Time Series Contextual Anomaly Detection for Detecting Stock Market Manipulation

by

Seyed Koosha Golmohammadi

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Anomaly detection in time series is one of the fundamental issues in data mining. It addresses various problems in different domains such as intrusion detection in computer networks, anomaly detection in healthcare sensory data, and fraud detection in securities. Though there has been extensive work on anomaly detection, most techniques look for individual objects that are different from normal objects but do not take the temporal aspect of data into consideration. We are particularly interested in contextual anomaly detection methods for time series that are applicable to fraud detection in securities. This has significant impacts on national and international securities markets.

In this thesis, we propose a prediction-based Contextual Anomaly Detection (CAD) method for complex time series that are not described through deterministic models. First, a subset of time series is selected based on the window size parameter, Second, a centroid is calculated representing the expected behaviour of time series of the group. Then, the centroid values are used along with correlation of each time series with the centroid to predict the values of the time series. The proposed method improves recall from 7% to 33% compared to kNN and random walk without compromising precision.

We propose a formalized method to improve performance of CAD using big data techniques by eliminating false positives. The method aims to capture expected behaviour of stocks through sentiment analysis of tweets about stocks. We present a case study and explore developing sentiment analysis models to improve anomaly detection in the stock market. The experimental results confirm the proposed method is effective in improving CAD through removing irrelevant anomalies by correctly identifying 28% of false positives.

To my parents, Mojtaba and Foroogh,
for their never-ending love and support.

Anomaly detection is the detective work of machine learning: finding the unusual, catching the fraud, discovering strange activity in large and complex datasets. But, unlike Sherlock Holmes, you may not know what the puzzle is, much less what suspects you're looking for.

– Ted Dunning, Chief Application Architect at MapR, 2014.

Acknowledgements

I would like to thank Osmar Zaiane for his support throughout my thesis. He has been a mentor, a friend, and a superb advisor. I have had the opportunity to get his advice on many technical issues. I enjoyed our long discussions on different algorithms, data mining techniques, and analytical evaluation of different approaches. Furthermore, I had the opportunity to discuss some of the philosophical issues and dilemmas of our time. I had the opportunity to travel with Osmar to a few conferences and learn collaboration, networking, and fair evaluation of the works of our peers first-hand. I have asked Osmar's thoughts and advice on the most important decisions in my life during the past five years. He has always been truly a living example of how to lead my life and to face tough decisions.

I would like to thank my father, Mojtaba, and my mother, Foroogh, who have always encouraged me to go forward. Their never-ending love and support has been a tremendous help to continue in my journey for the highest educational achievements.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Anomalies or outliers are individuals that behave in an unexpected way or feature abnormal properties [70]. The problem of identifying these data points or patterns is referred to as outlier/anomaly detection. The significance of anomaly detection lies in actionable information that they provide in different domains such as anomalous traffic patterns in computer networks which may represent intrusion [63], anomalous MRI images which may indicate the presence of malignant tumours [210], anomalies in credit card transaction data which may indicate credit card or identity theft [8], or anomalies in stock markets which may indicate market manipulation. Detecting anomalies has been studied by several research communities to address issues in different application domains [40].

Time series are indispensable in today's world. Data collected in many domains such as computer networks traffic, healthcare, flight safety, and fraud detection are sequences or time series. More formally, a time series $\{ x_t, t \in T_0 \}$ is the realization of a stochastic process $\{ X_t, t \in T_0 \}$. For our purposes, set T (i.e. the set of time points) is a discrete set and the real valued observations x_t are recorded on fixed time intervals. Though there has been extensive work on anomaly detection [40], the majority of the techniques look for individual objects that are different from normal objects but do not take the temporal

aspect of data into consideration. For example, a conventional anomaly detection approach based on values of data points may not capture anomalous data points in the ECG data in Figure 1.1. Therefore, the temporal aspect of data should be considered in addition to the amplitude and magnitude values. Though time series anomaly detection methods constitute a smaller portion of the body of work in anomaly detection, there have been many methods within this group that are designed for different domains. Time series outlier detection methods are successfully applied to different domains including management [211], detecting abnormal conditions in ECG data [138], detecting shape anomalies [236], detecting outlier light curves in astronomical data [245], and credit card fraud detection [78].

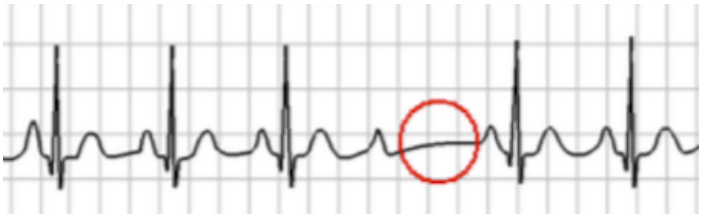


Figure 1.1: Anomaly in ECG data (representing second degree heart block)

These methods are shown to be effective in their target domain, but adapting the methods to apply to other domains is quite challenging. This is evidently due to the fact that the natures of time series and anomalies are fundamentally divergent in different domains. We are particularly interested in developing effective anomaly detection methods for complex time series that are applicable to fraud detection in securities (stock market). The detection of such anomalies is significant because by definition they represent unexpected (suspicious) periods which merit further investigations, as they are potentially associated to market manipulation.

1.1 Problem setting

The anomaly detection problem for time series data can be perceived in three settings:

1. Detecting anomalous time series, given a time series database: here, the time series is anomalous with respect to the training time series in the database. The time series in the database may be labelled or a combination of labelled and unlabelled samples.

2. Detecting anomalous subsequence: here, the goal is identifying an anomalous subsequence within a given long time series (sequence). This problem setting is also introduced in the works of Keogh et al., as detecting discords "the subsequences of a longer time series that are maximally different from the rest of the sequence" [113] in the time series [114]. Figure 1.2 shows an anomalous subsequence within a longer time series. It is not the low values of the subsequence which make it anomalous, as it appears in other places in the given time series, but its abnormal length.

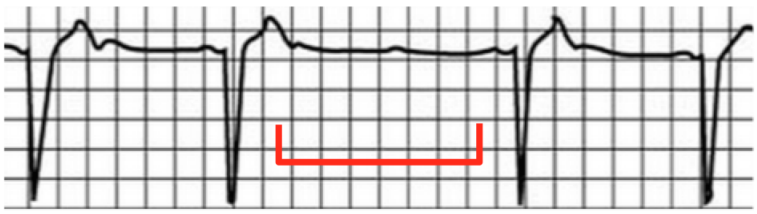


Figure 1.2: Anomalous subsequence within a longer time series

3. Detecting contextual or local anomalies: here, anomalies are data points that are anomalous in a specific context but not otherwise. For example, Edmonton's average temperature during 2013 (see Figure 1.3) was 4.03 degrees Celsius, while the same value during January would be an

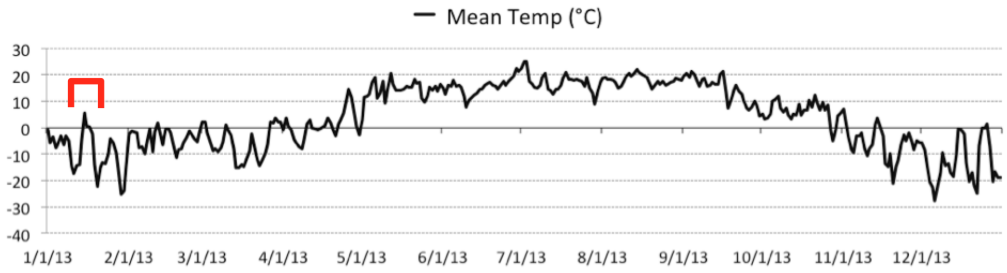


Figure 1.3: Average daily temperature of Edmonton during the year 2013

anomaly (i.e. contextual anomaly). Another example would be data points or periods in a time series that deviate from the expected pattern given a group of time series that are expected to have a similar pattern (e.g. heart rate of different horses or stock returns of similar companies).

In this thesis, we focus on contextual/local anomaly detection within a group of similar time series. The context is defined both in terms of similarity to the neighbourhood data points of each time series and similarity of time series pattern with respect to the rest of time series in the group. Local anomalies are different from global anomalies because a data point that is detected as an anomaly with respect to the neighbourhood data points may not be an anomaly with respect to all other data points in the dataset. Local anomaly detection methods are particularly useful in non-homogeneous datasets and datasets with changing underlying factors such as financial data. The major motivation for studying local anomaly detection is the development of methods for detecting local anomalies/outliers in complex time series that do not follow a seasonal pattern and are non-parametric, meaning it is difficult to fit a polynomial or deterministic function to the time series data. This is a significant problem in domains with complex time series such as stock market. Market manipulation periods have been shown to be associated with anomalies in the time series of assets [156] [209], yet the development of effective methods to detect such anomalies remains a challenging problem.

1.2 Anomaly detection to improve detecting stock market manipulation

Market capitalization exceeded $1.5 trillion in Canada and $25 trillion in USA in 2015 [1] (GDP of Canada and USA in 2015 were $1.5 and $17 trillion respectively). Protecting market participants from fraudulent practices and providing a fair and orderly market is a challenging task for regulators. 233 individuals and 117 companies were prosecuted in 2015, resulting in over $138 million in fines, compensation, and disgorgement in Canada. However, the effect of fraudulent activities in securities markets and financial losses caused by such practices is far greater than these numbers suggest as they impact public and market participants trust. "Securities fraud broadly refers to deceptive practices in connection with the offer and sale of securities". Securities fraud is divided into the following categories [2]:

- High yield investment fraud: these schemes typically offer guaranteed returns on low-risk or no-risk investments in securities instruments. Perpetrators take advantage of investors' trust and claim high returns to operate their funds. The most prevalent high yield investments appear in the form of: pyramid scheme, Ponzi schemes, prime bank scheme, advance fee fraud, commodities fraud (foreign currency exchange and precious metals fraud) and promissory notes.

- Broker embezzlement: these schemes include broker unauthorized and illegal actions to gain profit from client investments. This may involve unauthorized trading or falsification documents.

[1] http://data.worldbank.org/indicator/CM.MKT.LCAP.CD
[2] FBI report 2010-2011

- Late-day trading: these schemes involve trading a security after the market is closed.

- Market manipulation: these schemes involve individuals or groups attempting to interfere with a fair and orderly market to gain profit.

Market manipulation and price rigging remain the biggest concerns of investors in today's market, despite fast and strict responses from regulators and exchanges to market participants that pursue such practices [3]. Market manipulation is forbidden in Canada [4] and the United States [5]. The industry's existing approach for detecting market manipulation is top-down, and is based on a set of known patterns and predefined thresholds. Market data such as price and volume of securities (i.e. the number of shares or contracts that are traded in a security) are monitored using a set of rules and red-flags trigger notifications. Then, transactions associated with the detected periods are investigated further, as they might be associated with fraudulent activities. These methods are based on expert knowledge but suffer from two issues: i) detection of abnormal periods that are not associated with known symptoms (i.e. unknown manipulative schemes), and ii) adaption to changing market conditions whilst the amount of transactional data is exponentially increasing (this is due to the rapid increase in the number of investors and listed securities) which makes designing new rules and monitoring the vast data challenging. These issues lead to an increase in false negatives (i.e. there is a significant number of abnormal periods that are left out of the investigation). Data mining methods may be used as a bottom-up approach to detect market manipulation by identifying unusual patterns and data points that merit further investigation, as they are

[3] http://economictimes.indiatimes.com/markets/stocks/market-news/market-manipulation-continues-to-be-the-biggest-concerns-for-investors/articleshow/12076298.cms

[4] Bill C-46 (Criminal Code, RSC 1985, c C-46, s 382, 1985)

[5] Section 9(a)(2) of the Securities Exchange Act (SECURITIES EXCHANGE ACT OF 1934, 2012)

potentially associated with fraudulent activities. We are interested in methods that are based on offline processes (as opposed to online or stream mining).

Our thesis is that a data mining approach can effectively identify anomalies in securities markets to improve detection of market manipulation without expert knowledge. However, there are many challenges involved in developing data mining applications for fraud detection in securities markets including massive datasets, accuracy, performance measures, and complexity. The impacts on the market, on privacy and on the training of auditors are other issues that must be addressed, but are beyond the scope of this thesis. In this thesis we focus on time series with daily and weekly frequencies over a period of 40 years. This may be perceived as high frequency in the data mining community when compared to time series with monthly and quarterly frequency, however, should not be confused with High Frequency Trading (HFT) which we discuss in Section 1.3. For our purposes, we define market manipulation in securities (based on the widely accepted definition in academia and industry) as the following:

Definition 1.2.1. market manipulation involves intentional attempts to deceive investors by affecting or controlling the price of a security or interfering with the fair market to gain profit.

We divide known market manipulation schemes into three groups based on the above definition:

1. Marking the close: buying or selling a stock near the close of the day or quarter to affect the closing price. This might be done to help prevent a takeover or rights issue, to avoid margin calls (when a position is financed through borrowing funds) or to affect the performance of a fund manager's portfolio at the end of a quarter (*window dressing*). A typical indicator is trading in small amounts before the market closes.

2. Wash trades: pre-arranged trades that will be reversed later and impose no actual risk to neither buying or selling parties. These trades aim to give the appearance that purchase and sales have been made (*Pooling* or *churning* can involve wash sales or pre-arranged trades executed in order to give an impression of active trading in a stock).

3. Cornering the market (in a security): to gain control of sufficient amount of the security to control its price.

It has been shown that the manipulated prices revert towards their natural levels in the days following the manipulation date [48]. **The common characteristic of different types of market manipulation for data scientists would be the unexpected pattern/behaviour in securities data.** This is the primary motivation for using time series anomaly detection methods to identify market manipulation. The stock market is essentially a non-linear, non-parametric system that is extremely hard to model with any reasonable accuracy [234]. Consequently, detecting anomalies in stock market is a challenging task. We aim to investigate the following research questions in this thesis:

- What are the existing techniques and challenges in developing a data mining method for detecting securities fraud and market manipulation?
- How to detect anomalies in securities market, in particular stock market?
- How to handle unlabelled data for anomaly detection?
- How big data techniques, in particular using Twitter data, can improve anomaly detection?
- How to evaluate the performance of an anomaly detection method, which aims to detect stock market manipulation?

We tackle these research questions in the manuscript starting through i) highlighting challenges in anomaly detection, ii) introducing challenges in anomaly detection in stock market, iii) reviewing data mining techniques for detecting market manipulation, iv) reviewing anomaly detection methods, v) reviewing data mining methods for detecting market manipulation, vi) describing our proposed method and the underlying theories, vii) illustrating empirical results and discussions, vii) studying big data techniques to improve the performance of the proposed anomaly detection method.

1.3 Computational Challenges

There are many computational challenges in developing anomaly detection methods for time series. These challenges can be divided into two groups: i) general challenges in developing anomaly detection methods for time series, and ii) computational challenges that arise from the particular domain for which an anomaly detection method is developed. In this section, first, we introduce challenges that are involved in developing anomaly detection methods for time series, second, we describe our challenges in developing an effective anomaly detection method for non-parametric time series that are applicable to fraud detection in the stock market.

The most important challenges in developing anomaly detection methods for time series include:

1. Different ways to define anomalies in a time series as the anomaly within a given time series could be an event, a subsequence, or the entire time series.

2. Distinguishing outliers from noise in the data for domains that include noise in the given time series.

3. Scalability and computational complexity as time series are usually long in practical applications and there may be many time series to process.

4. Feature scaling as usually there are different scales when multiple time series are involved in real applications and many anomaly detection algorithms assume time series on similar comparable scales.

5. Unknown size of the window or length of anomalous periods when attempting to detect anomalous subsequences.

6. Different length of the training and testing time series.

7. Utilizing appropriate similarity measures (also known as distance measures). For example, Euclidean distance is not applicable in problems where the lengths of time series to be compared is different (see the Appendix C for a list of the most widely used distance measures for time series).

There are two fundamental challenges in developing effective anomaly detection methods in stock market. First, the time series in stock market (e.g. prices, return, volume) are non-stationary and non-linear time series that are extremely difficult to model [235], thus, difficult to devise deviations of observations from an underlying model. This means anomaly detection methods that assume the time series are generated based on an underlying model or process such as prediction based, Hidden Markov Model (HMM) based and segmentation based are not suitable for capturing anomalies in securities. Second, we need datasets with known anomalies to evaluate the performance of a given anomaly detection technique. Below, we describe how aforementioned computational challenges emerge in developing anomaly detection methods that are aimed at fraud detection in securities. Later in Section 1.4, we elaborate on our contributions and approach in tackling these issues.

1. Scalability: anomaly detection methods for time series vary in their computational complexity. This becomes important particularly in domains such as the stock market where both the number of time series and the lengths of time series are huge and increasing rapidly. High Frequency Trading (HFT) – an evolution of securities market that adopts sophisticated algorithms to automatically analyze and react to market data in milliseconds – is a phenomena that contributes substantially to the rapidly growing size of time series data in securities market [6].

2. Different Forms and Resources of Data: The data in the securities market comes from different resources and in different forms such as news data, analytical data (level I and level II data) and fundamental data (financial reports and filings). The data in securities market can be divided into two groups:

   - Unstructured data including news and financial Events (e.g. Factiva [7]), stock-chat message boards (e.g. stocktraderchat [8], yahoo message board [9]).
   - Structured data including trading data (e.g. Trade And Quote (TAQ) from NASDAQ [10]), stock analytics, companies' financial information (COMPUSTAT [11]), companies' insider activities (e.g. Thomson Reuters Insider Filings Data Feed (IFDF)).

[6] HFT are algorithms that could submit many orders in millisecond. HFT accounts for 35% of the stock market trades in Canada and 70% of the stock trades in USA according to the 2010 Report on regulation of trading in financial instruments: Dark Pools & HFT.

[7] global.factiva.com

[8] http://stocktraderschat.com

[9] http://finance.yahoo.com/mb/YHOO

[10] http://www.nasdaqdod.com

[11] http://www.compustat.com

Heterogeneous datasets and integrating data from different sources makes both data preparation and learning phase of the data mining process challenging. The most important and widely used features for anomaly detection in securities market are price and volume of assets [48] [164], however, big data methods and tools could be useful here to integrate other features and resources in the anomaly detection process.

3. Unlabelled Data: Labelled data for anomaly detection and fraud detection in securities market is very rare because (a) labelling data is very costly and typically requires investigation by auditors, and (b) the number of positive samples (fraud cases) constitutes a tiny percentage of the total number of samples. This is also known as the problem of imbalanced classes and has been attempted in literature by boosting the anomaly class (i.e. *oversampling*) and generating artificial anomalies (i.e. *data synthesize*) [44]. Koscsis et al. used Markov Decision Processes (MDPs) to generate synthetic samples (assuming there are very few positive samples) and used frequency that a player abates from the optimal policy as features to train the classifier in the modelling [125].

4. Performance Measure: Misclassification costs are unequal in fraud detection because false negatives are more costly. In other words, missing a fraud case by predicting it to be not-fraud, hurts performance of the method more than including a not-fraud case by predicting it to be fraud. The issue of identifying an appropriate performance measure in problems with unequal misclassification costs has been studied within different learning approaches. Some of the most effective performance measures that are proposed for fraud detection addressing unequal misclassification costs are listed below based on the learning approach:

    - Supervised Learning: threshold, ordering, and probability metrics

are effective performance measures for evaluating supervised learning methods for fraud detection [176]. According to our studies the most effective metrics include: Activity Monitoring Operating Characteristic (AMOC) [75] (average score versus false alarm rate), Receiver Operating Characteristic (ROC) analysis (true positive rate versus false positive rate), mean squared error of predictions, maximizing Area under the Receiver Operating Curve (AUC), minimizing cross entropy (CXE) [230] and minimizing Brier score [230].

- Semi-supervised Learning: entropy, conditional entropy, relative conditional entropy, information gain and information cost [132] are the most effective performance measures for fraud detection using semi-supervised learning methods.
- Unsupervised Learning: Hellinger and logarithmic scores [243] and t-statistic [26] are reported to have higher performances when using data mining methods that are based on unsupervised learning approaches.

1.4 Contribution

Our goal is to develop an effective contextual anomaly detection method for complex time series that are applicable to fraud detection in securities. Below, we elaborate on our contributions to specific computational challenges that are discussed in Section 1.3.

