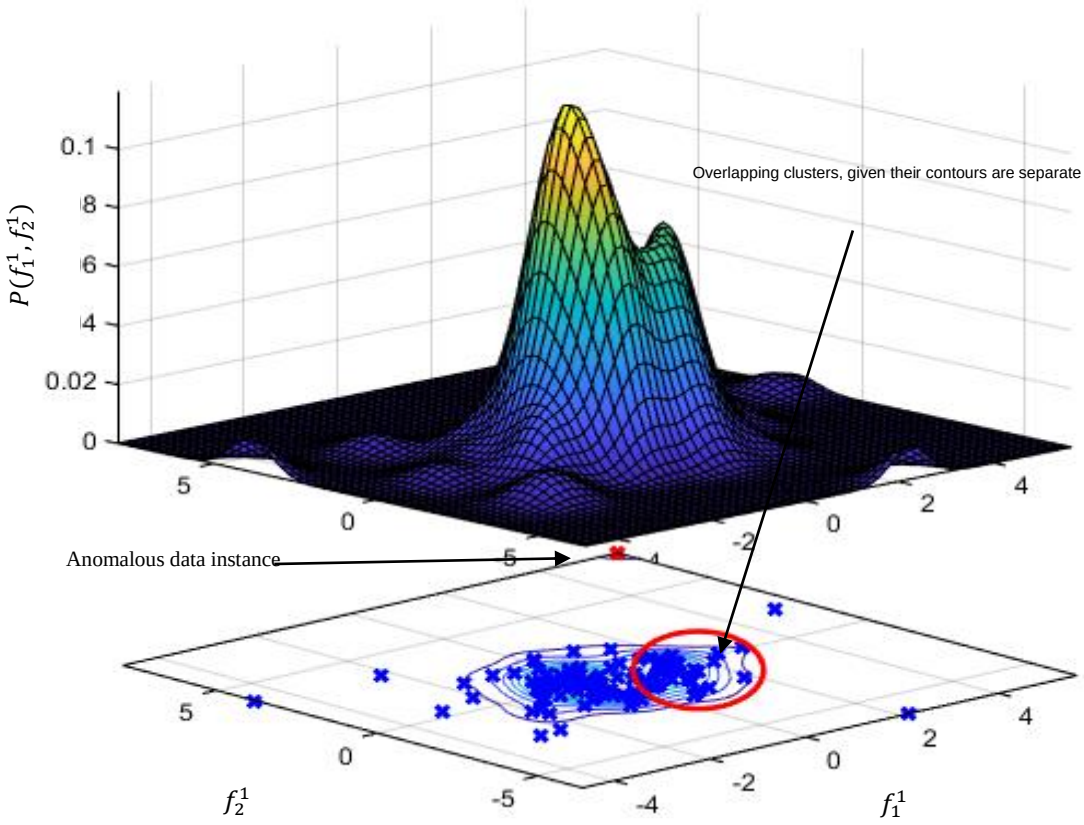


Figure 3-7: Probability distribution using kernel density estimate for a 2-D feature set $\epsilon \{f_1^1, f_2^1\}$ of Apple Stock for 100 data points along with its contour

For the remaining data points having a new estimate, the difference is again calculated, samples having difference less than the bandwidth (of the dataset under consideration) are again grouped into another cluster and the process continues. The algorithm is originally designed to deal with univariate data [76]. As there are seven dimensions extracted from KPCA in the financial feature F_t^7 set used here, seven separate smoothing parameters are obtained for each dimension. The first problem that should be tackled with is the estimated multi-modal PDFs in multi-dimensions. In such cases, when multi-modal PDFs are generated, it is difficult to determine one mean value for the whole set of data points, as there are several means generated for the given distribution, as shown in figure 3.6 (each peak in a multi-modal distribution). Now, each mean is considered separately along with multiple smoothing parameters for each dimension. The implemented algorithm can lead to multiple clusters even within a compact group of data points. In addition, the cluster values for one may overlap with the one adjacent to it (depending upon the value of the

Pseudo Code - Algorithm 1: Stock Price Feature Clustering and Anomaly Detection

1. For any specific stock, extract the feature set $F^5 = \{F_t^1, F_t^2, F_t^3, F_t^4, F_t^5\}$
2. Apply KPCA on the features considered and transform them into, $F_t^7 = \{f_t^1, f_t^2, \ldots . f_t^7\}$
3. For a selected window of samples (F), construct their joint probability distribution $\hat{P}(F, t; h)$; $F, h \in \mathbb{R}^7$ using multi-dimensional KDE approach [54] for bandwidth (h).
4. Construct the MKDE based clustering model for anomaly detection:
   a. Given: $F_t^7 = \{f_t^1, f_t^2, \ldots . f_t^7\}$, for $f^i \in \mathbb{R}$ is the input sample, $t \in [0, T]$.
   b. Set: $\mathcal{C} = \emptyset$, t = length (f); where $\mathcal{C}$ is a cluster.
   c. j = 0;
   d. WHILE $F \neq \emptyset$; % F is the set of data samples to cluster
      j = j + 1; % Iteration counter
      % define the bandwidth h and $\bar{f}$ is the mean(s) location of distribution for the data samples
   e. FOR i=1, 2, 3……t
      IF $|\bar{f} - f| < h$
      $\mathcal{C}_j = \mathcal{C}_j \cup f_i$; % Add the set of data for all the features f_i to the Cluster $\mathcal{C}_j$
      $f = f \backslash f_i$; % Remove the clustered data from the original set
      ENDIF
      ENDFOR
   ENDWHILE
5. In case of Multi-modal PDF as shown in figure 3.6, for the so-called clusters formed,
   FOR $\mathcal{C}_i = \mathcal{C}_1 : \mathcal{C}_j$ % for j, number of clusters
   $d = \min \|\mathcal{C}_i, \mathcal{C} \backslash \mathcal{C}_i\|$;
   IF $d < h$ && $\mathcal{C}_i \cap \mathcal{C} \backslash \mathcal{C}_i = \emptyset$ && $\frac{\mu(\hat{P}_{\mathcal{C}_i})}{\mu(\hat{P}_{\mathcal{C} \backslash \mathcal{C}_i})} < 0.7$ % For every cluster $\mathcal{C}_i$, if it is not overlapping with the rest of the clusters $\mathcal{C} \backslash \mathcal{C}_i$, and if the ratio of their individual PDF at their respective means is less than 0.7 i.e. if the ratio is greater than 70%, they will be treated as separate clusters or else combined into one.
   $\mathcal{C}_i = \mathcal{C}_i \cup \mathcal{C} \backslash \mathcal{C}_i$; % Merge the clusters
   $\mathcal{C} \backslash \mathcal{C}_i = \emptyset$;
   ENDIF
   ENDFOR

bandwidth selected). In such a situation, the clusters that overlap must form a single cluster, but not if one of them is an anomaly as shown in figure 3.7. Such problems are quite common while dealing with anomaly detection in financial data [105]. It is therefore necessary to define a highly illustrious feature that can resolve the two clusters separately and adapt the clustering approach in this case.

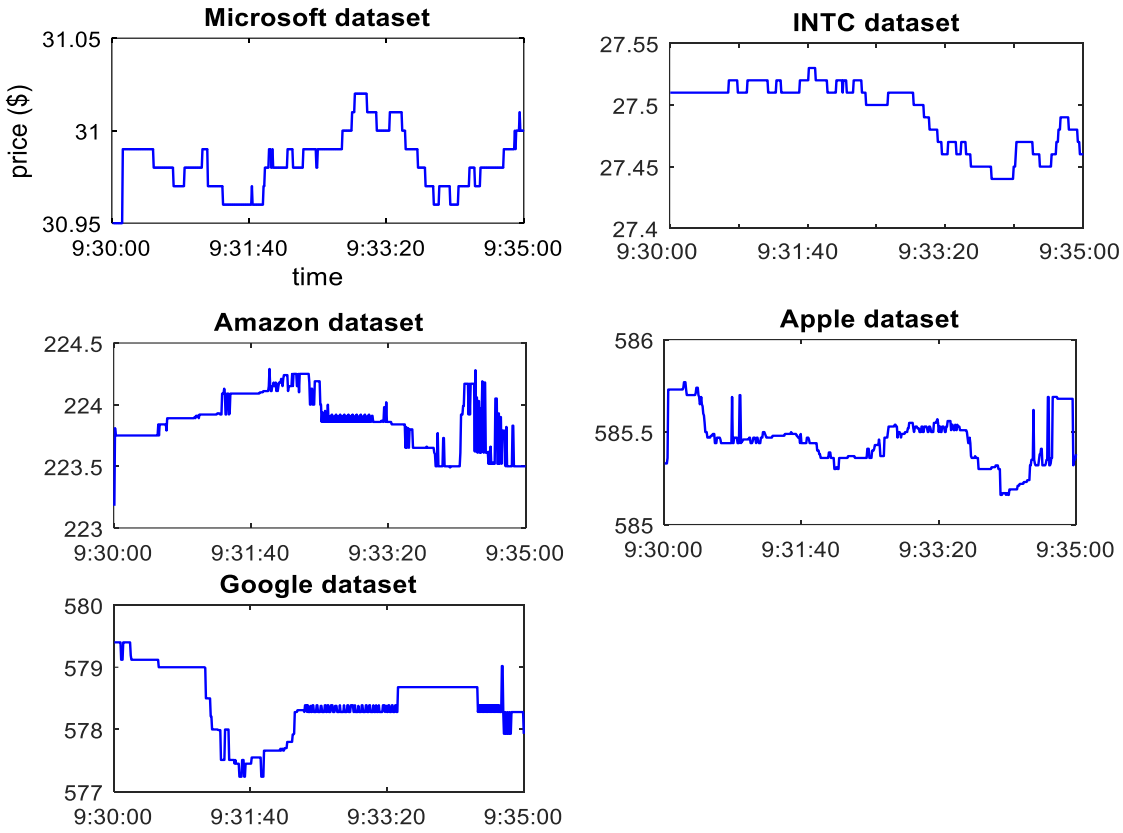


Figure 3-8: Varying bid prices of different stocks from 09:30:00 AM to 09:30:52 AM on 21st June 2012

It should also be noted that there are a number of methods (cubic spline, gradient ascent etc.) to cluster this kind of dataset but very few to distinguish between normal and abnormal data, which makes such a problem of clustering based anomaly detection, a challenging task. Algorithm 1 presents the possible solution to resolve such an issue of the multi-modal distributions.

After formal implementation of the above algorithm, two critical situations may arise in this case. First, if the number of left out data points considered are fairly large and more than one anomalous value in the distribution so obtained (forms a cluster of their own, given their separation, d' is more than the bandwidth). Such a problem can be avoided by using robust features and selecting a proper window size under consideration. Second, if the data instances are sparse as shown in figure 3.4, it is a possibility here that an anomalous trade may be clustered with the normal ones. To address such a situation, KPCA helps in reducing the sparsity of the dataset and is adapted to increase the spread among

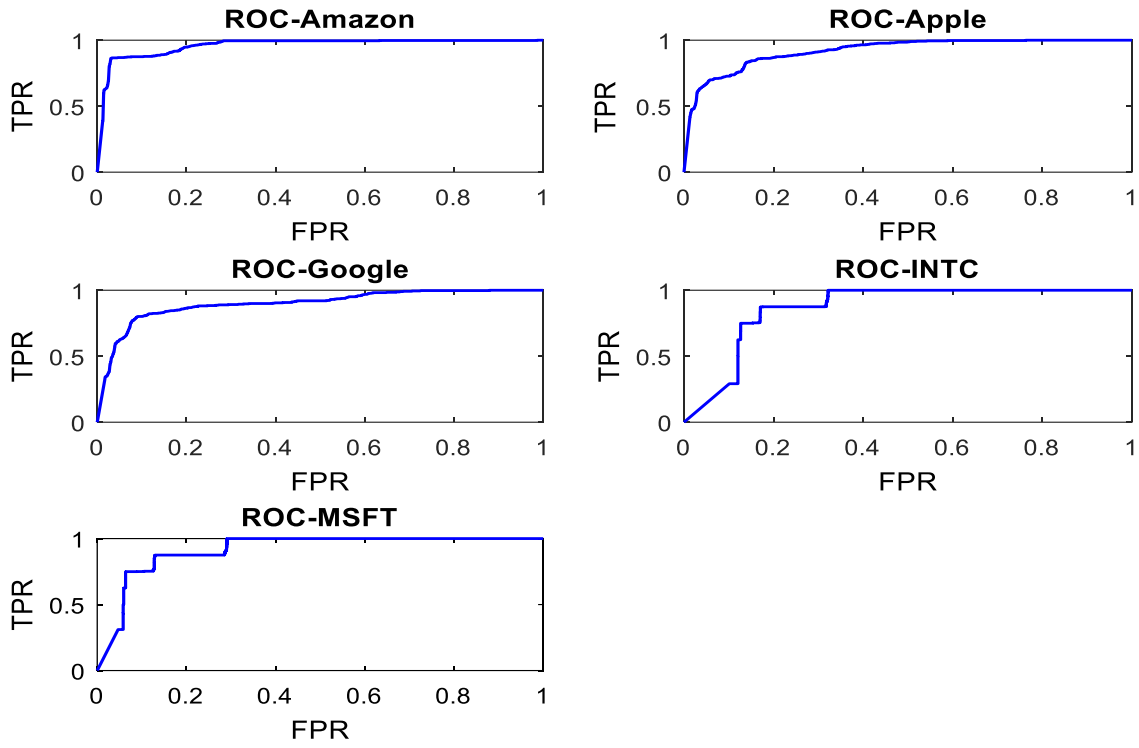


Figure 3-9: ROC curves for five different stocks. Group A (Amazon, Apple, Google) stocks show an identical behaviour in their performance that can be attributed to their smaller data size and the similar amount of anomalies injected whereas Group B (Microsoft, Intel Corp) stocks provide a different (almost similar performance within each other) compared to Group A attributing to the larger injection of anomalies into them

normal and abnormal data instances. The data instances that are not clustered are marked as anomalies. It should also be noted that the above described process is not totally focussed on devising a new clustering method, but rather an approach to narrow down anomaly detection problem.

3.4.5 Experimental Evaluation

The dataset varies in the size of each stock used, based on how they have been categorized into two groups; *Group I* has Apple, Amazon, Google, Intel Corp and Microsoft stocks, each converging itself within the range of 200,000 samples to a bit more than 800,000, for any one form of trade (Ask or Bid) from the LOBSTER project. *Group II* having the stocks taken from the NBS Bloomberg trading platform having more than 1 million trades in Bid/Ask for a given day. Prior to using *group I*, it is made sure no abnormal trading activity was detected [78] and reported by any regulatory organisation for these stocks on the given day [106], marking it as a normal dataset without any manipulation. In order to check the robustness of the proposed approach, three different types

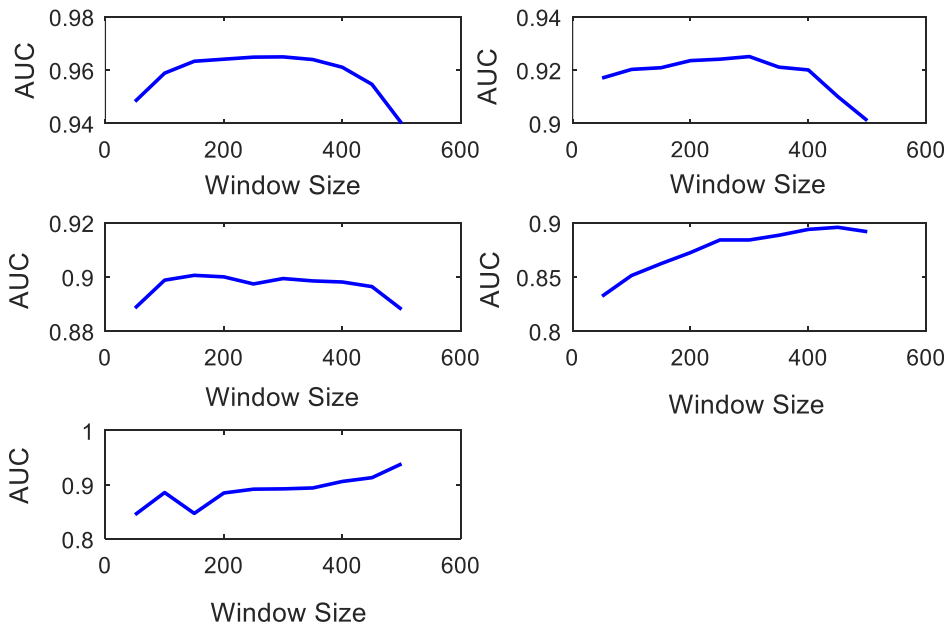


Figure 3-10: Varying AUC values with number of samples fed to multi-dimensional KDE clustering algorithm. Optimal window length for Amazon, Apple and Google can be observed around 300 sample. Intel and Microsoft's AUCs rises with increasing number of samples

of anomalies as shown in figure 1.1 complementary to the real life scenarios are injected [15] into this time series in significant amounts. The number of anomalies injected are also varied based on the size of stocks in each *group*. As the size of *group I* stocks varies considerably, it has been sub-categorised into *Group A* for Apple, Amazon and Google stocks as the average number of trades are limited to 200,000 and *Group B* for Intel Corp and Microsoft stocks having the average number of trades approximately equal to 800,000. Following this premise, *group A & B* stocks are injected with 100 and 200 anomalies/type, respectively making a total of 300 and 600 anomalies per stock with considerable spacing among them. For *group II*, since the size is almost comparable with Group B stocks, 200 anomalies of each type are injected in every stock making it 600 anomalies per stock. Such a configuration of synthetic data is practically accepted as per the business standards [107] and is then tested for the proposed model. To ensure comprehensive assessment of the approach, the detection is performed without a priori information about the location, amplitude and time span of the anomaly injected. It is also possible that a given anomaly will be followed by a rather similar, non-anomalous, waveform in shape but any prior knowledge about any succeeding or preceding samples is

totally avoided. Once the transformed feature vectors $\boldsymbol{F_t^7}$ are obtained from KPCA, they are windowed into a heuristic sample size of 500: $\boldsymbol{F_t^7} = \{f_t^1, f_t^2 \ldots\ldots f_t^7\}$ for $t \in [t, t+500)$ and are then supplied to MKDE clustering algorithm for manipulation detection. Such a condition is further explored, and the detection results are calculated by varying window sizes. Furthermore, to improve the robustness of the approach, the displacement between the added anomalies is varied to check how the model reacts, if two different anomalies are placed closed to each other.

Most of the proposed approaches described in the literature have claimed a considerable amount of detection accuracy in price manipulation. As some of the models [5, 53, 108] focused on the detection of a specific manipulation scheme rather than presenting a general detection model, an adept comparison with such proposed approaches is avoided. However, advance computational models like AHMMAS [78], Naïve Bayes based model [108], Probabilistic Neural Network (PNN) [55] and Peer Group Analysis [51] were selected as the benchmark approaches for the proposed model. An evaluation metric defined

Table 3.1: AUC comparison of proposed approach with benchmark approaches

	KPCA-MKDE	NB	PNN	AHMMAS	PGA
Microsoft	**0.9143**	0.8560	0.7977	0.7336	0.8289

Table 3.2: Comparison of AUC for all five stocks when the manipulation occurs within close vicinity of each other

	Anomalies placed far from each other **(1000ms apart)**	Anomalies placed close to each other **(6ms apart)**	% fall in AUCs
AAPL	0.9206	**0.8773**	4.70
AMZN	0.9602	**0.9539**	0.65
GOOG	0.8996	**0.8923**	0.81
INTC	0.8680	**0.7994**	7.90
MSFT	0.9143	**0.8804**	3.70

for the representation of the results is Receiver Operating Characteristics (ROC) curve and the Area Under its Curve [78, 109-111]. It is also worth mentioning that although ROC curve evaluation is often used with classification approaches trained using labelled data, there are various instances of it being used in totally unsupervised approaches [109-115].

Table 3.3: p-value for the MKDE estimate for first seven principal components calculated

	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7
Amazon	1.22e-08	1.62e-07	0.0001	2.24e-16	3.34e-14	3.56e-41	1.28e-31
Apple	5.91e-18	2.05e-35	7.36e-06	6.62e-18	9.57e-07	4.016e-66	7.52e-09
Google	1.81e-05	8.37e-42	0.0523	0.04301	3.49e-14	1.052e-08	9.078e-11
INTC	4.71e-26	0	2.59e-43	0	0.0068	2.31e-06	0
MSFT	8.59e-11	0.0052	2.05e-93	0	7.49e-22	2.56e-40	0.0037

Table 3.4: AUC comparison of KPCA-MKDE approach with benchmark techniques for anomaly detection

	KPCA-MKDE	kNN	PCA	K means	OCSVM	AHMMAS
Amazon	**0.9602**	0.7982	0.9013	0.5799	0.8933	0.5152
Apple	**0.9206**	0.7926	0.6902	0.5819	0.6603	0.5344
Google	**0.8996**	0.5612	0.7993	0.6328	0.5911	0.5119
INTC	**0.8732**	0.5469	0.868	0.5077	0.697	0.5169
MSFT	**0.9143**	0.5509	0.8655	0.5047	0.6419	0.6711

For the experimental setups described in this section, the following section discusses the obtained results and analyses the pertinence of the model in manipulation detection.

3.4.6 Results and Discussion

The ROC curves for five different stocks having added anomalies (300 and 600 for *Group A* & *B* respectively) are shown in figure 3.9. The tweaking factor that varies TPR and FPR values is the threshold applied on the output score. Here, the output score of the proposed approach is the difference between mean of each cluster and the corresponding sample that ultimately leads to the decision as to whether a sample is manipulative or normal. AUC values in table 3.1 for some of the existing benchmark approaches in stock price manipulation detection are calculated only for Microsoft dataset for 21st June 2012. This is due to the fact that the AUC results for LOBSTER stocks (except for Microsoft) are not available from other models, so only Microsoft stock is reported. However, for some state-of-the-art models like AHMMAS [78], where all the details about the parameters used for the same dataset (and using a combination of different anomalies), the proposed approach is again compared for the rest of the stocks in tables 3.4, 3.5 and 3.6. AUC Comparison for specific manipulation type with the existing state-of-the-art models is made impossible since most of the existing benchmark models have not provided results under specific manipulation type using same stocks and replicating their models is made impossible due to missing parameters values.

In order to check for the robustness of the proposed approach in detecting manipulations when two or more manipulative activities occur within a short duration of itself, the KPCA-MKDE based clustering model is applied on a dataset where the artificial anomalies are placed close to each other. Results are calculated after injecting same three anomalies described before, placed only 6 ms apart from each other. Table 3.2 shows a comparison of AUCs so calculated with the arrangement when they are separated 1000 ms apart on an average. It can be clearly seen from table 3.2 that the fall in AUC values for a situation when the anomalies are placed sufficiently close to each other is not more than 8%, (Intel Corp data [*Group B*]). For stocks like Amazon and Google [*Group A*], there is only a small fall in AUC as <1% change is encountered in both the stocks. The derived inference from such results is that although there is a vast

change between the two situations in terms of spacing among different remains intact.

The class discrimination capability of the principal components from KPCA was assessed using Kruskal-Wallis statistical test as it fits for mutually independent components and avoids the assumption that the underlying datasets are inherently normally distributed [116]. Chi-square is used as a test statistic here to evaluate the performance of the proposed method. In table 3.3, the *p*-values for every individual principal component obtained from KPCA for both normal and manipulative trading instances in *group I* stocks are presented. Smaller *p*-values (less than 0.05) obtained for every principal component proves the statistical significance of the proposed model using KPCA. However, since the significance levels are variable among all the

Table 3.5: F-measure comparison of KPCA-MKDE with other anomaly detection benchmark techniques

	KPCA-MKDE	kNN	PCA	K means	OCSVM	AHMMAS
Amazon	**0.5559**	0.1714	0.1568	0.0484	0.0284	0.0102
Apple	**0.6394**	0.0344	0.0457	0.0708	0.0045	0.0012
Google	**0.5651**	0.135	0.0806	0.0513	0.0196	0.0072
INTC	**0.6034**	0.1014	00085	0.0119	0.0126	0.0175
MSFT	**0.6216**	0.1148	0.0077	0.0141	0.0092	0.0279

Table 3.6: FAR comparison of KPCA-MKDE with other anomaly detection benchmark techniques

	KPCA-MKDE	kNN	PCA	K means	OCSVM	AHMMAS
Amazon	1.22	**0.14**	3.9	7.33	49.54	9.22
Apple	1.07	**0.45**	6.64	1.26	67.8	7.83
Google	1.62	0.68	7.22	9.95	75.2	**0.5**
INTC	0.54	0.23	57.29	**0.02**	59.08	1.15
MSFT	0.71	0.08	49.89	**0.02**	77.48	0.52

components, manipulation detection is not possible by defining a single threshold. The detection ability of the proposed approach between normal and abnormal classes is further evaluated using the following performance metrics: AUC, FAR [47, 53, 108-111] and F-measure [47, 53, 108, 112]. The corresponding values for AUC, F-measure and FAR are summarised and compared with the existing approaches in tables 3.4-3.6 respectively.

Furthermore, the proposed detection model is repeatedly applied over *group I* dataset by varying window sizes to MKDE based clustering. It is performed to reduce the amount of uncertainty over the number of samples to be used as an input to clustering. The evaluation assessment in such a case is again carried out using AUC as a performance measure. Figure 3.10 shows the variability of AUCs with different window sizes. It can be easily inferred from this figure that the AUC values for stocks: Amazon, Apple and Google rise with window sizes initially but falls when the number of samples exceeds a given value (300 samples /window). For Intel and Microsoft stocks, the AUC value continues to increase and is maximum when window size is 500. The average spacing among anomalies in this case is 1000 msec.