1. **Scalability**

   The problem of anomaly detection in securities involves many time series with huge length. This makes the computational complexity of anomaly detection methods important especially in presence of HFT where thou-

sands of transactions are recorded per second in each time series (i.e. stock). We attempt to propose a method that is linear with respect to the length of input time series. We conducted extensive experiments to study the computational complexity as a critical factor in developing the proposed method for contextual anomaly detection in time series. We studied the computational complexity of the proposed method as well as the competing methods that we use in the validation phase.

2. **Unlabelled Data and Injection of Anomalies**

   In an attempt to address the issue of unlabelled data we propose a systematic approach to synthesize data by injecting anomalies in real securities market data that is known to be manipulation-free. We use a dataset that is known to be anomaly-free (i.e. no market manipulation) then we inject random anomalies in the data. This is discussed in detail in Section 4.2.

3. **Performance Measure**

   We studied performance measures both theoretically and experimentally to identify impact of different factors and to propose a fair performance measure in problems with imbalanced classes and unequal misclassification costs. In Section 1.3, we described the issues with using conventional performance measures for evaluating anomaly detection methods in presence of unbalanced classes.

4. **Different Resources and Forms of Data and Big Data**

   We propose aggregating other resources of information (in addition to structured data related to each stock which is represented in time series) by leveraging *big data* tools and techniques to achieve insights on anomalies. The additional data resources include information such as news,

financial reports and tweets [12] that could be utilized in the anomaly detection process. We are particularly interested in integrating Twitter data in our analysis. The motivation to integrate other resources in the process is taking the anomaly detection a step further as will be enumerated later. For instance, confirming if there is a reason that may explain occurrence of the detected anomaly, can be accomplished using external information (e.g. large number of tweets before an event detected as anomaly may be the reason for the seemingly anomalous event/value).

[12]StockTwits is a platform to organize information about stocks on twitter. The StockTwits API could be used to integrate this information to improve fraud detection in securities.

Chapter 2

Related Work

Anomaly detection aims to address the problem of detecting data that deviate from an expected pattern or behaviour. More formally, given descriptive data $\{ x_1, x_2, \cdots, x_n \}$ about a phenomena there is a probability distribution $P(x)$. Data is assumed to follow the probability distribution under normal conditions. Given a set of i.i.d. data samples $\{ x_1, x_2, \cdots, x_n \}$ we can calculate their likelihood and determine if there is a deviation from the underlying phenomenon. This can trigger a reaction or raise an alarm. For example, unexpected sensory data of patients' vital signs or weather temperature. The motivation of detecting anomalies in real life is generally to initiate a decision making process to respond to such cases. However, in real situations, it is very difficult, if not impossible, to define the probability distribution $P(x)$ that describes the phenomenon. Typically, anomaly detection methods aim to circumvent this issue. Anomaly detection has a long history in statistics with early attempts on the problem in 1880s [59]. Anomaly detection has been adopted to in various domains such as credit card fraud detection [8], intrusion detection in computer networks [63], detecting anomalous MRI images which may indicate the presence of malignant tumours [210] and detecting stock market manipulation. These methods are typically designed for a specific domain and developing a generic method for different domains has remained a

challenging problem. This is evidently because of the fundamental differences that anomalies in different domains have.

In this chapter, we start by reviewing the literature on anomaly detection in Section 2.1. Then, we present an extensive literature review on anomaly detection methods for time series and how they are different to the proposed thesis in Section 2.2. In Section 2.3 We review data mining methods that are used to detect securities fraud and stock market manipulation.

2.1 Anomaly Detection Methods

In this section we review anomaly detection methods in a broader sense and based on different approaches that are applied to anomaly detection in the literature.

2.1.1 Classification based anomaly detection

These techniques are based on learning a classifier using some training data to identify anomalies from normals. These algorithms are also called One Class Classifiers (OCC) [220]. The anomaly class is assumed very rare and OCC is learned on assumingly normal samples. The new data point is compared with the learned distribution and if it is very different it would be declared anomalous. The classifier is learned by choosing a kernel and using a parameter to set the close frontier delimiting the contour of observations in the feature space. Martinez et al. show OCC can perform well in two-class classification problems with different applications [153]. There are some research works indicating that OCC can outperform standard two-class classifiers [106] [107].

OCC assume an approximate shape for the hypersphere and aim to adapt the shape to the training data while having the minimum coverage of the

input space. The Kernels are utilized in two forms: i) deducing a minimum volume hypersphere to the dataset in the feature space (Support Vector Data Description (SVDD) [219]), ii) identifying the maximum margin hyperpsphere to separates data from the origin [200]. Radial Basis Function (RBF) is a popular kernel that has been applied to both of these approaches and widely is referred to as reduced set Parzen Density Estimators [200]. These models are shown to be sensitive to potential anomalies in training data. Some variations of OCC are proposed to address the issue [208] [190].

Classification algorithms are also used for rule induction to capture normal behaviour/pattern through a set of rules. A given data instance would be declared anomalous if it does not match the rules. Some of the rule induction algorithms that are used include Decision Trees [9], RIPPER [47] CART [29], and C4.5 [197]. The rules have a confidence level representing the rate of correct classification by each rule on training data. A given test instance is run through the rules to identify the rule which captures the instance and the confidence value becomes the anomaly score of the instance. There has been some extended research work on these techniques [73] [95][133] [196] [221] [214]. The unsupervised learning approach in association rule mining has been adopted to generate rules for OCC [5]. The rules are produced from categorical data in the training data using unsupervised learning algorithms. A support threshold is typically used to filter rules with low support aiming to extract the most dominant patterns in data [214].

The OCC methods have four characteristics [220] that need to be considered when adopting them for anomaly detection:

1. **Simple configuration:** there are very few parameters that need to be set while there are established techniques to estimate them. It is important to follow these techniques and practices as the parameters

may impact the classification results greatly.

2. **Robustness to outliers:** the implicit assumption when adopting OCC methods, is the training data represents the target class. However, this assumption maybe inappropriate as there might be anomalies in the training data, especially in real-life data. It is important to devise a plan to mitigate the risk of anomalies in the input data.

3. **Incorporation of known outliers:** OCC can be improved by incorporating data from the second. Therefore it is recommended to include data from the other class in training.

4. **Computational requirements:** these methods are particularly slow as the computation is heavy and required for every test instance. These methods may not be appropriate for data although this aspect becomes less important with time, the fact that evaluating a single test point takes much time might make the model useless in practice.

2.1.2 Clustering based anomaly detection

Clustering methods utilize unsupervised learning algorithms to identify groupings of normals in the data [105] [214]. These methods are divided into three groups:

1. Methods that assume normal instances are near the closest centroid, thus data instances that are distant from the centroids are anomalous. First, a clustering algorithm is used to identify centroids, second the distance of every data instance with the closest centroid is calculated. This distance is the anomaly score of each instance. The centroids that are generated on the training data are used to identify the anomaly score of

a given test instance. Some of the clustering algorithms that are used for this technique include Expectation Maximization (EM), Self-Organizing Maps (SOM) and K-means [207]. The drawback of this technique is that it is unable to identify anomalies when they constitute a cluster.

2. Methods that assume normals are part of a cluster, thus data instances that do not belong to any cluster are anomalous. First a clustering algorithm is used to devise data points in clusters, second, data points that do not fall in any cluster are declared anomalous. These methods require clustering algorithms that do not force every data point in a cluster such as ROCK [89], DBSCAN [68], and SNN clustering [64]. Alternatively, it is possible to remove detected clusters from the input data and declare the remaining data points as anomalies. This approach was introduced in the FindOut algorithm [246] by extending the WaveCluster algorithm [205]. The drawback of these methods is they may have unreliable and inconsistent results since they are targeting anomalies while clustering algorithms are designed to identify clusters.

3. Methods that assume normal data points belong to dense and large clusters while anomalies belong to sparse and small clusters. Data instances that belong to small or low density groups are declared as anomalous after clustering on the data. There has been different research works which adopted a variation of this technique [212][66][94] [149] [167] [178].

2.1.3 Nearest Neighbour based anomaly detection

The principle idea in nearest neighbour based anomaly detection is that normal data instances occur in dense neighbourhoods, thus data instances that are distant from their closes neighbours are anomalous. These techniques require a

similarity measure (also known as distance measure or a metric) that is defined between two given data instances. We can divide nearest neighbourbased anomaly detection methods into the following two categories:

1. **Using kth Nearest Neighbour:** in these methods, the anomaly score of each instance is calculated based on the distance to its kth nearest neighbour. Then, typically a threshold on the anomaly score is used to verify if a test instance is an anomaly. This technique was first introduced to detect land mines on satellite ground images [33] and later was applied to other applications such as intrusion detection by identifying anomalous system calls [137]. This anomaly detection technique can also be used to identify candidate anomalies through ranking of the n instances with the largest anomaly score on a given dataset [186]. The core nearest neighbour based anomaly detection technique has been extended in three different ways:

    - Computing the anomaly score of a datapoint as the sum of distances to kth nearest neighbour [69] [66] [249]: An alternate method of calculating the anomaly score of a data instance would be to to count the number of nearest neighbour n that are less than or equal to d distance apart from the given data instance [121] [124] [122] [123].
    - Using various distance/similarity measures to handle different data types: Lee et al. proposed the hyper-graph based technique, HOT, in which the categorical values are modelled using a hyper-graph and the distance of two given instances are calculated based on the connectivity of the graph [238]. Otey et al. utilized distance of categorical and continuous attributes separately when the given dataset includes a mixture of categorical and continuous data at-

tributes [168]. Other forms of similarity measures have been applied to continuous sequences [170] and special data [126].

- Improving the efficiency of algorithm (time complexity of the generic technique is $O(N^2)$ for N instances) by reducing the search space through discounting the instances that cannot be anomalous or focusing on instances that are most likely to be anomalous: A simple pruning step on a randomized data is shown to reduce the average time of searching for nearest neighbour to linear time [19]. Sridhar Ramaswamy et al. introduced a partitioning technique where first, instances are clustered and the lower and upper bound distances to its kth nearest neighbour is calculated within each cluster, second, the bounds are used to discount partitions that cannot include the top k anomalies (i.e. pruning irrelevant partitions) [186].

Other similar clustering based techniques have been proposed to prune the search space for nearest neighbours [66] [218]. Within an attribute space that is partitioned into hypergrids of hypercubes, a pruning technique eliminates hypercubes that have many instances since these are most likely normals. If a given data instance belongs to a hypercube with few instances and neighbouring hypercubes with few instances, it is declared anomalous.

2. **Using Relative Density:** These methods aim to approximate neighbourhood density on the input data because a data instance on a low density neighbourhood is deduced as anomalous while an instance in a dense neighbourhood is deduced as normal. The distance to kth nearest neighbour for a given data instance is defined through a hypersphere centered at the data instance containing k other instances where the radius of the hypersphere represents the distance. Thus, the distance to

the kth nearest neighbour for a given data instance is equivalent to the inverse of its density. This makes these methods sensitive to regions of varying densities and may result in poor performance. To address this issue, some techniques compute the density of instances with respect to density of their neighbours. The ratio of average density of the k nearest neighbours of the data instance over the local density of the data instance itself is used in Local Outlier Factor (LOF) technique [30] [31]. The local density is computed using a hypersphere centered at the given data instance encompassing k nearest neighbours while the hyperphere radius is minimized. Then, k is divided by the volume of the hypersphere which gives the local density. A data instance that falls on a dense region would be normal and have a local density similar to its neighbours while an anomalous data instance would have a lower local density compared to its neighbours. Thus a higher LOF score for the anomalous instance.

Connectivity-based Outlier Factor (COF) is a variation of LOF where the neighbourhood for a given instance is computed incrementally [217]. First, the closest instance is added to the neighbourhood set given a data instance. Second, the next instance is added while the distance of members in the set remains the minimum. This process is repeated to grow the neighbourhood until reaching k. Third, the COF anomaly score is computed by dividing the volume of neighbourhood by k similar to LOF. LOF has also been adopted in other proposed methods for outlier detection [46] [90] [110] [172] [212] [45].

2.1.4 Statistical anomaly detection

The principal concept shaping the statistical anomaly detection methods is the basic definition of anomaly, "normal data instances occur in high probability

regions of an underlying stochastic model, while anomalies occur in the low probability regions of the stochastic model". Statistical techniques aim to fit a probability distribution to normal data and by inference declare a given data instances that does not follow the model, anomalous. The underlying reasoning is the low probability that is estimated for these data instances to be generated from the learned model. There are two approaches to fit a statistical model to data, parametric and non-parametric, and they both have been utilized for statistical anomaly detection. The primary difference of these approaches is that the parametric techniques assume some knowledge about the underlying distribution [53].

- **Parametric Techniques** assume the "normal data is generated by the probability distribution $P(x, w)$, where x is an observation and w is the parameter vector. The parameters w need to be estimated from given data" [65]. This is the main drawback of these methods because the parametric assumption typically does not hold. Furthermore, parameter estimation may be problematic in high dimensional datasets. The parametric technique can be divided into three groups based on the assumed distribution:

  - Gaussian Model Based Techniques, that assume the underlying Gaussian distribution generates the input data. Maximum Likelihood Estimates (MLE) is the classical approach for estimating the parameters. Some statistical tests have been proposed using Gaussian models to detect anomalies [16] [15].
  - Mixture Distribution Based Techniques, that provide a aggregated (mixture) of individual distributions representing the normal data. The model is used to examine if a given data instance belongs to the model and instances that do not follow the model are declared

anomalous [3]. The Poisson distribution is widely used as the individual models that are aggregated in the mixture to represent normal data [33]. Different variations of the mixture distribution based technique are used along with an extreme statistic to identify anomalies [188] [189].

– Regression Model Based Techniques, that fit a regression model to input data and compute the anomaly score of a given data instance based on its residual. The residual for a given test instance represents the value that is not explained by the model, thus its magnitude is used as the anomaly score (i.e. deviation from normal). There are some statistical tests to investigate anomalies with different confidence levels [11] [91] [223]. Regression model based techniques are well-studied in literature for time series data [1] [2] [80].

  The Akaike Information Content (AIC) - a measure to compare quality of statistical models on a given dataset - has been used to detect anomalies in the data during when fitting models [119]. The regression model based technique for anomaly detection is sensitive to potential anomalies in the input data since they impact the parameters. Robust regression is introduced to address the issue of anomalies in the data when fitting a model [191]. The classic regression model is not applicable to multivariate time series data, therefore, different variations of regression are proposed to address such problems through statistics on i) using Integrated Moving Average (ARIMA) model to detect anomalies in the multivariate time series [224], ii) using Autoregressive Moving Average (ARMA) model to detect anomalies by mapping the multivariate time series to a uni-

variate time series and detecting anomalies in the transformed data [81].

- **Non-parametric Techniques** that unlike parametric techniques, do not use a priori parameters defining the structure of the model but are built using the given data. These techniques typically do not require assumptions about the data (some time very few few assumptions are required). We divide non-parametric techniques for anomaly detection to two groups:

  - Histogram Based techniques, which simply use histograms to model normal data. These methods are heavily used for fraud detection [76] and intrusion detection [52][67] [65]. A histogram is generated based on different values of the feature in univariate data and a test data instance which does not fall in any bins is declared as anomalous. The height of the bins represents the frequency of data instances within each bin. The histograms can be generated for each data attribute in the case of multivariate data. The plain vanilla histogram technique can be extended by assigning an anomaly score to a given test data instance based on the height of the bin it falls into. The anomaly score is computed with the same analogy, for each attribute, in the case of multivariate data. The disadvantage of using histograms is they are sensitive to the bin size. Smaller bin sizes result in many false alarms (i.e. anomalies falling out of the bins or in rare bins) while large bins may produce high false negative rates (i.e. anomalies falling in frequent bins). Another disadvantage to using a histogram appears in multivariate data due to disregarding the relationships of data attributes.

  - Kernel Functions, which use a kernel function to fit a model to data.

These techniques typically use Parzen Density estimation [175]. A test instance that is distant from the model is declared anomalous. Kernel functions are also used to estimate the probability distribution function (PDF) of normal instances [53] and a given test instance falling in low probability regions of the PDF would be anomalous. These methods are sensitive to selected kernel function, kernel parameters and sample size. Appropriate kernels and parameters can improve performance of anomaly detection but a poor choice of the kernel and parameters may have significant negative impacts on performance of the method. Another disadvantage to using kernel-based techniques is that the sample size may grow exponentially in high dimensional data.

2.1.5 Information theoretic anomaly detection

Information theoretic based anomaly detection techniques are based on the assumption that anomalies produce irregularities in the information content of the dataset. These techniques utilize various measures in information theory such as entropy, relative entropy, and Kolmogorov Complexity.

We can define the basic form of information theory technique as a dual optimization where for a given dataset D with complexity $C(D)$ the the subset of instances I are minimized such that $C(D) - C(D - I)$ is maximized. Data instances in this subset are therefore labelled anomalous. The aim of the information theory technique in an optimization problem that has two objectives and does not have a single optimum, is to find a Pareto-optimal solution. In other words, this is a dual optimization of minimizing the subset size and maximizing the reduction in the complexity of the dataset. The brute-force approach to solving the problem has exponential time complexity. However,

different approximation methods are proposed to detect the most anomalous subset. Local Search Algorithm (LSA) [92] is a linear algorithm to approximate the subset using the entropy measure. A similar method is proposed using the information bottleneck measure [10].

Information theory based techniques are also applicable to datasets where data instances are ordered such as spatial and sequence data. Following the basic form of information theory anomaly detection, the problem is described as finding the substructure I such that $C(D) - C(D - I)$ is maximized. This technique has been applied to spatial data [141], sequential data [13][46][139] and graph data [163]. The complexity of the dataset D (i.e. $C(D)$) can be measured using different information measures, however, Kolmogorov complexity [135] has been used by many techniques [13]. Arning et al. used the regular expression to measure the Kolmogorov Complexity of data [13] while Keogh et al. used the size of the compressed data file based on a standard compression algorithm [116]. Other information theory measures such as entropy and relative uncertainty have been more popular in measuring the complexity of categorical data [10] [93] [92] [134].

Some of the challenges using information theory based methods include:

1. finding the optimal size of the substructure which is the key to detecting anomalies,

2. choosing the information theory measure since the performance of anomaly detection is highly dependent on the measure. These measures typically perform poorly when the number of anomalies in the data is not large, and

3. obtaining anomaly score for a specific test instance.

2.1.6 Spectral anomaly detection

Spectral techniques are based on the assumption that the input data could be transformed to a new feature space with lower dimensionality where normals and anomalies are distinguishable in the new space [4]. These techniques aim to represent the data through a combination of attributes that capture the majority of variability in data. Features that are irrelevant or unimportant are filtered out in the transformation phase where each data instance is projected to the subspace. A given test instance is declared anomalous (or novel) if the distance of its projection with other instances is above a threshold. Both supervised and unsupervised learning algorithms have been utilized to develop spectral-based anomaly detection methods in two forms:

1. **Utilizing distance of data instances:**

    - Learning Vector Quantization (LVQ) [208] which uses a competitive training rule to build a lattice of centres that model the normal data,
    - k-means [22] which uses the distance to the nearest centre as a distance metric, and,
    - Self Organizing Maps (SOM) [208] which uses the difference between a given data instance to its nearest node of the lattice as the detection feature.

2. **employing projection techniques to reconstruct data in a subspace:**

    - Principal Component Analysis (PCA) [111][226] which uses the most representative principle components of the data to map and reconstruct samples in the subspace. The orthogonal reconstruction error is used to detect anomalies,

- Kernel Principal Component Analysis (KPCA) [21] that reconstructs samples in a subspace similar to PCA but using the kernel trick [96],
- Autoassociative Neural Networks (AARNA) [106] which uses a single hidden layer neural network with fewer units in the hidden layer than the input dimensionality and the error in the output of the network represents the distance to the true distribution of data. AARNA has been shown to be equivalent to PCA when using a single hidden layer, and,
- Diabolo Networks [128] [240] which similar to AARNA uses neural networks but with more hidden layers to achieve nonlinear reconstruction subspaces. AARNA has been shown to be equivalent to the KPCA method [128]).