A more exhaustive evaluation of the proposed approach is made by including other performance measures like AUC, F-measure and false alarm rate for the same dataset. Tables 3.4, 3.5 and 3.6 mentions a comparative analysis of the KPCA-MKDE based approach using such measures. It can be easily interpreted from the tables that though the AUCs and F-measures for the proposed approach surpassed the existing anomaly detection approaches in unsupervised learning, there are some downsides when it comes to false positives. As mentioned in table 3.6, although some of the existing approaches have better FAR values than the proposed approach, the overall performance can still be appreciated as it provided significant improvement in terms of F-measures and AUC values.

The proposed approach is also applied on *group II* dataset taken from Bloomberg trading platform. A more recent dataset of 11 stocks from 12th November 2018 is also considered from Bloomberg Trading platform in Northumbria Business School (NBS). The stocks considered here are selected

because of their popularity and high trading frequency and the total number of trades (Average number of trades per stock per day ~ 1 million). Figure 3.11 and 3.12 shows the F-measure and the False Alarm Rates (FAR) also called as FPR obtained using the proposed model. As it can be observed, the proposed model proves to be efficient and clearly outperforms the existing models for stock price manipulation detection.

The experimental results obtained using KPCA – MKDE based approach achieves a higher rate of detection of manipulation of three types (Saw tooth, Spike & Square pattern) in stock price. Manipulation schemes like pump and dump, ramping and quote stuffing are carefully modelled by the time series following real life cases reported by SEC [13, 15, 16]. The results also outperformed some of the existing approaches for stock price manipulation detection and also some of the existing benchmark techniques for anomaly detection like PCA [82], K-means [117], kNN [65], OCSVM [65] and AHMMAS [78]. Such a performance can be attributed to the wider information content revealed due to the adaptation of the principal components from KPCA by increasing the spread of the data points. Such a spread is later exploited by MKDE to cluster normal trades. The robustness of the proposed approach can be explained from the decomposition of the feature sets for a given length of the samples in a window using KPCA. In reference to the cases of price manipulation for pump and dump and quote stuffing, a sudden flip in prices after a long held position of incremental rise (within the selected window of data samples) in prices arouses an uncertainty over the sample length for clustering of the dataset. To further explore such an issue, variable window sizes were considered during the experiment with stocks and the results so calculated. While for *Group A* stocks: Amazon, Apple and Google, AUC achieved maximum attainment around an optimal window length of 300 samples per window for MKDE based clustering approach, *Group B* stocks: Amazon and Intel Corp. on the other hand continues to rise even if the window size is increased up to 500. Although, the research cannot contribute in explaining the possible rationale behind such variations in AUCs, further investigation into such a behaviour of the model reveals that Intel and

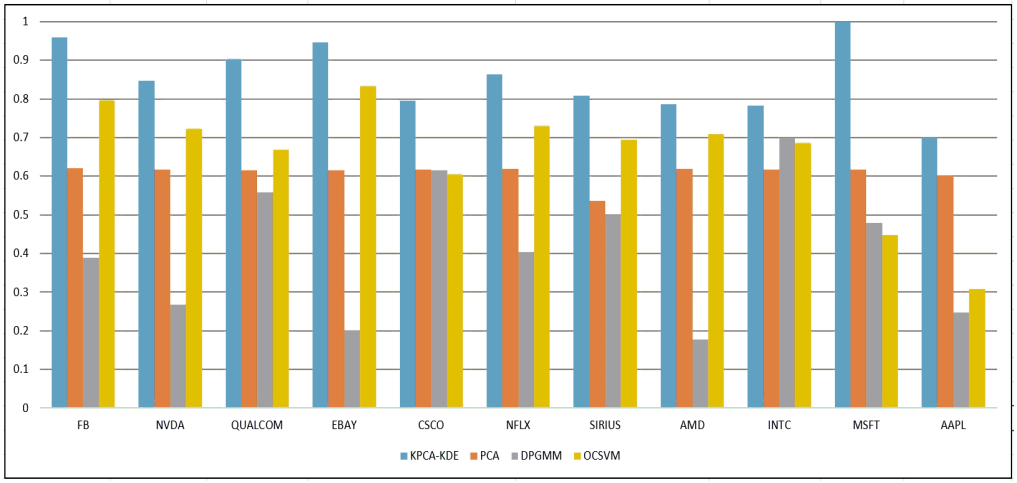


Figure 3-11: F-measure comparison for KPCA-MKDE approach on NBS dataset with existing benchmark anomaly detection approaches

Microsoft stock prices usually sustain a given value (piecewise constant) for a considerable amount of time rather than frequent variation as in *Group A* stocks. Figure 3.8 shows such a behaviour during a same period from 09:30:00 AM to 09:30:52 AM for all the stocks prices. The robustness of the KPCA-MKDE approach is capable to achieve higher detection rates even when several manipulation schemes occur successively. Only a small change (<1% fall) in AUCs is observed for Amazon and Google stocks when the anomalies are placed close to each other (6 ms apart) as compared to when they are sufficiently far apart. Even the least AUC value achieved for Intel data (0.7992) in the former case is still close 0.8, which is considered better performance for a classifier [74]. The proposed model for manipulation detection performed

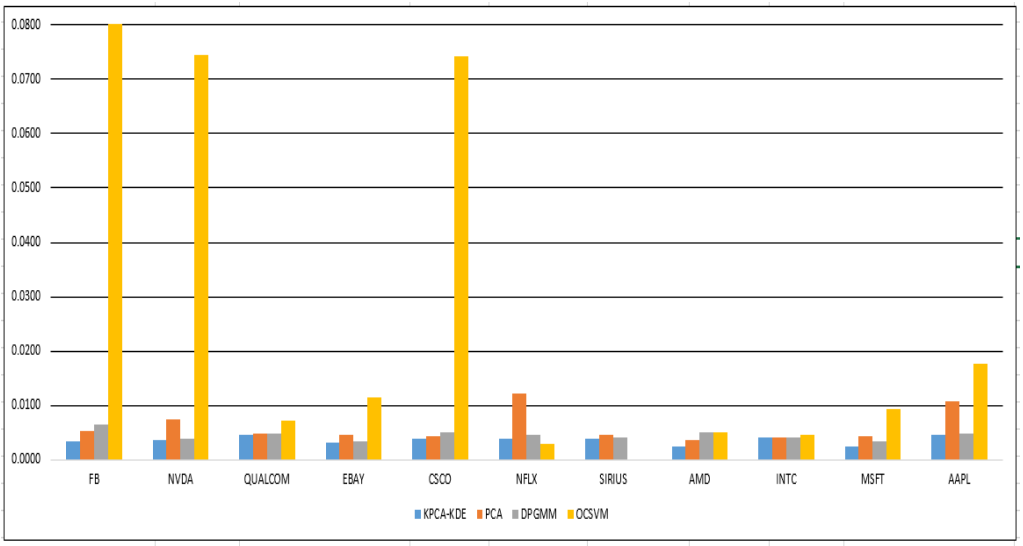


Figure 3-12: FAR comparison for KPCA-MKDE approach on NBS dataset with existing benchmark anomaly detection approaches

variably for some of the data sets and did not attain very high values of AUC as it did for Amazon, Apple and Microsoft stocks. This can be attributed to the high variability of the data and the possible overlapping of the anomalies with similar waveforms that created large False Positives (FPs), nevertheless still managed to get AUC values higher than the rest of the existing approaches. Apart from the AUC results, the values from table 3.5 and 3.6 elaborates the detection outcomes. It is observed that the F-measure values for the proposed approach are not very significant (although comparatively) in values (<0.65). Further investigation into such an issue reveals the degraded detection performance of the approach towards spoofing manipulation schemes. This is due to the drawback of the level-1 tick data being used, as it does not contain the order cancellation information. This is crucially informative as it correlates the price fluctuation (usually assumed high for spoof trading) with the volume change. This information can be included in a future investigation using level-2 order book implying the price volatility associated with the order cancellation may lead to improved F-measure and false alarm rates.

To test the validity and robustness of the proposed algorithm, it is further tested on a recent dataset acquired from Bloomberg trading platform, NBS having 11 different stocks. The F-measure and FAR values generated are shown in figure 3.11 and 3.12. It can easily be interpreted from the figures that the proposed approach outperforms the benchmark anomaly detection techniques like PCA, OCSVM and DPGMM. The major contribution to such a performance is attributed to the ability of the multidimensional KPCA-MKDE algorithm to distinguish between normal and manipulative trades and to the less volatile nature of the stocks included. However, the FAR value, figure 3.12 for Netflix (NFLEX) and SIRIUS stocks is degraded compared to OCSVM but is accompanied with a considerable compensation for the same stocks in terms of F-measure, as can be seen from figure 3.11. It is worth mentioning here that the computational complexity of such an adept approach is $O(m^3)$ to decompose the m-dimensional input data using KPCA using RBF kernel. Upon proper selection of the principal components, the total number of dimensions of the KPCA output have been reduced to l dimensions. Furthermore, it

requires only $O(N.l.\log(N) + 2^j)$ calculations for clustering using Multivariate KDE via diffusion [99] with 'N' samples in a given window, l variables and j clusters.

3.5 Conclusion

Both models proposed in this chapter presents a strong rationale behind their application and achieves a substantial solution to the underlying stock price manipulation detection problem in an unsupervised setting. The first model presented an innovative approach for detecting stock price manipulation based on EMD and KDE clustering. This research envisages two types of manipulations existing in the stock markets, which relates to different categories of price manipulation and strives to work upon their detection using unsupervised learning. To achieve this, a large open source database, which is known for not having any manipulation, is considered. To test the validity of the proposed approach, a very large number of artificially generated anomalies are then injected to it making the input dataset, a mixture of both normal and manipulated instances. Based on the extracted features, instantaneous mode functions (IMFs) were computed using the EMD algorithm. Once IMFs are obtained for a given stock, the dataset is then windowed before passing these to the KDE clustering algorithm for manipulation detection.

KDE clustering algorithm groups the input data set, based on the density estimate defined within a bandwidth parameter, into clusters. A threshold value set up on the value of the pdf for a given cluster separates the normal and anomalous samples. It is found that the proposed model outperforms the existing approaches by a maximum of 84% higher than the AUC for some stocks.

The second model generalised the detection approach to three different types of manipulative schemes namely pump and dump, spoof trading and quote stuffing using a combination of KPCA and multi-dimensional KDE clustering techniques. Principal components were computed, through a non-linear transformation using the kernel trick, upon a set of features extracted from the stock prices. The dataset is then time-windowed before passing the selected components to the MKDE clustering algorithm for manipulation detection. The

MKDE clustering algorithm groups the multivariate input dataset into clusters based on the density estimate defined within a bandwidth parameter. A threshold value set up on the clustered region separates the normal and anomalous trading instances. To test the validity of the proposed model, two real world stock datasets comprising of 16 different datasets (13 stocks in total) were used and augmented using artificially generated manipulation cases. Different performance metrics such as AUC, F-measure and FAR were used to evaluate the performance of the proposed approach. A comparative analysis of the proposed approach results is performed with existing price manipulation detection researches and also with existing unsupervised anomaly detection techniques.

It can be easily observed that the proposed model outperformed existing manipulation detection techniques in terms of improving the AUC, enhancing the F-measure and reducing the false alarm rates while totally avoiding the labelling information. Such an improvement in the results was leveraged from the non-linear decomposition of stock prices using KPCA and further adaptation of the decomposed components. This helped in increasing the gap between the normal and abnormal stock trades in the transformed kernel domain. For further research, the performance of the proposed approach can be evaluated by varying the kernel functions for both KPCA and MKDE. In addition, the inclusion of the volume information for the cancelled orders using level-2 data can be considered for further enhancement of the detection performance.

This chapter highlighted the importance of decomposed independent features in detecting manipulative instances in stock prices. However, reducing false positive rates is a major concern and an attempt to further reduce it is made in chapter 4 by following an example that defends a human body against any abnormalities. Chapter 4 explains the proposed approach to detect stock price manipulation detection using Immune inspired dendritic cell algorithm. It tries to mimic the innate immune system by following danger theory in which any abnormal cell death in traced by dendritic cells.

Chapter 4: Stock Price Manipulation Detection using Bio-inspired Artificial Immune Systems.

4.1 Introduction

Artificial Immune Systems (AIS) are computational intelligence techniques inspired by the biological immune system. An AIS trains a set of pattern detectors based on normal data [40]. It assumes or defines an inductive bias (a set of patterns) only for normal data, which also evolves over time. Dendritic Cell Algorithm (DCA) is an immune response inspired sub-category of AIS. It follows the similar concept of a human body's defence system that tracks, learns and identifies a threat to the body using Danger theory [118]. It can be explained as the immune system categorisation of objects that can cause damage and the objects that cannot. This is independent from the idea It recently gained a lot of popularity in computational analysis of data involving abnormality detection in bio-medical engineering, error detection in robotics and network intrusion detection. In this research, to address the stock price manipulation detection problem, which has sufficient samples of normal trading records, and much fewer examples of manipulative cases, anomaly detection is a suitable technique due to its advantage on problems with the above-mentioned features [8].

Proposed by Greensmith et al [119], DCA mimics the human immunological defence mechanism using the danger theory. It proposes to capture the abnormal trends in a dataset without any labelling information. It is applied on UCI Wisconsin dataset for breast cancer detection and even for Iris classification in [20, 47]. Mokhtar et al [120] implemented a modified DCA on simulated robotic units for online error detection. An anomaly detection model using DCA has been developed by [6] in a real time environment in traditional intrusion detection technologies. In [121] DCA was implemented to detect malicious activities in wireless sensor networks. Alizadeh et al [122] proposed a recent work on sensor fault detection in wind turbine and achieved promising performance using DCA based approach. Chelly et al [123] recently presented a survey that compiles all the works actively done using DCA in anomaly detection.

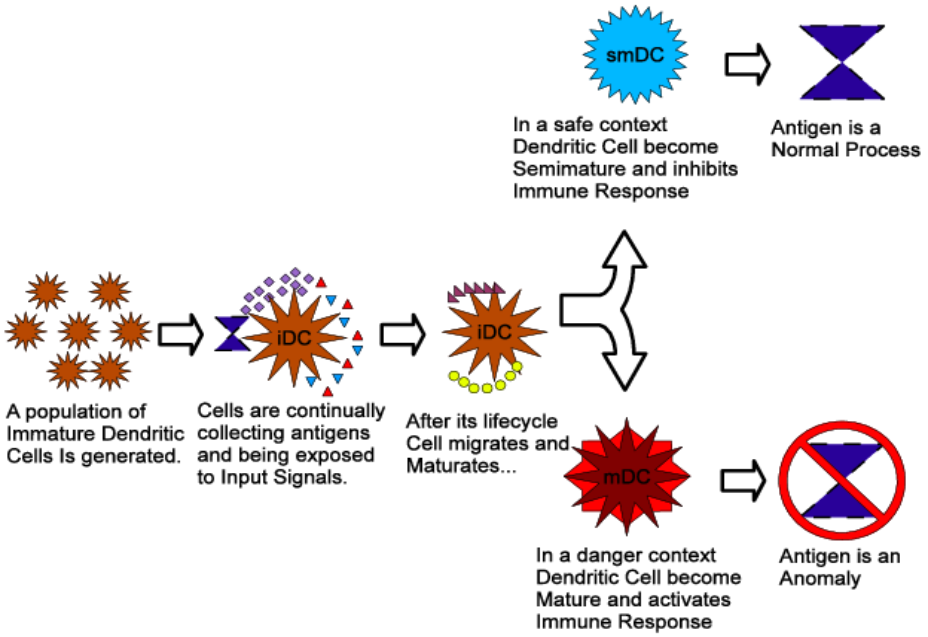


Figure 4-1: Correlation between HIS and AIS [118]

The proposed work introduces a semi-supervised learning method where an artificial immune system-based approach, Dendritic Cell Algorithm (DCA) is altered and followed by Kernel Density Estimation based clustering technique. DCA mimics the natural immune system present within a human body by following the danger theory model. An important advantage of this approach is that the DCA is adapted in scaling down the dimension of the input data set into a set of only three outputs which are then clustered using multi-dimensional KDE clustering. The model avoids the confusion of assigning different threshold parameters as in a conventional DCA and hence makes the detection process, automatic. Another important advantage is that during the pre-processing phase of DCA the proposed method avoids the annotated data in signal categorization. The rationale behind using DCA is the fact that it is a population based anomaly detection approach that does not require a priori supervised training and that the output is generated from a weighted equation that defines the strength of normal (semi-mature) and abnormal (mature) cytokine concentration. It can also be considered as a binary clustering problem [40].

This research introduces a detection model for Stock price manipulation for anomalies that act as the basis of manipulation schemes like Pump and Dump, Gouging or Spoof Trading. A summary of contribution made is as follows; this

research recommends the combination of an altered DCA and kernel density estimation (KDE) based clustering. It performs anomaly detection for a selective set of outputs obtained from DCA while examining two types of manipulation patterns. The uniqueness of this approach from the existing benchmark approaches (unsupervised and supervised learning) for anomaly detection is that it is a data driven approach and avoids the confusion in selecting the thresholds for the parameters calculated. It is also not biased towards a particular type of price manipulation scheme only as any knowledge about the anomalies injected is not provided to the model a priori, neither the location of any anomalous instance nor its magnitude. The distinctiveness of the results obtained can be observed while comparing the area under the ROC curve and the false alarm rate for the experiments performed.

4.2 Artificial Immune System

4.2.1 Negative selection and positive selection algorithms

A human body has formidable line of defences that protect itself from different foreign agents that may represent themselves in the form of mechanical injury, disease causing germs or any other pathogens. This line of defence is called as Immune system that acts as a barrier against any foreign bacteria in the form of an army of cells roving the body ready to detect and defend any attack. A fundamental categorisation of immune system can be made in terms of general and objective description of it i.e., Innate immune system and adaptive immune system. Innate or general immune system is a non-specific immunity, a body develops to ward off any attack. It is a primary line of defence that works irrespective of the type of the anomaly and the objective is prevent the intruder, however without distinguishing from entering the body. Starting from the largest organ in the body, the skin is the first component of the defence followed by the sticky, mucous lining of all the organs. This is still supported by other similar chemical blockades such as the lysozyme in the eyes, stomach acid, genitourinary tract along with flora or microbial community of every other organ that prohibits pathogens attempting to enter.

The next line in the defensive system is the inflammation due to mast cells which releases information in the form of histamine molecules when coming across

any abnormal objects. It is responded by the rush of the blood to the infected region (inflammation) and subsequently followed by the white blood cells or leukocytes that kills the pathogen irrespective of its type. There are multiple types of leukocytes including phagocytes, neutrophils and lymphocytes (T-cells and B-cells). However, neutrophils are the most abundant cells in the human body that wanders around in search of suspicious objects. They also have the capacity to consume 100 pathogens at a time and can also detect body's own cells that behave abruptly such as cancer cells, a trait exploited by the positive/negative selection algorithm of AIS. The last line of defence are the dendritic cells that also roams around in the body and sends a signal in case of a sudden cell death, a phenomenon explained in detail in the next section.

Dendritic cells bridge the gap between the innate and adaptive immune system commonly present in places that are constantly in touch with the outside environment. A response signal from the dendritic cell is in the form of an antigen. Antigens are the molecules present on the surface of the pathogens and are used by immune system to recognise the type of it. Once such a signal is generated or when an infection has already started, a dendritic cell informs the T-cells and a cell mediated response is initiated. At this stage, B-cells can also come into effective rescue depending on the type of infection initiating a humoral immune response. B-cells release antibodies that attaches themselves to the antigen of the pathogen in an attempt to label it and asks for the B and T-memory cells to come and kill the pathogen. T-memory cells are generated once an infection has occurred in the past and a record of it is maintained.

Negative selection algorithm (NSA) [124] is leveraged upon a type of human immune systems (HIS) based on the fact that all newly formed immature T cells in an HIS must go through a process of negative selection in the thymus, where self-reactive T cells binding with self-proteins are removed. As a result, when mature T-cells are discharged into the bloodstream, they can only connect to non-self antigens. The negative selection process in AIS collects a set of self-strings (class in data science) that defines the monitored system's normal state before generating a set of detectors that only identify nonself strings. This detector set is used to track anomalous data changes in the system in order to

classify them as self or non-self. Positive selection is similar to negative selection with the only difference in which self-string detectors are evolved rather than non-self-string detectors. Despite the benefits of fault tolerance, adaptation, and self-monitoring, the above mentioned types of AIS algorithms have shown certain shortcomings in identifying novel attacks, making them unsuitable for intrusion detection systems as network traffics change their behaviour over time [125]. The above mentioned two algorithms are one of the primitive ones introduced types of AIS and the idea is confined within the definition of self and non-self i.e., any self entity in an immune system can detect an entity which is non-self.

AIS has been extensively used in various anomaly detection applications including intrusion detection [171], experimental datasets like Mackey-Glass time series [40], fault detection [172], spam email detection [173], remote data auditing [174] etc. Recently, Hosseini et al [171] proposed a novel approach for intrusion detection using a combination of negative selection and several classification algorithms to improve the detection accuracy and reduce computations in time. It proposes to use NSA as a feature selection step to prepare the training dataset as only including features that have a higher correlation with a target normal vector beyond a threshold and reject others. Yang et al [175] improved the approach by optimising the parameters in negative/positive selection algorithm using evolutionary algorithms, however making the process slightly computationally expensive but effectively improving the detection rates.

It should be noted here that both negative and positive selection algorithms are supervised methods and rely on the data annotations. Most of existing applications of these approaches that focus on anomaly detection claims significant detection rates and are further improved when supported with some optimisation methods. Besides relying on the data labelling, none of the proposed approaches attempts to reduce false positives. With the development in immunology, a theory that does not support self/non-self classification was introduced in 1994 as danger theory [118]. As mentioned before, it relied on the concept of a cell death due to a pathogen attack rather than its presence. It can

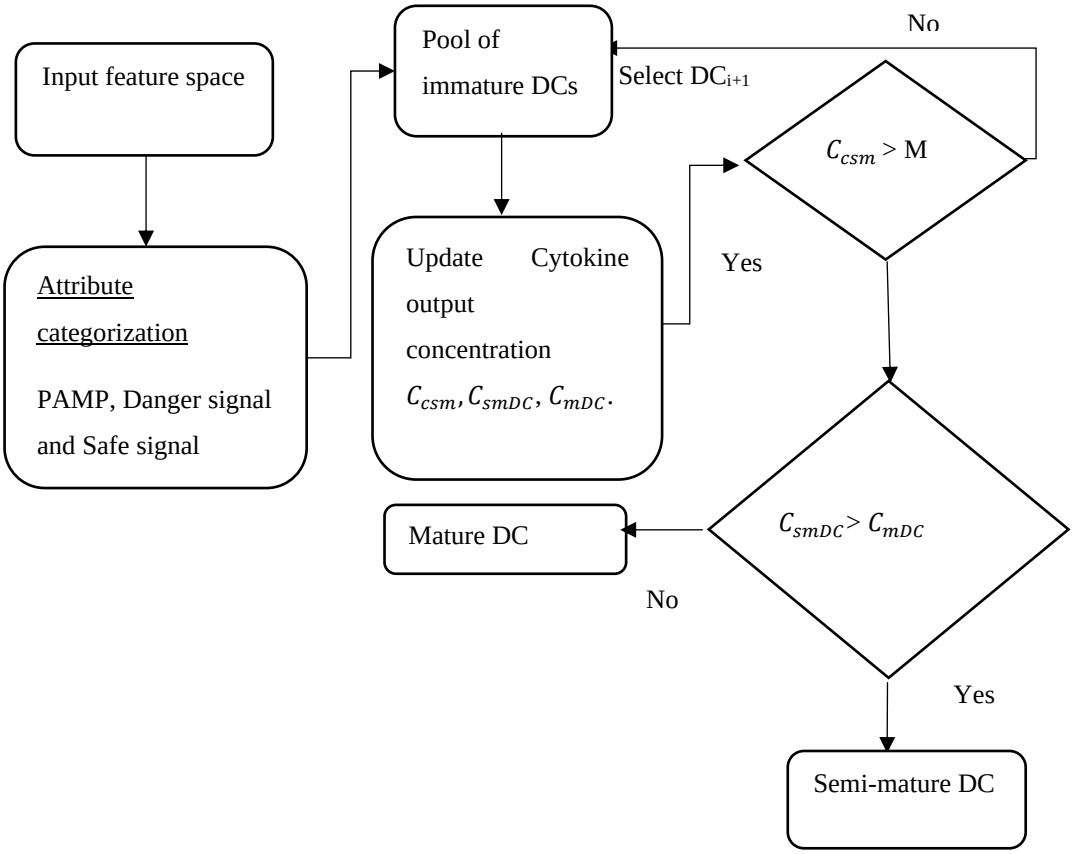


Figure 4-2: Operational Flow of Original Dendritic Cell Algorithm

be translated into data analytics as an approach that can focus on false positive detection and elimination. The next section explains the danger theory supported by dendritic cells and provides an insight of how it can be used for stock price manipulation detection.

4.2.2 Dendritic cell algorithm based stock price manipulation detection.

Our body consists of several lines of defense that become active when attacked by a foreign agent in the form of a virus, bacterial infection or an injury. These foreign agents are called antigens and the guards that counter their effect are anti-bodies. DCA in computational data science mimics similar traits of the innate immune system (IS) following the concept of danger theory. It states that the IS reacts to any danger to the human body, not based on the detection of intruder cells but on the death of a natural cell. Dendritic Cells (DCs) play a pivotal role in the innate IS as well as in initiating adaptive immune responses. DCs are responsible for early detection of any foreign invader through processing input signals along with the antigens. The processing of the input

Table 4.1: Weights for the DCA output cytokine function

Concentration of → **Weights of different categories of signals ↓**	Co-stimulation molecules (C_{csmDC})	Semi-mature cytokines (C_{smDC})	Mature cytokines (C_{mDC})
W_P (for PAMP signals C_P)	2	1	2
W_D (for Danger signals C_D)	0	0	2
W_S (for Safe signals C_S)	2	1	-3.9

signals is only sufficient for determining whether a tissue compartment (dendritic cell) is currently under attack or not. Figure 4.2 depicts the mechanism about its flow of operation.