Reconstruction methods are more practical compared to distance-based techniques, however, they perform poorly on noisy data. Various methods are proposed to address this issue such as analyzing projection of each data instance along the principal component with the lowest variance [174]. Data instances with low correlation with such principle component will have low values as they meet the correlation structure of data, thus, data instances with large values are declared anomalous as they do not follow the structure. Huber et al. proposed using robust PCA [103] to estimate principal components from the covariance matrix of the normal data for anomaly detection in astronomy [206].

2.1.7 Stream anomaly detection

The techniques that have been discussed so far in this chapter are not designed for processing a continuous stream of data. Stream data mining techniques are typically based on online learning methods and only use a chunk of data that comes in instead of using the whole data. The problem of anomaly detection in stream mining can be described as identifying the change in the stream of data when the process generating the stream changes. Stream mining based anomaly detection has numerous practical applications such as web traffic analysis, robotics, fault detection in industrial machinery [74] [112] [153] [154] [247], credit fraud detection, intrusion detection, medical anomaly detection, etc. [41]. Batch processing is not suitable for such applications.

Anomalies appear in two forms within stream data:

1. temporal change in the source generating the stream where tracking subsequent changes is desirable, and

2. permanent change in the source generating the stream where tracking subsequent changes after detecting the change is not required. This often appears in systems where the change should trigger some action to avoid undesirable outcomes or revert to normal conditions before the change.

We can define anomaly detection in stream data as $\alpha_i = (\tau_i, l_i)$ where i is the starting data instance and li the interval [6]. The stream anomaly detection method outputs a signal in a di interval using the data that is observed so far where $\tau i \leq d_i \leq \tau_i + l_i$, and $(d_j - \tau i)$ is minimal. Anomaly detection in stream data is challenging because:

- the anomalous events may be rare and their length may vary. Therefore, these techniques develop rules for identifying normals (learning normals),

- the data may include drift (i.e. stream of data that is slightly different from normal patterns) and the techniques should adapt to such characteristic to reduce false alarms, and

- the data stream may be complex thus the technique should adapt to complex decision boundaries to identify anomalies.

Principal subspace tracking algorithms have been utilized to detect anomalies based on deviations from the subspace of normals [57] [130]. These techniques are particularly useful when dealing with nonlinear data. This approach was utilized for feature space modelling by adapting a Kernel Recursive Least Squares [6]. This approach was adapted to one-class SVM on non-stationary data [34], however, it performs poorly in high dimensional data due to its high computational complexity.

Promising results have been obtained using a fully probabilistic model for stream anomaly detection [244]. Though this method requires a predetermined assumption about data distribution. Other approaches include classification trees [216] and clustering methods [60] that were adopted for stream anomaly detection where batches of data are stored to update the anomaly detection model. This results in issues related to storage and response time . The drift technique [58] [61] [233][240] has been adopted to address this issue by first capturing the probability distribution of data stream continuously, secondly detecting changes on it. One-class classifiers are utilized to approach the first step, however, this requires capturing the whole dataset for training and in memory processing which is not a realistic assumption for practical stream data. There are few online learning methods that are appropriate for stream anomaly detection [196] [136].

2.2 Anomaly Detection in Time Series

We reviewed the literature on different data mining methods for detecting securities market manipulation in an earlier work [85]. In this section, we focus on characteristics and drawbacks of existing methods. We elaborate on our approach towards addressing limitations of the existing methods. Anomaly detection methods for detecting contextual outliers in time series can be classified along two orthogonal directions: i) the way the data is transformed prior to anomaly detection (transformation dimension), and ii) the process of identifying anomalies (anomaly detection technique). Table 2.1 describes a list of existing methods for detecting local outliers in time series along these two dimensions.

Transformation is the procedure that is applied to data before anomaly detection. There are two motivations for data transformation: i) to handle high dimensionality, scaling and noise, and ii) to achieve computational efficiency. The transformation procedures include:

- Aggregation that focuses on dimensionality reduction by aggregating consecutive values. A typical approach for aggregation is replacing a set of consecutive values by a representative value of them (usually their average).

- Discretization which converts the given time series into a discrete sequence of finite alphabets. The motivation of using discretization is using existing symbolic sequence anomaly detection algorithms and improving computation efficiency [140].

- Signal Processing which maps the data to a different space as sometimes detecting outliers in a different space is easier and the mapping may reduce the dimensionality (e.g. Fourier transforms [71], wavelet transforms

Table 2.1: Anomaly Detection Methods for Time Series

Transformation → Technique ↓	**Aggregation**	**Discretization**	**Signal Processing**
Window Based	kNN [42], SVM [146]	kNN [109]	
Proximity Based	PCAD [187], [181]		
Prediction Based	Moving Average [43], AutoRegression [43], Kalman Filters [120], SVM [147]	FSA [155]	Wavelet [250] [145]
HMM based	[144]	[183] [252]	
Segmentation	[39] [38] [195]		

[250]).

There are some issues and risks that need to be considered when using transformation techniques. The time series are in a different format after aggregation, therefore, the values after transformation correspond to a set of data points in the original time series. This is particularly problematic in time series that do not follow a uniform distribution. Although discretization may improve computational efficiency, the dimensionality of symbolic representations remains the same after transformation. Most discretization techniques need to use the entire time series to create the alphabet. Furthermore, the distance measures on symbolic representation may not represent a meaningful distance in the original time series. Transformation using signal processing techniques may also suffer from the issue of distance measure in the new space. We avoid the transformation process in the proposed outlier detection method and we use original values of all data points in the given time series. The time series in securities fraud detection are typically processed offline (there is no noise in recorded values) and are aligned time series. Below, we briefly review five groups of anomaly detection methods for detecting local/contextual outliers in time series and we highlight their disadvantages:

1. Window based: a time series is divided to fixed window size subse-

quences. An anomaly score is calculated by measuring the distance of a sliding window with the windows in the training database. Chandola et al. use the distance of a window to its kth nearest neighbour as the anomaly score [42] while Ma and Perkins use the training windows to build one class SVMs for classification (the anomaly score for a test window is 0 if classified as normal and 1 if classified as anomalous) [146].

- Disadvantage: the window based outlier detection methods for time series suffer from two issues: i) the window size has to be chosen carefully (the optimal size depends on the length of anomalous subsequence), and ii) the process can become computationally expensive (i.e. $O((nl)^2)$ where n is the number of samples in testing and training datasets and l is the average length of the time series.
- In our proposed method, we divide the given time series to fixed window size periods and look for outliers within that period (i.e. neighbourhood) but there is no sliding window (thus lower time complexity). Furthermore, the size of windows in the proposed method (e.g. 1 year) is much longer than the length of anomalies. We use overlapping of a few time stamps to avoid missing outliers on the border of the windows. The length of the overlapping is set to 4 data points in our experiments.

2. Proximity based: the assumption here is that the anomalous time series are different to other time series. These methods use the pairwise proximity between the test and training time series using an appropriate distance/similarity kernel (e.g. correlation, Euclidean, cosine, DTW measures). Unlike the window based method, instead of rolling a window the similarity measure is used to measure the distance of every two given sequences. A kNN or clustering method (k-means) is used where

the anomaly score of each time series is the distance to the kth nearest neighbour in the dataset in the former case, and the distance to the centroid of the closest cluster in the latter case [182] [187].

- Disadvantage: these methods can identify anomalous time series, but cannot exactly locate the anomalous region. They are also highly affected by the similarity measure that is used, and in the problems that include time series misalignment the computational complexity may significantly increase.
- Our proposed method, like any outlier detection method which uses a distance measure, is affected by the type of distance measure, however, it has been shown that the Euclidean distance (the similarity measure that we use) outperforms most distance measures for time series [83]. As we indicated in Section 1, the time series in our problem are discrete and the values are recorded in fixed time intervals (i.e. time series are aligned). Unlike proximity based methods, which assign an anomaly score based on the distance of two given sequences, our proposed method assigns an anomaly score based on the distance of predicted value for each data point and its actual value, thus enables detecting the location of anomalous data point/region.

3. Prediction based: these methods assume the normal time series is generated from a statistical process but the anomalous data points do not fit the process. The time series based models such as Moving Average (MA) [43] Auto Regressive (AR) [43], Autoregressive Integrated Moving Average (ARIMA) [177] and ARMA [158] as well as non-time series based models such as linear regression [159], Gaussian process regression [242] and support vector regression [147] are used to learn the parameters of

the process. Then, the model derived from a given time series is used to predict the (n+1)th value using previous n observations.

- Disadvantage: there are two issues in using such prediction based methods for outlier detection in time series: i) the length of history that is used for prediction is critical in locating outliers, and ii) performance of these methods are very poor in capturing outliers if the data is not generated by a statistical process.
- The assumption that the normal behaviour of any given time series is generated from a model and such a model could be derived from history of the time series, does not hold in some domains such as securities market. Therefore, outlier detection methods based on this assumption (i.e. prediction based, Hidden Markov Model and segmentation based) are inappropriate in detecting anomalies in complex time series such as securities.

4. Hidden Markov Model (HMM) based: the assumption here, is the underlying process creating the time series is a hidden Markovian process (i.e. the observed process creating the original time series is not necessarily Markovian) and the normal time series can be modelled using an HMM [185] [109]. The training data is used to build an HMM which probabilistically assigns an anomaly score to a given test time series.

   - Disadvantage: the issue in using HMM based methods is the assumption that there is a hidden Markovian process generating the normal time series. Therefore, this method fails if such a process does not exist.

5. Segmentation based: first a given time series is partitioned into segments. The assumption here is that there is an underlying Finite State

Automaton (FSA) that models the normal time series (the states and transitions between them in FSA is constructed using the training data) and segments of an anomalous time series do not fit the FSA [39] [38].

- Disadvantage: segmentation based methods may suffer from two issues: i) the state boundaries are rigid and may not be robust to slight variations in the data during the testing phase, and ii) segmentation technique may fail in detecting outliers in problems where the assumption "all training time series can be partitioned into a group of homogeneous segments" does not hold.

2.3 Data Mining Methods for Detecting Market Manipulation

There has been an increasing number of research works on detecting market manipulation using data mining methods in the past few years. We presented a comprehensive literature review [85] to identify (a) the best practices in developing data mining techniques (b) the challenges and issues in design and development, and (c) the proposals for future research, to detect market manipulation in securities market. We identified five categories based on specific contributions of the literature on the data mining approach, goals, and input data.

1. **Social Network Analysis**

   Traditional data mining methods (e.g. classification, clustering, association rules) often consider samples as independent data points [82]. However, these methods cannot leverage the relationship between samples in datasets that are richly structured and mostly heterogeneous. Such structured data can be represented in the form of a social network

where nodes correspond to data samples (i.e. objects/individuals), and edges represent relationships and dependencies between objects. Mapping, understanding, analyzing and measuring interactions across such a network is known as Social Network Analysis (SNA). Using SNA to find correlations that indicate fraud in securities market begins with transforming the market events to a graph (preprocessing). The most interesting application of SNA in securities market fraud is detecting brokers that collaborate to: a) inflate/deflate the price of a security by putting prearranged orders with other brokers and manipulating the volume, b) move stocks between accounts for tax reasons, and c) get credibility in the market with high number of transactions. Blume et al. combined SNA and interactive visualization to identify malicious accounts in an exchange [23]. Authors designed indicators of fraudulent activities (based on the textual description of typical fraud cases) that can be detected using SNA:

- Circular trading: characterizes consistently buying and selling more or less the same volume of a stock
- Primary-Secondary indicator: marks accounts buying low and selling high. Network centrality can help to find the primary account; a function of f is calculated for every vertex representing the size of the account and comparing the price of the transaction with average price (in the past c transactions)
- Prominent edge indicator: identifies transferring stocks from one account to another which happens when an edge (transaction) between two vertices appears several times

SNA provides many algorithms (e.g. algorithms for identifying central nodes and cycles) that are effective in finding collaborative efforts to

manipulate market as well as methods for monitoring interactions of traders in the market.

2. **Visualization**

   The goal of visualization in the context of securities fraud detection is producing visualizations that go beyond conventional charts enabling auditors to interact with the market data and find malicious patterns. Visualization of the market data is both important for real-time monitoring and off-line investigations. Visualization can help auditors identify suspicious activities in securities and traders' transactions. The input data includes historical trading data or real-time stream of data about securities/traders transactions. Securities market investigators use different charts and figures to monitor the market. However, in our discussions with Canadian securities market auditors and regulators we found great interests in finding data visualization techniques that are beyond charts/tables which permit one to see the patterns within the data or other information not readily discernible. Stockare is a visual analytics framework for stock market [101], which combines a 3D Treemap for market surveillance, and a behaviour-driven visualization using SNA for monitoring the brokers' activities. In the 3D visualization each cell represents a security, the size of a cell is proportional to the market capitalization and the colour code of a cell indicates the change in the price (e.g. green for increase and red for decrease in the price). The 3D visualization provides a tool for the real-time monitoring (15 minutes delay) of raw trading flow (price and volume). Trading details are compared to a set of parameters and an alert is raised if they are out of range. Analysis of the trading network aims to reveal the social structure among traders and identify suspected trading patterns. Nodes represent traders, the area

around each node represents the trading value, and directional edges indicate the flow and weight of trades/exchanges. A database of malicious trading patterns is used as a reference to compare with events in the trading network and identify suspicious activities. Liquidity, returns and volatility are higher for the manipulated stocks, therefore, charting these parameters in parallel with the same time alignment helps regulators in identifying suspicious patterns and trends [54]. Isolated jumps in liquidity can indicate suspicious trades when returns are within the normal ranges. Li et al. combine SAX and the chaos game bitmaps representation of sequences to develop an outlier detection method. The bitmap representation is a visualization method that includes two steps, first, the frequency counts of substrings of length L are mapped into a 2L by 2L matrix, second, the frequency counts are colour-coded [237]. SAX is used to discretize a time series with real values (with alphabet size of four) because the bitmap representation is applicable to discrete sequences. The distance of bitmaps is used as the distance of two sliding windows to derive anomaly scores.

3. **Rule Induction**

   The goal of these methods is extracting rules that can be inspected and used by auditors/regulators of securities market. The input data includes historical trading information for each trader account as well as trader accounts that are labelled to be suspicious for fraudulent activity. It is also possible to extract rules that identify unknown patterns and irregularities using unlabelled data (i.e. using unsupervised learning algorithms). Data mining methods that generate rules are of particular interest because of the intrinsic features that rules provide for fraud detection in securities market. High transparency, easily comparable to existing reg-

ulatory/auditing rules, and easily integrable to existing tools, are only a few features that make using rules very compelling among auditors and investigators in securities market. Abe et al. introduced an approach for rule induction by temporal mining of data. First, time series data is cleaned (preprocessing) in two steps: a) the period of subsequence is determined, and b) the temporal pattern extraction is performed using a clustering algorithm (EM and K-means). Also relevant data attributes are selected manually or by using attribute selection algorithms. Second, a rule induction algorithm such as C4.5 [184], AQ15 [98] or Version Space [157] is used to produce if-then rules. An environment is developed using the proposed method and tested using a dataset that consists of temporal price data (price, volume, high, low, etc.) of 9 stocks from Japan's stock market from January 5, 2006 to May 31, 2006. The buy/sell decisions on each stock is determined using the clustering method and is used for testing on a different stock. Experimental results show that the introduced method for pattern extraction is promising as it outperforms the baseline. A crucial issue in rule induction methods is identifying effective rules from the set of generated rules. There are numerous objective rule interestingness measures that can be used for this purpose. An extensive experiment comparing over 70 different objective measures to describe rule interestingness using a dataset in healthcare identified Recall [165], Jaccard [215], Kappa [215], Collective Strength (CST) [215], X2-M [165] and Peculiarity [166] as the most effective objective measures. However, such ranking may be different in experiments on financial data and to the best of our knowledge there has not been a work that compares objective measures for rule interestingness on financial data.

4. **Pattern Recognition using supervised learning methods**

The goal of using these methods is detecting patterns that are similar to the trends that are known to represent fraudulent activities. This can be pursued in two different levels: a) detecting suspicious traders with fraudulent behaviour, b) detecting securities that are associated with fraudulent activities. The input data includes historical trading data for each trader account (in the former case) or for each security (in the latter case) and a set of patterns/trends that are known to be fraud (labels). Pattern recognition in securities market typically is performed using supervised learning methods on monthly, daily or intraday data (tick data) where features include statistical averages and returns. Ogut et al. used daily return, average of daily change and average of daily volatility of manipulated stocks and subtracted these numbers from the same parameters of the index [164]. This gives the deviation of manipulated stock from non-manipulated (index) and higher deviations indicate suspicious activities. The assumption in this work is price (consequently return), volume and volatility increases in the manipulation period and drops in the post-manipulation phase. The proposed method was tested using the dataset from Istanbul Stock Exchange (ISE) that was used in a related work to investigate the possibility of gaining profit at the expense of other investors by manipulating the market [7]. Experimental results show that ANN and SVM outperform multivariate statistics techniques (56% compared to 54%) with respect to sensitivity (which is more important in detecting price manipulation as they report correctly classified manipulated data points). Diaz et al. employed an "open-box" approach in application of data mining methods for detecting intraday price manipulation by mining financial variables, ratios and textual sources [54]. The case study was built based on stock market manipulation cases pursued by the US Securities and Exchange Commission (SEC) during

2009. Different sources of data that were combined to analyze over 100 million trades and 170 thousand quotes in this study include: profiling info (trading venues, market capitalization and betas), intraday trading info (price and volume within a year), and financial news and filing relations. First, using clustering algorithms, a training dataset is created (labelling hours of manipulation, because SEC does not provide this information). Similar cases and Dow Jones Industrial Average (DJI) were used as un-manipulated samples. Second, tree generating classification methods (QUEST, C5.0 and CART) were used and tested using jackknife and bootstrapping. Finally, the models were ranked using overall accuracy, measures of unequal importance, sensitivity and false positives per positives ratio. A set of rules were generated that could be inspected by securities investigators and be used to detect market manipulation. The results indicate:

- liquidity, returns and volatility are higher for the manipulated stocks than for the controlling sample
- although, it is possible to gain profit by manipulating the price of a security to deflate its price (short selling), most market manipulators attempt to increase the stock price
- closing hours, quarter-ends and year-ends are "common preconditions for the manipulations"
- sudden jumps in volume of trading and the volatility of returns are followed by price manipulation in most cases

These findings are in line with our understanding of the problem where a market manipulation activity would appear as an anomaly/outlier in the data.

5. **Anomaly Detection**

The goal of these methods is detecting observations that are inconsistent to the remainder of data. These methods can help in discovering unknown fraudulent patterns. Also, spikes can be detected effectively using anomaly and outlier detection according to the market conditions, instead of using a predefined threshold to filter out spikes. Similar to the supervised learning methods, outlier detection can be performed both in security and trader levels for fraud detection. The input dataset is the historical transactional data of each trader, or the transaction and quote data for each security. Many anomaly detection methods are based on clustering algorithms and do not require labelled data, however, the performance evaluation of such methods are debatable. Ferdousi et al. applied Peer Group Analysis (PGA) to transactional data in stock market to detect outlier traders [78]. The dataset consists of three months of real data from the Bangladesh stock market that is claimed to be an appropriate dataset as securities fraud mostly appears in emerging markets [78] such as Bangladesh stock market. The data is represented using statistical variables (mean and variance) of buy and sell orders under fixed time periods. The npeer is set as a predefined parameter describing the number of objects in a peer group and controls the sensitivity of the model. A target object is decided a member of a peer group if members of the peer group are the most similar objects to the target object. After each time window (5 weeks) peer groups are summarized to identify the centroid of the peer group. Then, the distance of peer group members with the peer group's centroid is calculated using t-statistic, and objects that deviate significantly from their peers are picked as outliers. Trader accounts that are associated with these objects are flagged as suspicious traders that suddenly behaved differently to their peers.

IBM Watson Research Center proposed an efficient method for detecting burst events in stock market [231]. First, a burst is detected in financial data based on a variable threshold using the skewed property of data (exponential distribution), second, the bursts are indexed using Containment-Encoded Intervals (CEIs) for efficient storing and access in the database. This method can be used for fraud detection or identifying fraudulent behaviour in the case of triggering fraud alarms in real-time. The burst patterns of stock trading volume before and after 9/11 attack is investigated using the proposed approach and the experimental results confirm that the method is effective and efficient compared to B+tree. We elaborate on anomaly detection methods on time series, as this is the focus of our proposed method.