It is extremely relevant first to have an understanding of some of the basics of the terms used in Fig 4.2. In danger theory, the death of a cell is defined in its environmental context and the immune system reacts to it appropriately. If a cell dies normally, a process called apoptosis, no natural immune system reaction is needed and if the death is under abnormal circumstances like injury, cell disease or failure of the blood supply (necrosis), the immune system is notified by dendritic cells and appropriate action is taken [118]. As shown in figure 4.1, Dendritic cells (DCs) are the main building block in a danger model as they are responsible for having the first interaction with the incoming antigens and guiding them to the immune system [176]. The surface of a DC comprises of co-stimulatory molecules (CSMs) that restrict the number of antigens they can sample during necrosis.

During an innate immune response for a cell death, the three different classes of signals that a DC collects are pathogen associated molecular patterns (PAMP), Safe signals (SS) and Danger signals (DS). Where PAMP are the proteins generated by pathogenic molecules like a virus from an injury or a disease. Hence, the presence of a PAMP signal indicates an anomalous behaviour in data science. A safe signal will clearly indicate a normal behaviour but a danger signal on the other hand indicates an anomalous behaviour but with lower confidence and hence creates an ambiguity in the detection. Due to such

a confusion, once a given DC reaches the limit of either the number of antigens it is exposed to or the cytokine concentration for the costimulatory molecules, it is migrated to lymph node, where the decision whether it is matured (abnormal) or semi-matured (normal) is taken based on the concentration of signals it is exposed to.

To mimic the same in data processing, the algorithm is divided into four different segments or phases. Pre-processing phase where DCA first categorizes the input data or antigens using feature selection for different input attributes into three different signals namely – Pathogen associated molecular particle (PAMP), Danger and Safe signals using mutual information based on kernel estimates. A detection phase where the concentration of co-stimulation molecules (C_{csmDC}), semi-mature (C_{smDC}) and mature (C_{mDC}) cytokines based on an output cytokine equation for every antigen in a given DC is calculated as follows,

$$C_{[csmDC,smDC,mDC]} = \frac{((W_P * C_P) + (W_S * C_S) + (W_D * C_D))}{W_P + W_S + W_D} \qquad (4.1)$$

Where W_P, W_S, W_D are weights of different categories of signals, PAMP (C_P), Safe (C_S) and Danger (C_D) respectively and are taken from the pre-defined Table 4.1 [121] and the output is the cytokine concentration for all three stages attained by a DC. It has been extensively researched to optimize the weights as per the input data and the output labels [126]. Although, this further improves the performance of the approach, it remains supervised and slow in terms of training.

The concentration of these three values decides if a given data instance is normal or abnormal. Further, if C_{csmDC} exceeds the migration threshold (M) as shown in figure 4.2, that DC is then migrated to a stage where the decision is taken based on the cumulative concentrations of C_{smDC} and C_{mDC}. If the semi-mature concentration of cytokines for that antigen exceeds mature concentration (i.e. $C_{smDC} > C_{mDC}$), the given antigen is then assigned a binary value of 0, otherwise it is assigned a value of 1. Such a phase is called context assessment where the context of the migrated DC is assessed. Finally, for all of the antigens processed by the migrated DCs, the total number of times each antigen has been assigned binary value of 1 is calculated and then analysed using a Mature Context Antigen

Value (MCAV). MCAV is a parameter representing the ratio of the number of times an antigen is assigned '1' to the total number of data instances present in a dataset. During this classification phase, an antigen is marked as anomalous if its MCAV exceeds a heuristically setup threshold and normal otherwise.

- *Signal categorization using kernel estimation based mutual information* --

Several automated signal categorization techniques have been introduced and applied in the past including PCA based categorization [127, 128], RST-DCA [129], Entropy [126] etc. However, they still either lack in their total dependency on the labelled data or are not coherent with the signal categorization, which led to some deteriorating results [130]. One of the most common unsupervised approaches is PCA based signal categorization. Despite the ability of PCA to categorize the signal attributes based on the decreasing order of variances, its interpretability remains confined by the fact that components generated by standard PCA are often noisy and exhibit no substantial valid pattern that can be well represented in a linear subspace [56, 81, 93].

This research proposes the use of mutual information (MI) based on kernel estimates between the original feature attributes $\boldsymbol{F} = \{F_1, F_2, F_3 \ldots\}$ and target data or class $\boldsymbol{C}$ using only 5% of the original feature space [131]. This is done in order to avoid the total dependency on the labelling information and to avoid the curse of dimensionality for a huge dataset. The formal definition of MI is given by the following equation,

$$I(F_i; C) = \sum_i p(F_i, C). \log\left(\frac{p(F_i, C)}{p(F_1). p(C)}\right) \tag{4.2}$$

which calculates the degree of dependence between feature vectors and the output class [132]. The joint probability distribution is estimated using Gaussian kernel. The value of MI is equal to zero when F_i and C are statistically independent of each other, positive when the given attribute is strongly relevant and negative when irrelevant. Based on the different levels of mutual information calculated in decreasing order an attribute can be categorized into different signals as safe, danger and PAMP.

- *Kernel Density Estimation based Clustering.*

Clustering using kernel density estimates is a mutated approach for anomaly detection where a probability distribution of a given data is generated using non-parametric density estimation [76]. Among the different clustering techniques available, it has an advantage that it does not require the knowledge of the number of clusters a priori. The proposed method recommends the use of kernel based density estimation for a set of data instances and cluster them based on the following algorithm. As per the fundamental approach for an input data sample F having n instances,

$$F = \{F_1, F_2, F_3 \dots F_n\}$$

The kernel density estimator calculates the probability density $\hat{P}(F)$ which is given by,

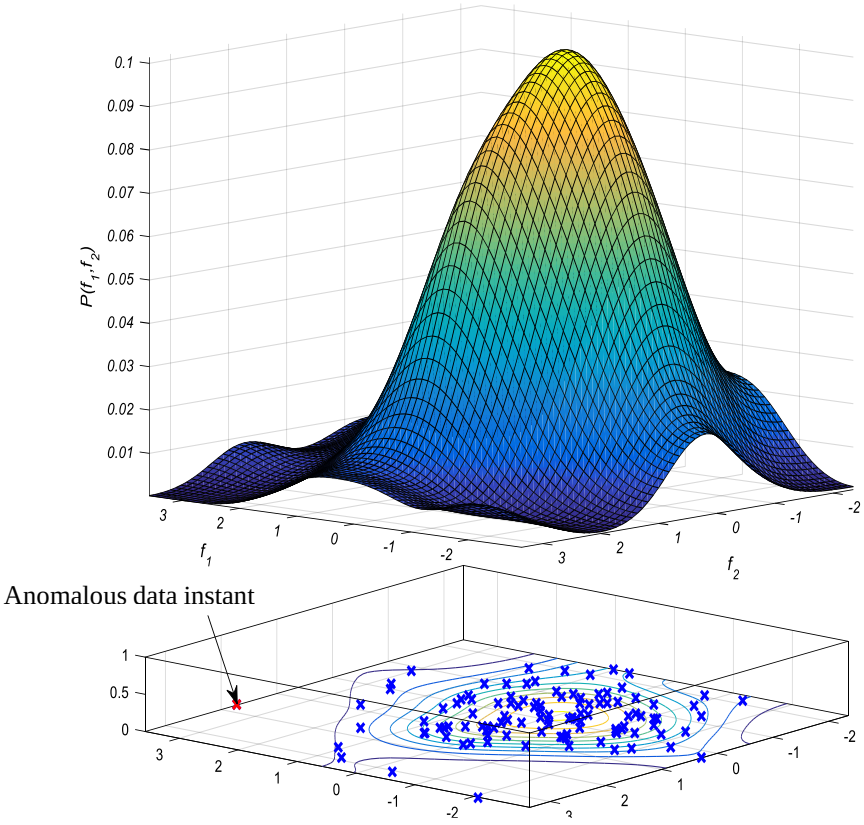


Figure 4-3: Probability distribution for a cluster along with its contour

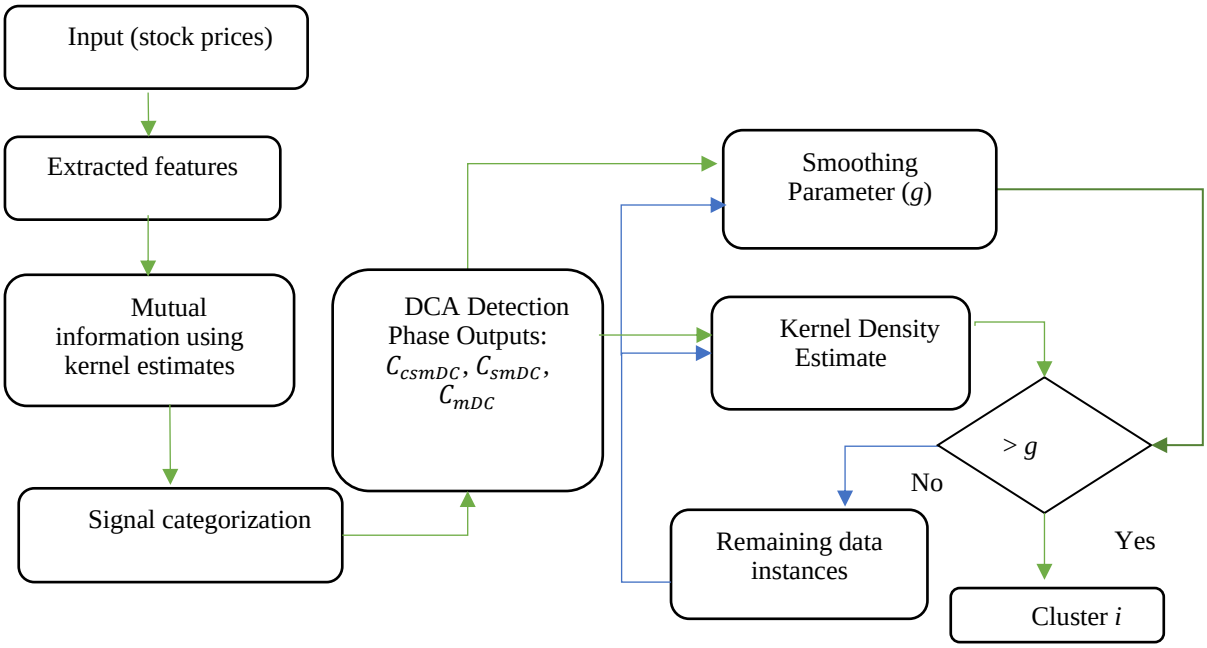


Figure 4-4: Block Diagram of the proposed detection model

$$\hat{P}(F) = \frac{1}{ng}\sum_{i=1}^{n} K\left(\frac{F - \overline{F_i}}{g}\right) \tag{4.3}$$

Where n is the total size of the data to be clustered, g is the smoothing parameter or bandwidth, the kernel function, K that is Gaussian here,

$$K(F) = \frac{1}{2\pi} exp\left(-\frac{F^2}{2}\right) \tag{4.4}$$

Given F be a multivariate data having d dimensions for clustering, $F \in \mathbb{R}^d$; the bandwidth g is defined as follows,

$$g = 1.06\sigma n^{-1}\alpha; \text{ for Gaussian kernel} \tag{4.5}$$

Where α is a parameter calculated for the kernel density estimation and whose value is set to 5 as it minimizes the mean square error between the estimated density distribution and the original distribution as proposed by Silverman [133]. σ is the standard deviation of F whereas n is the varying length of F that will change at every iteration. Initially, an empty or null cluster $\not{C}$ is considered and the bandwidth parameter g is calculated (4.5) for the complete

input sample. For a set of data samples whose difference between the mean of each distribution shown in fig. 4.3 and the sample points is less than g are grouped into one cluster, $\mathcal{C}_i$. For the rest of the instances whose difference is greater than g, a new bandwidth parameter g', a new distribution and a separate mean is again obtained, and a new cluster is formed. The process repeats itself until all the data points within the dataset are clustered. For each cluster so formed, there is a different distribution and a different shaped contour for the set of data points clustered as shown in fig. 4.3. The values on the horizontal axis represented by the feature set F_1, F_2 and on the vertical axis, probability density for the same. Each data instance now can be associated with a given cluster and can be tracked upon. Using this approach, the exact location in time, where each data instance is clustered can be identified. The pseudo code of the clustering algorithm is explained below. Data instances left un-clustered are marked as anomalies.

The next section will explain and discuss the rationale for the proposed approach and combines the outputs from the two methods explained so far. Following to which the later section will further provide details of the experimental results and dataset used followed by its discussion and conclusion.

4.2.3 Detection model

The flow of the operation for detecting abnormal patterns follows a sequence of the stages explained in fig 4.4. In order to capture the effect of such patterns, a pre-processing step of removing any artifacts that restrain anomaly detection process such as periodicity [31] among the original waveform is applied before feature extraction as shown in fig. 4.5. The filtered stock price values so obtained, $x(t)$ and a new feature vector $w(t)$ is also considered. Such a feature is commonly known as the Wilson's amplitude [134].

As the manipulation schemes in stock prices are more prone to high frequencies [135], low frequency components in the signal are removed using wavelet based inverse denoising i.e., only the high frequency components are considered as a feature, $\hat{x}(t)$ [83] where $x(t)$ is the input time series (stock prices). Further, slope of the price i.e., the rate of change of prices and the new

Pseudo Code: KDE Clustering Algorithm

$\boldsymbol{F} = \{F^1, F^2, \ldots . F^d\}$; % for $F^i \in \mathbb{R}^d$ is the feature sample having d dimensions.

$\mathcal{C} = \emptyset$, t = length (F); % where $\mathcal{C}$ is a cluster.

j = 0; % Cluster loop initiate

WHILE $F = \emptyset$

j = j + 1;

$g = 1.06\sigma n^{-1}\alpha$; % define the bandwidth $\boldsymbol{g}$

FOR i=1,2,3……t % t =number of data instances

IF $|\overline{\boldsymbol{F}} - \boldsymbol{F}| < \boldsymbol{h}$ % $\overline{\boldsymbol{F}}$ is the mean(s) location of distribution for the data samples

$\mathcal{C}_j = \mathcal{C}_j \cup \boldsymbol{F}_i$;

$\boldsymbol{F} = \boldsymbol{F} \backslash \boldsymbol{F}_i$; % clustered instances removed from the input sample set.

ENDIF

ENDFOR

feature, $\frac{\partial(w(t))}{\partial t}$ that further magnifies the change are also used as the feature sets. A total of five feature including original time series are used for the approach,

$$F = \left[x(t), \hat{x}(t), w(t), \frac{\partial\big(w(t)\big)}{\partial t}, \frac{\partial(x(t))}{\partial t}\right]$$

The whole feature set has already been described the Chapter 3. As per the original DCA, the proposed model suggests the computational evaluation of the stock prices by collecting the statistical features and assigning a rank to each attribute using mutual information. Feature categorization is then carried out on it and the feature having highest rank is assigned to the safe signal (SS) and the ones with the lowest rank are assigned to the PAMP signal while the remaining features are assigned to the danger signals (DS). The proposed approach suggests using DCA transforming the input features to three DCA outputs (C_{csmDC}, C_{smDC}, C_{mDC}) calculated during the detection phase and context assessment phase that assigns an MCAV value of 5 to each data instance for all three DCA outputs from detection phase [119,121]. Such an output is further subjected to KDE clustering.

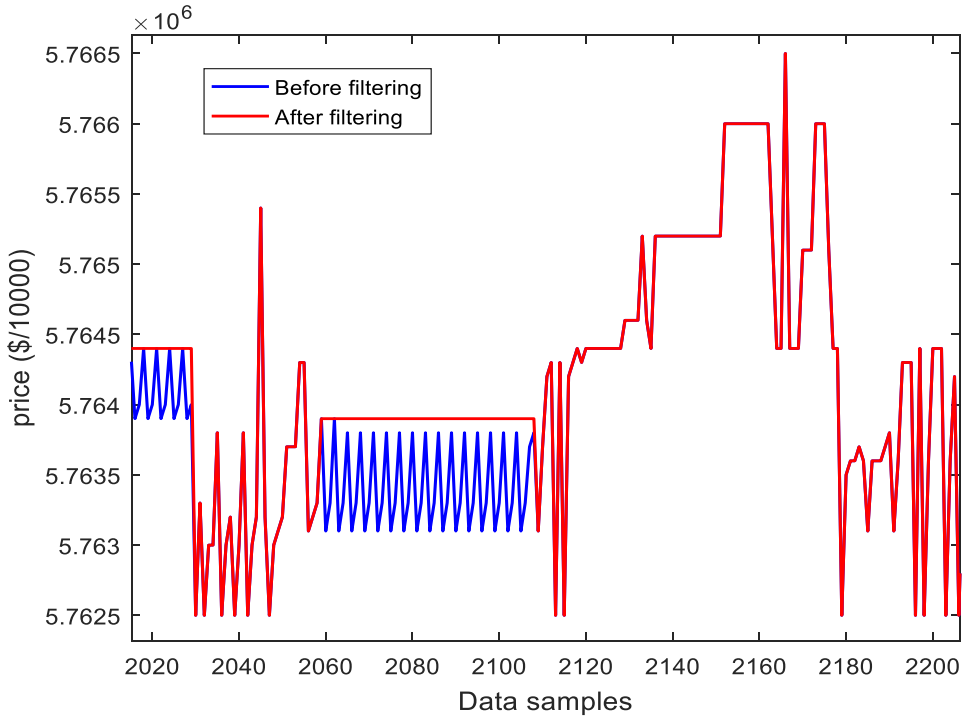


Figure 4-5: Stock prices before and after periodicity removal. The red ones after the pre-processing

As the stock prices are non-stationary and highly volatile with variable sampling frequency in nature, its statistical properties like mean, variance and standard deviation continuously vary with time. Based on this, the distribution of stock prices also deviates from normality i.e., the uncertain disparity between the normal and anomalous instances. Hence, it is not viable to assume the original data only to be a part of normal distribution. Such saliences of any dataset can be captured using adept and detailed specific features and by creating a data driven population density model. In view of this fact, the output data from DCA context assessment phase is then grouped into clusters by KDE clustering while fitting a kernel-based distribution and without forth specifying the number of clusters [76]. Analysis of the data now becomes easier as the size of a cluster is smaller, and detection of price manipulation can be performed. One of the important advantages of using such a data driven approach is its decision-making capability based on analysing the patterns that are being subjected as an anomaly. Subsequent sections will show and discuss the obtained results.

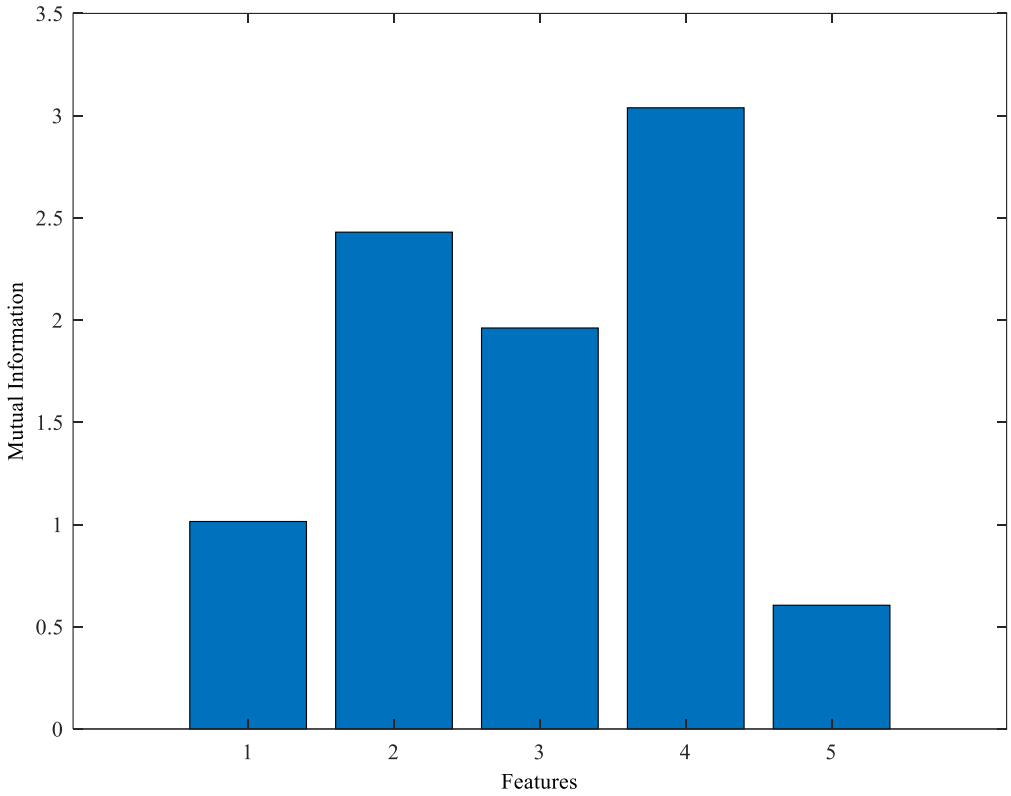


Figure 4-6: Feature categorization using Mutual Information based on its values. The feature with the maximum value of mutual information is assigned to safe signal, the minimum feature as PAMP and the remaining ones in between as danger signals.

4.2.4 Experiments and results

The experimental data set used is provided by an open source LOBSTER database [48] and is detailed in the previous chapters. Artificial manipulation is injected by using two different types of anomalies in order to test the validity of the proposed approach as shown in fig. 2.1. This is due to the fact that it is extremely difficult to acquire data annotation, due to the market confidentiality policies and even the data cannot be achieved without paying a hefty amount annually. Both types of anomalies described before are injected into the normal dataset, type 1: a synthetic anomalous waveform having a saw-tooth like fall of 16 bps in 95 ms, an imitation of a real life example of spoof trading and type 2: a rise and then sudden fall of 30 bps in a time span of 0.1 sec is a reconstruction of the Pump & Dump from 14th Dec, 2011. For an input of five features, signal categorization is performed and subsequently three outputs (4.1) from DCA detection phase are generated for the three set of signals used (Safe, Danger and PAMP). The input data window to the KDE clustering is heuristically selected

as 100 samples. For all of the proposed and existing approaches, their performance is evaluated using the AUC (Area Under the Curve) calculated from the Receiver Operating Characteristics (ROC) curve and the false positive rate or false alarm ratio (FAR). An ROC is a curve between True Positive Rate (TPR) and False Positive Rate (FPR) where TPR and FPR are calculated while varying the threshold of the output from the KDE clustering.

Table 4.2 & 4.3 shows the performance evaluation of DCA based approach and its improvement over the k-means based [117], the PCA-based [82], the K-nearest neighbour based [65] and the OCSVM based [65] anomaly detection techniques which are some of the most commonly used methods in unsupervised and supervised learning. Table 4.4 shows the comparison of the proposed approach with the existing benchmark approaches in market manipulation detection. For this purpose, only existing approaches that claims the generality of their algorithms to detect different manipulation schemes and have not used supervised learning in stock prices [6, 8] are selected.

Table 4.2**:** Comparison of AUC with existing benchmark techniques for anomaly detection

AUC	Amazon	Apple	Google	INTC	MSFT
DCA-KDE	**0.9337**	**0.9841**	0.8415	**0.995**	**0.9963**
k-means [4]	0.5799	0.5819	0.6328	0.5077	0.5047
PCA [7]	0.9013	0.6902	**0.8793**	0.8732	0.8655
OCSVM [1]	0.8933	0.6603	0.5911	0.697	0.6419
kNN [1]	0.5993	0.5623	0.5876	0.5469	0.5509

Table 4.3: Comparison of False Alarm ratio with existing benchmark techniques for anomaly detection

FAR	Amazon	Apple	Google	INTC	MSFT
DCA-KDE	**0.10**	**0.19**	**0.68**	**0**	**0**
k means [30]	7.33	1.26	9.95	0.02	0.02
PCA [7]	3.9	6.64	7.22	57.29	49.89
OCSVM [1]	49.54	67.8	75.2	59.08	77.48
kNN [1]	0.14	0.45	0.69	0.23	0.08

4.2.5 Discussion

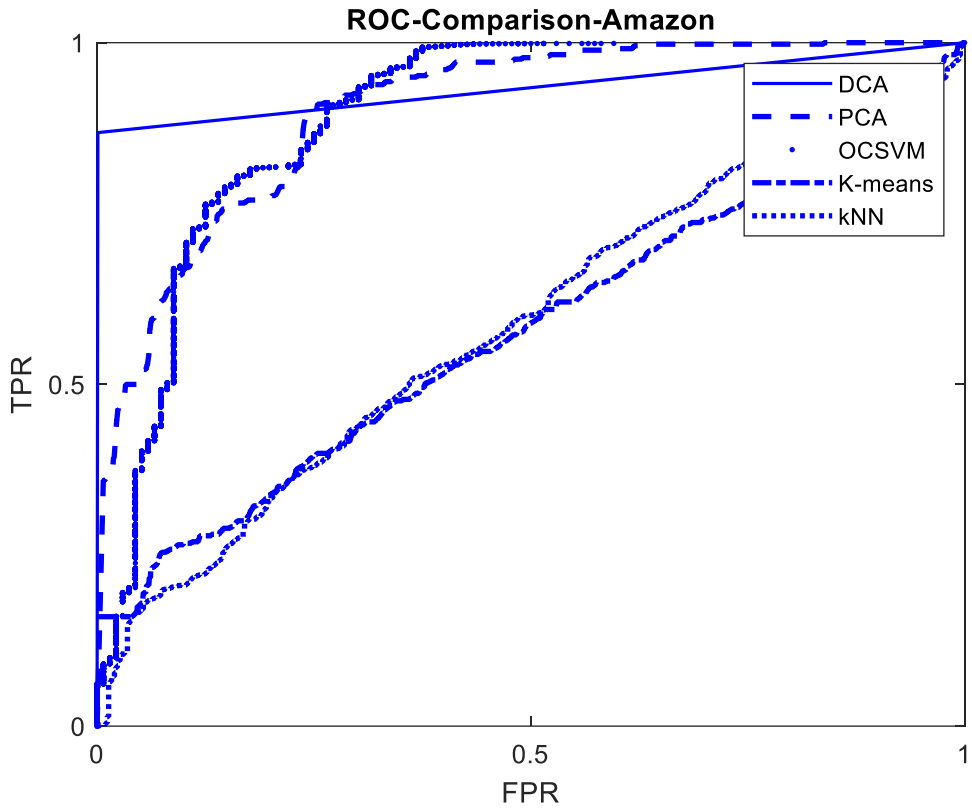
The experimental results shown in Table 4.2 & Table 4.3 obtained by the DCA-KDE clustering based anomaly detection approach presents a promising development in the detection of two types of price manipulation (Spoof trading and Pump & Dump). Such manipulative schemes are carefully selected, as they seem to provide similar impact on the price of a stock as the ones depicted by these added anomalies. Figures 4.7 (a)– (e) shows a comparative analysis of ROC curves and as is evident from the AUC values of the DCA-KDE approach on stocks like Amazon, Apple, INTC and MSFT that shows a major improvement on it (all above 0.9) except for the Google stock. As a measure of performance, AUC values for DCA based approach showed an improvement at least by 3.47% for Amazon, 29.86% for Apple, 12.42% for INTC and 13.12% for MSFT stock when compared with the rest of the existing techniques in anomaly detection. Figure 4.8 and 4.9 shows graphically such an enhanced improvement.

Similarly, the false alarm ratio for the same stocks shows significant results having fewer false positives but again not for Google stock relatively. One of the possible reasons that can be attributed to this is the volatility in the Google stock and the possible overlapping of normal patterns similar to the injected

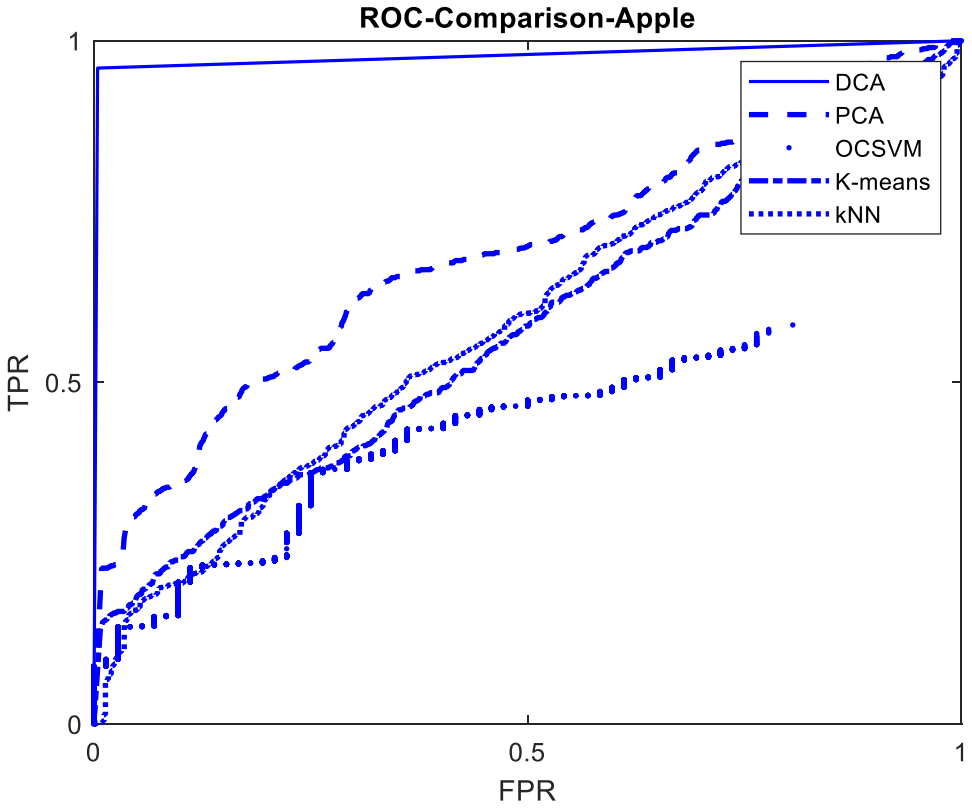
Table 4.4: Comparison of AUC with existing approaches towards market manipulation detection.

AUC	Amazon	Apple	Google	INTC	MSFT
DCA-KDE	**0.9337**	**0.9841**	**0.8415**	**0.9950**	**0.9963**
AHMMAS	Not Reported	0.8142	0.8025	0.8971	0.7336
EMD-KDE	0.9226	0.7946	0.7896	0.8805	0.8903

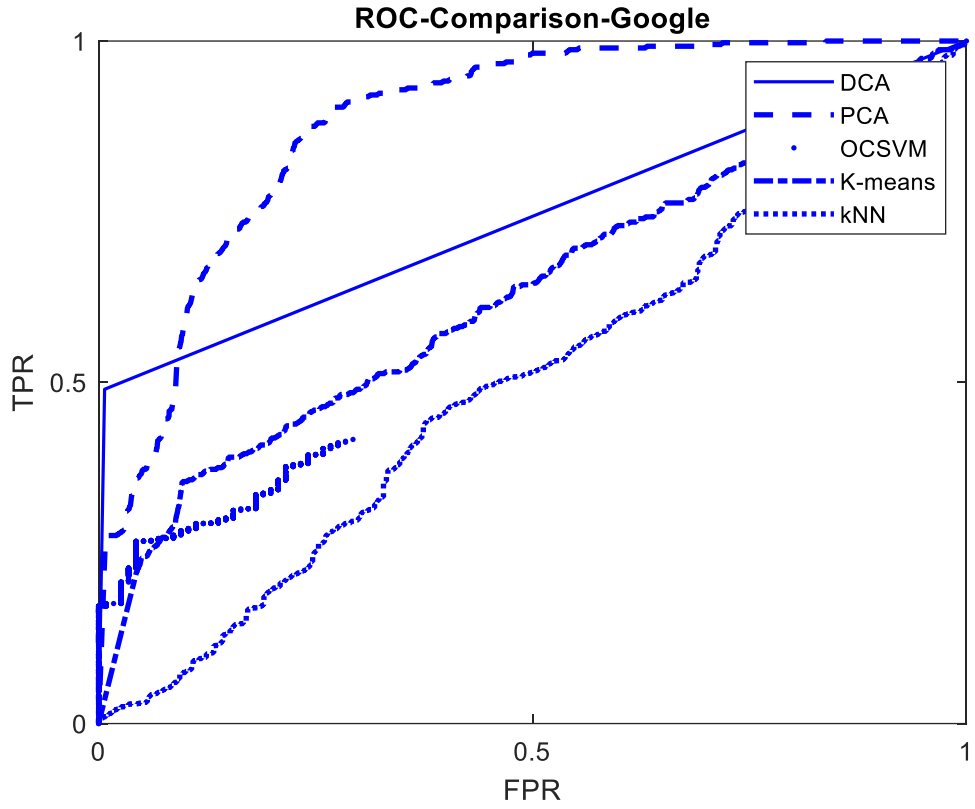
anomalies. Collectively, since false alarm ratio for Google achieved an

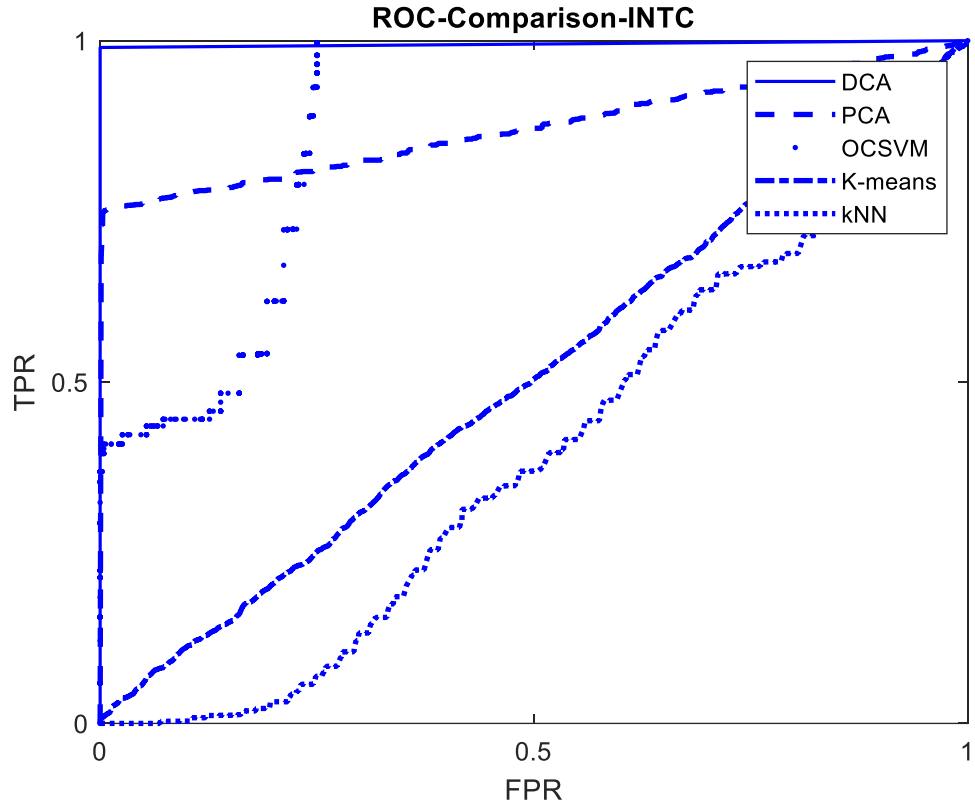


(a)



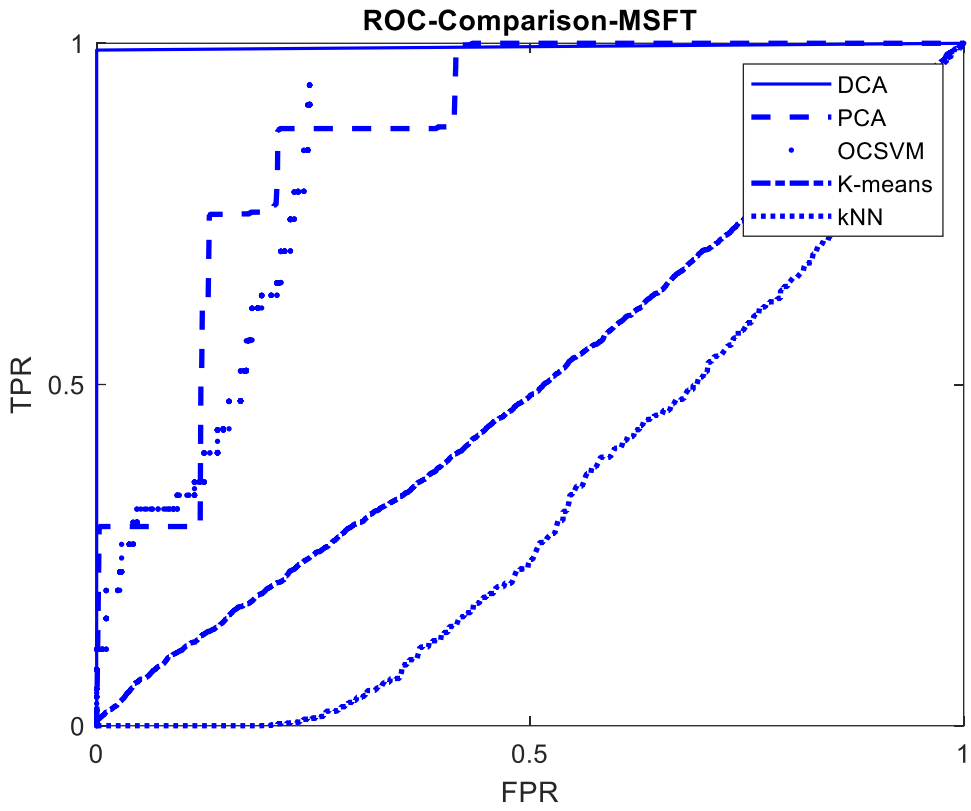
(b)





(d)

improvement over the rest of the techniques, the overall performance of the



(e)

Figure 4-7 (a)-(e): ROC comparison with selective anomaly detection techniques in time

approach can be justified well. Comparatively though, they showed promising advantage as the even the minimum gain is 28.57% for Amazon, 57.77% for Apple, 1.45% for Google and almost 100% for both INTC and MSFT stocks over the rest of the techniques in anomaly detection as shown in Fig. 4.9.

The results also surpassed some of the existing approaches in market manipulation detection using both supervised and unsupervised training as shown in Table 4.4. Although the results performed variably for the AUC values of DCA-KDE based approach. Stocks like Amazon, Apple, INTC and MSFT again performs better except for Google, still an AUC over 0.8 is considered better performance [74]. A slightly lower AUC value is achieved on Google stock, however the obtained AUC (0.8415) still represents an improvement over AUC values achieved by the existing competitive approaches, namely 0.7896 for the EMD-KDE (authors' previous work on market manipulation detection) and 0.8025 for the AHMMAS approach [20]. The robustness in performance of the proposed approach can be explained from the non-linear data transformation

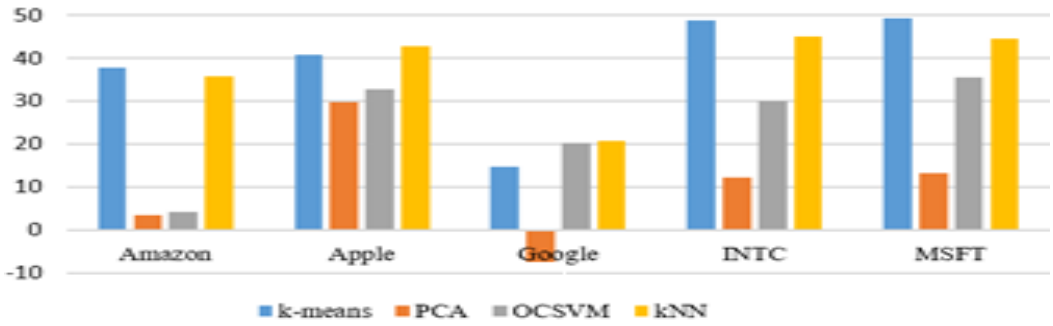


Figure 4-8: Percentage improvement in AUC values for the proposed DCA-KDE approach over existing benchmark techniques for anomaly detection

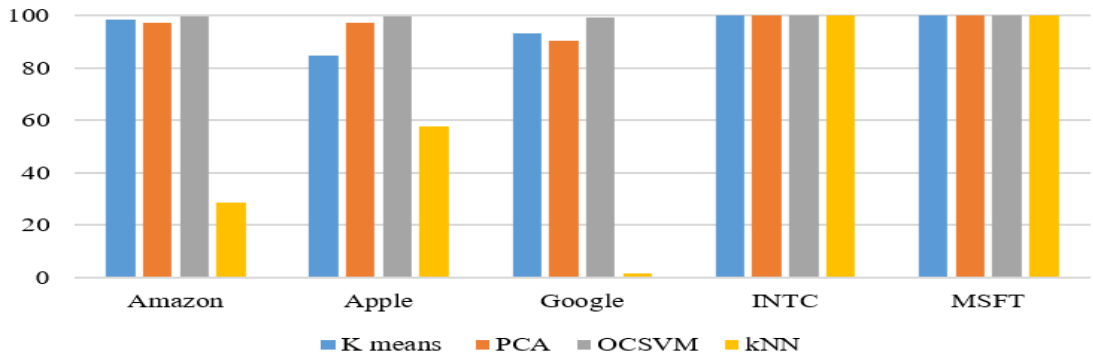


Figure 4-9: Percentage improvement in FAR values for the proposed DCA-KDE approach over existing benchmark techniques for anomaly detection

using (4.1) that narrows down the amount of data to be processed by KDE clustering approach. Additionally, feature categorisation section helps in deciding the escalation level of an individual feature which is missing from both previously proposed EMD-KDE and KPCA-KDE approaches in chapter 3. This further helped in reducing the number of false positives as can be seen from the results. Another possible rationale that contributes to the robustness of the proposed method is the automatic selection of the smoothing parameter from the dataset that establishes the cluster boundaries and can probably divide the normal and anomalous boundary using KDE clustering based approach. Furthermore, the run time involved with some of the supervised training methods in optimizing the results can also be saved using the proposed method.

4.3 Conclusion

The research mentioned in this chapter presents another innovative approach for detecting stock price manipulation based on the combination of DCA and KDE clustering. The research proposed to avoid dependencies on the data annotation, as it is extremely expensive and hard to achieve due to clauses of confidentiality

policy in the markets. An explanation of artificial immune systems and its types with a brief literature of its applications has also been presented along with their limitations. The proposed method envisaged two types of manipulations existing in the stock markets, which relates to different categories of price manipulation and their detection using semi-supervised learning. To achieve this, a large open source database, which is known for not having any manipulation, is considered. To test the validity of the proposed approach, a significant number of artificially generated manipulations are then injected to it making the input dataset, a mixture of both normal and abnormal instances. For a very small number of extracted features, mutual information was calculated with the output class. Based on which, they were categorized into PAMP, Danger and Safe signals. After the context assessment phase in DCA, the outputs so obtained are then subjected to a KDE clustering algorithm which groups the data set based on the density estimate defined within a bandwidth parameter. The data instances left un-clustered are then marked as anomalies. The results have been compared with the existing benchmark techniques in both supervised/unsupervised anomaly detection and also with the existing models in market manipulation detection including authors previous work (EMD-KDE). It is found that the proposed model outperforms the existing techniques in anomaly detection by a significant margin in terms of AUC and FAR values for stocks considered. It also outperforms author's previous contribution, chapter 3 being simple, robust and also in terms of improving upon the false positives. However, the approach is still limited in its representation of data being less diverse (volume information), its size and the detection of normal and manipulative instances overlap that further contributes of more false positives. The research is limited by the scope of the feature set considered and can be further improved if the stock volume information can also be included. There is also a need either to vary most of the heuristic data especially, number of data instances provided to KDE clustering or an algorithm to implicitly select such values while maintaining the satisfactory results.

Chapter 5 presents two different techniques using deep features. It provides detailed analysis of manipulation detection by learning affinity among the trades

using autoencoders and tries to reduce the false positives caused by normal/abnormal overlap by observing the trades under a defined context.

Chapter 5: Deep Learning Based Stock Price Manipulation Detection

5.1 Introduction

There is an increasing demand of analysing stock price data at most of the stock exchanges around the world. One of the key objectives in doing so is the establishment of a detection model that can identify manipulative instances caused by the market manipulators or market abusers. Stock price manipulation can be explained as the illicit trade transactions made by the manipulator that represents falsifying market prices using illegal means [3]. This is due to the fact that it diminishes the investor confidence as it creates a false impression about the manipulated stock and eventually effects the stature of the market as well. To accomplish such an objective, the stock price data needs to be thoroughly studied, analysed and a optimum decision boundary needs to be established between normal and abnormal patterns. One of the key constraints here is the unavailability of the annotated datasets having both normal and manipulative trades required to train a given machine learning model. Due to which, it becomes difficult to analyse and provide specific parameters to a detection model. This leads us to propose a fully unsupervised model that can determine the exact location of the manipulative instances without much human intervention.

In order to make such a prediction, it is crucial to comprehend the problem from its basics. Stock price manipulation is an act of manipulating stock prices by using some predefined strategies like pump & dump [119] and spoof trading [10]. Pump & dump is a scheme where the manipulator deceives the investors by pumping the price of a given stock through the creation a false demand for the same stock which leads to several added investors who believe the demand to be genuine. However, the manipulator then sells its own investment bought at a cheaper price (bid) when the desired price is achieved. Figure 1.1 (a) & (b) represents such a situation explaining the progress of a spoofing case in 2012. It should be kept in mind that unlike pump & dump, spoofing can occur at a deeper (although visible) level of the order book.

This chapter provides a detailed description of two deep learning based approaches applied to solve the stock price manipulation problem in a progressive manner. First, a static model of manipulation detection is implemented that learns the inter-relationship among stock trades distributed over time. Such inter-relationship is learned by an autoencoder that is trained upon an adapted loss function that represents the inherent density estimate of the input features. In the second detection model, a dynamic approach is adopted that aims the detection of manipulated instances that skip the detection under the guise of a normal pattern. In other words, the second approach addresses the issue of overlap among normal and abnormal trades using contextual learning. The model constitutionalised leverages the learning of a pattern defined by KDE clustering by combining Temporal Convolutional Network (TCN) and Generative Adversarial Network (GAN) into a tempGAN model.

5.2 Stock Price Manipulation Detection Based on Under-Complete Autoencoder Learning of Stock Trades Affinity.

In this section, we aim to capture the above-mentioned manipulations in a dataset, acquired from an open source database by training autoencoders (AEs). Autoencoders are neural network models that approach to learn the specifics of an underlying dataset in an unsupervised manner generally used for data denoising [137] or dimensionality reduction [138]. The goal here is to encode the input stock price data using an encoding function, to further reconstruct the input using a decoding function and to minimize the reconstruction error by optimizing the loss function. Some of the recent research [139-142] attempt to use anomaly detection by testing an AE (only trained on normal data) providing high irregularities in the output only over the abnormal instances. Although most of the existing research using AE for anomaly detection claimed substantial improvements in the results, very few of them explored the spatial aspect of the time-series dataset under consideration. It becomes extremely important for a robust model to learn the space-time representation of the dataset and its evolution with time whether it is normal or a combination of both normal and abnormal stock trades. Unlike past approaches, this proposed research will first envisage the spatio-temporal characteristics of the dataset and then train an AE

further upon it. The validity of any model can be determined from its ability to detect the anomalies (market manipulations here) and minimum amount of human effort required in detecting them. Following are the key contributions made by the proposed approach;

• *Affinity matrix describing the relationship among data points* - A new dataset is generated describing the affinity among all the input stock price data instances (Size - $N*d$, $d \in \mathbb{R}^d$) given length, N and d, dimensions. Although, a number of affinity matrix based clustering techniques exist [143–146], all of them asks for pre-defined parameters including the number of clusters. The research proposes to describe the innate relationship or affinity among stock prices through a graph Laplacian representation [145]. Such a matrix is suitable for explaining the relationship among all the stock prices. The research proposes to describe the innate relationship among stock prices within a given dataset using a Euclidean distance measure. It is also useful in describing the affinity of a normal data instance towards normal/abnormal data instance and vice-versa.

• *Optimization of under-fitting Autoencoder (AE) using kernel density estimates* - An under-fitting AE is well suited for reducing the dimensions of the input dataset while optimizing the loss function for a minimum reconstruction error. The aim is to extract most significant features that can represent the stock price data. The input dataset here is the affinity matrix, size ($N * N$) for N data instances and a single hidden layer. Such an AE is optimized while fitting the inherent data distribution using kernel density estimate (KDE) as an objective/likelihood function [146]. This helps in preserving the inherent characteristics of the input stock price data in the extracted features from hidden layer.

The following sections explain the understanding and implementation of affinity matrix and AE. Thereafter, the proposed work plan along with the processing of the output from the AE using Multidimensional KDE (MKDE) clustering technique is described. Experimental results for price manipulation detection on

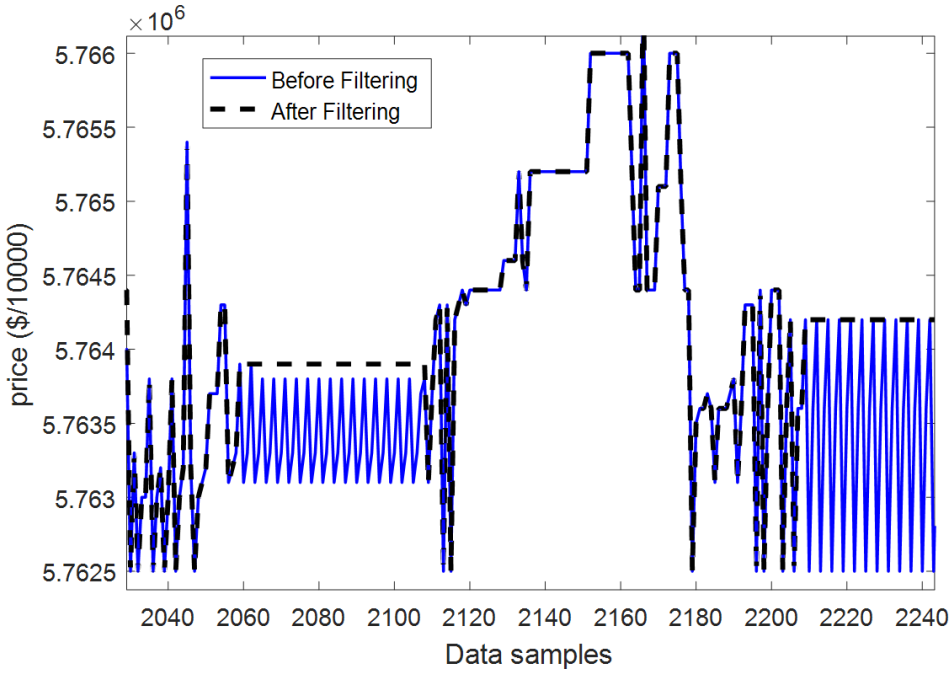


Figure 5-1: Stock prices before and after periodicity removal [30]. The red ones after the pre-processing overlap original prices shown in blue

the stocks used are presented and discussed in subsequent section and finally, conclusions are drawn.