Chapter 3

Detecting Stock Market Manipulation using Supervised Learning Algorithms

The standard approach in application of data mining methods for detecting fraudulent activities in securities market is using a dataset that is produced based on the litigation cases. The training dataset would include fraudulent observations (positive samples) according to legal cases and the rest of observations as would be normal (negative samples) [54] [164] [118] [203]. We extend the previous works through a set of extensive experiments, adopting different supervised learning algorithms for classification of market manipulation samples using the dataset introduced by Diaz et al. [54]. We adopt different decision tree algorithms [248], Naive Bayes, Neural Networks, SVM and kNN.

We define the classification problem as predicting the class of $\{Y \in 0, 1\}$ based on a feature set of $\{X_1, X_2, \ldots, X_d | X_i \in \mathbb{R}^2\}$ where Y represents the class of a sample (1 implies a manipulated sample) and X_i represents features such as price change, number of shares in a transaction (i.e. volume), etc. The dataset is divided to training and testing dataset. First, we apply supervised learning algorithms to learn a model on the training dataset, then, the models are used to predict the class of samples in the testing dataset.

3.1 Case Study

We use the dataset that Diaz et al. [54] introduced in their paper on analysis of stock market manipulation. The dataset is based on market manipulation cases through SEC between January and December of 2003. The litigation cases that include the legal words related to market manipulation ("manipulation", "marking the close" and "9(a)" or "10(b)") are used as manipulated label for that stock and is added to the stock information such as price, volume, the company ticker etc. Standard and Poor's[1] COMPUSTAT database is employed for adding the supplementary information and also including non-manipulated stocks (i.e. control samples). The control stocks are deliberately selected from stocks that are similar to manipulated stocks (the selection is based on similar market capitalization, beta and industry sector). Also, a group of dissimilar stocks were added to the dataset as a control for comparison of manipulated and non-manipulated cases with similar characteristics. These stocks are selected from Dow Jones Industrial (DJI) companies. The dataset includes 175,738 data observations (hourly transactional data) of 64 issuers (31 dissimilar stocks, 8 manipulated stocks and 25 stocks similar to manipulated stocks) between January and December of 2003. There are 69 data attributes (features) in this dataset that represent parameters used in analytical analysis. The dataset includes 27,025 observations for training and the rest are for testing. We only use the training dataset to learn models for identifying manipulated samples.

[1]Standard and Poor is an American financial services and credit rating agency that has been publishing financial research and analysis on stocks and bonds for over 150 years.

3.2 Methods

A. Decision Trees

Decision trees are easy to interpret and explain, non-parametric and typically are fast and scalable. Their main disadvantage is that they are prone to overfiting, but pruning and ensemble methods such as random forests [28] and boosted trees [198] can be employed to address this issue. A classification tree starts with a single node, and then looks for the binary distinction, which maximizes the information about the class (i.e. minimizing the class impurity). A score measure is defined to evaluate each variable and select the best one as the split:

$$score(S,T) = I(S) - \sum_{i=1}^{p} \frac{N_i}{N}\, I(S_i) \tag{3.1}$$

where T is the candidate node that splits the input sample of S with size N into p subsets of size N_i $(i = 1, \ldots, p)$ and $I(S)$ is the impurity measure of the output for a given S. Entropy and Gini index are two of the most popular impurity measures and in our problem (i.e. binary classification) are:

$$I_{entropy}(S) = -(\frac{N_+}{N}\, log\frac{N_+}{N}) - (\frac{N_-}{N}\, log\frac{N_-}{N}) \tag{3.2}$$

$$I_{gini}(S) = \Big[\frac{N_+}{N}\,(1 - \frac{N_+}{N})\Big] + \Big[\frac{N_-}{N}\,(1 - \frac{N_-}{N})\Big] \tag{3.3}$$

where N_+ represents the number of manipulated samples (i.e. positive samples), N_- represents the number of non-manipulated samples (negative samples) in a given subset. This process is repeated on the resulting nodes until it reaches a stopping criterion. The tree that is generated

through this process is typically too large and may *overfit*, thus, the tree is pruned back using a validation technique such as cross validation. CART [29] and C4.5 [197] are two classification tree algorithms that follow the greedy approach for building the decision tree (above description). CART uses the Gini index and C4.5 uses the entropy as their impurity function (C5.0 that we used in our experiments is an improved version of C4.5).

Although pruning a tree is effective in reducing the complexity of the tree, generally it is not effective in improving the performance. Algorithms that aggregate different decision trees can improve performance of the decision tree. Random forest [28] is a prominent algorithm that builds each tree using a bootstrap sample. The principle behind random forest is using a group of *weak learners* to build a *strong learner*. Random forest involves an ensemble (*bagging*) of classification trees where a random subset of samples is used to learn a tree in each split. At each node a subset of variables (i.e. features) is selected and the variable that provides the best split (based on some objective function) is used for splitting. The same process is repeated in the next node. After training, a prediction for a given sample is done through averaging votes of individual trees. There are many decision tree algorithms but it has been shown random forest, although very simple, generally outperforms other decision tree algorithms in the study on different datasets by Rich Caruana et al. [35]. Therefore, experimental results using random forest provide a reasonable proxy for utilizing decision trees in our problem.

B. Naive Bayes

Applying the Bayes theorem for computing $P(Y = 1|X)$ we have

$$P(Y = 1|X = x_k) = \frac{P(X = x_k|Y = 1)\ P(Y = 1)}{\sum_j P(X = x_k|Y = y_j)\ P(Y = y_j)} \tag{3.4}$$

where the probability of Y given kth sample of X (i.e. x_k) is divided by sum over all legal values for Y (i.e. 0 and 1). Here the training data is used to estimate $P(X|Y)$ and $P(Y)$ and the above Bayes rule is used to resolve the $P(Y|X = x_k)$ for the new x_k. The Naive Bayes makes the conditional independence assumption (i.e. for given variables X, Y and Z, $(\forall i, j, k)\ \ P(X = x_i|Y = y_j;\ Z = z_k) = P(X = x_i|Z = z_k)$) to reduce the number of parameters that need to be estimated. This assumption simplifies $P(X|Y)$ and the classifier that determines the probability of Y, thus

$$P(Y = 1|x_1, \ldots, x_n) = \frac{P(Y = 1)\ \prod_i P(X_i|Y = 1)}{\sum_j P(X|Y = y_j)\ \prod_i P(X_i|Y = y_j)} \tag{3.5}$$

The above equation gives the probability of Y for the new sample $X\langle X_1, \ldots, X_n\rangle$ where $P(X_i|Y)$ and $P(Y)$ are computed using the training set. However we are only interested in the maximum likelihood in the above equation and the simplified form is:

$$\widehat{y} = \arg\max_{y_k} P(Y = y_k)\ \prod_i P(X_i|Y = y_k) \tag{3.6}$$

C. Neural Networks

An Artificial Neural Network in contrast to Naive Bayes estimates the posterior probabilities directly. A Neural Network to learn a model for classification of manipulated samples can be viewed as the function, $F : \mathbb{R}^d \rightarrow \{0, 1\}$, where X is a d-dimensional variable. This is a function that minimizes the overall mean squared error [173]. The output

of the network can be used as the sign predictor for predicting a sample as positive (i. e. manipulated). We adopted the back propagation algorithm of neural networks [193]. The principle behind neural networks, taken from the function of a human neuron, is a nonlinear transformation of the activation into a *prescribed reply*. Our neural network consists of three layers, input layer (the number of nodes in this layer is equal to the number of features, X_i), hidden layer (it is possible to consider multiple hidden layers) and output layer (there is a single node in this layer representing Y). Each node is a neuron and the network is fully connected (i.e. all neurons, except the neurons in the output layer have axioms to the next layer). The weight of neurons in each layer is updated in the training process using $a_j = \sum_{i=1}^{d} X_i W_{ij}$ and the response of a neuron is calculated using the sigmoid function, $f(a_j) = \dfrac{1}{1 + exp(-a_j)}$ which is fed forward to the next layer. The weights are updated in the training process such that the overall mean squared error, $SSE = \dfrac{1}{2} \sum_{j=1}^{N} (Y - \widehat{Y})^2$ is minimized, where Y is the actual value, $\widehat{Y}$ is the network output and N is the number of samples.

D. Support Vector Machines

We adopt binary SVM for classification [32] of manipulated samples where $Y \in -1, 1$ (i.e. 1 represents a manipulated sample). The main idea behind SVM is finding the *hyperplane* that maximizes the marginal distance (i.e. sum of shortest distances) to data points in a class. The samples in input space are mapped to a feature space using a kernel function to find the *hyperplane*. We use the linear kernel in our experiments (other widely used kernels for SVMs are polynomial, radical basis function (RBF) and sigmoid [99]). The SVM is trying to find w and b in the

hyperplane $w \cdot x_i - b = \pm 1$ which means the marginal distance of $\frac{2}{\|w\|}$ should be maximized. This is an optimization problem of minimizing $\|w\|$ subject to $y_i(w \cdot x_i - b) \geq 1$. A simple trick to solve the optimization problem is working with $\frac{1}{2}\|w\|^2$ to simplify derivation. The optimization problem becomes $argmin_{w,b}\ \frac{1}{2}\|w\|^2$ subject to $y_i(w \cdot x_i - b) \geq 1$ and this can be solved through standard application of the Lagrange multiplier.

E. k-Nearest Neighbour

kNN [49] is a simple algorithm that assigns the majority vote of k training samples that are most similar to the to the new sample. There are different similarity measures (i.e. distance measures) such as Euclidean distance, Manhattan distance, cosine distance, etc. kNN is typically used with Euclidean distance. The linear time complexity of Euclidean distance ($O(n)$) makes it an ideal choice for large datasets. We use kNN with Euclidean distance as the similarity measure of the k nearest samples for binary classification.

F. Performance Measure

Misclassification costs are unequal in fraud detection because false negatives are more costly. In other words, missing a market manipulation case (i.e. positive sample) by predicting it to be non-manipulated (i.e. negative sample), hurts performance of the method more than predicting a sample as positive while it is actually a negative sample (i.e. manipulated case). Threshold, ordering, and probability metrics are effective performance measures for evaluating supervised learning methods for fraud detection [176]. According to our studies the most effective metrics to

evaluate the performance of supervised learning methods in classification of market manipulation include Activity Monitoring Operating Characteristic (AMOC) [76] (average score versus false alarm rate), Receiver Operating Characteristic (ROC) analysis (true positive rate versus false positive rate), mean squared error of predictions, maximizing Area under the Receiver Operating Curve (AUC), minimizing cross entropy (CXE) [230] and minimizing Brier score [230].

We use ROC analysis in our experiments reporting sensitivity, specificity and F_2 measure. Let True Positive (TP) represent the number of manipulated cases classified correctly as positive, False Positive (FP) be the number of non-manipulated samples that are incorrectly classified as positive, True Negative (TN) be the number of non-manipulated samples that are correctly classified as positive and False Negative (FN) be the number of manipulated samples that are incorrectly classified as negative, the *precision* and *recall* are $P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$ respectively. Sensitivity or *recall* measures the performance of the model in correctly classifying manipulated samples as positive, while the Specificity, $SPC = \frac{TN}{TN+FP}$ measures the performance of the model in correctly classifying non-manipulated samples as negative. We use F_2 measure because unlike F_1 measure, which is a harmonic mean of precision and *recall*, the F_2 measure weights *recall* twice as much as precision. This is to penalize misclassification of TP more than misclassification of TN. The *F-Measure* is defined as

$$F_\beta = (1+\beta^2) * \frac{P * R}{(\beta^2 * P) + R} = \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + (\beta^2 * FP) + FP} \quad (3.7)$$

and F_2 *measure* is a special case of *F-Measure* where β is equal to 2.

Table 3.1: Stock Market Anomaly Detection using Supervised Learning Algorithms

Algorithm	Sensitivity	Specificity	Accuracy	F2 measure
Naive Bayes	***0.89***	***0.83***	***0.83***	***0.53***
CART	0.54	0.97	0.94	0.51
Neural Networks	0.68	0.81	0.80	0.40
CTree	0.43	0.95	0.93	0.40
C5.0	0.43	0.92	0.89	0.35
Random Forest	0.32	0.96	0.92	0.30
kNN	0.28	0.96	0.93	0.26

3.3 Results and Discussion

Diaz et al. [54] and some previous works used the raw price of securities as a feature in their modelling. We argue that although the price is the most important variable that should be monitored for detecting market manipulation, it should not be used in its raw form. The price of a stock does not reflect the size of a company nor the revenue. Also, the wide range of stock prices is problematic when taking the first difference of the prices. We propose using the price percentage change (i.e. return), $R_t = (P_t - P_{t-1})$ or $log(P_t - P_{t-1})$ where R_t and P_t represent return and price of the security at time t respectively. Furthermore, this is a normalization step, which is a requirement for many statistical and machine learning methods (the sample space of R_t is $[-1, M]$ and $M > 0$). We used stock returns in our experiments and removed the raw price variable from the datasets.

The baseline F_2 measure on the testing dataset (6,685 positive/manipulated samples and 137,373 negative samples) is 17%. If a hypothetical model (this would be also ineffective) predicts all samples as manipulated, clearly the *recall* is 100% but the specificity would be 4%, thus, F_2 measure of 17%. Some related works report the accuracy [54] or overall specificity and sensitivity (i.e. combining performance measures on training and testing datasets or including the performance of models in correctly classifying non-manipulated

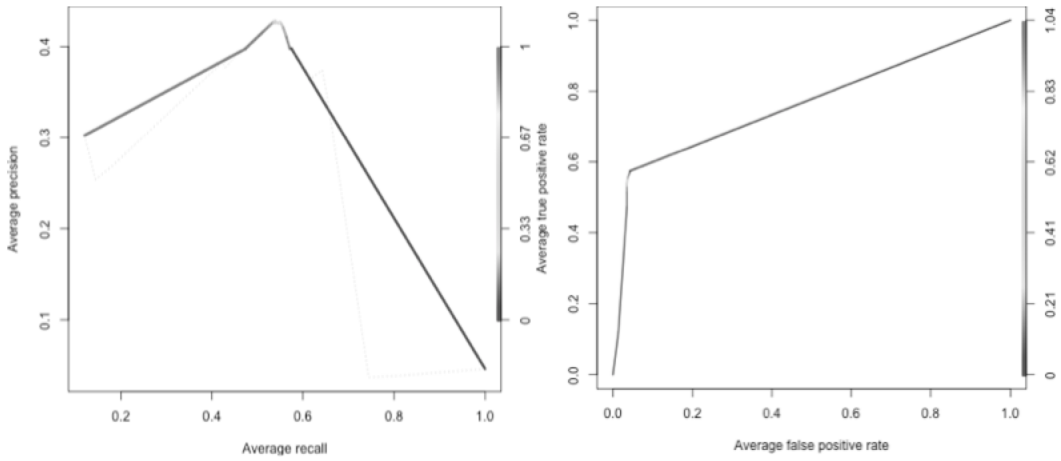


Figure 3.1: Performance results using CART - (a) comparing average precision and recall (b) comparing average TP and FP rates

samples). We emphasize that these numbers may be misleading (some of the worst models that we built in our experiments with respect to correctly classifying manipulated samples, easily exceed accuracy rates of 90%) because a) the misclassification costs for manipulated and non-manipulated cases are unequal, and, b) the number of samples in the manipulated class is typically significantly lower than the number of samples in the non-manipulated class. In our experiments, we focus on performance of the models on correctly classifying manipulated samples.

Table 3.1 describes a summary of performance measures of the supervised learning algorithms that we adopted to detect market manipulation on the testing dataset. All the algorithms listed in the table outperform the baseline significantly but SVM which fails to improve the baseline (fine-tuning parameters and using other kernel functions are expected to improve results and we will pursue this avenue in our future work). Decision trees generally produce models that rank high in our experiments. These models are relatively fast and it is possible to improve the results slightly with tweaking the parameters (we did not find significant performance improvements) or using a grid to optimize the parameters. We avoided exhaustive search for best parameters as it is a

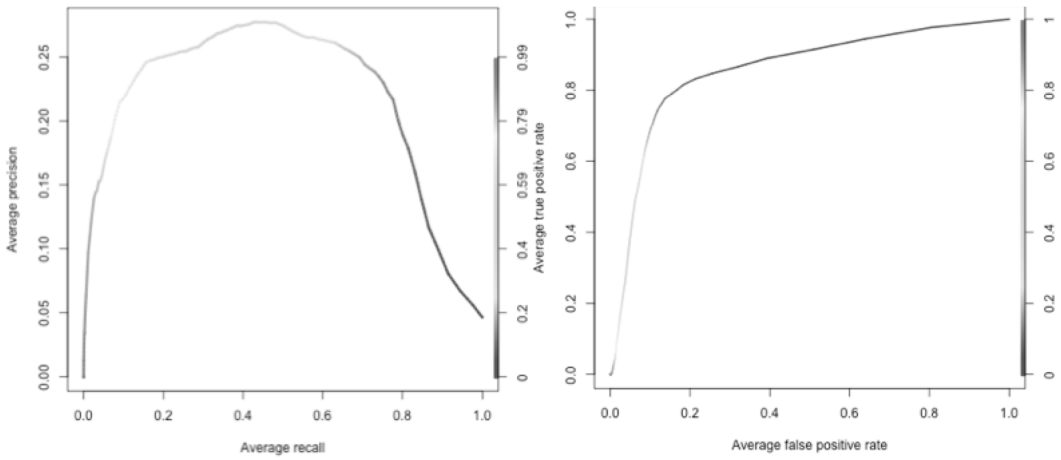


Figure 3.2: Performance results using Random Forest - (a) comparing average precision and recall (b) comparing average TP and FP rates

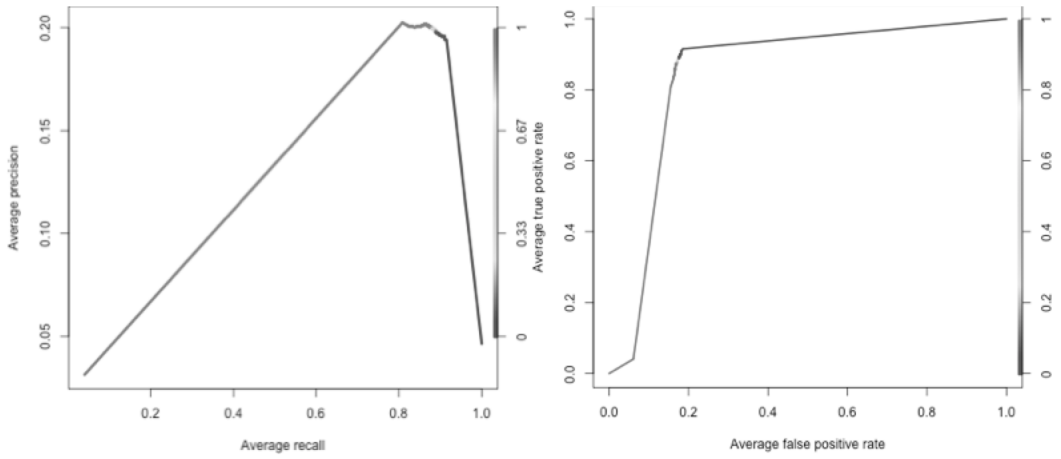


Figure 3.3: Performance results using Naive Bayes - (a) comparing average precision and recall (b) comparing average TP and FP rates

risk factor for overfitting. The Naive Bayes outperform other algorithms in our experiments with sensitivity and specificity of 89% and 83% respectively. Figures 3.1, 3.2 and 3.3 illustrate ROC curves describing the performance of models based on CART, Random Forest and Naive Bayes.

We use kNN with equal weights and this most likely gives the lower bound performance of kNN on the testing dataset. A future work may use weighted kNN [202] to allow different weights for features (e.g. using Mahalanobis distance [239] to give more weight to features with higher variance). The same

principle can be pursued in regression decision trees using a regularizer term to assign different weights to features. Furthermore, we tackle the issue of imbalanced classes by boosting the number of manipulated samples in our datasets through SMOTEBoost [44] and applying decision tree algorithms to the new datasets. The initial results using SMOTEBoost improves performance of the models but the improvements are not significant. We are working on other approaches for boosting the number of samples in the minority class that is highly desired in developing data mining methods for detecting market manipulation. The results indicate adopting supervised learning algorithms to identify market manipulation samples using a labelled dataset based on litigation cases is promising.

Our studies show that supervised learning algorithms are i) straightforward to implement and interpret, and ii) provide high performance results in classifying market manipulation cases from normal cases. However, this approach has some drawbacks which make it impractical for identifying potential market manipulation in stock market including:

1. nonlinear time complexity resulting in computationally expensive methods,

2. relying on labelled data.

The requirement of labelled data is the key drawback that makes supervised learning approaches inappropriate for detecting potential stock market manipulation, because, the outcomes are based on very limited set of samples compared to the number of stocks and variability of different industry sectors in stock market. Furthermore, as we explained in Chapter 1, labelled data for stock market manipulation is generally not available in large scale. In Chapter 4.5 we attempt to address disadvantages of adopting supervised learning algorithms by developing an unsupervised learning algorithm for identifying

anomalies in complex time series. The proposed method is particularly useful for detecting potential market manipulation in stock market due to its low time complexity.

Chapter 4

Contextual Anomaly Detection

The classic approach in anomaly detection is comparing the distance of given samples with a set of normal samples and assigning an anomaly score to the sample. Then, samples with significant anomaly scores are labelled as outliers/anomalies. Anomaly detection approaches can be divided into two categories: i) searching a dictionary of known normal patterns and calculating distances (supervised learning methods), and ii) deriving a normal pattern based on characteristics of the given samples (unsupervised learning methods).

The problem of distinguishing normal data points or sequences from anomalies is particularly difficult in complex domains such as the stock market where time series do not follow a linear stochastic process. Previously, we developed a set of prediction models using some of the prominent existing supervised learning methods for fraud detection in securities market on a real dataset that is labelled based on litigation cases [87]. In that work, we adapted supervised learning algorithms to identify outliers (i.e. market manipulation samples) in stock market. We used a case study of manipulated stocks during 2003 that David Diaz introduced in his paper on analysis of stock market manipulation [54]. The dataset is manually labelled using SEC cases. Empirical results showed that Naive Bayes outperformed other learning methods achieving an F_2 measure of 53% while the baseline F_2 measure was 17% (Table 3.1 shows

a summary of the results). We extended the existing work on fraud detection in securities by adopting other algorithms, improving the performance results, identifying features that are misleading in the data mining process, and highlighting issues and weaknesses of these methods. The results indicate that adopting supervised learning algorithms for fraud detection in securities market using a labelled dataset is promising (see Chapter 3 for details of the methods and experimental results). However, there are two fundamental issues with the approach: first, it may be misleading to generalize such models to the entire domain as they are trained using one dataset, and second, using labelled datasets is impractical in the real world for many domains, especially securities market. This is because theoretically there are two approaches for evaluating outlier detection methods: i) using a labelled dataset, and ii) generating a synthetic dataset for evaluation. The standard approach in producing a labelled dataset for fraud detection in securities is using litigation cases to label observations as anomaly for a specific time and taking the rest of observations as normal. Accessing labelled datasets is a fundamental challenge in fraud detection and is impractical due to different costs associated to manually labelling data. It is a laborious and time consuming task, yet all existing literature on fraud detection in securities market using data mining methods, are based on this unrealistic approach [54] [118] [203] [164].

In an attempt to address challenges in developing an effective outlier detection method for non-parametric time series that are applicable to fraud detection in securities, we propose a prediction-based Contextual Anomaly Detection (CAD) method. Our method is different from the conventional prediction-based anomaly detection methods for time series in two aspects: i) the method does not require the assumption of time series being generated from a deterministic model (in fact as we indicated before, stock market time series are non-parametric and researchers have not been able to model these

time series with reasonable accuracies to date [235]), and ii) instead of using a history of a given time series to predict its next consecutive values, we exploit the behaviour of similar time series to predict the expected values.

The input to CAD is the set of similar time series $\{ X_i | i \in \{ 1, 2, \cdots, d \} \}$ such as stock time series within an industry sector of S&P and the window size parameter win. These time series are expected to have a similar behaviour as they share similar characteristics including underlying factors which determine the time series values. First, a subset of time series is selected based on the window size parameter (we call this step chunking), Second, a centroid is calculated representing the expected behaviour of time series of the group within the window. The centroid is used along with statistical features of each time series X_i (e.g. correlation of the time series with the centroid) to predict the value of the time series at time t (i.e. $\widehat{x_{it}}$).

We determine the centroid time series within each chunk of time series by computing the central tendency of data points at each time t. Figure 4.1 describes the stocks return in energy sector of S&P 500 during June 22 to July 22 of 2016. The red point represents the mean of values at the timestamp t. In an earlier work we showed using mean for determining the centroid of time series in an industry sector within a chunk is effective [86]. In this chapter we also explore other aggregation functions to determine centroid time series and their impact in anomaly detection including median (i.e. middle value in a sorted list of numbers), mode (the most frequent number in a list of numbers) and maximum probability.

Kernel Density Estimation (KDE) is a powerful non-parametric density estimation model especially because it does not have the issue of choice of binning in histograms (the binning issue results in different interpretation of data). We use KDE to estimate the probability of x_{it}:

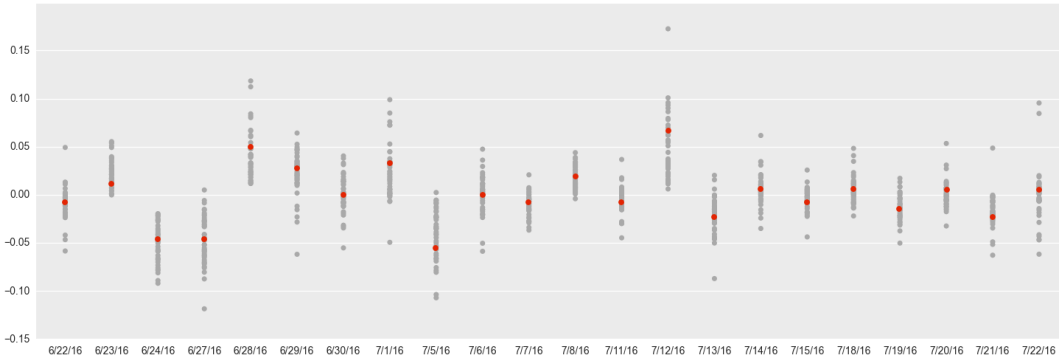


Figure 4.1: Stocks return distributions and means in energy sector of S&P 500

$$P(x) = \frac{1}{Nh}\sum_{n=1}^{N} K(\frac{x - x_n}{h}) \tag{4.1}$$

where N is the total number of time series (thus N values at each time t) and h is the bandwidth parameter (the function $K(.)$ is the kernel). The expectation of the equation gives the expected value of the probability:

$$E(P(x)) = \frac{1}{Nh}\sum_{n=1}^{N} E(K(\frac{x - x_n}{h})) = \frac{1}{h}\, E(K(\frac{x - x_n}{h})) = \frac{1}{h}\int K \cdot P(x')dx' \tag{4.2}$$

We use the Gaussian kernel for KDE (there are some other kernels such as tophat, exponential and cosine) which results in recovering a smoother distribution. Using Gaussian kernel as the kernel on univariate values on a given time t we get:

$$P(x) = \frac{1}{Nh}\sum_{n=1}^{N}(2\pi)^{-\frac{1}{2}}\ \ e^{-\frac{1}{2}(\frac{x - x_n}{h})^2} \tag{4.3}$$

The above kernel density is an estimate of the shape of distribution of values at t using the sum of Guassians surrounding each datapoint. Figure 4.2 describes KDE distribution on the energy stocks returns of S&P 500 during June 22 to July 22 of 2016 (the input to this figure is the same as Figure 4.1). The red points represent the values that have the maximum probability given

Figure 4.2: Centroid calculation using KDE on stocks return in energy sector of S&P 500

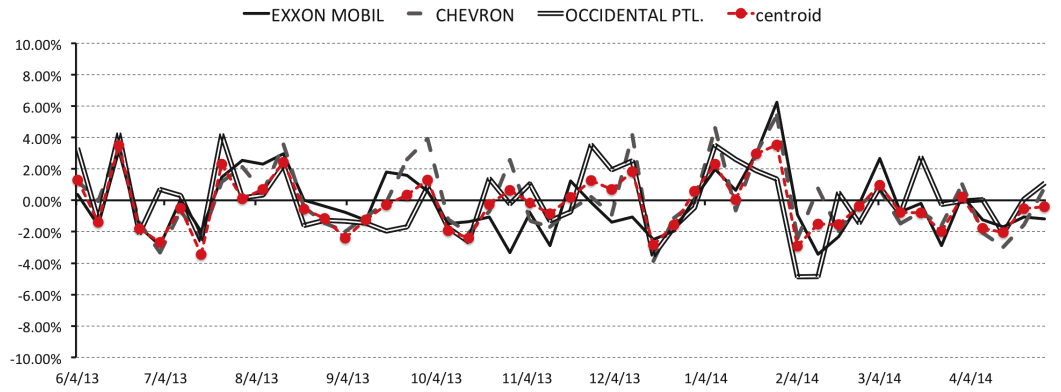


Figure 4.3: Centroid time series given stocks in S&P 500 energy sector

the distribution at the time t.

The centroid time series C is computed within each chunk of time series using the aggregate function as $\{C_j | j \in \{1, 2, \ldots, t\}\}$. Figure 4.3 shows the centroid of energy sector.

Algorithm 1 describes the CAD algorithm. This is a lazy approach, which uses the centroid along with other features of the time series for predicting the values of $\widehat{X_{it}}$:

$$\widehat{X_{it}} = \Psi(\Phi(X_i), C) + \epsilon \tag{4.4}$$

where $\widehat{X_{it}}$ is the predicted value for the time series X_i at time t, $(\Phi(X_t))$ is a function of time series features (e.g. the value of X_i at time stamp $t-1$, drift,

auto regressive factor, etc.), Ψ specifies the relationship of a given time series feature with the value of centroid at time t (i.e. c_t), and ϵ is the prediction error (i.e. $\sqrt{(\widehat{X_{it}} - X_{it})^2}$). In this thesis we use the value of the given time series at time $t-1$ as the time series feature (i.e. x_{it-1}) to represent ($\Phi(X_t)$). The centroid time series C is the expected pattern (i.e. $E(X_1, X_2, \cdots, X_d)$) which can be computed by taking the mean or any aggregate function that aims to determine the central tendency of values of time series X_i at each time stamp t.

We define Ψ as the multiplication of time series value at time $t-1$ (i.e. $\Phi(X_t)$) and correlation of time series X_i and the centroid ($\rho(X_i, C)$ is the correlation of time series X_i and C in Algorithm 1). The correlation is determined using the Pearson correlation of a given time series and the centroid (i.e. $\rho(X_i, C) = \frac{cov(X_i, C)}{\sigma_{X_i}\sigma_C}$ where cov is covariance and σ is standard deviation). We use the correlation of each time series with the centroid to predict values of the time series because if the centroid correctly represents the pattern of time series in a group (i.e. industry sector), the correlation of individual time series with the centroid is an indicator of time series values. Third, we assign an anomaly score by taking the Euclidean Distance of the predicted value and the actual value of the given time series (the threshold is defined by the standard deviation of each time series in the window). It has been shown that the Euclidean Distance, although simple, outperforms many complicated distance measures and is competitive in the pool of distance measures for time series [83] [115]. Moreover, the linear time complexity of Euclidean distance makes it an ideal choice for large time series. Finally, we move the window and repeat the same process. Figure 4.3 depicts the centroid time series within three time series of S&P energy sector with weekly frequency and a window size of 15 data points.

Algorithm 1 CAD Algorithm

Require: A set of similar time series

Input: Time series $\{X_i | i \in \{1, 2, \ldots, d\}\}$, window size and overlap size (overlap is set to 4 data points in our experiments). $strt \in \mathcal{N}$ is the start of window, $end \in \mathcal{N}$ is the end of window, $win \in \mathcal{N}$ is the window size and $\{olap \in \mathcal{N} | olap < win\}$ is the length of windows overlap

Output: Set of anomalies on each time series

Initialization $strt = olap$
while $strt \leq end - win$ **do**
  $strt = start - olap$ {calculate the time series centroid C of X_i}
  for $i = 0$ to d **do**
    $c_i = \rho(X_i, C)$
    for $j = 0$ to win **do**
      predict data point x_{ij} in X_i using c_i
      if $dist_{Euclidean}(x_j, \widehat{x_j}) > std(X_i)$ **then**
        return x_j
      end if
    end for
  end for
  $strt = strt + win$
end while

There are different methods to compute the expected behaviour of similar time series such as taking the mean value of all time series at each time stamp t. We used median and mode in addition to mean in our experiments. Furthermore, we explored using maximum likelihood using KDE to capture a value which maximizes the probability within the distribution of time series values at each time stamp t.

4.1 Time Complexity

The problem of anomaly detection in securities involves many time series with huge length. This makes the computational complexity of anomaly detection methods important especially in presence of High Frequency Trading (HFT) where thousands of transactions are recorded per second in each time series (i.e. stock). The proposed method is linear with respect to the length of input

time series. The centroid can be calculated in $O(n)$ and using the Euclidean distance adds another $O(n)$ to the computation leaving the overall computational complexity of the method in linear order (including other statistical features of a given time series such as drift and autoregressive factor in the predictive model will have the same effect on the computational complexity). However, there are constants such as the number of time series d and the number of local periods (e.g. 1-year periods that are used to capture outliers within that period of the original time series) that are multiplied to the total length of time series n. The constants are expected to be much smaller than the input size thus should not affect the order of computational complexity.

It is possible to use time series anomaly detection methods which have higher time complexity. However, these methods would be inappropriate for detection potential market manipulation in securities market because i) there are thousands of stocks in the market and this number is growing, ii) the number of transactions are enormous and rapidly increasing especially with the introduction of HFT a few years ago which resulted in billions of transactions per day, iii) there are many other financial instruments that are traded in the market and are subject to market manipulation similar to stocks (e.g. bonds, exchange traded funds, etc.).

4.2 Unlabelled Data and Injection of Outliers

We propose a systematic approach to synthesize data by injecting outliers in real securities market data that is known to be manipulation-free. The market data that we use - S&P constituents' data is fraud-free (i.e. no market manipulation) thus considered outlier-free in the context of our problem. This is due to many reasons, most importantly, these stocks are:

- the largest companies in USA (with respect to their size of capital) and very unlikely to be cornered by one party or a small group in the market,
- highly liquid (i.e. there are buyers and sellers at all times for the security and the buy/sell price-spread is small) thus practically impossible for a party to take control of a stock or affect the price in an arbitrary way,
- highly monitored and regulated both by analysts in the market and regulatory organizations.

These are the major reasons which make S&P stocks a reliable benchmark for risk analysis, financial forecasting and fraud detection with a long history in industry and in numerous research works [62] [102] [164].

In our proposed approach, values of synthetic outliers for a given time series are generated based on the distribution of subsequences of the given time series (e.g. in periods of 1 year). It is important to note that our proposed outlier detection method follows a completely different mechanism and is not affected by the process of outlier injection in any way (we elaborate more on this at the end of this section). The conventional approach in defining outliers for a normal distribution $N(\mu, \sigma^2)$, is taking observations with distance of three standard deviation from the mean (i.e. $\mu \pm 3\sigma$) as outliers. However, when the distribution is skewed we need to use a different model to generate outliers. We adopted Tukey's method [225] for subsequences that do not follow a normal distribution. It has been shown that Tukey's definition for outliers is an effective approach for skewed data [204]. Formally, we propose generating artificial outliers using the following two-fold model:

$$\tau(x_{it}) = \begin{cases} \mu + [Q_3 \pm (3 * IQR)] & if \quad \gamma_1 > \epsilon \\ \mu \pm 3\sigma & if \quad N(\mu, \sigma^2) \end{cases} \tag{4.5}$$

where Q_1 is the lower quartile (25th percentile), Q_3 is the upper quartile

(75th percentile), IQR represents the inter-quartile (i.e. Q_3-Q_1) of the data, and γ_1 represents the skewness or third moment of the data distribution:

$$\gamma_1 = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{\sum_1^k (x_i - \mu)^3}{n} \tag{4.6}$$

and k is the length of the subsequence of time series X_i (i.e. number of data points in the subsequence). γ_1 is 0 for a normal distribution as it is symmetric. The values in a given time series are randomly substituted with the synthetic outliers $\tau(x_{it})$. We emphasize that the process of injecting outliers to create synthesized data using the real market data is completely separate from our anomaly detection process. Anomalies are injected randomly and this information is not used in the proposed anomaly detection process. The injected outliers in a time series are based solely on the time series itself and not the group of time series. Furthermore, the outlier detection method that we propose is an unsupervised learning method and the ground truth that is based on the synthetic data, is only used to evaluate performance of the proposed method and the competitive methods after capturing outliers. Injecting anomalies for evaluating outlier detection methods has been attempted in different domains such as intrusion detection [72]. One may ask, assuming the above model defines outliers, can we use this same two-fold model approach to identify outliers for a given set of time series? The answer is no, because the statistical characteristics of the time series such as mean, standard deviation and skewness are affected by outliers, therefore, these values may be misleading as the input time series include outliers.

We use the market data from S&P constituents datasets that are considered outlier-free. The process to synthesize artificial outliers described in this section is used to inject outliers in the real datasets. These datasets are used as the input data for the outlier detection methods in our experiments. We use the performance measures precision, recall and F-measure in our experiments.

If the null hypothesis is that all and only the outliers are retrieved, absence of type I and type II errors correspond to maximum precision (no false positives) and maximum recall (no false negatives) respectively. Precision is a measure of exactness or quality, whereas recall is a measure of completeness or quantity. We compare performance of the proposed method with two competing algorithms for time series anomaly detection, Naive predictor (Random walk) and kNN. In this thesis we identified three criteria for effective anomaly detection methods in stock market: i) have $O(n)$ or close to linear time complexity, ii) be able to detect individual anomalous data points, iii) rely on an unsupervised learning approach. The proposed method is designed to satisfy these criteria. Random walk and kNN are carefully selected as competing methods satisfying these criteria. Random walk is a widely accepted benchmark for evaluating time series forecasting [84], which predicts x_{t+1} through a random walk (a jump) from x_t. Random walk is equivalent to ARIMA (0,1,0) (Auto-Regressive Integrated Moving Average) [27]. This model does not require the stationary assumption for time series, however, assumes that the time series follow a first-order Markov process (because the value of X_{t+1} depends only on the value of X at time t). x_{t+1} is anomalous if it is significantly deviated from its prediction. We use kNN as a proximity based approach for outlier detection. Furthermore, kNN, although simple, reached promising results in the work on detecting stock market manipulation in a pool of different algorithms including decision trees, Naive Bayes, Neural Networks and SVM. For each data point p we calculate $D^k(p)$ as the distance to all other kth nearest points (using Euclidean Distance). A data point p would be anomalous if $D^k(p)$ is significantly different from other data points q with $D^k(p)$ (i.e. larger than three standard deviation).

4.3 Performance Measure

The conventional performance measures are inappropriate for anomaly detection because the misclassification costs are unequal. The second issue which makes performance evaluation challenging is unbalanced classes. Anomaly detection for detecting stock market manipulation encompasses both properties because i) false negatives are more costly, as missing a market manipulation period by predicting it to be normal hurts performance of the method more than including a normal case by predicting it to be market manipulation, and, ii) the number of market manipulations (i.e. anomalies) constitute a tiny percentage of the total number of transactions in the market. We argue the performance measure should focus on correctly predicting anomalies and avoid including results of predicting normals because the performance evaluation should primarily target predicting anomalies. We use F-measures, similar to Chapter 3.3, with higher β values to give higher weights to recall of correctly identifying anomalies:

$$F_\beta = (1+\beta^2) * \frac{P * R}{(\beta^2 * P) + R} = \frac{(1+\beta^2) * TP}{(1+\beta^2) * TP + (\beta^2 * FP) + FP} \tag{4.7}$$

where P and R represent the precision and recall respectively ($P = \frac{TP}{TP+FP}$ and $R = \frac{TP}{TP+FN}$), TP is true positives (the number of anomalies predicted correctly as anomalies), FP is false positives (the number of normal data points that are predicted as anomalies), TN is true negatives (the number of normal data points that are predicted as normal), FN is false negatives (the number of anomalies that are incorrectly predicted as normal), and $\beta \in \mathbb{N}$ and $\beta > 0$. In our experiments, we set β to 4 and report F-measures for all algorithms and experimental setups consistently. We chose the value 4 to illustrate the impact of giving a higher weight to recall while consistently reporting F-2

measure which is widely used in literature. It is possible to use higher β values and in our case it would improve the aggregated F-measure as the recall of the proposed method is substantially higher than its precision.