5.2.1 Distance Based Affinity Matrix

Affinity matrix can be described as a technique that explores the relationship among data points. Also known as similarity matrix, it is also used to explore the similarity among data points by using Euclidean distance as a measure. The idea is to compute affinity among stock price data points, apply feature selection and then group the extracted features using proposed clustering techniques. A number of approaches for calculating the affinity based clustering techniques have been proposed in the literature [141-143], although most of them require the number of clusters to be specified a priori. The process of creating an affinity matrix is taken from the fact that every stock price data instance within a similar group is strongly correlated to each other compared to the ones that are far apart. One can also understand this as the manifold creation within graphs, where the contiguous stock price data instances have similar labelling information and the distant stock price data instances differ. For a set of n stock price data instances

under consideration $x = (x_1; x_2;: x_n) : x_i \in \mathbb{R}^d$ and considering the affinity matrix to be non-negative matrix, $W: W \geq 0$ can be explained as follows,

$$dist = \|d(x_i, x_j)\| \tag{5.1}$$

$$A_{ij} = exp\left(\frac{dist}{2 * \sigma^2}\right) \tag{5.2}$$

where $d(x_i, x_j)$ is the *l*2-*norm* distance metric between every stock price data instance x_i and x_j across multiple dimensions. Such a matrix can also be termed as adjacency matrix here as it calculates a correlation factor between all the stock prices within the dataset. The non-negative adjacency matrix, A_{ij} (2) is sufficient to make the resulting matrix graph Laplacian $L = D - A$ where $D \subset \mathbb{R}^{n*n}$ is a diagonal matrix whose entries being $D_{ii} = \sum A_{ij}$ positive semidefinite which makes the task computationally inexpensive [147]. Interpretation of A, in most of the existing researches, a sparse representation is preferred to avoid spurious connections between far away stock price data points (disjoints) [148]. Although, such a technique becomes insensitive to outliers and hence is avoided in this research.

5.2.2 Under-Complete Autoencoders

Out of several AEs available, standard under-fitting AEs were found suitable for detecting anomalies. This is due to its advantage over the other AEs that it minimizes the influence of small variations in the data during the learning of the model by avoiding any regularization/penalty terms as in Contractive, Sparse or Denoising AE [149]. The autoencoder is trained upon the dataset in a way that the inherent distribution of the dataset is efficiently learned. For this purpose, the dataset is modelled using kernel density estimates and to best fit the parameters of AE to the stock price data, the loss function here is selected as the kernel density estimation of the dataset under consideration.

An AE will learn the distribution pattern present for a given dataset and will try to maximize the log-likelihood $l(f(x))$ as shown in equation (10) to optimize the learning. For a given stock price dataset, $x = (x_1; x_2;: x_n) : x_i \in \mathbb{R}^d$, a kernel density estimated function can be described as follows,

$$P(x; g) = \frac{1}{ng} \sum_i K\left(\frac{x - X_i}{g}\right) \tag{5.3}$$

at the location X_i, g is computed via the diffusion process [99], K is the Gaussian kernel shown below,

$$K(x) = \left(\frac{1}{2\pi}\right)^{-d/2} exp\left(\frac{x^T x}{2}\right) \tag{5.4}$$

The selection of such a function is based on the better adaptability of the AE to learn the underlying stock price data set [150]. The value of X_i is selected as a linear combination of the latent (hidden) layer output and the output bias, (the rationale for selecting a linear relationship proves to provide a better optimization of the parameter values while minimizing the reconstruction error [150])

$$X_i = b + W * h(x_j) \tag{5.5}$$

where $h(x_j)$ is the latent layer output for the j^{th} variable, W are the weights, assuming similar weights between input-latent and latent-output layers and b as the output bias. As explained in the details above, in order to make the AE learn and adapt to the dataset under consideration, it is proposed to select the loss function as the density estimate of the data obtained from (5.3).

$$f(x) = P(\left(\hat{x}\middle|h(x)\right)) = P(\hat{x}; g) \tag{5.6}$$

Substituting the value from (5.5) in (5.3),

$$P(\hat{x}; g) = \frac{1}{ng} \sum_i K\left(\frac{\hat{x} - (b + W * h(x_j))}{g}\right) \tag{5.7}$$

Let $\hat{x}$ be the output of the decoder. Let also consider the latent input layer relationship to be linear (5.8),

$$h(x_j) = a + W * x_i \tag{5.8}$$

From (5.7), (5.8) and (5.4), following conclusion is made for the log-likelihood,

$$l\left(f(x)\right) = \text{-log}(P(\hat{x}; g)) \tag{5.9}$$

$$l(f(x)) = \frac{1}{(2\pi)^{-\frac{d}{2}} * ng^2} \sum_i \frac{\|\hat{x} - (W^2 x_i + C)\|^2}{2} \quad (5.10)$$

where

$$C = b + a * W \quad (5.11)$$

As the added bias C and the weights W^2 are a linear transformation of the input x_i in (5.10), the loss function can be regarded similar to the *l*2-*norm* (sum of the Euclidean distances) as with a standard autoencoder for real inputs. Such an AE is first trained upon the dataset having normal trades. Once trained the same AE is then used upon the test stock price data, containing both normal and abnormal trades.

5.2.3 Detection Model

As mentioned briefly in the introduction, the proposed research aims to create a clear description of data distribution for clustering algorithms to follow for manipulation detection. The approach allows statistics of the dataset to be processed in such a way that the separation between normal and abnormal trades becomes clearly distinguishable. For the purpose of achieving so, firstly a pre-processing step of removing artifacts such as periodicity [31] from the stock

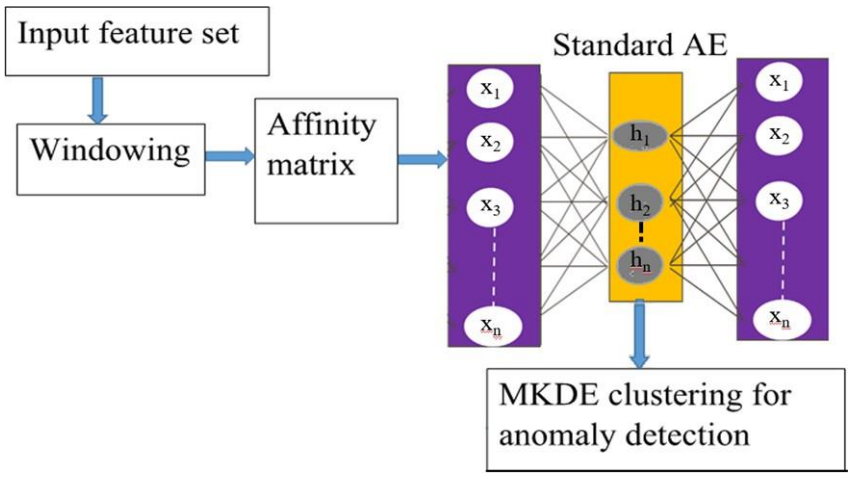


Figure 5-2: Proposed Architecture for Price Manipulation Detection

prices is applied as illustrated in Figure 5.1. As shown in Figure 5.2, the approach computes the features relevant in capturing the effect of anomalies from the pre-processed time series. As the high frequency elements in the stock

price data is more prone to anomalies [8] wavelet transform is applied to analyse only the high frequency elements in the data and neglect the low frequency elements i.e. for input stock prices, $x(t) : t \in (t;\ t + n)$ for n number of data instances within the window, only its high frequencies portion $\hat{x}(t) : t \in (t;\ t + n)$ is selected. In other words, low frequency components in the signal are removed using wavelet based inverse denoising i.e., only the high frequency components are considered as a feature, $\hat{x}(t)$ [15] where $x(t)$ is the input time series (stock prices). Further, slope of the price i.e., the rate of change of prices and the new feature, $\frac{\partial w(t)}{\partial t}$ that further magnifies the change are also used as the feature sets along with Wilson's amplitude [30], stock traded volume information $v(t)$ and the slope of traded stock volume $\frac{\partial v(t)}{\partial t}$are also considered as features. A total of five feature including original time series are used for the approach,

$$X = \left[x(t), \hat{x}(t), w(t), \frac{\partial w(t)}{\partial t}, \frac{\partial x(t)}{\partial t}, v(t), \frac{\partial v(t)}{\partial t}\right]$$

The architecture of the proposed work allows such time specific features as the input to being divided into windows of fixed length. Windowing the whole dataset into smaller set of samples reduces the number of computations for the affinity matrix to be calculated next. Once windowed, each set of features are now transformed into an affinity matrix using the proposed method explained before. The output is now processed through an under-fitting single layered autoencoder pre-trained upon the normal dataset. Following which, the 6 encoded features are extracted from the AE are then passed to the MKDE based clustering approach without stipulating the amount of clusters required up front [76]. It should be noted here that to estimate the distribution of a non-stationary and volatile stock price dataset where the mean and variance regularly varies with time, it is not reasonable to assume the stock price data to be normally distributed. In order to extract meaningful information, a data driven population distribution estimate needs to be created. Hence the density estimate of the extracted features from the AE is created by fitting a kernel based distribution prior to be processed by the proposed MKDE clustering approach [30]. The

MKDE based clustering is summarised in the Algorithm mentioned in section 4.2.3 of Chapter 4.

The results obtained after the implementation of the above mentioned proposed research are presented and discussed in the following section along with the dataset used.

5.2.4 Results and Discussion

The datasets used in this approach are tick data for level 1 orderbook taken from the LOBSTER project, an open source and include stocks like Apple, Amazon, Google, Microsoft and Intel corporation for June 12, 2012 operating on NASDAQ, USA [48]. The dataset provides stock prices and volume information versus time. The rationale behind selecting such stocks is the popularity of each of them with the amount of influence they have on the market and are reported to have no manipulative trades [89]. The fact that acquiring labelled dataset is extremely difficult because of data confidentiality regulations and the hefty sum one has to pay annually, artificial manipulation of two different types as shown in Figure 1.1 (a) & (b) is preferred to test the robustness of the detection model.

A saw-tooth like waveform having a rise of 7 bps in 95 msecs creates the impression of a real life example of trading activity by Demonstrate LLC condemned for spoof trading on 25th Sept, 2012 [82]. Type 2 is an example of pump and dump manipulation strategy for WAB prices having a rise and fall of 30 bps in a duration of 0.1 sec on 14th Dec, 2011 [78]. As the number of data instances varies among the stocks, the amount of manipulative instances injected is also varied. The number of manipulations injected in Apple, Amazon and Google stocks are 100 anomalies/type and for Microsoft and Intel corp stocks, 200 anomalies/type. To ensure the effectiveness of the detection model, a random injection of the manipulation in the original stock price dataset is practiced making a combination of both normal and abnormal trading patterns. Initially, a total of seven time specific features are extracted from the synthetic dataset. An affinity matrix $L_{500*500}$ is then generated by considering the window length of 500 data instances for the input feature set. Following which, a pre-

Table 5.1: AUC performance comparison against a selection of existing manipulation detection techniques

Dataset	Proposed Approach	kNN [2]	PCA [5]	K-means [10]	OCSVM [2]
Apple	**0.9981**	0.7926	0.6902	0.5819	0.6603
Amazon	**0.9998**	0.7982	0.9013	0.5799	0.8933
Google	**0.8215**	0.5612	0.7993	0.6328	0.5911
Intel Co	**0.9701**	0.5469	0.8680	0.5077	0.6970
MSFT	**0.9989**	0.5509	0.8655	0.5047	0.6419

trained AE (upon normal dataset) is used to process the affinity matrix and extract six encoded features before being processed by the MKDE clustering approach to cluster normal and manipulative trades separately. The input to the MKDE clustering is a dataset of size 500 by 6 using a Gaussian kernel without specifying the number of clusters up front. The proposed approach is evaluated by using area under the receiver operating curve [74] as the performance measure along with false positive ratio and F-measure. Table 5.1 shows the comparative assessment of stocks with K-means based approach [117], PCA based [82], K nearest neighbour based [65] and OCSVM based manipulation detection techniques [65] in terms of AUC. Such techniques are selected for

Table 5.2: FAR performance comparison against a selection of existing manipulation detection techniques

Dataset	Proposed Approach	kNN [2]	PCA [5]	K-means [10]	OCSVM [2]
Apple	**0.0054**	0.45	6.64	1.26	67.8
Amazon	**4.9197E -04**	0.14	3.9	7.33	49.54
Google	**0.0288**	0.68	7.22	9.95	75.2
Intel Co	**0**	0.23	57.29	0.02	59.08
MSFT	**0**	0.08	49.89	0.02	77.48

Table 5.3: F-score performance comparison against a selection of existing manipulation detection techniques

Dataset	Proposed Approach	kNN [2]	PCA [5]	K-means [10]	OCSVM [2]
Apple	**0.3689**	0.1344	0.1457	0.1708	0.0450
Amazon	**0.4704**	0.1714	0.1568	0.1484	0.0284
Google	**0.2566**	0.135	0.1806	0.1513	0.0196
Intel Co	**0.5836**	0.1014	0.2085	0.1119	0.0126
MSFT	**0.5934**	0.1148	0.2077	0.2141	0.0920

comparison being some of the commonly used methods in both unsupervised and supervised learning for manipulation detection and an optimum selection of the parameters is carefully carried out to assure a fair comparison. Similarly, table 5.2, & 5.3 shows the comparative assessment of the proposed stock price manipulation detection method against k-NN, PCA, K-means and OCSVM approaches in terms of FAR and F-score. Finally, the proposed model is also assessed on the basis of its comparison in terms of AUC values with existing benchmark research in stock price manipulation detection [78], [31] as shown in table 5.4. Given the fact that only methods that aim to generalize their detection

Table 5.4: AUC performance comparison against existing benchmark stock price manipulation detection methods

Dataset	Proposed Approach	AHMMAS [6]	EMD-KDE [31]
Apple	**0.9981**	0.8142	0.7946
Amazon	**0.9998**	Not Reported	0.9226
Google	**0.8215**	0.8025	0.7896
Intel Co	**0.9989**	0.8971	0.8805
MSFT	**0.9701**	0.7336	0.8903

model towards different manipulation schemes with unsupervised learning and have used the similar datasets are selected.

It can be easily observed that the proposed approach outperforms a selection of existing manipulation detection techniques (both supervised and unsupervised) along with existing research in stock price manipulation detection. It is also important to notice the significant enhancements in terms of false alarm rates as most of the values calculated are two decimal places below zero. To assess the performance in Table 5.1, the AUC value of the proposed approach in stocks surpassed existing manipulation detection methods by 25.92% for Apple, 10.92% for Amazon, 2.77% for Google, 11.76% for Intel corp and 15.41% for Microsoft stocks. It is also worth analysing the low AUC value for Google stock (comparatively to other stocks) which can be attributed to the volatility and also the overlapping of normal and manipulative trading behaviours. However, this can be improved by a thorough analysis of the time series in Google stock price and carefully selecting the injection locations of the manipulative data. In addition, from Table 5.2 & 5.3, the performance comparison of the proposed approach in terms of FAR and F scores, shows a dramatic improvement of no less than 95.76% and 90.07% respectively, for all the stocks.

As is evident from Table 5.4, the results also outperform the existing research in stock price manipulation detection in terms of AUC values by a maximum of 22.3% over the same stocks. It can also be observed from table 5.4 that higher AUC values are obtained for stocks like Apple, Amazon, Intel corp and Microsoft, however it decreases slightly for Google stock. The comparatively lower AUC value for Google stock still justifies the effectiveness of the proposed model as it shows an improvement over existing researches namely, 0.7896 for EMD-KDE approach [31] (authors previous research), 0.8025 for AHMMAS [78]. The rationale behind the effectiveness of the proposed approach can be explained by making the autoencoder learn the relationships among stock prices captured by the affinity matrix. In addition, the optimization of the autoencoder parameters while selecting the kernel density estimate of the dataset as the loss function leads to an improvement in the learning of the model. Moreover, the automatic selection of the KDE clustering based parameters

including independent selection of the number of clusters adds to the robustness of the proposed model.

5.3 Stock Price Manipulation Detection Using Contextually Learned Similarity Metric for Anomalous Trades.

Market manipulation can be explained as influencing legitimate trading rules with fraudulent practices pertaining to manipulators personal gains [1]. Over the years, markets have evolved with diversification in trading practices, globalisation, shear competition with more modern businesses being added every day. Since the markets are an integral part of the modern business world, they play a significant role in the economy of their respective countries. Indeed, most of the stock exchanges are being under constant monitoring by several regulatory authorities, market analysts and researchers in a bid to detect and identify market manipulation. However, it is computationally expensive both in terms of manpower and time. For example, nearly five years after the flash crash of 2010, US state department of justice arrested the man responsible of the trillion dollar crash in 2015 [2]. Stock price manipulation, as part of the trade-based manipulation [3] is related to influencing the trading price of financial security within a stock exchange using abusive schemes which consequently effects the faith and the predicted gross return from a stock. The process of manipulation detection and further conviction used was later described as ‘bicycles to try and catch Ferraris’ by Bloomberg [4]. Stock price manipulation can be described as the inflation or deflation of stock prices using illegitimate means. Some of those schemes that have been focused in this research include spoofing/layering, quote stuffing, pump and dump [9]. A description of their appearance and patterns for some of the real world cases has already been provided in chapter 1. It can be comprehended from the definition of such schemes that the objective behind such a process is to create a confusion about the amount of trading activity and to delay the normal processing of the investors.

The effect of most of these manipulation schemes is rippled across the stock price time series data. Detection and analysis of such schemes possess an

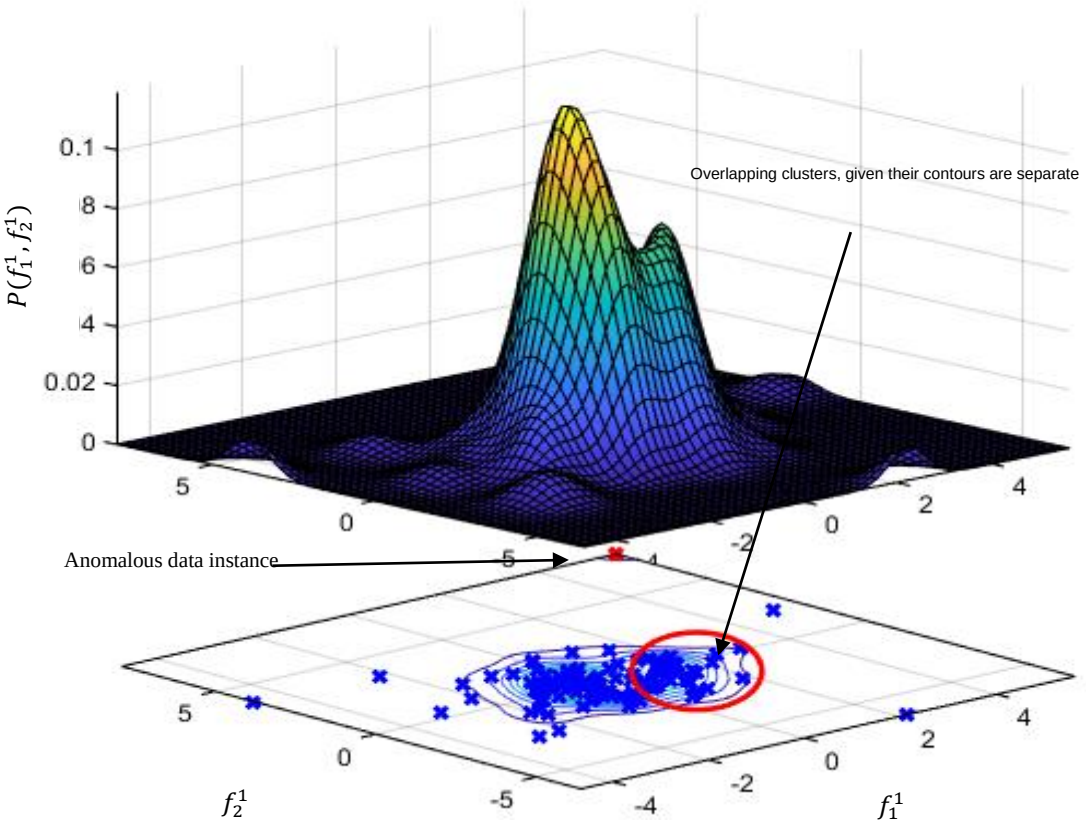


Figure 5-3: Probability distribution using kernel density estimate for a 2-D feature set ϵ $\{f_1^1, f_2^1\}$ of Apple Stock for 100 data points along with its contour

extremely difficult challenge for an overlap among normal and abnormal trades. This research proposes a solution to address and detect such anomaly trades. The idea is to define a context from the meaningful information extracted from the input and learn our proposed model upon it. It is proposed to combine a temporal convolutional network (TCN) with a general adversarial network (GAN), in the sense of utilising the temporal aspect of the TCN while using the learned discriminator representation of the GAN input to minimise the learned similarity metric for TCN. However, discriminative models struggle with anomalies for a complex data distribution especially in the case where there is no class decision boundary defined. The issue can be rectified by training the discriminator only on normal training samples and generative model on a mixture distribution of novel and normal data. So that for a fixed discriminator, any optimal generator having a mixture of normal and abnormal trades will be treated as novel by the discriminator.

This research is leveraged upon the work proposed in [151] to adapt a learned similarity metric avoiding the element-wise error. It is proposed to learn a similarity metric rather than using the general consensus of reconstruction error for an encoder-decoder network while incorporating convolutions in the temporal domain. The rationale for choosing such an alternative is that element-wise errors do not model the properties of human visual perception as a normal stock price variation could be a subject of major change when decomposed contextually. This is performed by collapsing the decoder of an encoder-decoder TCN (ED-TCN) and the generator of a GAN into one. To accomplish, an ED-TCN and GAN are jointly trained while using the discriminator representation of normal stock prices.

The contextual information input to the ED-TCN is a feature map from an auditory block within the model that also computes temporary cluster labels to it. A readily available clustering algorithm has been used that creates a distinction between various cluster patterns based on abnormality. A pattern so generated with normal and abnormal regions forms the basis of a contextual cue which can be learned by the model implemented further (illustrated in sec 3). The contributions made are summarised as follows:

- Combination of an ED-TCN and GANs into an unsupervised generative model which can learn the distinction between a normal and an abnormal distribution when efficiently trained on the latent variables that also reduces the complex computations significantly.
- This research demonstrates the learning and detection ability of a model for normal and abnormal overlapping data instances, can be significantly improved under a defined context.
- The model proposes an improved similarity learning metric for the TCN model and demonstrates that a generative model generates better results than the conventional reconstruction error based model.
- The research model learns the established clustering patterns as a context defining a spatial normal-abnormal data pattern. The clustering algorithm

used is a multidimensional kernel density estimation based clustering (MKDE) [29]

5.3.1 Contextual learning

The rationale behind contextual learning for manipulation detection originates from the image segmentation/ object detection either in static or dynamic domain [152]. As the stock price time series is volatile, evolved manipulation strategies influence the normal trading in a way that does not raise alarms alerting the regulatory authorities i.e., they are overshadowed by normal trading waveforms. As uncertain number of factors can influence the stock prices, it is infeasible to pick and predict manipulation just by looking at them. In order to detect such anomalies, contextual analysis is applied in time domain while leveraging the similarity metric for the model. It has become a well-accepted fact that contextual information can influence a decision about abnormalities as it enhances the perception and understanding of the data [152]. The utilisation of such information can be used to improve the overall performance if carefully exploited.

5.3.1.1 Contextual estimation: Multi-dimensional kernel density based clustering

The first step in estimating the context for the model is defined using multi-dimensional kernel density estimation clustering. MKDE based clustering is one of the authors previous work on manipulation detection [29]. With the added advantage of not assuming number of clusters a priori, MKDE clustering assigns normal and abnormal labels to the data. The method suggests calculating a non-parametric density estimate for a dataset and group them into different clusters based on the Algorithm 1 shown in chapter 3 (section 3.4.3).

A cluster of stock price instances is created if the difference between the mean of the kernel density estimates and the data instances if less than the bandwidth g. For all of the remaining instances, the process is repeated with new mean and the bandwidth until convergence. In case of a multimodal distribution, each mode is associated with its bandwidth parameter which leads to multiple clusters so formed. Such a situation is reviewed again once all the clusters are formed. The clusters that are overlapping each other, as shown in Figure 5.3 are

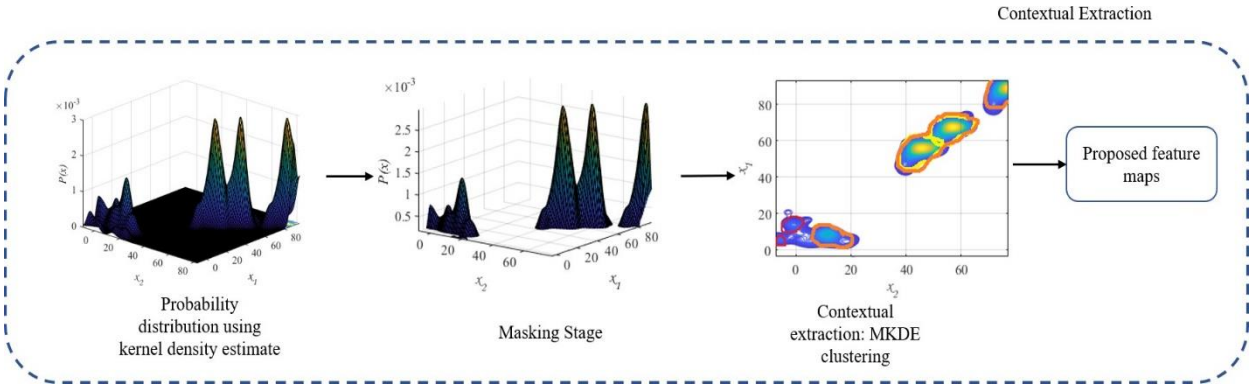


Figure 5-4: Analysis of the clusters generated using kernel density estimate from Algorithm 1 for extracting contextual relationship incorporated within proposed feature maps

combined into one if (i) the separation among them is less than the bandwidth and (ii) the ratio of their peak density estimate is less than 0.7. The instances left unclustered will be considered as anomalous.