4.4 Data

We use several datasets from different industry sectors of S&P 500 constituents (see Appendix B for more information on S&P sectors). We use these datasets in two different granularities of daily and weekly frequencies. The S&P 500 index includes the largest market cap stocks that are selected by a team of analysts and economists at Standard and Poor's. The S&P 500 index is the leading indicator of US equities and reflects the characteristics of top 500 largest market caps. As we indicated in Section 4.2, these stocks (time series) are assumed to have no anomalies (i.e. no manipulations), as they are highly liquid and closely monitored by regulatory organizations and market analysts. We use 10 different datasets including 636 time series over a period of 40 years. To the best of our knowledge, this study surpasses the previous works in terms of both the duration and the number of time series in the datasets. Table 4.1 describes the list of datasets that we extracted from Thompson Reuters database for experiments to study and validate our proposed method (the CSV files are available at www.ualberta.ca/~golmoham/thesis). The table includes the total number of data points with a finite value (excluding NaN) in each dataset. These time series are normalized (by taking the percentage change) in a preprocessing step of our data mining process. Normalizing and scaling features before the outlier detection process is crucial. This is also a requirement for many statistical and machine learning methods. For example, consider the price, which is the most important feature that should be monitored for detecting market manipulation in a given security. The price of

Table 4.1: List of datasets for experiments on stock market anomaly detection on S&P 500 constituents

S&P Sector	Number of time series	Number of data points [weekly frequency]	Number of data points [daily frequency]
Energy	44	63,000 +	315,000 +
Financials	83	117,000 +	587,000 +
Consumer Discretionary	85	111,000 +	558,000 +
Information Technology	66	80,000 +	395,000 +
Consumer Staples	40	64,000 +	323,000 +

a security would include the trace of market manipulation activities because any market manipulation scheme seeks profit from deliberate change in price of that security. However, the price of a stock neither reflects the size of a company nor the revenue. Also, the wide range of prices is problematic when taking the first difference of the prices. A standard approach is using the price percentage change (i.e. return), $R_t = (P_t - P_{t-1}/P_{t-1})$ where R_t and P_t represent return and price of the security at time t respectively. The sample space of R_t is $[-1, M]$ and $M > 0$. The ratio of artificial outliers that are injected in the outlier-free dataset (see section 4.2) is 0.001 of the total number of data points in each dataset.

4.5 Results and Discussion

We studied the performance of CAD through a set of comprehensive experiments. We ran experiments with different window sizes (15, 20, 24, 30 and 35) on all 10 datasets in 5 industry sectors of S&P 500 to compare performance of CAD with comparable linear and unsupervised learning algorithms, kNN and Random Walk. Table 4.1 describes the list of datasets in the experiments along with the number of time series in each dataset (i. e. stocks). Table 4.2 shows CAD performance results along with kNN and Random Walk for datasets with

weekly frequency using window size 15. CAD-mean, CAD-median and CAD-mode represent CAD algorithm using different central tendency measures of mean, median and mode for computing the centroid time series within each chunk. CAD-maxP utilizes KDE to determine the centroid time series by computing a dta point which maximizes the probability under KDE distribution curve at each time t.

Table A.1 in the appendix includes performance results for all window sizes.

Table 4.2: Comparison of CAD performance results with kNN and Random Walk using weekly S&P 500 data with window size 15 (numbers are in percentage format)

Dataset	**Algorithm**	**Prec.**	**Rec.**	**F2**	**F4**
Consumer Staples	CAD-maxP	0.32	36.84	1.53	4.74
	CAD-mean	0.33	34.70	1.59	4.86
	CAD-median	0.31	32.70	1.47	4.52
	CAD-mode	0.35	31.39	1.70	5.11
	kNN	0.28	6.02	1.17	2.71
	RandomWalk	0.24	1.65	0.75	1.22
Consumer Dis.	CAD-maxP	0.35	37.99	1.68	5.17
	CAD-mean	0.33	34.49	1.58	4.83
	CAD-median	0.33	32.29	1.59	4.84
	CAD-mode	0.38	31.83	1.79	5.37
	kNN	0.29	6.26	1.24	2.86
	RandomWalk	0.25	1.72	0.79	1.28
Energy	CAD-maxP	0.27	33.85	1.32	4.10
	CAD-mean	0.33	34.70	1.59	4.86
	CAD-median	0.30	31.39	1.47	4.49
	CAD-mode	0.31	29.66	1.51	4.57
	kNN	0.29	6.36	1.23	2.86
	RandomWalk	0.34	2.39	1.09	1.77
IT	CAD-maxP	0.33	41.03	1.60	4.96
	CAD-mean	0.34	33.69	1.63	4.98
	CAD-median	0.34	32.95	1.61	4.91
	CAD-mode	0.37	32.28	1.75	5.27
	kNN	0.33	6.83	1.40	3.19
	RandomWalk	0.32	2.14	1.00	1.60
Financials	CAD-maxP	0.32	33.25	1.55	4.73
	CAD-mean	0.34	35.47	1.65	5.05
	CAD-median	0.36	33.94	1.74	5.27
	CAD-mode	0.34	31.02	1.63	4.92

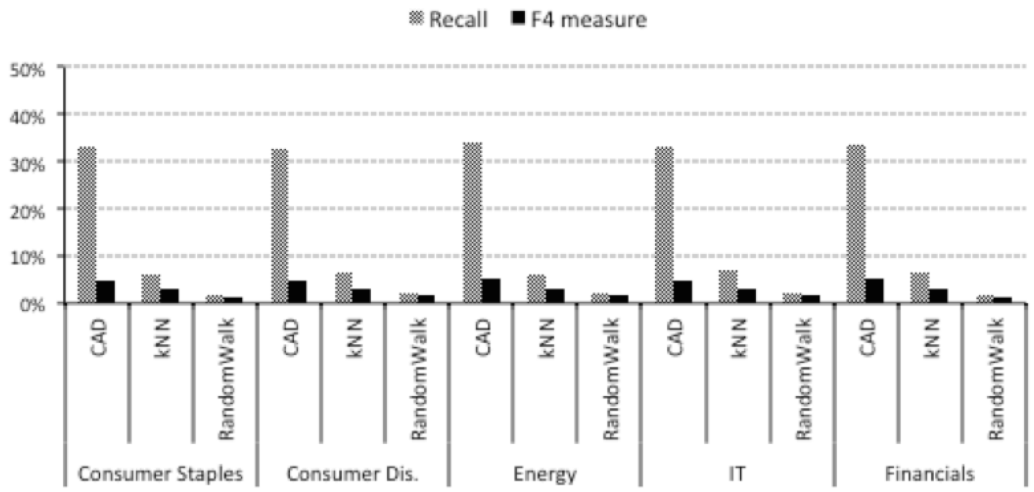


Figure 4.4: Average recall and F4-measure on weekly data of S&P sectors

kNN	0.34	7.18	1.42	3.27
RandomWalk	0.38	2.62	1.20	1.94

Figure 4.4 describes the average recall and F4-measure of the anomaly detection methods on each dataset with weekly frequency. It shows a similar trend where CAD clearly outperforms its contenders.

Figure 4.5 illustrates the average performance results over all window sizes of each anomaly detection method on each dataset with daily frequency. As can be noted, the results are stable regardless of the window size and dataset in the experiments. Our method, CAD, outperforms the other two methods on recall (i.e. it is superior at finding anomalies). A hypothetical predictor that predicts all data points as anomalies would reach an F4-measure of 0.016 since the injected outliers only represent 0.001 of the total number of data points. Our objective is maximizing recall without compromising precision. The precision is about 0.5% for all three algorithms while CAD reaches much higher recall in predicting anomalies. The baseline for precision (by predicting all data points as anomalies) is less than 0.04% because the total number of

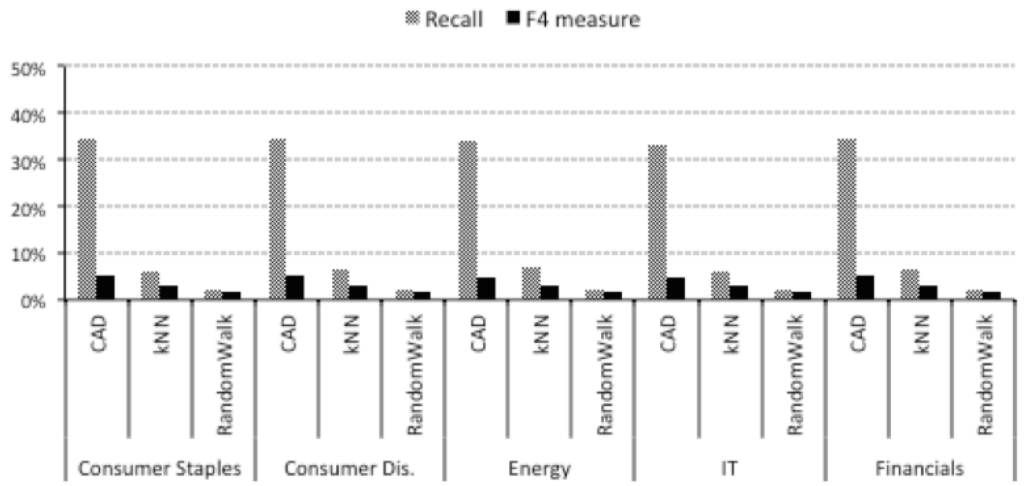


Figure 4.5: Average recall and F4-measure on daily data of S&P sectors

anomalies constitutes less than 0.1% of data, which drops to 0.04% after data preprocessing. Although avoiding false positives is generally desirable, it is not the primary focus in detecting stock market manipulation because missing an anomaly (potential market manipulation) hurts the method much more than incorrectly predicting a sample as anomalous (false positive). We emphasize that the objective of the proposed method is improving recall without compromising the precision measures using other applicable methods (precision of kNN and Random Walk is less than 0.5%). The experimental results show that CAD improves recall from 7% to 33% without compromising the precision.

In Chapter 4, we present a novel and formalized method to reduce the number of false positives in CAD by integrating information from resources other than market data using big data techniques.

Chapter 5

Big Data Techniques to Improve CAD Performance

We adopted big data techniques to improve performance of the Contextual Anomaly Detection (CAD) method by eliminating false positives. A formalized method is developed to explore the market participants' expectation for each detected datapoint. This information is used to filter out irrelevant items (false positives). Big data techniques are often used to predict consumer behaviour, primarily using social network services such as Twitter, Facebook, Google+ and Amazon reviews. We utilized big data for a novel application in time series anomaly detection, specifically stock market anomalies, by extracting information from Twitter. This information can be integrated into the anomaly detection process to improve the performance of the proposed anomaly detection by eliminating irrelevant outliers. Although outliers that are captured using outlier detection methods represent anomalous data points and periods, some of them may be irrelevant, because there might be a reasonable cause for the anomaly outside time series of market data (for example a news release about a company before the event may explain the abnormal stock return).

Big data is defined in Gartner glossary [1] as "high-volume, high-velocity and

[1] http://www.gartner.com/it-glossary/big-data/

high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making". Volume, Variety, and Velocity, also known as three V's, are three aspects of data management that are widely adopted to describe big data [129]. Volume refers to the size of data while velocity refers to the rate that data is generated and needs to be processed. It is challenging to set a threshold on volume and velocity to distinguish big data from regular data because it may depend on the type of data (e.g. imagery or video data versus sensory data) that dictates processing complexity. Furthermore, the perception of big volumes of data may change with advancements in hardware and software technologies. For example, in 90s one gigabyte of data was perceived as large volume data while in 2015 people typically would not refer to a dataset big if its size is smaller than a terabyte. Variability refers to the structure of data. For instance, video files often require a more complex and lengthier processing compared to tabular data. Big data techniques are methods to extract insights and knowledge from big data.

There have been numerous applications of Big data techniques in the past decade to improve predictions. The Netflix challenge is a classic example of the significance of using other resources of data to improve predictions and achieve insights. Netflix launched the Netflix prize in 2007 to improve predictions on user rating for movies. They offered one million dollars to developers of an algorithm that could improve prediction results of their existing system, Cinematch, by 10% given a dataset of users and ratings of movies and using only this matrix. The winning algorithm introduced in 2011 was never implemented in practice due to its complexity. However, it was shown that the prediction results could improve further using other data resources such as IMDB[2] that is readily available online containing other information such as movie genre,

[2]www.imdb.com

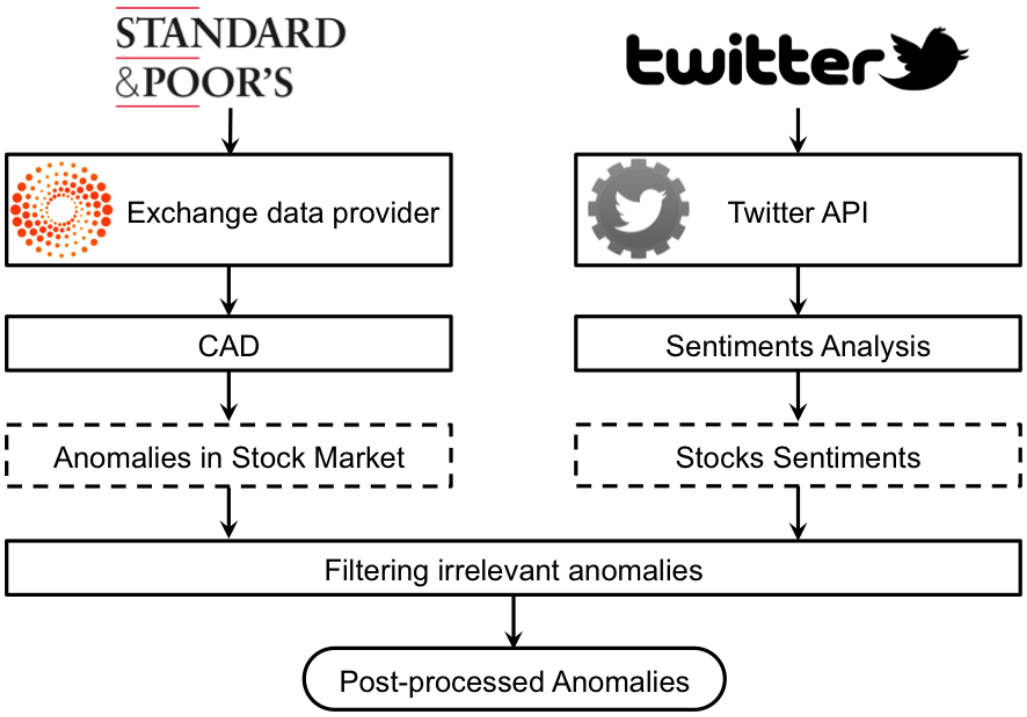


Figure 5.1: Utilizing Twitter Data to Improve Anomaly Detection in the Stock Market

actors, relationships, etc. Using a similar approach to integrate additional information to improve anomaly detection is particularly challenging in securities fraud detection, which are typical challenges in big data problems - velocity, volume, and variability. We are specifically interested in big data techniques to extract information from unstructured data from tweets.

We developed a case study to investigate sentiment analysis on Twitter to improve anomaly detection in the stock market. Figure 5.1 describes a high level overview of the process flow in the case study:

- **A.1)** extracting market data for Oil and Gas stocks of S&P 500,
- **A.2)** predicting anomalies in the Oil and Gas stocks,
- **B.1)** extracting tweets from Twitter for the Oil and Gas stocks in S&P 500,

- **B.2)** preparing a training dataset by extracting tweets for stocks in the Information Technology sector (this data is manually labelled as negative, neutral and positive by an individual who was not involved in developing the methods to preserve fairness of the study),

- **B.3)** building a model for sentiment analysis of tweets that is trained and tested using tweets on stocks (i.e. labelled tweets on the Information Technology sector),

- **B.4)** predicting sentiment of each stock per day using the sentiment analysis model (this produces a time series of sentiments for any given stock returns time series), and,

- **C.1)** filtering irrelevant anomalies based on the respective sentiment on the previous day of every detected anomaly.

In this chapter, we present a formalized method and a case study on the Oil and Gas industry sector of S&P 500 to reduce the number of false positives of the proposed Contextual Anomaly Detection (CAD) method in Chapter 3 of this thesis. We adopted sentiment analysis methods to develop a classifier specifically for sentiment analysis of tweets for the stock market. We extracted tweets and deduced daily sentiment for each company (i.e. positive, neutral and negative). This information is used to eliminate false positives of the contextual anomaly detection.

5.1 Sentiment Analysis

Sentiment analysis is the process of computationally identifying and categorizing people's opinions towards different matters such as products, events, organizations, etc. [20]. The impact of sentiment analysis has been significantly

increasing in the past few years in light of social networks and micro-blogging along with advancements in cloud computing and distributed data processing. The social interactions that users have on Facebook, Twitter, Pinterest, ebay, Amazon etc. make sentiment analysis invaluable for commercial interests in understanding the collective human opinion and behaviour online. Several experiments confirm prediction capabilities of sentiment analysis of social media content such as predicting size of markets [25] and unemployment rate [12]. User interactions in social media has attracted a great deal of attention in the past few years for predicting financial markets. Some research works suggest analyzing news and social media such as blogs, micro-blogs, etc. to extract public sentiments could improve predictions in the financial market [131] [201]. Feldman et al. [77] proposed a hybrid approach for stock sentiment analysis based on news articles of companies.

Twitter is the most popular micro-blogging platform. Twitter's technology and popular brand enable millions of people to share their opinions on variety of topics such as their well-being, politics, products, social events, market conditions and stock market. The flexible architecture and APIs enable researchers and industry to use twitter for various prediction purposes. Twitter was used to predict movie ticket sales in their opening week with accuracy of 97.3% [14]. We utilize Twitter in this thesis to identify people's opinions about stocks.

Sentiment analysis techniques could be used to automatically analyze unstructured data such as tweets in the neighbourhood time period of a detected anomaly. Textual analysis has a long history in the literature [55], however categorization through sentiments is recent [50] [160] [171] [222] [227] [241]. The typical approach for representing text for computational processes is based on a the bag-of-words (BOW) [55] where each document is represented by a vector of words. This bag-of-words is called a collection of unigrams. This

approach assumes a euclidean space of unigrams that are independent from each other. Thus, documents can be represented as a matrix where each row represents a document. Sentiment analysis methods can be divided into two groups while both use BOW:

1. **lexicon based method** [213] [56] that is an unsupervised approach where a polarity score is assigned to each unigram in the lexicon and the sum of all polarity scores of the text identifies the overall polarity of the text,

2. **machine learning approach** [169] that is a supervised approach where the unigrams or their combinations (i.e. N-grams) are used as features by classifiers.

Choosing lexicon elements for inclusion in the sentiment analysis is critical. There have been studies analyzing different parts-of-speech (e.g. adverbs, adjectives) for lexicon elements [169] [127]. The role of emotions has also been investigated in the past few years to enhance sentiment analysis of micro-blogs [179] [97] [143] [253]. Fersini et al. studied the use of adjectives, emoticons, emphatic and expressive lengthening as expressive signals in sentiment analysis of micro-blogs and concluded these signals can improve feature space and result in higher performing sentiment classification [79].

Some of the more sophisticated algorithms, include the context of where a message was published in addition to the message itself in the sentiment analysis [104][228] [18] [199]. The context is captured by different parameters such as the message author, network of the author (friends/followers) and the structure of the network. Hu et al. proposed using social relations to improve sentiment analysis of noisy and short texts [100]. Another example of including the context, is clustering of users, tweets and features for sentiment analysis in twitter [254]. Pozzi et al. explored authors network to estimate user polarities

by aggregating the messages with approvals in the network [180]. Saiff et al. improved sentiment analysis by utilizing semantic features in addition to message features [194].

There are some research works that focus on new sentiment classification methods instead of improving feature space such as utilizing ensemble methods (Bagging, Random Subspace and Boosting) which are shown to outperform base learners empirically [232]. Bayesian Model Averaging ensemble method is another research work in this body of work which is shown to outperform both traditional and ensemble methods [232]. Carvalho et al. improved classification accuracy by using genetic algorithm to identify subsets of words within a set of paradigm words [36].

5.2 Sentiment Analysis on Twitter

The recent technological advancements and internet along with massive data have drastically changed how people access and consume information for different purposes, particularly in social and economic sciences. Social media are increasingly reflecting and influencing the behaviour of other complex systems such as the stock market. Users interactions in social media is generating massive datasets that could explain the collective behaviour in a previously unimaginable fashion [117] [229]. We can identify interests, opinions, concerns and intentions of the global population with respect to various social, political, cultural and economic phenomena. Twitter, the most popular micro-blogging platform on internet, is at the forefront of the public commenting about different phenomena.

Twitter data is becoming an increasingly popular choice for financial forecasting [88] [162] [108]. Researchers have investigated whether the daily number of tweets predicts the S&P 500 stock return [152]. Ruiz et al. used a

graph-based view of Twitter data to study the relationship between Twitter activities and the stock market [192]. Some research works utilize textual analysis on twitter data to find relationships between mood indicators and the Dow Jones Industrial Average (DJIA) [25] [24] [151]. However, the correlation levels between prices and sentiments on twitter remains low in empirical studies especially when textual analysis is required. More recently, Bartov et al. found aggregated opinions on twitter can predict quarterly earnings of a given company [17]. These observations suggest a more complicated relationship between sentiments on twitter and stock returns.

Every day, a huge number of messages are generated on Twitter which provide an unprecedented opportunity to deduce the public opinions for a wide range of applications [142]. We intend to use the polarity of tweets to identify the expected behaviour of stocks in the public eyes. Here are some example tweets upon querying the keyword "$xom".

- $XOM flipped green after a lot of relative weakness early keep an eye on that one shes a big tell.
- #OILALERT $XOM >>Oil Rises as Exxon Declares Force Majeure on #Nigeria Exports
- Bullish big oil charts. No voice - the charts do the talking. `http://ln.is/www.youtube.com/ODKYG` $XLE $XOM $CVX $RDS $HES $OIH $SLB @TechnicianApp
- Barclays' Q2 Energy Earnings Expectations For Oil Majors & Refiners $XOM $COP $CVX $USO `http://benzinga.com/z/8203908`

The combination of the $ sign along with a company ticker is widely used on Twitter to refer to the stock of the company. As shown, the retrieved tweets may be about Exxon Mobil's stock price, contracts and activities. These

messages are often related to people's sentiments about Exxon Mobil Corp., which can reflect its stock trading.

We propose using twitter data to extract collective sentiments about stocks to filter false positives from detected anomalies in stocks. We study the sentiment of stocks at time $t-1$ where t is the timestamp of a detected anomaly. A sentiment that aligns with the stock return at time t confirms the return (i.e. aligns with expected behaviour) thus, indicates the detected anomaly is a false positive. We introduce a formalized method to improve anomaly detection in stock market time series by extracting sentiments from tweets and present empirical results through a case study on stocks of an industry sector of S&P 500.

5.2.1 Data

We use two datasets in this case study: Twitter data and market data. We extracted tweets on the Oil and Gas industry sector of S&P 500 for 6 weeks (June 22 to July 27 of 2016) using the Twitter search API. Table 5.1 shows the list of 44 Oil and Gas stocks in S&P 500 and respective number of tweets constituting 57,806 tweets.

Table 5.1: Tweets about Oil and Gas industry sector in S&P 500

Ticker	Company	cashtag	Tweets
APC	ANADARKO PETROLEUM	$APC	1052
APA	APACHE	$APA	1062
BHI	BAKER HUGHES	$BHI	1657
COG	CABOT OIL & GAS 'A'	$COG	736
CAM	CAMERON INTERNATIONAL	$CAM	255
CHK	CHESAPEAKE ENERGY	$CHK	4072
CVX	CHEVRON	$CVX	3038
COP	CONOCOPHILLIPS	$COP	1912
CNX	CONSOL EN.	$CNX	1023
DNR	DENBURY RES.	$DNR	1008

DVN	DEVON ENERGY	$DVN	1459
DO	DIAMOND OFFS.DRL.	$DO	1227
ESV	ENSCO CLASS A	$ESV	825
EOG	EOG RES.	$EOG	1149
EQT	EQT	$EQT	669
XOM	EXXON MOBIL	$XOM	5613
FTI	FMC TECHNOLOGIES	$FTI	511
HAL	HALLIBURTON	$HAL	2389
HP	HELMERICH & PAYNE	$HP	838
HES	HESS	$HES	917
KMI	KINDER MORGAN	$KMI	2138
MRO	MARATHON OIL	$MRO	2063
MPC	MARATHON PETROLEUM	$MPC	950
MUR	MURPHY OIL	$MUR	689
NBR	NABORS INDS.	$NBR	384
NOV	NATIONAL OILWELL VARCO	$NOV	827
NFX	NEWFIELD EXPLORATION	$NFX	779
NE	NOBLE	$NE	1102
NBL	NOBLE ENERGY	$NBL	583
OXY	OCCIDENTAL PTL.	$OXY	671
OKE	ONEOK	$OKE	651
BTU	PEABODY ENERGY	$BTU	186
PSX	PHILLIPS 66	$PSX	1205
PXD	PIONEER NTRL.RES.	$PXD	955
QEP	QEP RESOURCES	$QEP	713
RRC	RANGE RES.	$RRC	860
RDC	ROWAN COMPANIES CL.A	$RDC	476
SLB	SCHLUMBERGER	$SLB	1962
SWN	SOUTHWESTERN ENERGY	$SWN	1912
SE	SPECTRA ENERGY	$SE	421
TSO	TESORO	$TSO	1086
RIG	TRANSOCEAN	$RIG	1846
VLO	VALERO ENERGY	$VLO	1464
WMB	WILLIAMS COS.	$WMB	2471

There are two options for collecting tweets from Twitter: the Streaming API and the Search API. The Streaming API provides a real-time access to tweets through a query. It requires a connection to the server for stream of tweets. The free version of Streaming API and the Search API provide access

to a random sampling of about 1% of all tweets [3]. While the syntax of responses for the two APIs is very similar, there are some differences such as limitation on language specification on queries in Streaming API. We used the Search API to query recent English tweets for each stock in the Oil and Gas industry sector of S&P 500 using its cashtag. Twitter unveiled the cashtag feature in 2012 enabling users to click on a $ followed by a stock ticker to retrieve tweets about the stock. The feature has been widely adopted by users when tweeting about equities. We account for the search API rate limits by sending many requests for each stock with 10 second delays. The batch process runs daily to extract tweets and store them in a database.

The market data for stocks in Oil and Gas industry sector is extracted from Thompson Reuters following the same approach in Section 4.4. The stock returns are calculated as $R_t = (P_t - P_{t-1}/P_{t-1})$ where R_t, is the stock return and P_t and P_{t-1} are the stock price on days t and $t-1$ respectively.

5.2.2 Data Preprocessing

The response for a query on Twitter APIs includes several pieces of information such as username, time, location, retweets, etc. Figure 5.2 describes the JSON response for searching "$msft" representing a tweet about Microsoft's stock. For our purposes, we focus on the timestamp and tweet text. We store tweets in a mongoDB database ensuring each unique tweet is recorded once. mongoDB is an open source NoSQL database which greatly simplifies tweet storage, search, and recall eliminating the need of a tweet parser.

Tweets often include words and text that are not useful and potentially misleading in sentiment analysis. We remove URLs usernames and irrelevant texts and symbols. Our preprocessing includes three processes:

[3]The firehose access on Streaming API provides access to all tweets. This is very expensive and available upon case-by-case requests from Twitter.

```
{
    "_id": ObjectId("57677c67eb11ddc37749fac1"),
    "contributors": null,
    "truncated": false,
    "text": "RT @daytradingninja: Microsoft's $MSFT stock resumes trade, drops 4.2%
        premarket after LinkedIn buyout deal",
    "is_quote_status": false,
    "in_reply_to_status_id": null,
    "id": NumberLong(744652927445385217),
    "favorite_count": NumberInt(0),
    "entities": {
        "symbols": [{
            "indices": [
                NumberInt(33),
                NumberInt(38)
            ],
            "text": "MSFT"
        }],
        "user_mentions": [{
            "id": NumberInt(57849762),
            "indices": [
                NumberInt(3),
                NumberInt(19)
            ],
            "id_str": "57849762",
            "screen_name": "daytradingninja",
            "name": "DayTrading Ninja"
        }],
        "hashtags": [

        ],
        "urls": [

        ]
    },
    "retweeted": false,
    "coordinates": null,
    "source": "<a href=\"https://mobile.twitter.com\" rel=\"nofollow\">Mobile Web (M5)</a>",
    "in_reply_to_screen_name": null,
    "in_reply_to_user_id": null,
    "retweet_count": NumberInt(3),
    "id_str": "744652927445385217",
    "favorited": false,
    "retweeted_status": {
        "contributors": null,
        "truncated": false,
        "text": "Microsoft's $MSFT stock resumes trade, drops 4.2% premarket after LinkedIn
            buyout deal",
        "is_quote_status": false,
        "in_reply_to_status_id": null,
        "id": NumberLong(742339223072571393),
        "favorite_count": NumberInt(1),
        "entities": {
```


Figure 5.2: Sample JSON response for a tweet about Microsoft ($MSFT)

- **Tokenization** that involves extracting a list of individual words (i.e. bag of words) by splitting the text by spaces. These words are later used as features for the classifier.

- **Removing Twitter Symbols** which involves filtering irrelevant text out such as the immediate word after @ symbol, arrow, exclamation mark, etc.

- **Removing Stopwords** that involves removing words such as "the", "to", "in", "also", etc. by running each word against a dictionary.

- **Recording smiley faces** which involves translating smiley and sad faces to a positive and negative expression in the bag of words.

5.2.3 Modelling

We adopted three classifiers for determining sentiment of tweets including Naive Bayes, Maximum Entropy and Support Vector Machines. The same features are applied to all classifiers. The anomalous time series of $\{ \eta_k, 0 \leq k \leq n \}$ for the time series $\{ x_1, x_2, \ldots, x_n \}$ where η_k represents an anomaly on day k in the time series (i.e. stock) X. We check sentiment of the stock on day $k-1$ given η_k. We consider the detected anomaly as a false positive, if the sentiment confirms the change in stock return on day k, however, a sentiment that is in disagreement with the return on the next day implies unexpected stock behaviour, thus anomaly.

We study the proposed method for filtering out false positives within detected anomalies by first, running Contextual Anomaly Detection (CAD) method on an anomaly-free dataset (see Chapter 3 for CAD algorithm), second, removing detected anomalies in the first step that do not conform with their respective sentiment on Twitter. Figure 5.3 describes an example of

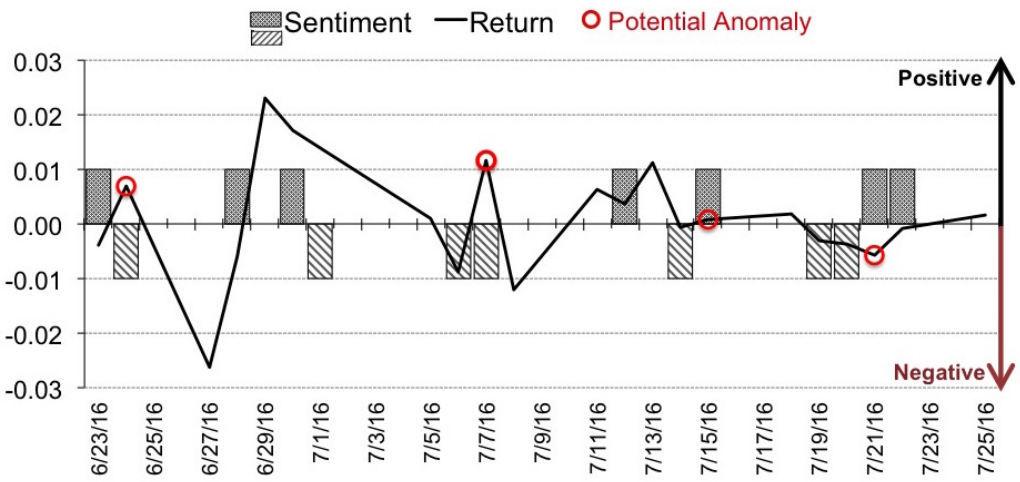


Figure 5.3: Identifying false positives in detected anomalies on Exxon Mobil (XOM)

stock sentiments on Twitter and anomalies that are detected on XOM (Exxon Mobil). The figure shows 4 anomalies (represented with red circles) that are detected on XOM along with the stock sentiment on Twitter for each day (days with no bars have the neutral sentiment). The data points on June 24 and July 21 are declared irrelevant because the stock's sentiments on the day before these dates confirm the change direction on the next day. However, other two anomalies (July 7 and 15) remain relevant because the sentiments on the day before the anomalies do not confirm the change direction in the stock return.

We found through our preliminary experiments that sentiment analysis using classifiers that are trained on movie reviews or generic tweets that are widely used in literature perform poorly for stock tweets. This is due to different corpus and linguistics that are specific to stock market. We developed a training dataset that is labelled manually to address this issue. Table 5.2 shows the list of 66 stocks in the Information Technology industry sector of S&P 500 and respective number of tweets constituting over 6,000 tweets. We manually labelled over 2,000 tweets by querying cashtags of the Information

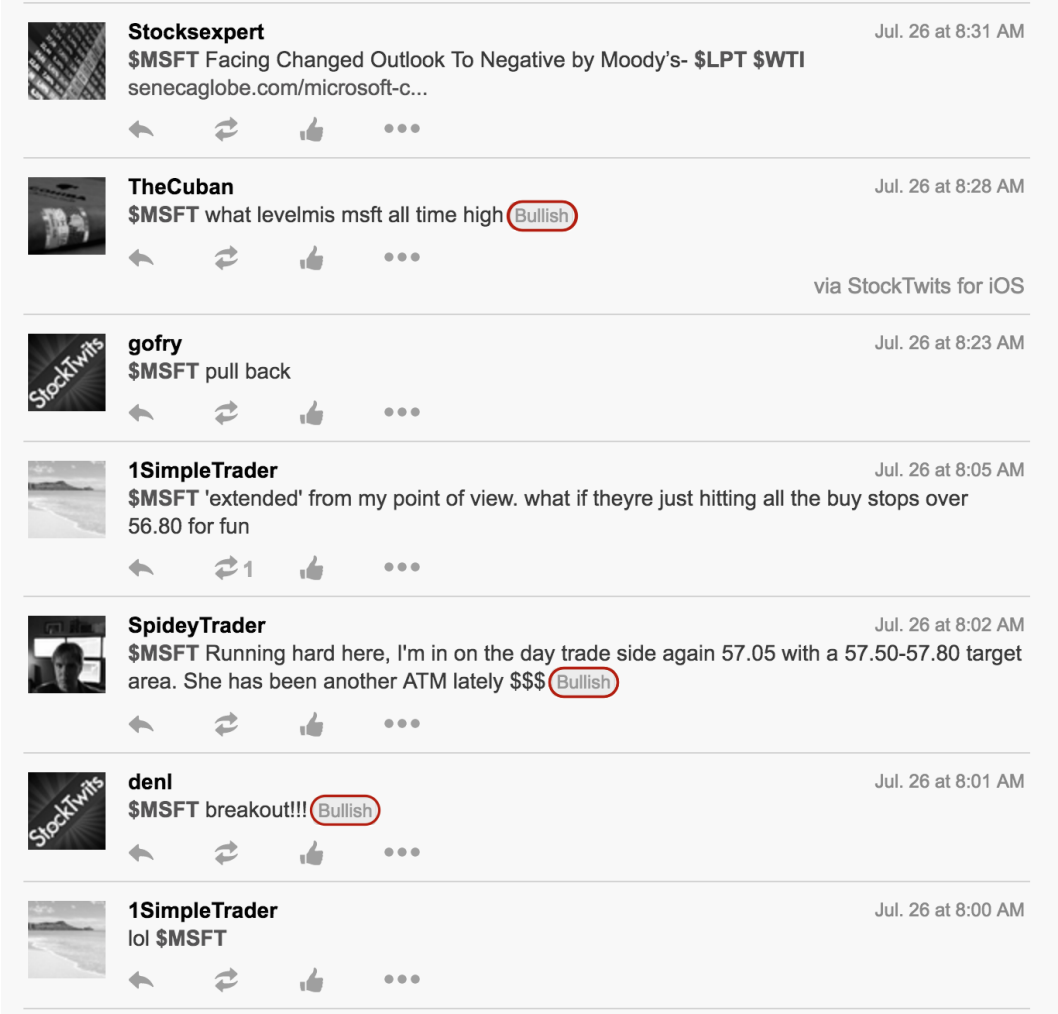


Figure 5.4: Micro-messages about the stock of Microsoft (MSFT) on StockTwits website and respective sentiments

Technology Stocks. We also used StockTwits [4], a widely popular social media platform that is designed for sharing ideas between investors and traders, to extract messages that are labelled by stock market participants. We developed a tool to query StockTwits for a given stock and extract relevant messages. Then, messages that are labelled by their poster as *Bearish* and *Bullish* are mapped to negative and positive sentiments in our code. Figure 5.4 shows example messages on StockTwits about the stock of Microsoft on July 26, 2016.

[4]http://stocktwits.com/

The training data is labelled manually with three sentiment labels: negative, neutral and positive. This data is used to train the classifiers that we used. The testing dataset is tweets about the Oil and Gas industry sector of S&P 500. Table 5.1 shows the list of 44 Oil and Gas stocks in the testing dataset S&P 500 and respective number of tweets constituting 57,706 tweets in total.