5.3.1.2 Analysis of the cluster pattern for contextual estimation

The clustering arrangement so formed is further analysed towards the generation of an intrinsic feature map with assigned labels as part of the contextual estimation. Figure 5.4 describes steps in the analysis and the incorporation of the contextual information in the form of contextual feature maps. First, the established clustering pattern is evaluated and passed through a masking stage. Then the identified normal and abnormal regions within the density matrix are isolated by suppressing or masking probability densities of other regions to zero. The masked intrinsic relationships represented by the spatial arrangement of normal and abnormal instances separated in the form of clusters is defined as a context. This research avoids generating contextual scores for different regions as some of the existing research [152] given that it requires annotated data for generating heat maps for each class with the probability intensity used as scores [152]. For each of the clusters, the multidimensional kernel density distribution $P(x; g) \sim \mathcal{N}(\mu_c, \Sigma_c)$; for mean, μ_c and covariance, Σ_c can be obtained from the previous stage of MKDE clustering. Where,

$$\mu = [\mu_{x_1}, \mu_{x_2}, \dots \mu_{x_n}], \Sigma_c = \begin{bmatrix} \sigma_{x_1} & 0 & & 0 \\ 0 & \sigma_{x_2} & \dots & 0 \\ \vdots & 0 & & \vdots \\ 0 & 0 & & \sigma_{x_n} \end{bmatrix} \forall C$$

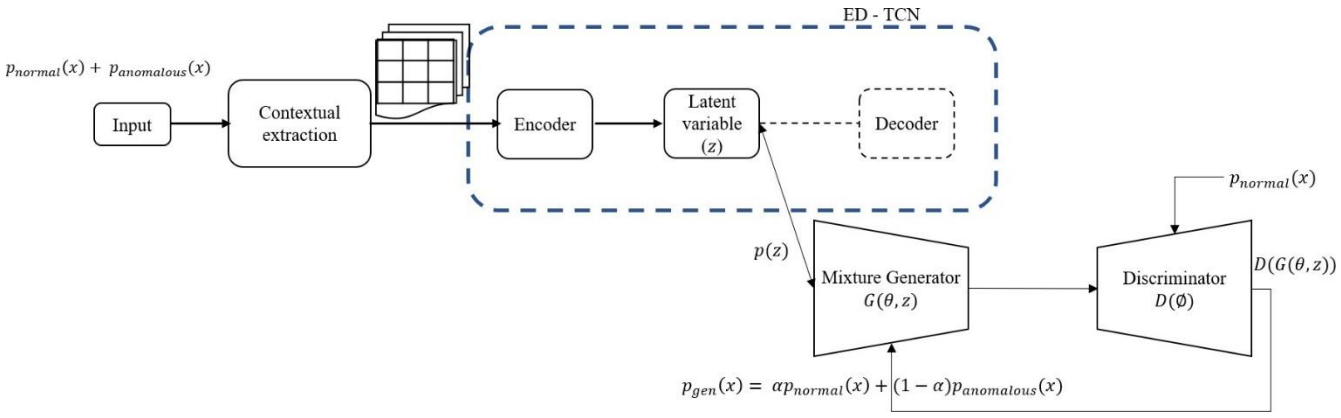


Figure 5-5: Overview of the proposed model architecture. A combined TCN-GAN model is shown with processing of the proposed feature maps with contextual information embedded into it.

The feature map $\forall C$ generated for an underlying dataset is now reduced to,

$$F(x) = \begin{cases} \sum P_C(x; g) & \forall C \\ 0 & otherwise \end{cases} \tag{5.12}$$

For n dimensional input, the output feature maps are $(w * n * B)$, where w is the batch size of the input time series and B is the tensor that represents the size of the density matrix for the input. It should be noted that the output at this stage is a mixture of multivariate kernel density based distribution for all of the non-masked normal and abnormal clusters. As an example, for Amazon stock a tensor of 926 MB storage is created.

5.3.1.3 Contextual learning: Temporal convolutional network (TCN)

The TCN model used in this research is essentially an encoder - decoder (ED-TCN) network involving convolutions in the temporal domain while encoding and deconvolution in decoding. The idea is to encode an input data sample x into a latent variable y and then decode it back. The original concept of an ED-TCN is explained in [153] in which a convolution layer followed by pooling during encoding and upsampling followed by deconvolution during decoding. In a conventional TCN, convolutions are calculated simultaneously for all of the data instances within a fixed length of time. Computations are carried out layer wise, followed by pooling that efficiently works with long term temporal

patterns. However, in this research pooling and upsampling are avoided based on the rationale explained in section 4. For an input x, the encoder, decoder output at any layer l for the TCN can be expressed as,

$$E_l = f(W * E^{l-1} + b); \tag{5.13}$$

Where $f(\,)$ is an activation function, $W = \sum W^{(i)}$ is a collection of convolution filters at layer l, convoluted $(*)$ with the encoder output from the previous layer and added with a corresponding bias b. This output from the encoder can also be expressed in the form,

$$E^l = q(y|x), D^l = q(\tilde{x}|y) \tag{5.14}$$

Where $q(y)$ represents the marginal distribution of the encoded or latent output y. Lets focus on the TCN loss function for the encoder that can be expressed in the form of marginal likelihood for every latent data instance,

$$\log q(x) = \mathcal{L}_{TCN}(\theta, x) + D_{KL}(q(y|x)\|p_\theta(y)) \tag{5.15}$$

Where, θ is the set of encoder parameters and D_{KL} represents the Kullback-Leibler divergence [154] of the approximate posterior. The first term in (5.15) can also be written as,

$$\mathcal{L}_{TCN}(\theta, x) = \mathbb{E}_{p(y|x)}[-\log p(y|x) + \log q_\theta(x, y)] \tag{5.16}$$

which can be further simplified to,

$$\mathcal{L}_{TCN}(\theta, x) = D_{KL}(q(y|x)\|p_\theta(y)) + \mathbb{E}_{p(y|x)}[\log q_\theta(x, y)] \tag{5.17}$$

5.3.1.4 Contextual learning: Semi-supervised generative adversarial networks (GAN)

GAN proposed by [155] is a network architecture composed of two individual neural networks, a generative network $G(\theta, y)$ and a discriminative one $D(\emptyset; \tilde{x})$, both competing with each other. Where the generative network $G(\theta, y)$ maps the input latent data sample y to the data point x, given that this latent variable is sampled from a marginal distribution $p(y)$ and parameters θ. As mentioned before in the introduction, the decoder of the ED-TCN model and generator are sharing the parameters, it is presumed to consider the same θ for the generative

modelling as well. The discriminative network, having parameters $\emptyset$ on the other hand discriminates the generative output against the original training input and assigns a probability $p_{disc}(x)$ that x is generated from original training input and 1 - $p_{disc}(x)$ that x is from the generator based on the latent input y with $p_{gen}(y)$. The objective here is to optimise the whole network using the loss function based on binary cross entropy,

$$\mathcal{L}_{GAN} = \min \log D(\emptyset, \tilde{x}) + \max \log (D(G(\theta, y))) \quad (5.18)$$

GANs were originally proposed for generative modelling by [155] but ever since there has multiple variants of GANs including CycleGANs [156], StyleGAN [157], AnoGAN [158] etc. along with multiple applications like text to image generation [159], super resolution video [160] etc. This research is adapted to implement GANs in a semi-supervised learning environment and use it for anomaly detection by leveraging upon the work done by [161]. Saliman et al. (2016) [162] proposed and proved that a semi-supervised classification can be achieved using GANs by optimising a feature matching loss L_{f_m},

$$\mathcal{L}_{f_m} = \min_{\theta} \left\| \mathbb{E}_{x \sim p(x)}[\mathbf{f}(x)] - \mathbb{E}_{y \sim p(y)}[\mathbf{f}(G(\theta, y))] \right\| \quad (5.19)$$

where $\mathbf{f}(x)$ is the discriminator output used a feature representation of x and $p(x)$ is the nominal data distribution. This proof of concept was later confirmed by Dai et al. (2017) [163] by defining a complement generator that generates samples from a scattered around and in a low density region of the nominal data distribution manifold and claims that the discriminator classification rate improves for such novelties. This research also leverages upon the work done by Dai et al., 2017 [163] that states a unique generator that can generate a mixture of nominal and anomalous distributions $p_{gen}(y) = \alpha * p_{nominal}(y) + (1 - \alpha) p_{anomalous}(y)$ where a portion of $p_{anomalous}(y)$ is either separated from $p_{nominal}(y)$ by a given Euclidean distance or is concentrated in the low density regions of it. The concept that can be borrowed from the MKDE clustering explained in the contextual estimation section. Such a mixture generator can be optimised using a loss function that computes the KL-divergence between the generative distribution and the real data distribution. More specifically (5.18) can be written as,

$$\mathcal{L}_{GAN} = \min_{\theta}\left[-\mathcal{H}(p_{gen}(y,\theta))\right] + \mathbb{E}_{y\sim p_{gen}(y)}\log p(x) + \mathcal{L}_{f_m} \qquad (5.20)$$

where $\mathcal{H}(\,)$ is the entropy function and $p(x) \in (nominal;\ anomalous)$ is the real data distribution. (5.11) defines a loss function that is able to create a generative distribution that is close to the normal data manifold ($\mathcal{L}_{f_m}$) of the discriminator and also tries to minimise the distinction between generator distribution and the real data. Fortunately, to train a GAN based on (5.11), the real data estimate can be provided from MKDE clustering with clusters that defines normal and anomalous regions within a context. It should be noted that the discriminator used here is only trained on normal class of data i.e., if it classifies a context as fake, it is likely that it consists of anomalous instances. To summarise, this research proposes to implement a GAN that generates data instances as a mixture of nominal class distribution along with the ones scattered in the low density regions of it. Such a GAN is trained alongside feature matching loss and the anomaly detection ability of the discriminator is judged based on a threshold applied to the ratio between the nominal and fake class probability.

5.3.2 Detection model

As mentioned briefly in the introduction section, the discriminator can objectively distinguish a normal data sample from a manipulative one when only trained on normal data. In other words, a GAN discriminator has to learn a similarity metric that can help classify normal data samples from the ones containing a mixture of normal and abnormal samples. The research model shown in Figure 5.5 explains the processing of the data samples beginning from contextual extraction using MKDE clustering followed by the masking stage. The output of such a stage are density matrices wherein the non-cluster regions, are suppressed to zero. The spatial arrangement of density distribution described in the output stage governs the contextual relationship and also explains probabilistic position, size and Euclidean distance among normal and abnormal data instances. This contextual relationship among normal and abnormal clusters is further explored using the tempGAN model where the similarity metric is determined by using the kernel convoluted input as the latent space y to the generative network of GAN. The convoluted output of the encoder in temporal

domain is considered as the latent variable for the generative model and since both decoder (ED-TCN) and generator (GAN) map the latent variable y to x, their parameter θ are also shared. This accumulates the advantage of both GAN as a semi-supervised generative model and ED-TCN that encodes the contextual relationship onto a latent space y. For a conventional temporal convolution network, the convolution layer is followed by pooling to reduce or down-sample the input dimensions. However, this research avoids the loss in the temporal resolution that can originate using fewer dimensions and also force upon the generative estimation by using fixed size latent input.

This research also avoids the conventional reconstruction error for the ED-TCN model. Instead, it is proposed to learn a similarity metric using GAN discriminator. This can be achieved by calculating the expected log-likelihood of $p(\mathrm{f}(x)|y)$, where $\mathrm{f}(x)$ is the discriminator output assumed to be concentrated around $\mathrm{f}(\tilde{x})$ as the mean, $\tilde{x}$ being the sample from the decoder. The ED-TCN loss in (5.18) can now be written as,

$$\mathcal{L}_{TCN}(\theta, x) = D_{KL}(q(y|x)\|p_\theta(y)) + \mathbb{E}_{q(y|x)}[\log q_\theta(\mathrm{f}(x)|y)] \qquad (5.21)$$

The model can now be trained by optimising the combined loss function given by,

$$\mathcal{L} = \mathcal{L}_{GAN} + \mathcal{L}_{TCN} \qquad (5.22)$$

Although the whole model is trained altogether, the optimisation of $\mathcal{L}_{GAN}$ and $\mathcal{L}_{TCN}$ are two separate processes i.e., the generator network is optimised by minimising $\mathbb{E}_{q(y|x)}(\log q_\theta(\mathrm{f}(x)|y))$ and $\mathbb{E}_{y\sim p_{gen}(y)} \log p(x)$ with weights being updated based on the parameter α. Similarly, the discriminator trying to optimise by minimising the feature matching loss updating the parameters $\emptyset$ and avoiding the backpropagation from $\mathcal{L}_{GAN}$ to TCN encoder as suggested by [151].

5.3.3 Experiments

5.3.3.1 Dataset Used

This research is validated on the stock prices and volumes taken from two different source; order-books from LOBSTER project [48] and from the Bloomberg platform at Newcastle Business School (NBS), Northumbria

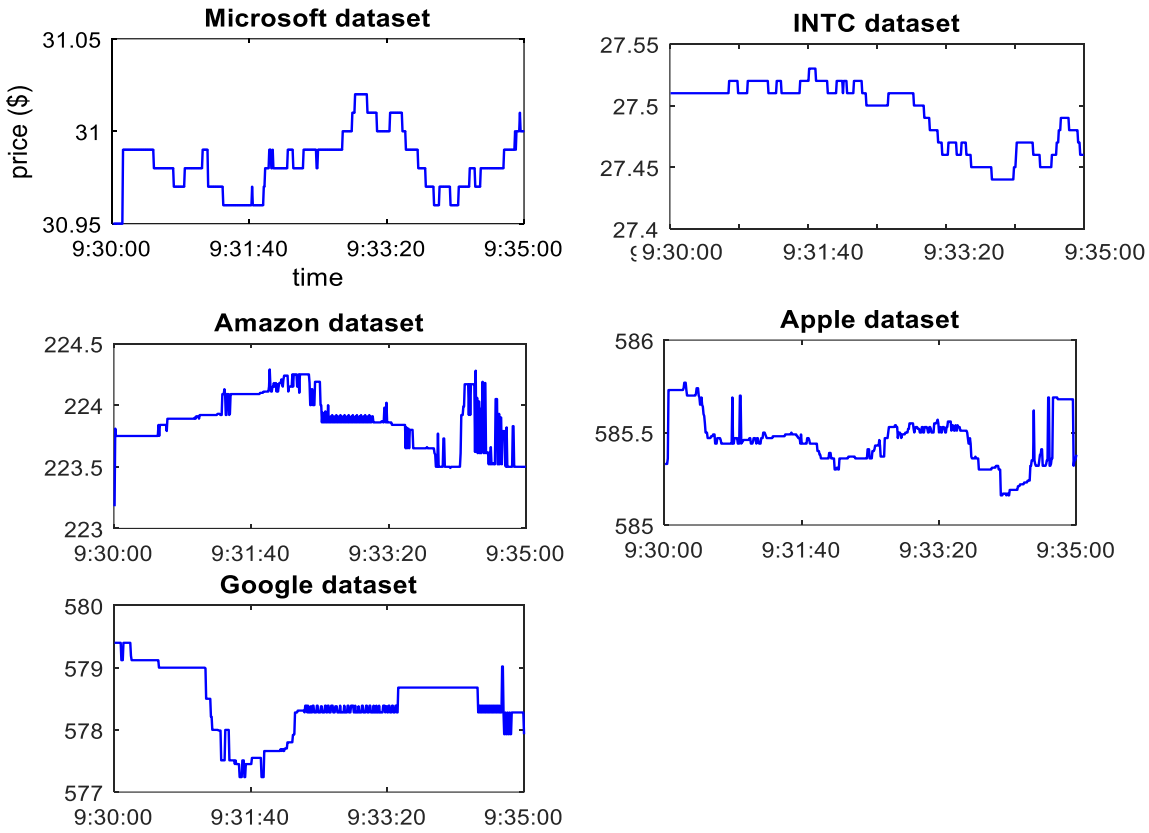


Figure 5-6: Varying bid prices of different stocks from 09:30:00 AM to 09:30:52 AM on 21st June 2012

University, UK. The dataset from LOBSTER project consists of level-1 tick data for Apple, Amazon, Google, Intel Corp and Microsoft stock prices recorded on 21st June 2012 and is a standard dataset used for multiple researches [29, 33, 78, 164]. Whereas the dataset from NBS includes level-1 tick data from Apple, Amazon, Microsoft, Google, Intel Corp, SIRI, EBAY, Cisco, Nvidia, Facebook, Netflix, QUALCOM and AMD recorded on 12th November 2018. The rationale behind specifically selecting these stocks is the high trading frequency, high volatility, and the massive trading amount (>~ 1 million trades/day) for each stock has been observed, making it more vulnerable to manipulation. Additionally, the stocks from LOBSTER project and NBS dataset are not reported for any manipulation [29, 48, 78]. Figure 5.6 shows the variation in stock bid prices from LOBSTER project over a certain period of time. Since both the datasets are free from manipulation, in order to test the validity of the proposed model, synthetic anomalies that mimics the exact representation of major manipulation schemes are added to these stocks. Manipulation schemes

focused in this research includes spoofing, pump and dump, quote stuffing and are randomly injected along the time series in significant amounts. The data is also categorised into two different types Group I: LOBSTER dataset (200,000 - 800,000 average trades bid/ask) and Group II: NBS Bloomberg dataset (>~ 1 million average trades bid/ask) based not just due to the source but also due to varying size. For the same reason of varying number of trades, the number of injected anomalies/type also varies). Added anomalies in Group I: 200 anomalies * 3 types = 600 manipulation schemes, Group II: 300 anomalies * 3 types = 900 manipulation schemes. Such an arrangement of data distribution is acceptable as per the business standards [107, 165] and has been extensively used by [29, 55, 78] provided that none of the regulatory authorities have reported any abnormalities with both LOBSTER and NBS datasets.

5.3.3.2 Experimental Setup

To ensure an impartial evaluation of the detection model none of the relevant information including the location, size and duration of the manipulative waveforms were provided *a priori* to the approach. The manipulations are also randomly inserted which makes the detection more difficult as it possible that the manipulation is followed/overlapped in time by a similar pattern which makes could further increase false positives. The input time series is heuristically windowed into a fixed sample size of 500 instances prior to providing to the model. The contextual relationship extracted using MKDE clusters and masking in the form of density matrix of size 500*500 are provided to the tempGAN model for further evaluation. Here ED-TCN parameters are learned by using two layers of 1-D convolutional filters of size [64,96] in encoder and its reciprocal in decoder/generator. The convolution of input with the filters is performed in an acausal manner, $X_{t-d/2}$ to $X_{t+d/2}$, for a filter of length d at any time instant, resulting in predictions in time will be a function of not just the previous but also future instances. The discriminator network architecture includes two convolutional layers of size [64,128] followed by a fully connected layer with normalised ReLU,

$$normReLU = \frac{ReLU(x)}{ReLU(x) + \epsilon} \tag{5.23}$$

for any input vector x and $\epsilon = 1e-5$ and linear activation function, respectively. The rationale behind the selection of such activation functions is their extensive use on time series data [153, 166, 167] including market manipulation detection [33, 55]. The research also finds the best results using a normReLU activation function throughout the model i.e., encoder, decoder/generator, and discriminator.

The detection results reported in the form of F-measure, Area under the ROC curve and false alarm ratios (FAR) are compared with the existing manipulation detection models, some of which claimed significant detection results. However, for a comprehensive assessment of the approach, existing models that focused only on detecting a specific manipulation scheme rather than making it generic are avoided for comparison. For this purpose, contemporary state-of-the-art models like AHMMAS [78], Affinity based detection [33] and KPCA-KDE method [29] for stock price manipulation detection were selected. Although, the evaluation metrics considered in this research AUC and ROC curves are often related to supervised learning with labels, it has often been used in unsupervised domains [109, 110].

The next section discusses and examine in detail the results obtained for the experimental setup described here and finally concluded in the subsequent one.

5.3.4 Results and Discussion

The types of the manipulation schemes injected into the original dataset creates a typical pragmatic case of manipulation that translates the real life manipulation ROC curves for the proposed approach as shown in Figure: 5.7 are also varied over input window length to capture the effects of the changing contextual relationship within on the manipulation detection. The approach is repeatedly applied over the dataset with varying window sizes. It is essentially carried out to avoid the amount of uncertainty over the input sequence length. However, it is difficult to observe a major change in the detection capability of the model. This can be explained by the process of collecting contextual information and robustness of the model towards varying receptive fields which can be extremely helpful in overcoming the number of computations associated with large amount of data. As mentioned in the previous section, the comparison of the proposed

Table 5.5: Comparison of Stocks in terms of AUC values with a selection of existing manipulation detection techniques

AUC	Amazon	Apple	Google	Microsoft	Intel Corp
tempGAN	**0.8180**	**0.8200**	**0.7712**	**0.8302**	**0.7991**
PCC [5]	0.7510	0.5751	0.7327	0.7212	0.7276
OCSVM [2]	0.7444	0.5502	0.4925	0.5349	0.5808
Knn-LOF [168]	0.5448	0.5685	0.5246	0.5059	0.5469
Knn-COF [2]	0.5310	0.5365	0.5084	0.5147	0.5809
BGM [169]	0.5122	0.5245	0.5119	0.5592	0.5169
K-means– Bergman Divergence [10]	0.5271	0.5384	0.5650	0.5122	0.5077

model is made with the existing state of the art anomaly detection models both in stock price and generic. For an input stock prices of 500 data instances, a contextual feature map of $F_{500*500}$ having probability density, only for the observed clusters is generated. This information classed into normal and abnormal densities is then provided to the tempGAN model where the objective is to establish a similarity metric contemporary to the common convention of reconstruction. It can be observed from Table 5.5 to 5.7 that the proposed approach outperforms the existing anomaly detection approaches for all the stocks in LOBSTER project in terms of AUCs. It should be noted that most of the compared anomaly detection techniques such as PCC [82], Knn- LOF [168], Knn-COF [65], OCSVM [65], Bayesian Gaussian mixture model using drichilet process (BGM) [169], K-means– Bergman Divergence [117] were originally proposed as generic detection models. For a fair comparison proper replication is also taken care for stock price data. In terms of AUC, it can be easily interpreted from the table that the proposed approach outperforms the existing anomaly detection models with the best possible explanation of exploring the class discrimination among normal and abnormal instances under a context. The class discrimination capability of the proposed approach is also compared with the state-of-the-art manipulation detection models in Table 5.7. The comparison is however limited in terms of the evaluation metrics and the specific manipulation type, as replicating those models is restricted by the lack of parameter values missing from their research methodologies. It can be easily

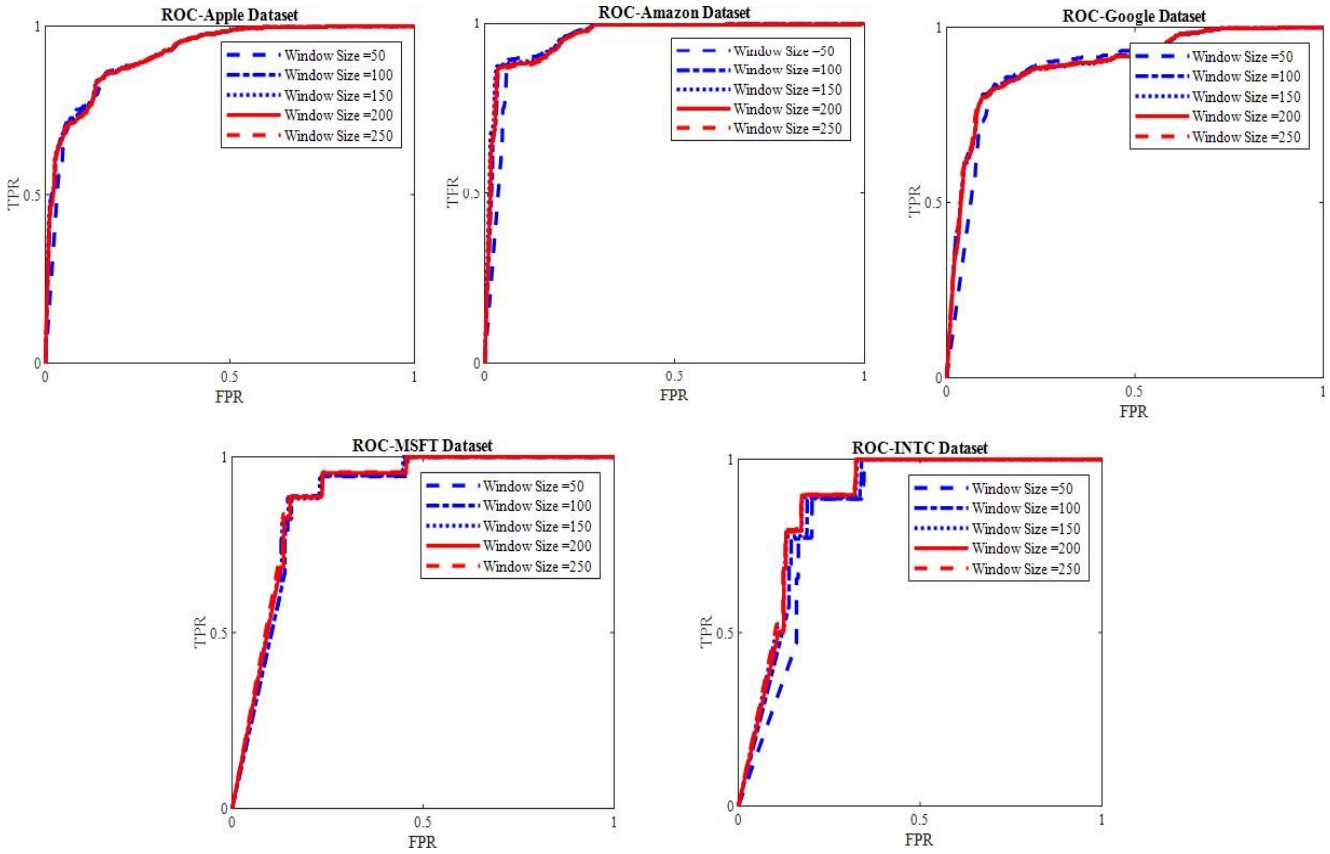


Figure 5-7: Effects of varying window length ⊂ [50,250] on Group I stocks' ROC

observed from Table 5.6 that for all stocks, tempGAN model performance is comparable with KPCA-KDE method provided that the KPCA-KDE detection model is totally unsupervised. However, it is also noted that the performance of KPCA-KDE detection model decreases with the rising amount of data and with increasing number of manipulation instances whereas for a tempGAN model, the performance seems to be independent of the size of input data as is visible in Figure: 5.7. This can be confirmed by following up the results from the original KPCA-KDE model proposed in [29]. It is worth exploring the comparatively low AUC value for Google over other stocks which can be associated with the extremely high volatility within. However, the detection rate for such behaviours can be improved with observing the time series under a regular sampling rate which can instate the order-book into a continuous time series rather than a discrete one. More importantly, the FAR results from Table 5.8 reimburse the performance by improving the score by at least 1.2 %.

A further evaluation of the proposed approach is carried out in terms of False alarm rates (%) and F-measure. The corresponding values are summarised in table 5.7 & 5.8. The F-score from Table 5.7 also suggests the significant improvement of the proposed model over the existing anomaly detection methods. It is observed each individual improvement is 2.697% for Amazon stock, 9.70% for Apple, 17.39% for Google stock, 12.11% for Microsoft stock and 20.52% for Intel Corp stock. It can be easily interpreted from the tables that although tempGAN model surpassed the existing anomaly detection models in

Table 5.6: AUC comparison of tempGAN with the selected state-of-the-art manipulation detection models in stock prices

AUC	tempGAN	KPCA-KDE	AHMMAS
Amazon	**0.8180**	0.7723	0.7652
Apple	**0.8200**	0.7671	0.8044
Google	**0.7919**	0.7496	0.7812
Microsoft	**0.8302**	0.7619	0.7311
Intel Corp	**0.7991**	0.7233	0.8169

Table 5.7: Comparison of stocks in terms of F-Score values with a selection of existing manipulation detection techniques

F-Score	Amazon	Apple	Google	Microsoft	Intel Corp
tempGAN	**0.3655**	**0.5667**	**0.5137**	**0.6378**	**0.6379**
PCC [5]	0.3559	0.5161	0.4375	0.5688	0.5292
OCSVM [2]	0.1568	0.0380	0.0616	0.0106	0.0106
Knn-LOF [168]	0.0284	0.0037	0.0163	0.0127	0.0157
Knn-COF [2]	0.1714	0.0286	0.1125	0.2566	0.2142
BGM [169]	0.1840	0.0510	0.1148	0.2641	0.2236
K-means– Bergman Divergence [10]	0.0102	0.0010	0.0061	0.0387	0.0218

terms of AUC and F-score, it requires an explanation for the degraded values in terms of FAR for all of stocks. Further study into this issue identifies the false positives associated with spoof trading manipulation. However, one of the shortcomings here is the use of level-1 tick data given details about the order cancellation are missing. Such information is critically important as the price fluctuation (which is seemingly high for spoof trading) is deeply correlated with the volume change.

It is also evident from Table 5.8 that Knn-LOF method suggested by [168] provides better results in terms of FAR. Although, Knn-LOF model also considers the local or contextual densities, the ambiguity in selecting the value of several parameters in Knn is a slight disadvantage to its credibility while processing the large amounts of data such as the stock prices. Nevertheless, the overall performance of the tempGAN model can be appreciated as is substantial for AUC, F-score and FAR for all of the stocks i.e., tempGAN model improves the detection rate whilst lowering the false positives up to a considerable extent.

The tempGAN model is also applied on to Group II: The stock prices for Eleven stocks obtained from Bloomberg trading platform at NBS. The rationale behind the selection of such stocks is already mentioned in section V. Table 5.9 and 5.10 shows the comparative analysis in terms of F-score and False alarm rates for the stocks. As is evident from tables the proposed model justifies being efficient and clearly outperforms the existing models for stock price manipulation detection. It is worth mentioning here that only few anomaly detection techniques are used for comparison i.e., K-means–Bergman Divergence and Knn based (used for Group I) methods were avoided since the results obtained were minimal and hence non-comparable.

To summarise, the experimental results obtained by the proposed tempGAN model computes a higher detection rate in stock price manipulation when executed for three different schemes viz. Spoof trading, Pump and Dump and Quote stuffing. Such manipulation schemes were carefully observed and replicated following the real life manipulation cases reported by SEC [20–22]. The proposed work is compared with the existing state-of-the-art manipulation detection techniques independent of the specific manipulation type. The experimental performance is also compared with the existing anomaly detection techniques in time series and proved to outperform them. Such a significant performance by the proposed work can be associated with the macro management of the manipulation behaviour exploited within a context. The application of the kernel density based clustering on the input data revealed a contextual relationship between normal and manipulative trading. Such relationship in the form of an adapted density matrix is further explored by a

Table 5.8: Comparison of Stocks in terms of FAR values with a selection of existing manipulation detection techniques

FAR	Amazon	Apple	Google	Microsoft	Intel Corp
tempGAN	2.12	2.09	2.54	1.62	1.63
PCC [5]	2.32	2.17	2.72	1.81	1.64
OCSVM [2]	5.0	7.74	8.32	50.99	58.39
Knn-LOF [168]	50.64	68.9	76.3	78.58	60.18
Knn-COF [2]	**1.24**	1.55	**1.78**	**1.18**	**1.33**
BGM [169]	2.87	**1.465**	2.217	5.27	5.30
K-means–Bergman Divergence [10]	10.32	8.93	16.23	16.75	22.25

temporal convolutional network combined with a generative adversarial network. The robustness of the proposed tempGAN model can be explained by the similarity metric established and the independence from element-wise metrics such as mean squared reconstruction error.

For manipulation schemes such as spoofing and pump and dump, a sudden snap after a long-held position of linear/nonlinear rise or decline in stock prices raises concerns about the length of the input. This research explored into this issue by repeating the same experiment for variable lengths of the input data instances. It is found that the proposed model remains intact to the detection results in AUC ignoring minor change in values for INTC and Microsoft stocks. Although the research is unable to explain the variations in them but further investigation into the same reveals the less volatile nature or a piece-wise constant for a significant amount of time in them compared to the rest of the stocks.

5.3.5 Conclusion

This chapter presents two computationally intelligent approaches for price manipulation detection. The first proposed approach is based on the blend of affinity matrix and KDE clustering independent of the annotated data. The research aimed at the detection of two different types of manipulation schemes using fully unsupervised learning. For this purpose, a standard dataset

(reportedly free from manipulation) is considered and to evaluate the effectiveness of the approach, it was injected with significant number of manipulative trade instances. Such a dataset having a combination of both normal and abnormal trades was further processed to compute an affinity matrix from a small set of features extracted. An autoencoder pre-trained using the density distribution of the normal dataset was used to extract the encoded features while providing the affinity matrix as an input. The encoded data was then subjected to a proposed KDE approach for clustering and the data instances left un-clustered are treated as manipulation. Finally, the obtained results were compared with a selection of existing manipulation detection techniques like kNN based, PCA based, OCSVM based, and K-means based. In order to check the robustness of the proposed approach, it was evaluated in terms of AUC, F-Score and FAR. The approach was also compared with some existing benchmark research in stock price manipulation detection.

It was observed that the proposed approach clearly outperformed the existing methods in terms of AUC values, improved the F-score and reduced the false positives while avoiding the annotated data. The significant improvement in results can be attributed to improved learning of the AE using the information captured by the affinity matrix and preserving the information in the encoded features. However, there is possibility to further improve the proposed detection approach by identifying the type of manipulation being detected. In addition, an independent selection of some parameters like window length and threshold values for feature extraction is also a matter of future research.

A second novel approach towards stock price manipulation detection using semi-supervised learning. A brief review of the updated literature in detecting market manipulation has also been presented as part of the research. The proposed model is validated on stock price data from two real-life datasets taken from LOBSTER project and Bloomberg trading platform at NBS, Newcastle upon Tyne, UK. A total of 13 different stock were considered based on the volume and messages generated within a day. These stocks were injected with manipulative instances mimicking three different types of manipulation schemes. This research studies a cluster information for stock price manipulation

Table 5.9: Comparison of group II Stocks in terms of F-Score values with a selection of existing manipulation detection techniques

F-Score	FB	NVDA	QUALCOM	EBAY	CSCO	NFLX	SIRIUS	AMD	INTC	MSFT	AAPL
tempGAN	0.6937	0.6302	0.7429	0.7549	0.5849	0.6420	0.6990	0.6715	0.5812	0.7693	0.565
KPCA-KDE	0.7332	0.5647	0.6888	0.6912	0.5068	0.5678	0.6327	0.6057	0.5255	0.7083	0.4997
PCC	0.5582	0.2823	0.4137	0.4733	0.2946	0.3422	0.4291	0.3384	0.3092	0.6064	0.2940
BGM	0.3882	0.2670	0.3759	0.2021	0.6161	0.2031	0.2966	0.1769	0.3002	0.4786	0.2477
OCSVM	0.2617	0.1084	0.2319	0.0338	0.4252	0.0854	0.1786	0.0496	0.1224	0.3748	0.1301

detection where the proposed model learns an inter-relationship of normal and abnormal clusters introduced as a context. The proposed model is a combination of temporal convolutional network and generative adversarial network designed to capture the high-level structure of the data distribution using an independent similarity metric avoiding the conventional element wise squared reconstruction error. The model can be considered as a basic encoder-decoder network with the added temporal feature, critical to the stock price manipulation detection. The proposed model is trained upon a combination of normal and anomalous instances by treating the decoder network as a combination of autoencoder's decoder and a GAN generator. The resulting output being judged as normal or fake at the discriminator stage.

It is interesting to note that the discriminator output is leveraged by using convolutional features as the latent space whilst providing the updated similarity measure. This is also be accredited to the exploitation of the time series in a contextual domain and clearly establishing a boundary among normal and abnormal data samples. It can be easily observed that the proposed model outperformed the existing manipulation detection techniques in terms of

Table 5.10: Comparison of group II Stocks in terms of FAR values with a selection of existing manipulation detection techniques

FAR	FB	NVDA	QUALCOM	EBAY	CSCO	NFLX	SIRIUS	AMD	INTC	MSFT	AAPL
tempGAN	0.0366	0.0376	0.0501	0.0345	0.0397	0.0418	0.0407	0.0261	0.0428	0.0261	0.0470
KPCA-KDE	0.0570	0.0541	0.0840	0.0521	0.0625	0.0471	0.0663	0.0449	0.0552	0.0361	0.0528
PCC	0.5378	0.4818	0.6432	0.5854	0.5714	0.5301	0.5769	0.5276	0.4734	0.6399	0.5767
BGM	0.0652	0.0383	0.0496	0.0352	0.0501	0.0459	0.0409	0.0518	0.0420	0.0350	0.0491
OCSVM	0.3452	0.2481	0.5814	0.1977	0.2514	0.2188	0.3756	0.4487	0.8033	0.1409	0.5644

improving AUC, F-score and diminishing false alarm rates. The robustness of the proposed model was also validated over varying input length. However, it would be interesting to investigate the dilated TCN model instead of ED-TCN employed in this research and to explore the additional details of order cancellation at a level-2 depth of order-book. Moreover, the detection can be further improved by classifying the detected manipulation into its specific type. In principle, to further improve the detection performance, subjectively identify the type of the abuse with a degree of certainty.

Chapter 6: Conclusions and Future Work

6.1 Summary

Stock market surveillance has gained immense popularity since the market crash of 2010. Even years after the incident that wiped out nearly $1 trillion of the market globally in a span of few minutes, market is still recovering from the panic and shock that deprived investors of their faith and trust. One of the main causes of flash crash was manipulative schemes in high frequency trades. To prevent such incidents and others alike, major market regulations and policies over manipulation has been updated regularly and adopted by emerging markets over the course of time. Despite the agility of the market regulators and researchers that study and propose algorithms and methodologies that claim to address such issues, market manipulation detection still is an extremely challenging task for reasons like bias in the detection models using supervised approaches, models that only address specific manipulative schemes and the choice of parameters being heuristic, lack of quantitative analysis of manipulative cases that can describe characteristics/features of manipulated time series. It is therefore necessary to analyse inherent distribution of stock prices and consequently propose detection algorithms that are independent of manipulative schemes. The scope of this PhD thesis is focussed on broadly analysing different cases of trade based manipulation and develop surveillance algorithms by treating manipulative instances as anomalies considering the practical limitations of manipulation cases being rare.

To address the issues mentioned throughout the thesis, Chapter 2 presents a detailed review of literature that first explores manipulative schemes with example cases along with research that presented solutions for them either individually or targeting a few of them as a generic approach. Initially, an introduction three categories of market manipulation; action based, information based, and trade based manipulation was presented followed by state of the art research that claimed significant improvements in price manipulation detection rates. A total of five different categories of research ranging from Bio-inspired models, detection using decomposition based mixture models, clustering based

detection techniques, models using multiple algorithms for comparative analysis to deep learning models for manipulation detection were presented. Additionally, anomaly detection approaches that have been frequently used for time series anomaly detection and can potentially be implemented over stock prices were also presented. All of the existing research mentioned in this chapter, either associated with market manipulation detection or anomaly detection were thoroughly reviewed and limitations of them were highlighted.

Chapter 3 explained manipulation detection using decomposition techniques including principal component analysis, Dirichlet process Gaussian mixture models, empirical mode decomposition and kernel principal component analysis. It introduced two novel detection models using empirical models and kernel principal component analysis focussed on manipulative schemes like pump and dump and spoof trading. To address the types of manipulations, case studies for specific manipulative schemes were studied and relevant features that can capture such effects were extracted. Based on the extracted features, instantaneous mode functions (IMFs) were computed using the EMD algorithm which are further subjected to the proposed KDE clustering algorithm for manipulation detection. The chapter also proposed a detection model using kernel principal component analysis in higher dimensions and scaling the data points in the kernel space to increase the spread and forwarding this effect onto the transformed space. Such scaling helps the multi-dimensional kernel density estimation based clustering to group normal and manipulative instances. The proposed work also discussed issues related to the choice of the kernels, parameters, and the length of the input samples to be fed to the model. The research presented significant improvements in terms of detection rates and in reducing false positives. To prove the robustness of the proposed model, several parameters within the model were also varied and the effect of them on the results were explored. It is found that the proposed model was immune to the variation and the results were not significantly affected.

Chapter 4 presented an immune inspired Dendritic Cell Algorithm approach for detecting stock price manipulation by visualising two types of manipulation schemes existing in the stock market and strives to work upon their detection

using semi-supervised learning. To achieve this, a large open source database, which is known for not having any manipulation is injected with a significant number of artificially generated manipulation instances. In this chapter a small set of extracted features were categorized into PAMP, Danger and Safe signal based on mutual information, calculated with the output class. The outputs so obtained are then subjected to KDE clustering that assigns data instances that form a cluster of unit size as anomalies. It was found that the proposed model outperforms the existing techniques in anomaly detection and with the existing models in market manipulation detection by a significant margin in terms of AUC and FAR values, for the stocks considered.

The proposed approach is however, limited by the scope of the feature set considered and can be further improved if the trading volume information for each stock, as a feature can be included. There is also a need either to vary most of the heuristically selected parameters especially, number of data instances provided to KDE clustering or an algorithm to implicitly select such parameters while maintaining satisfactory results.

Chapter 5 discussed manipulation detection using deep features. The chapter presents two different approaches for manipulation detection: (a) by learning the affinity among trades using an autoencoder and (b) by observing every trading instance under a defined context. In the first model, it was observed that the models' detection accuracy improves by learning the inter-relationship among trades i.e., by calculating similarities among stock price instances. An autoencoder was trained upon the learned affinity matrices for normal trades using the inherent density estimate as the log-likelihood. In the second model, the overlap between normal and abnormal trades was explored to reduce the false positives in manipulation detection. A deep learning model is proposed that could learn upon the normal and abnormal behaviour of the stock prices using a learned similarity metric. The proposed similarity metric is learned by using the compressed deep features from a temporal convolutional network, which forms the underlying dataset for the generative model. The algorithm potentially recognised the manipulative instances and helped in reducing the false positives. The model was validated on real trading orders from LOBSTER project and

Bloomberg trading platform, Newcastle Business School (NBS) from New York Stock Exchange (NYSE).

6.2 Contributions Summary

This thesis contributes to stock price manipulation detection using five different machine learning approaches which were published in six conferences and journals after being peer-reviewed or are under progress. This section highlights the limitations and remaining challenges of the contributions which should be the objectives for future work.

- Chapter 3 presents two decomposition based techniques for stock price manipulation detection including empirical mode decomposition and kernel principal component analysis. Both approaches were published in Intellisys Conference – 2017, London and IEEE Access Journal – 2020 respectively. However, the approach was limited in terms of diversity of features used and lack of informative analysis in the transformed domain.

- Chapter 4 explains a bio-inspired approach of artificial immune system that translates the process of detecting a pathogen or any foreign agent in human bodies to stock price manipulation detection treating the manipulative instance as a pathogen. The proposed approach was accepted and published in Intellisys Conference – 2018 and IEEE-Congress on Evolutionary Computing, 2019. The approach, however impressive in improving the detection rate and reducing false positives up to a significant level failed to investigate the overlap among normal and abnormal stock price samples and needs to work upon better features.

- Chapter 5 adds two contributions in stock price manipulation detection using deep learning that includes training an autoencoder using the affinity among normal trading instances. The proposed work improves the detection rate up to a significant level and expands feature set with volume information, but robustness of the approach should be tested by varying kernel functions. The final approach defines a context under which a set of trading instances should be observed to detect the overlap

among normal and abnormal trading events. The first approach is published in IEEE Joint Conference on Neural Networks – 2020 and the second approach is under progress targeted for IEEE transaction on Neural Networks.

6.3 Future Work

This thesis proposed five different approaches for detecting stock price manipulation each with a progressive rise of accuracy in results and a fall in false positives. The research mentioned also targeted to include three manipulative schemes in an attempt to make each approach generic towards detection. However, despite the approaches there are few potential issues that should be included in any future work.

- **A transformation technique that allows alteration in the inherent probability distribution of the raw features**

This thesis introduced the non-linear transformation of input features into principal components of higher dimensions using adaptive KPCA algorithm. Further to which, the transformed principal components, were treated by a multi-dimensional clustering technique. However, the clustering technique could have also been applied to the input features and the results would not have been much different if a conventional KPCA method had been used. The reason here is that KPCA does not add much complexity to the probability distribution of the transformed components. Therefore, a future work should include a transformation technique that allows more informative insight into the transformed features density distribution.

- **Creating a labelled database of stock price manipulation**

It is highly necessary in this modern era of digitalisation and computational efficiency that a labelled database highlighting normal and abnormal trades is available across all of the major stock exchanges in the world. It would be extremely helpful in quantitative analysis of the data and create an ease of model training, evaluation and testing of the proposed approaches. More importantly, as the trading events evolve very quickly in any financial market, a comparative analysis with the labelled dataset would help understand the context associated

with the changing manipulative trades and could lead to more robust models for stock price manipulation detection.

- **Non-heuristic selection of kernel functions**

Throughout this thesis multiple approaches have applied kernel density estimation based clustering techniques both in unidimensional and multi-dimensional. However, for future work the performance of the proposed approaches can also be evaluated by varying the kernel functions in estimating the density distribution and further clustering the input.

- **Real time detection of stock price manipulation**

In financial markets, the role of regulatory authorities is questioned if it takes an exhaustive amount of time to detect a manipulative instant. It should also be noted that for major market exchanges, the amount of trades and in-effect the amount of manipulative trades are significantly large. It would be an outstanding contribution to the field if real-time detection of stock manipulation is attempted as future work using computationally intelligent techniques.

References

[1] Financial Conduct Authority: MAR 1.6 Market Abuse, F. C. A. London, 2014.

[2] T. C. W. Lin, "The New Market Manipulation," *Emory Law Journal*, vol. 66, pp. 1253-1314, 2017.

[3] F. Allen and D. Gale, "Stock Price Manipulation," *The Review of Financial Studies,* vol. 5, no. 3, pp. 503-528, 1992.

[4] J. Bates, "Post Flash Crash, Regulators Still Use Bicycles to Catch Ferraris: Blaming the Flash Crash on a UK man who lives with his parents is like blaming lightning for starting a fire," *Traders Magazine Online News*, vol. April 24, 2015.

[5] M. J. Aitken, F. H. Harris, J. B. McKinnon, D. Cumming, T. Smith, and K. Venkataraman, "Trade-Based Manipulation and Market Efficiency: A Cross-Market Comparison Chair of Capital Markets Technologies," *22nd Australasian Finance and Banking Conference*, pp. 1-43, November 2009.

[6] D. Y. Yeung and Y. Ding, "Host-Based Intrusion Detection Using Dynamic and Static Behavioral Models 1 Intrusion Detection Problems," *Pattern Recognition,* vol. 36, no. 1, pp. 229-243, 2003.

[7] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "An Experimental Evaluation of Novelty Detection Methods," *Neurocomputing,* vol. 135, pp. 313-327, 2014.

[8] S. Neupane, S. G. Rhee, K. Vithanage, and M. Veeraraghavan, "Trade-based manipulation: Beyond the prosecuted cases," *Journal of Corporate Finance,* vol. 42, pp. 115-130, 2017.

[9] Z. Wei, J. Xun, and X. Wang, "One-Class Classification-Based Finance News Story Recommendation," *Journal of Computational Information Systems,* vol. 6, pp. 1625-1631, 2009.

[10] E. J. Lee, K. S. Eom, and K. S. Park, "Microstructure-Based Manipulation: Strategic Behavior and Performance of Spoofing Traders," *Journal of Financial Markets,* vol. 16, no. 2, pp. 227-252, 2013.

[11] *SEC vs Hold Brothers* Litigation Release No. 67924, 2012.

[12] B. Theodoulidis and D. Diaz, "Financial Markets Monitoring and Surveillance: A Quote Stuffing Case Study," *SSRN Electronic Journal*, 2012.

[13] "*SEC vs ATG Capital LLC.*," ed, 2019, pp. 1-100.

[14] N. Hautsch and R. Huang, "The market impact of a limit order," *Journal of Economic Dynamics and Control*, vol. 36, no. 4, pp. 501-522, 2012.

[15] Nanex, Denver, CO, USA. (2012). Whac-a-Mole is Manipulation [Online]. Available: http://www.nanex.net/aqck2/3598.html

[16] *SEC vs Lidingo Holdings LLC et al*, U. S. S. A. E. COMMISSION Litigation No. 17-cv-01600, 2017.

[17] Y. S. Abu-Mostafa, A. F. Atiya, M. Magdon-Ismail, and H. White, "Introduction to the Special Issue on Neural Networks in Financial Engineering," *IEEE Transactions Neural Networks*, vol. 12, no. 4, pp. 653-656, 2001.

[18] I. Domowitz, *Market abuse and surveillance* (Foresight: EIA17). Economic Impact Assessment Foresight-Government Office for Science, 2012, pp. 1-22.

[19] S. Friederich and R. Payne, "Computer-based trading and market abuse," *Driver Review DR20, Foresight, Government Office for Science*, pp. 1--36, 2012.

[20] *SEC vs Lek Securities Corporation*, US Securities and Exchange Commission Litigation No. 17-cv-01789, 2017.

[21] *SEC vs CSIR Group LLC*, US Securities and Exchange Commission, ed, 2018, p. 24113.

[22] *SEC vs Mustafa David Sayid*, U. S. Securities and Exchange Commission Litigation Release No. 23805, 2017.

[23] "Amgen vs Connecticut Retirement Plans and Trust," in *S. Ct.* vol. 133, ed: Supreme Court, 2013, p. 1184.

[24] "Basic Inc. v. Levinson," in *US* vol. 485, ed: Supreme Court, 1988, p. 224.

[25] "United States v. Stein," in *F. 2d* vol. 456, ed: Court of Appeals, 2nd Circuit, 1972, p. 844.

[26] R. R. Solicitors, "A primer on Market Manipulation Regulations in the U.S. and U.K.," ed: Rahman Ravelli, 2021, pp. 1-43.

[27] *H. Rept. 111-517* - Dodd-Frank Wall Street Reform and Consumer Protection Act, G. U. S. G. P. Office, 2010.

[28] "Regulation No. 596/2014 of the European Parliament and of the Council on Market Abuse," *Official Journal of the European Union,* vol. 2013, no. April, pp. 1-61, 2014.

[29] B. Rizvi, A. Belatreche, A. Bouridane, and I. Watson, "Detection of Stock Price Manipulation Using Kernel Based Principal Component Analysis and Multivariate Density Estimation," *IEEE Access,* vol. 8, pp. 135989-136003, 2020.

[30] B. Rizvi, A. Belatreche, and A. Bouridane, "A Dendritic Cell Immune System Inspired Approach for Stock Market Manipulation Detection," *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings,* 2019, pp. 3325-3332.

[31] B. Abbas, A. Belatreche, and A. Bouridane, "Stock Price Manipulation Detection Using Empirical Mode Decomposition Based Kernel Density Estimation Clustering Method," in *Intelligent Systems and Applications: Intellisys*, Cham, 2018, pp. 851-866.

[32] B. Rizvi, A. Belatreche, and A. Bouridane, "Immune Inspired Dendritic Cell Algorithm for Stock Price Manipulation Detection," Cham, 2019, pp. 352-361.

[33] B. Rizvi, A. Belatreche, A. Bouridane, and K. Mistry, "Stock Price Manipulation Detection based on Autoencoder Learning of Stock Trades Affinity," *Proceedings of the International Joint Conference on Neural Networks*, 2020.

[34] "Securities and Exchange Commission 2015: Case: 1:15-cv-05456," ed, 2015.

[35] R. Aggarwal, B. Gerard, J. Sokobin, and A. V. Thakor, "Stock Market Manipulation-Theory and Evidence," *Capital Markets: Market Microstructure e-Journal,* no. November 2019, 2003.

[36] M. J. White, "The SEC in 2014," in *41st Annual Securities Regulation*, Institute Coronado, California.

[37] "SEC vs Diversified Corp," *US Securities and Exchange Commission*, ed, 2004.

[38] "SEC vs Whittemore," in *F. 3d* vol. 659, Court of Appeals, Dist. of Columbia Circuit, 2011, p. 1-8.

[39] "US v. Delgado," in *F. 3d* vol. 672, ed: Court of Appeals, 5th Circuit, 2012, p. 320.

[40] F. Gonzalez, "A Study of Artificial Immune Systems Applied to Anomaly Detection," Doctor of Philosophy, The University of Memphis, May 2003.

[41] V. Lee and X. Yang, "An Artificial Immune System Based Learning Algorithm for Abnormal or Fraudulence Detection in Data Stream," presented at the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, 16-18, April 2006.

[42] W. Ze-jun, C. Jia, Y. Huan, L. Lin, and W. Xin-an, "An Artificial Immune Model for Abnormal Fluctuation of Stock Price," in *2008 International Symposium on Computational Intelligence and Design*, 2008, vol. 1, pp. 274-277.

[43] V. Lee and X. Yang, "An Artificial Immune System Approach for Real-Time Discovery of Rare Patterns In Data Stream," *WSEAS Transactions on Systems*, vol. 5, no. 6, pp. 1476-1481, 2006.

[44] H. Qi and J. Wang, "A Model for Mining Outliers from Complex Data Sets," in *Proceedings of the 2004 ACM symposium on Applied computing*, 2004, pp. 595-599.

[45] C. Luo, Y. Zhao, L. Cao, Y. Ou, and C. Zhang, "Exception Mining on Multiple Time Series in Stock Market," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2008, vol. 3, pp. 690-693: IEEE.

[46] F. Yang, H. Yang, and M. Yang, "Discrimination of China's Stock Price Manipulation Based on Primary Component Analysis," *Proceedings of 2014 IEEE International Conference on Behavioral, Economic, Socio-Cultural Computing, BESC 2014*, no. 11, 2014.

[47] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, "A Hidden Markov Model with Abnormal States for Detecting Stock Price

Manipulation," *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, pp. 3014-3019, 2013.

[48] LOBSTER project, Atlanta, GA, USA. Limit Order Book System, 2012 [Online]. Available: https://lobsterdata.com/info/DataSamples.php

[49] G. K. Palshikar and M. M. Apte, "Collusion Set Detection Using Graph Clustering," *Data Mining and Knowledge Discovery*, vol. 16, no. 2, pp. 135-164, 2008.

[50] M. N. Islam, S. M. R. Haque, K. M. Alam, and M. Tarikuzzaman, "An Approach to Improve Collusion Set Detection Using MCL Algorithm," *ICCIT 2009 - Proceedings of 2009 12th International Conference on Computer and Information Technology*, no. Iccit, pp. 237--242, 2009.

[51] Z. Ferdousi and A. Maeda, "Unsupervised Outlier Detection in Time Series Data," *22nd International Conference on Data Engineering Workshops ICDEW06*, pp. 121, 2006.

[52] Y. Kim and S. Y. Sohn, "Stock Fraud Detection Using Peer Group Analysis," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8986-8992, 2012.

[53] D. Diaz, B. Theodoulidis, and P. Sampaio, "Analysis of Stock Market Manipulations Using Knowledge Discovery Techniques Applied to Intraday Trade Prices," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12757-12771, 2011.

[54] H. Ögüt, M. M. Doganay, and R. Akta¸s, "Detecting Stock-Price Manipulation in an Emerging Market: The Case Of Turkey," Expert Syst. Appl., vol. 36, no. 9, pp. 11944-11949, 2009

[55] T. Leangarun, P. Tangamchit, and S. Thajchayapong, "Stock Price Manipulation Detection using a Computational Neural Network Model," *8th International Conference on Advanced Computational Intelligence*, pp. 337-341, 2016.

[56] Q. Wang, W. Xu, X. Huang, and K. Yang, "Enhancing Intraday Stock Price Manipulation Detection by Leveraging Recurrent Neural Networks With Ensemble Learning," *Neurocomputing*, vol. 347, pp. 46-58, 2019.

[57] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. September, pp. 1-58, 2009.

[58] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. J. N. c. Williamson, "Estimating the Support of A High-Dimensional Distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443-1471, 2001.

[59] G. L. Grinblat, L. C. Uzal, and P. M. J. e. s. w. a. Granitto, "Abrupt Change Detection with One-Class Time-Adaptive Support Vector Machines," vol. 40, no. 18, pp. 7242-7249, 2013.

[60] T. Harris, "Quantitative Credit Risk Assessment Using Support Vector Machines: Broad Versus Narrow Default Definitions," vol. 40, no. 11, pp. 4404-4413, 2013.

[61] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient Algorithms for Mining Outliers from Large Data Sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427-438.

[62] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-Based Local Outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93-104.

[63] K. Zhang, M. Hutter, and H. Jin, "A New Local Distance-Based Outlier Detection Approach for Scattered Real-World Data," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2009, pp. 813-822.

[64] Y. Li, X. J. C. Zhang, and I. L. Systems, "Diffusion Maps Based K-Nearest-Neighbor Rule Technique for Semiconductor Manufacturing Process Fault Detection," *Chemometrics and Intelligent Laboratory Systems*, vol. 136, pp. 47-57, 2014.

[65] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[66] G. Münz, S. Li, and G. Carle, "Traffic Anomaly Detection Using K-Means Clustering," in *GI/ITG Workshop MMBnet*, 2007, pp. 13-14.

[67] S. Z. Li and A. Jain, "Mahalanobis Distance," in *Encyclopedia of Biometrics. Springer US, Boston, MA*: Springer, 2009, p. 953.

[68] H. S. Bhat and N. J. E. J. o. O. R. Kumar, "Option Pricing Under a Normal Mixture Distribution Derived from The Markov Tree Model," vol. 223, no. 3, pp. 762-774, 2012.

[69] E. Bigdeli, M. Mohammadi, B. Raahemi, and S. Matwin, "Incremental Cluster Updating Using Gaussian Mixture Model," in *Canadian Conference on Artificial Intelligence*, 2015, pp. 264-272.

[70] Lishuai Li, R. John Hansman, Rafael Palacios, and Roy Welsch. "Anomaly Detection via a Gaussian Mixture Model for Flight Operation And Safety Monitoring." *Transportation Research Part C: Emerging Technologies*, vol. 64, 2016, pp. 45-57.

[71] C. Pirrong, "The Economics of Commodity Market Manipulation: A survey," *Journal of Commodity Markets*, vol. 5, no. February, pp. 1-17, 2017.

[72] D. Cumming and M. Aitken, *High Frequency Trading and End of Day Manipulation* (Driver Review DR22). UK: Goverment Office for Science, 2012.

[73] L. C. Matioli, S. R. Santos, M. Kleina, and E. A. Leite, "A New Algorithm for Clustering Based on Kernel Density Estimation," *Journal of Applied Statistics*, vol. 45, no. 2, pp. 347-366, 2018.

[74] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, 2006.

[75] S. Alexander, O. Mangasarian, and B. J. T. R. Schölkopf, "Sparse kernel feature analysis," 1999.

[76] L. C. Matioli, S. R. Santos, M. Kleina, and E. A. Leite, "A new algorithm for clustering based on kernel density estimation," *Journal of Applied Statistics*, vol. 45, no. 2, pp. 347-366, 2017.

[77] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of computational graphical statistics*, vol. 9, no. 2, pp. 249-265, 2000.

[78] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, "Adaptive hidden Markov model with anomaly states for price

manipulation detection," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 26, no. 2, pp. 318-330, 2015.

[79] I. Jolliffe, *Principal Component Analysis* (Springer series in Statistics). Springer, 2011, pp. 1094-1096.

[80] H. Abdi and L. J. Williams, "Principal Component Analysis," *Wiley interdisciplinary reviews: computational statistics,* vol. 2, no. 4, pp. 433-459, 2010.

[81] G. C. Ranger and F. B. Alt, "Choosing principal components for multivariate statistical process control," *Communications in Statistics - Theory and Methods,* vol. 25, no. 5, pp. 909-922, 1996.

[82] M. L. Shyu, S. C. Chen, K. Sarinnapakorn, and L. Chang, "A Novel Anomaly Detection Scheme Based on Principal Component Classifier," *3rd IEEE International Conference on Data Mining,* no. 03, pp. 353-365, 2003.

[83] E. Haven, X. Liu, and L. Shen, "De-Noising Option Prices with The Wavelet Method," *European Journal of Operational Research,* vol. 222, no. 1, pp. 104-112, 2012.

[84] D. Donoho, I. Johnstone, and I. M. Johnstone, "Ideal Spatial Adaptation by Wavelet Shrinkage," *Biometrika,* vol. 81, pp. 425-455, 1993.

[85] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences,* vol. 454, no. 1971, pp. 903-995, 1998.

[86] D. P. Mandic, Z. Wu, and N. E. Huang, "Empirical Mode Decomposition-Based Time-Frequency Analysis of Multivariate Signals," *IEEE Signal Processing Magazine,* vol. 30, no. 6, pp. 74-86, 2013.

[87] L. Hong, "Decomposition and forecast for financial time series with high-frequency based on empirical mode decomposition," *Energy Procedia,* vol. 5, pp. 1333-1340, 2011.

[88] D. Labate, "Empirical Mode Decomposition vs Wavelet Decomposition for the Extraction of Respiratory Signal from Single-Channel ECG: A

Comparison," *IEEE Sensors Journal,* vol. 13, no. 7, pp. 2666-2674, 2013.

[89] J. Tse, X. Lin, and D. Vincent, "High Frequency Trading—Measurement, Detection and Response. Credit Suisse, Zürich," in "Technical Report " 2012.

[90] F. Allen and G. Gorton, "Price Manipulation, Market Microstructure and Asymmetric Information," *European Economic Review,* vol. 36, no. 2–3, pp. 624-630, 1992.

[91] D. Giannone, M. Lenza, and G. E. Primiceri, "Economic Predictions with Big Data: The Illusion of Sparsity," *CEPR Discussion Papers,* no. 12256, pp. 1-25, 2017.

[92] Y. K. Kwok, Mathematical Models of Financial Derivatives, 2nd ed. *Springer-Verlag Berlin Heidelberg,* 2008, p. 530.

[93] T. Kourti and J. F. MacGregor, "Process analysis, monitoring and diagnosis, using multivariate projection methods," *Chemometrics and Intelligent Laboratory Systems,* vol. 28, no. 1, pp. 3-21, 1995.

[94] Q. Wang, "Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models," *CoRR,* vol. abs/1207.3, 2012.

[95] B. Scholkopf, A. Smola, and K. R. M, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computing and Applications,* vol. 10, pp. 1299-1319, 1998.

[96] K. Heafield, "Detecting Network Anomalies With Kernel Principal Component Analysis," *Pasadena, CA, May,* pp. 1-16, 2006.

[97] W. Li, H. H. Yue, S. Valle-Cervantes, and S. J. Qin, "Recursive PCA for adaptive process monitoring," *Journal of Process Control,* vol. 10, no. 5, pp. 471-486, 2000.

[98] J. Ni, C. Zhang, S. X. Yang, S. Member, and A. High-voltage, "An Adaptive Approach Based on KPCA and SVM for Real-Time Fault Diagnosis of HVCBs," *IEEE Transactions on Power Delivery,* vol. 26, no. 3, pp. 1960-1971, 2011.

[99] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, "Kernel density estimation via diffusion," *Annals of Statistics*, vol. 38, no. 5, pp. 2916-2957, 2010.

[100] M. Thomas, K. D. Brabanter, and B. D. Moor, "New bandwidth selection criterion for Kernel PCA: Approach to dimensionality reduction and classification problems," *BMC Bioinformatics*, vol. 15, no. 1, 2014.

[101] B. Y. G. R. Terrell and D. W. Scott, "Variable Kernel Density Estimation," *Annals of Statistics*, vol. 20, no. 3, pp. 1236-1265, 1992.

[102] M. C. Jones and D. F. Signorini, "A comparison of higher-order bias kernel density estimators," *Journal of the American Statistical Association*, vol. 92, no. 439, pp. 1063-1073, 1997.

[103] J. S. Marron and M. P. Wand, "Exact Mean Integrated Squared Error," *The Annals of Statistics*, vol. 20, no. 2, pp. 712-736, 1992.

[104] M. P. Wand and M. C. Jones, "Comparison of smoothing parameterizations in bivariate kernel density estimation," *Journal of the American Statistical Association*, vol. 88, no. 422, pp. 520-528, 1993.

[105] B. B. Maillet and P. M. Merlin, "Outliers Detection, Correction of Financial Time-Series Anomalies and Distributional Timing for Robust Efficient Higher-Order Moment Asset Allocations," *SSRN Electronic Journal*, 2009.

[106] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognition*, vol. 40, no. 3, pp. 863-874, 2007.

[107] L. Cao, Y. Ou, and P. S. Yu, "Coupled Behavior Analysis with Applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1378-1392, 2012.

[108] K. Golmohammadi, O. R. Zaiane, and D. Diaz, "Detecting stock market manipulation using supervised learning algorithms," *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 435-441, 2014.

[109] R. A. Craig and L. Liao, "Phylogenetic tree information aids supervised learning for predicting protein-protein interaction based on distance matrices," *BMC Bioinformatics*, vol. 8, no. 1, p. 6, 2007.

[110] I. Smal, M. Loog, W. Niessen, and E. Meijering, "Quantitative comparison of spot detection methods in fluorescence microscopy," *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 282-301, 2010.

[111] J. Wu and X. Zhang, "Maximum Margin Clustering Based Statistical VAD With Multiple Observation Compound Feature," *IEEE Signal Processing Letters*, vol. 18, no. 5, pp. 283-286, 2011.

[112] P. E. Evangelista, M. J. Embrechts, P. Bonissone, and B. K. Szymanski, "Fuzzy ROC curves for unsupervised nonparametric ensemble techniques," presented at the Proceedings. 2005 IEEE International Joint Conference on Neural Networks, Montreal, QC, Canada, 2005.

[113] R. Kharghanian, A. Peiravi, and F. Moradi, "Pain detection from facial images using unsupervised feature learning approach," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 419-422.

[114] A. G. Roy and D. Sheet, "\ DASA: \ Domain Adaptation in Stacked Autoencoders using Systematic Dropout," *CoRR*, vol. abs/1603.0, 2016.

[115] S. Wang, D. Li, N. Petrick, B. Sahiner, M. G. Linguraru, and R. M. Summers, "Optimizing area under the ROC curve using semi-supervised learning.," *Pattern recognition*, vol. 48, no. 1, pp. 276-287, 2015.

[116] J. H. McDonald, *Handbook of Biological Statistics*: Sparky House Publishing, vol. 2, pp. 6-59, 2009

[117] S. Chawla and A. Gionis, "k-means-: A Unified Approach to Clustering and Outlier Detection," pp. 189-197, 2013.

[118] P. Matzinger, "The danger model: a renewed sense of self," *Science*, vol. 296, no. 5566, pp. 301-305, 2002.

[119] "SEC on Pump Scheme," ed, 2017, pp. 1-50. https://www.sec.gov/rss/your_money/pump_and_dump.htm

[120] M. Mokhtar, R. Bi, J. Timmis, and A. M. Tyrrell, "A modified dendritic cell algorithm for on-line error detection in robotic systems," in *2009 IEEE Congress on Evolutionary Computation (CEC)*, 2009, pp. 2055-2062.

[121] J. Greensmith, "The Dendritic Cell Algorithm," *Thesis for the Degree of Doctor of Philosophy*, no. October, pp. 1-316, 2007.

[122] E. Alizadeh, N. Meskin, and K. Khorasani, "A Dendritic Cell Immune System Inspired Scheme for Sensor Fault Detection and Isolation of Wind Turbines," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 545-555, 2018.

[123] Z. Chelly and Z. Elouedi, "From the general to the specific: Inducing a novel dendritic cell algorithm from a detailed state-of-the-art review," *International Journal of Pattern Recognition Artificial Intelligence*, vol. 30, no. 03, p. 1659009, 2016.

[124] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, "Self-nonself discrimination in a computer," in *Proceedings of 1994 IEEE computer society symposium on research in security and privacy*, 1994, pp. 202-212.

[125] J. Kim and P. J. Bentley, "An evaluation of negative selection in an artificial immune system for network intrusion detection," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, 2001, pp. 1330-1337.

[126] N. Elisa, L. Yang, and N. Naik, "Dendritic Cell Algorithm with Optimised Parameters Using Genetic Algorithm," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1-8.

[127] F. Gu, "Theoretical and Empirical Extensions of The Dendritic Cell Algorithm," Citeseer, 2011.

[128] F. Gu, J. Greensmith, R. Oates, and U. Aickelin, "PCA 4 DCA: The Application of Principal Component Analysis to The Dendritic Cell Algorithm," *Available at SSRN 2830357*, 2009.

[129] Z. Chelly and Z. Elouedi, "RST-DCA: A Dendritic Cell Algorithm Based on Rough Set Theory," in *International Conference on Neural Information Processing*, 2012, pp. 480-487.

[130] Z. Chelly and Z. Elouedi, "A Survey of The Dendritic Cell Algorithm," *Knowledge and Information Systems*, vol. 48, no. 3, pp. 505-535, 2016.

[131] T. M. Cover and J. A. Thomas, "Elements of Information Theory, 2nd E dition," ed: New York, Wiley-Interscience, 2006.

[132] A. A. Margolin *et al.*, "ARACNE: An Algorithm for The Reconstruction Of Gene Regulatory Networks In A Mammalian Cellular Context," in *BMC bioinformatics*, 2006, vol. 7, no. 1, pp. 1-15.

[133] B. W. Silverman, Density Estimation for Statistics and Data Analysis, Routledge, 2018, pp. 1-175.

[134] B. Abbas, O. Farooq, Y. Uzzaman, A. A. Khan, and A. L. Vyas, "Enhancing Classification Accuracy of Wrist Movement by Denoising sEMG signals," in 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka. Japan, 2013, pp. 5762-5764.

[135] J. Kim, P. Bentley, C. Wallenta, M. Ahmed, and S. Hailes, "Danger is ubiquitous: Detecting malicious activities in sensor networks using the dendritic cell algorithm," in *International Conference on Artificial Immune Systems*, 2006, pp. 390-403.

[136] R. Jarrow, S. Fung, and S. C. Tsai, "An Empirical Investigation of Large Trader Market Manipulation in Derivatives Markets," *Review of Derivatives Research*, vol. 21, no. 3, pp. 331-374, 2018.

[137] V. Alex, K. Vaidhya, S. Thirunavukkarasu, C. Kesavadas, and G. Krishnamurthi, "Semisupervised Learning Using Denoising Autoencoders for Brain Lesion Detection and Segmentation," *Journal of Medical Imaging*, vol. 4, no. 4, p. 041311, 2017.

[138] M. Sakurada and T. Yairi, "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction," in *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 2014, pp. 4-11.

[139] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic," *IEEE Sensors Letters*, vol. 3, no. 1, pp. 1-4, 2018.

[140] T. Amarbayasgalan, B. Jargalsaikhan, and K. H. Ryu, "Unsupervised Novelty Detection Using Deep Autoencoders with Density-Based Clustering," *Applied Sciences*, vol. 8, no. 9, pp. 1468, 2018.

[141] D. Singh and C. K. Mohan, "Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder," *IEEE Transactions on Intelligent Transportation Systems,* vol. 20, no. 3, pp. 879-887, 2019.

[142] C. Zhou and R. C. Paffenroth, "Anomaly Detection with Robust Deep Autoencoders," in *KDD '17: 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax NS Canada, 2017, pp. 665-674.

[143] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Transactions on pattern analysis machine intelligence,* vol. 22, no. 8, pp. 888-905, 2000.

[144] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," in *Proceedings of the seventh IEEE international conference on computer vision*, 1999, vol. 2, pp. 975-982.

[145] X. Y. Stella and J. Shi, "Multiclass Spectral Clustering," in *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, vol. 1, pp. 313-319.

[146] G. Alain and Y. Bengio, "What Regularized Auto-Encoders Learn from The Data-Generating Distribution," *The Journal of Machine Learning Research,* vol. 15, no. 1, pp. 3563-3593, 2014.

[147] F. R. Chung and F. C. Graham, "Spectral Graph Theory", *American Mathematical Soc.*, no. 92, 1997.

[148] X. J. Zhu, "Semi-Supervised Learning Literature Survey," 2005. Available online: https://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf

[149] K. Konda, R. Memisevic, and D. Krueger, "Zero-Bias Autoencoders and The Benefits of Co-Adapting Features," *arXiv preprint: 1802.10560,* 2014.

[150] N. Kourentzes, D. K. Barrow, and S. F. Crone, "Neural Network Ensemble Operators for Time Series Forecasting," *Expert Systems with Applications,* vol. 41, no. 9, pp. 4235-4244, 2014.

[151] A. B. L. Larsen, S. K. Snderby, H. Larochelle, and O. Winther, "Autoencoding Beyond Pixels Using a Learned Similarity Metric," *33rd*

International Conference on Machine Learning (ICML) 2016, vol. 4, pp. 2341--2349.

[152] K. Bardool, T. Tuytelaars, and J. Oramas, "A Systematic Analysis of a Context Aware Deep Learning Architecture for Object Detection," *CEUR Workshop Proceedings*, vol. 2491, pp. 1-15, 2019.

[153] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks for Action Segmentation and Detection," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1003-1012.

[154] J. M. Joyce, "Kullback-Leibler Divergence," *International Encyclopedia of Statistical Science*, vol. 720, p. 722, 2011.

[155] I. Goodfellow *et al.*, "Generative Adversarial Nets," presented at the Advances in neural information processing systems, 2014.

[156] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223-2232.

[157] T. Karras, S. Laine, and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4401-4410.

[158] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery," in *International conference on information processing in medical imaging*, 2017, pp. 146-157: Springer.

[159] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative Adversarial Text to Image Synthesis," in *International Conference on Machine Learning*, 2016, pp. 1060-1069.

[160] C. Ledig *et al.*, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *Proceedings of the IEEE*

conference on computer vision and pattern recognition, 2017, pp. 4681-4690.

[161] M. Kliger and S. Fleishman, "Novelty Detection with GAN," *ArXiv preprint arXiv:1802.10560*, 2018.

[162] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANS," *Advances in neural information processing systems*, vol. 29, pp. 2234-2242, 2016.

[163] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. Salakhutdinov, "Good Semi-Supervised Learning That Requires a Bad GAN," *arXiv preprint arXiv:.09783*, 2017.

[164] J. Zhai, Y. Cao, and X. Ding, "Data Analytic Approach for Manipulation Detection In Stock Market," *Review of Quantitative Finance and Accounting*, vol. 50, no. 3, pp. 897-932, 2018.

[165] NANEX, Denver, CO, USA. (2013). Exploratory Trading—Top 8 HFT Take Liquidity 59% of the Time [Online]. Available: http://www.nanex.net/aqck2/4136.html

[166] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal Convolutional Networks: A Unified Approach to Action Segmentation," *European Conference on Computer Vision*, vol. abs/1608.0, pp. 47-54, 2016.

[167] S. W. Fu, T. W. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-To-End Waveform Utterance Enhancement for Direct Evaluation Metrics Optimization by Fully Convolutional Neural Networks," *IEEE/ACM Transactions on Audio, Speech, Language Processing*, vol. 26, no. 9, pp. 1570-1584, 2018.

[168] D. Pokrajac, A. Lazarevic, and L. J. Latecki, "Incremental Local Outlier Detection for Data Streams," in *2007 IEEE symposium on computational intelligence and data mining*, 2007, pp. 504-515.

[169] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A Comparative Evaluation of Outlier Detection Algorithms: Experiments and Analyses," *Pattern Recognition*, vol. 74, pp. 406-421, 2018.

[170] L. Liu et al, "Clustering and Hybrid Genetic Algorithm based Intrusion Detection Strategy, " Indonesian Journal of Electrical Engineering and Computer Science vol. 12, pp. 762-770, 2012.

[171] Soodeh Hosseini and Seilani Hossein, "Anomaly Process Detection Using Negative Selection Algorithm and Classification Techniques, " Evol. Syst. vol. 12, pp. 769-778, 2021.

[172] C.A. Laurentys, G. Ronacher, R.M. Palhares, W.M. Caminhas, "Design of an Artificial Immune System for fault detection: A Negative Selection Approach", Expert Syst. App., vol. 37, pp. 5507-5513, 2010.

[173] I. Idris, A. Selamat, S. Omatu, "Hybrid Email Spam Detection Model with Negative Selection Algorithm and Differential Evolution", Eng. Appl. Artif. Intell., vol. 28, pp. 97-110, 2014.

[174] W. Hualong, Z. Bo, "Overview of Current Techniques In Remote Data Auditing", Appl. Math. Nonlinear Sci., pp. 145-158, 2016.

[175] Tao Yang, Wen Chen and Tao Li, "A Real Negative Selection Algorithm with Evolutionary Preference for Anomaly Detection" Open Physics, vol. 15, no. 1, pp. 121-134, 2017.

[176] Lei Ding, Fei Yu and Zhenghua Yang, "Survey of DCA for Abnormal Detection, " J. Softw. vol. 8, pp. 2087-2094, 2013.