Table 5.2: Tweets about Information Technology industry sector in S&P 500

Ticker	Company	cashtag	Tweets
ACN	ACCENTURE CLASS A	$ACN	111
ADBE	ADOBE SYSTEMS	$ADBE	74
AKAM	AKAMAI TECHS.	$AKAM	81
ADS	ALLIANCE DATA SYSTEMS	$ADS	153
ALTR	ALTERA	$ALTR	70
APH	AMPHENOL 'A'	$APH	57
ADI	ANALOG DEVICES	$ADI	80
AAPL	APPLE	$AAPL	387
AMAT	APPLIED MATS.	$AMAT	82
ADSK	AUTODESK	$ADSK	82
ADP	AUTOMATIC DATA PROC.	$ADP	58
BRCM	BROADCOM 'A'	$BRCM	54
CA	CA	$CA	125
CSCO	CISCO SYSTEMS	$CSCO	103
CTXS	CITRIX SYS.	$CTXS	71
CTSH	COGNIZANT TECH.SLTN.'A'	$CTSH	60
CSC	COMPUTER SCIS.	$CSC	159
GLW	CORNING	$GLW	87
EBAY	EBAY	$EBAY	117
EA	ELECTRONIC ARTS	$EA	87
EMC	EMC	$EMC	63
FFIV	F5 NETWORKS	$FFIV	67
FB	FACEBOOK CLASS A	$FB	239
FIS	FIDELITY NAT.INFO.SVS.	$FIS	144
FSLR	FIRST SOLAR	$FSLR	86
FISV	FISERV	$FISV	83
FLIR	FLIR SYS.	$FLIR	64
GOOGL	GOOGLE 'A'	$GOOGL	162
GOOG	GOOGLE 'C'	$GOOG	260

HRS	HARRIS	$HRS	67
HPQ	HEWLETT-PACKARD	$HPQ	101
INTC	INTEL	$INTC	166
IBM	INTERNATIONAL BUS.MCHS.	$IBM	160
INTU	INTUIT	$INTU	77
JBL	JABIL CIRCUIT	$JBL	89
JNPR	JUNIPER NETWORKS	$JNPR	85
KLAC	KLA TENCOR	$KLAC	71
LRCX	LAM RESEARCH	$LRCX	80
LLTC	LINEAR TECH.	$LLTC	95
LSI	LSI DEAD - ACQD.BY 54332K	$LSI	63
MA	MASTERCARD	$MA	98
MCHP	MICROCHIP TECH.	$MCHP	65
MU	MICRON TECHNOLOGY	$MU	107
MSFT	MICROSOFT	$MSFT	387
MSI	MOTOROLA SOLUTIONS	$MSI	70
NTAP	NETAPP	$NTAP	74
NVDA	NVIDIA	$NVDA	154
ORCL	ORACLE	$ORCL	1029
PAYX	PAYCHEX	$PAYX	61
QCOM	QUALCOMM	$QCOM	121
RHT	RED HAT	$RHT	77
CRM	SALESFORCE.COM	$CRM	125
SNDK	SANDISK	$SNDK	61
STX	SEAGATE TECH.	$STX	91
SYMC	SYMANTEC	$SYMC	79
TEL	TE CONNECTIVITY	$TEL	57
TDC	TERADATA	$TDC	73
TXN	TEXAS INSTS.	$TXN	89
TSS	TOTAL SYSTEM SERVICES	$TSS	87
VRSN	VERISIGN	$VRSN	63
V	VISA 'A'	$V	349
WDC	WESTERN DIGITAL	$WDC	121
WU	WESTERN UNION	$WU	77
XRX	XEROX	$XRX	79
XLNX	XILINX	$XLNX	77
YHOO	YAHOO	$YHOO	106

A. Feature Selection

Feature selection is a technique that is often used in text analysis to improve performance of results by selecting the most informative features

(i.e. words). Features that are common across all classes contribute little information to the classifier. This is particularly important as the number of features grow rapidly with increasing number of documents. The objective is using the words that have the highest information gain. Information gain is defined as the frequency of the word in each class compared to its frequency in other classes. For example, a word that appears in the positive class often but rarely in the neutral and negative classes, is a high information word. Chi-square is widely used as a measure of information gain by testing the independence of a word occurrence and a specific class:

$$\frac{N(O_{w_pc_p} * O_{w_nc_n} - O_{w_nc_p} * O_{w_pc_n})^2}{O_{w_p} * O_{w_n} * O_{c_p} * O_{c_n}} \tag{5.1}$$

Where $O_{w_pc_p}$ is the number of observations of the word w in the class c and $O_{w_pc_n}$ is the number of observations of the word w in other classes (i.e. class negative).

This score is calculated for each word (i.e. feature) and used for ranking them. High scores indicate the null hypothesis H_0 of independence should be rejected. In other words, the occurrence of the word w and class c are dependent thus the word (i.e. feature) should be selected for classification. It should be noted that Chi-square feature selection is slightly inaccurate from statistical perspective due to the one degree of freedom. Yates correction could be used to address the issue, however, it would make it difficult to reach statistical significance. This means a small number of features out of the total selected features would be independent from the class. Manning et al. showed these features do not affect performance of the classifier [150].

B. Classifiers

- **Naive Bayes:** A Naive Bayes classifier is a probabilistic classifier based on the Bayes Rule $P(c|\tau) = \frac{P(\tau|c)P(c)}{P(\tau)}$ where $P(c|\tau)$ is the probability of class c being negative, neutral or positive given the tweet τ. The best class is the class that maximizes the probability given tweet τ:

$$C_{MAP} = \arg\max_{c \in C} P(\tau|c)P(c) \tag{5.2}$$

where $P(\tau|c)$ can be calculated using the bag of words as features resulting in

$$C_{MAP} = \arg\max_{c \in C} P(x_1, x_2, \ldots, x_n|c)P(c) \tag{5.3}$$

$P(c)$ can be calculated based on the relative frequency of each class in the corpus or dataset. There are two simplifying assumption in Naive Bayes which make calculating $P(x_1, x_2, \ldots, x_n|c)$ straightforward, i) position of the words do not matter, and ii) the feature probabilities $P(x_i|c_j)$ are independent given the class c:

$$P(x_1, x_2, \ldots, x_n|c) = P(x_1|c) \bullet P(x_2|c) \bullet \cdots \bullet P(x_n|c) \tag{5.4}$$

in other words we have the Multinomial Naive Bayes equation as

$$C_{NB} = \arg\max_{c \in C} P(c_j) \prod_{x \in X} P(x|c) \tag{5.5}$$

- **Maximum Entropy:** MaxEnt eliminates the independence assumptions between features and in some problems outperforms Naive Bayes. MaxEnt is a probabilistic classifier based on the Principle

of Maximum Entropy. Each feature corresponds to a constraint in a maximum entropy model. MaxEnt classifier computes the maximum entropy value from all the models that satisfy the constraints of the features for the given training data, and selects the one with the largest entropy. The MaxEnt probability estimation is computed using

$$P(c|f) = \frac{1}{Z(f)}\ exp\left(\sum_{i} \lambda_{i,c}\ F_{i,c}(f,c)\right) \tag{5.6}$$

where $Z(f)$ is a normalization function and $F_{i,c}$ is a binary function that takes the input feature f for the class c. λ is a vector of weight parameters that is updated iteratively to satisfy the tweets feature while continuing to maximize the entropy of the model [51]. The iterations eventually converge the model to a maximum entropy for the probability distribution. The binary function $F_{i,c}$ is only triggered when a certain feature exists and the sentiment is hypothesized in a certain class:

$$F_{i,c}(f,c') = \begin{cases} 1 & \text{if } n(f) > 0 \text{ and } c' = c \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

- **Support Vector Machines (SVM):**

  SVM is a linear classification algorithm which tries to find a hyperplane that separates the data in two classes as optimally as possible. the objective is maximizing the number of correctly classified instances by the hyperplane while the margin of the hyperplane is maximized. Figure 5.5 describes such a hyperplane in a 2D-space separating black and white points. The hyperplane representing the decision boundary in SVM is calculated by

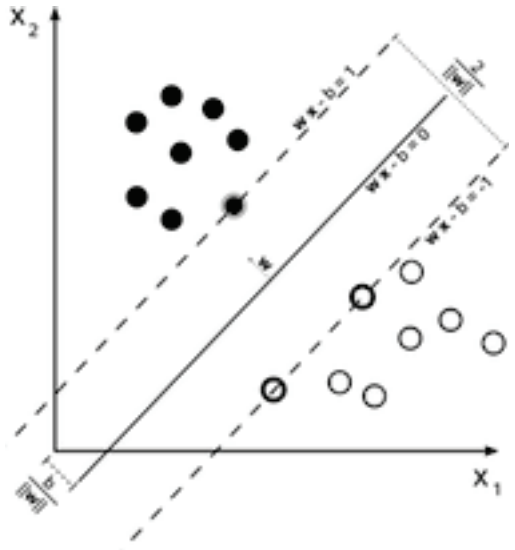


Figure 5.5: The minimum w in SVM gives the decision boundary with maximum margin

$$(\vec{w} \cdot \vec{x}) + b = \sum_i y_i \alpha_i (\vec{x_i} \cdot \vec{x}) + b = 0 \tag{5.8}$$

where weight vector $\vec{w} = (w_1, w_2, .., w_n)$ which is the normal vector defining the hyperplane is calculated using the n-dimensional input vector $\vec{x_i} = (x_{i1}, x_{i2}, .., x_{in})$, outputting the value y_i. α_i terms are the Lagrangian multipliers.

Calculating w using the training data gives the hyperplane which can be used to classify the input data instance $\vec{x_i}$. If $\vec{w} \cdot \vec{x_i} + b \geq 0$ then the input data instance is labelled positive (the class we are interested in), otherwise it belongs to the negative class (all of the other classes). It should be noted that although SVM is a linear classifier (as Naive Bayes and Maximum Entropy are) it is a powerful tool to classify text because text documents are typically considered as a linear dataset. It is possible to use Kernel functions for datasets that are not linearly separable. The Kernel is used to map the dataset to a higher dimensional space where the data could

be separated by a hyperplane using classical SVM.

There are two approaches for adopting SVM for a classification problem with multiple classes such as sentiment analysis with the classes negative, neutral and positive: i) one-vs-all where an SVM classifier is built for each class, and ii) one-vs-one where an SVM classifier is built for each pair of classes resulting in $\frac{M(M-1)}{2}$ for M classes. We used the latter for classifying sentiments using SVM.

In the one-vs-all approach the classifier labels data instances positive for the class that we are interested in and the rest of instances are labelled negative. A given input data instance is classified with classifier only if it is positive for that class and negative for all other classes. This approach could perform poorly in datasets that are not clustered as many data instances that are predicted positive for more than one class, will be unclassified. The one-vs-one approach is not sensitive to this issue as a data instance is categorized in the class with the most data instances, however, the number of classes can grow rapidly for problems with many classes (i.e. higher numbers of M).

C. Classifier Evaluation:

We trained classifiers using specifically stock tweets that are carefully labelled manually. We asked a person who has not been involved with training data to label the testing dataset. The testing data includes 1332 stock tweets that are manually labelled. We used 5-fold cross validation for training the classifiers that is sampling the data into 5 folds and using 4 folds for training and 1 fold for testing. This process is repeated 5 times and the performance results are averaged. We used precision and recall

for each class in addition to classification accuracy as performance measures to evaluate the classifiers. Precision of a classifier ($\frac{TP}{TP+FP}$) for a given class represents the fraction of the classified tweets that belong to the class, while recall ($\frac{TP}{TP+FN}$) represents the fraction of tweets that belong to the class out of all tweets that belong to the class. The precision for a class measures the exactness or quality, whereas recall measures the completeness or quantity. The classifier with the highest performance is used to predict sentiment of stocks in Oil and Gas industry sector (see Table 5.1 for the list of stocks in the Oil and Gas sector).

D. Calculating Polarity for each Stock:

The Twitter sentiments of stocks are predicted using an SVM classifier that is trained using labelled tweets about stocks. First, the classifier predicts sentiment of each tweet (i.e. negative, neutral and positive). Then, the polarity for each stock is computed using time series of negative, neutral and positive tweets:

- Negative tweets, tw_d^-: the number of negative tweets on day d
- Neutral tweets, tw_d^0: the number of neutral tweets on day d
- Positive tweets, tw_d^+: the number of positive tweets on day d

The polarity for each stock on a given day is the difference between the number of positive and negative tweets as a fraction of non-neutral tweets [251]. More formally

$$P_{s_d} = \frac{tw_d^+ - tw_d^-}{tw_d^+ + tw_d^-} \tag{5.9}$$

where P_{s_d} is the polarity of stock s on day d. Figure 5.6 shows the aggregated polarity of Exxon Mobil. The red dashed lines represent the

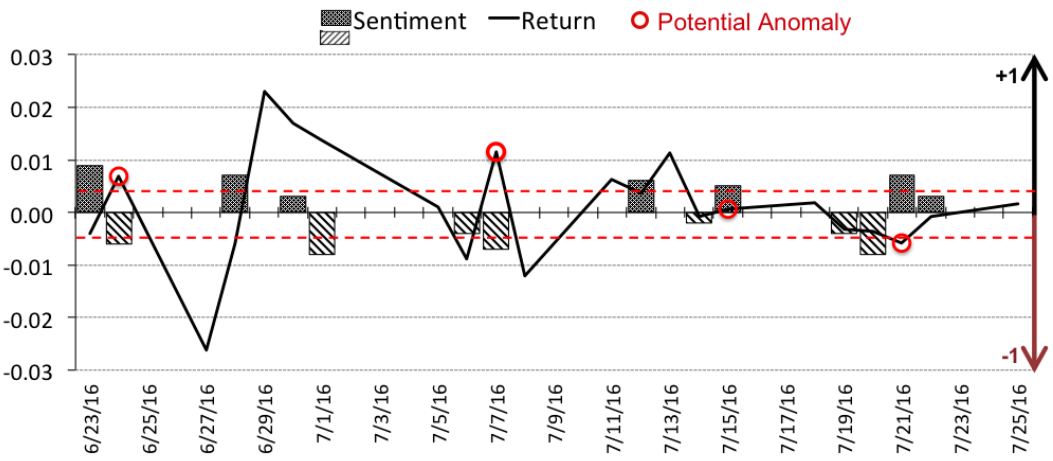


Figure 5.6: Polarity of Exxon Mobil stock per day along with potential anomalies that CAD produces

parameter *sentThreshold* that we define to control for the minimum magnitude of polarity that is required for declaring a potential anomaly a false positive. For example, the method would not include the polarity of Exxon Mobil on July 14 as an indicator to accept or reject the potential anomaly on July 15 as a false positive because its value is below the threshold. This parameter can be set during preliminary tests by trying a grid on *sentThreshold* (e.g. 0.2, 0.3, etc.) as we show in our experiments in Section 5.2.4.

5.2.4 Results and Discussion

We propose a two-step anomaly detection process. First, the anomalies are predicted on a given set of time series (i.e. stocks in an industry sector) using Contextual Anomaly Detection (CAD). Second, the anomalies are vetted using sentiment analysis by incorporating data in addition to market data. This process gives a list of anomalies that are filtered using data on Twitter. The first step, is based on an unsupervised learning algorithm due to various challenges with availability and applicability of labelled data for anomaly

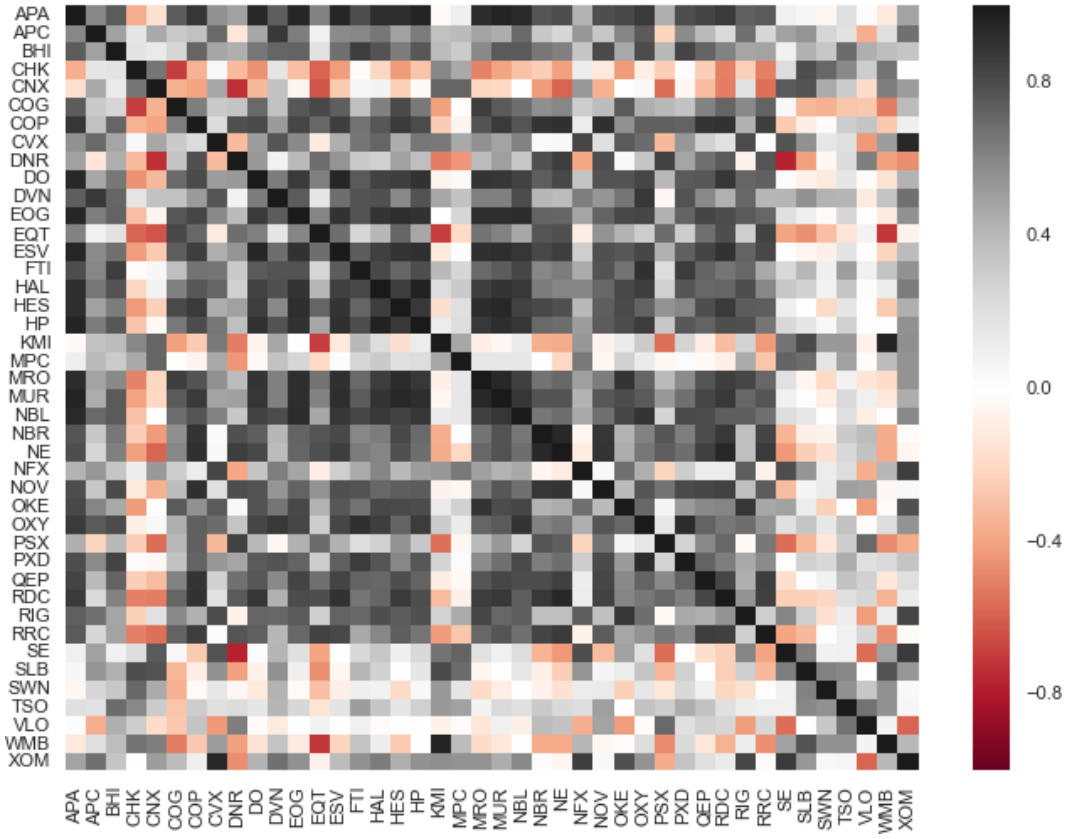


Figure 5.7: Correlation of stocks in Oil and Gas sector of S&P 500

detection in stock market (see Section 1.3 for discussion on challenges in identifying stock anomalies). The second step, relies on state-of-the-art supervised learning algorithms for sentiment analysis on unstructured data on Twitter.

We developed a set of experiments for this case study on the Oil and Gas sector of S&P 500 for the period of June 22 to July 27. Table D.1 in the Appendix shows the statistics on stock returns in Oil and Gas sector during this period. The correlation of stocks during this 6-week period for the case study is quite high as we expect within an industry sector. Figure 5.7 illustrates a heatmap of the correlation of stocks in the Oil and Gas sector of S&P 500 (see Table 5.1 for the list of companies and stocks in the Oil and Gas sector).

We studied several other classifiers in addition to the three classifiers that we introduced in Section 5.2.3 (i.e. Multinomial Naive Bayes, MaxEnt, also

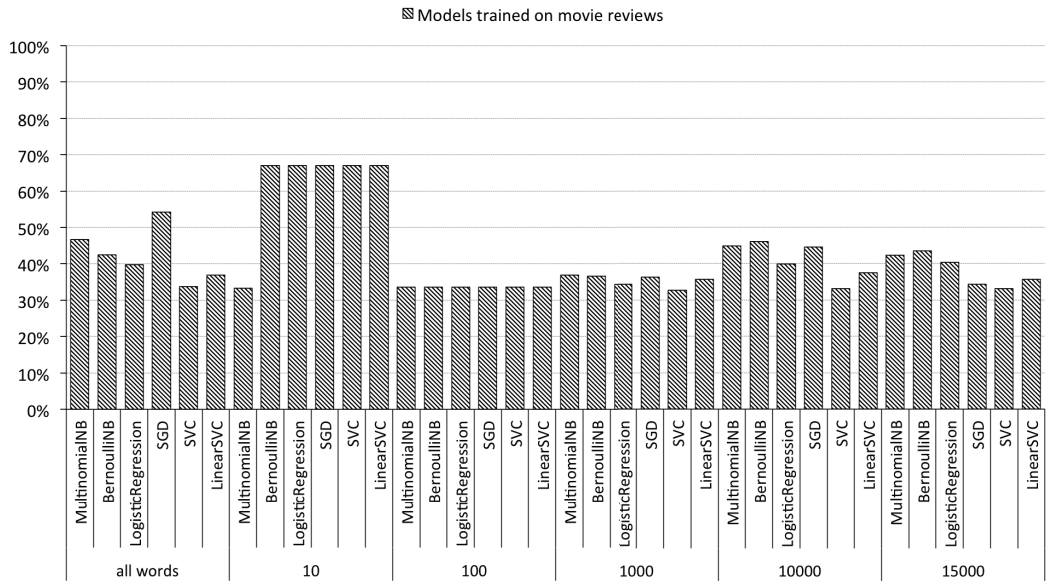


Figure 5.8: Accuracy of sentiment analysis models using training datasets in movie reviews

known as Logistic Regression, and SVM) to build a sentiment analysis model including Bernoulli Naive Bayes, Stochastic Gradient Descent (SGD) and C-Support Vector (SVC). Furthermore, we investigated the performance of sentiment analysis models using different number of features (i.e. 10, 100, 1000 etc. words).

Movie reviews data is typically used for sentiment analysis of short reviews as well as tweets [148]. This dataset includes movie reviews that are collected from IMDB [5]. Our experiments show that sentiment analysis models for stock tweets that are trained using this standard dataset perform poorly (see Figure 5.8 for an overview of accuracy measures when we use models that are trained on movie reviews). Table 5.3 describes the performance of different classifiers using movie reviews data and various sets of features based on accuracy, precision and recall. The results confirm our hypothesis that training data that is out of context is inappropriate for sentiment analysis of short text samples, particularly on Twitter.

[5]http://www.imdb.com/reviews/

Table 5.3: Classification results using different classifiers that are trained on movie reviews data

			Negative		Positive	
	Classifier	**Accuracy**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
all words	MultinomialNB	46.71	85.27	24.61	37.48	91.40
	BernoulliNB	42.51	83.16	17.67	35.78	92.76
	LogisticRegression	39.82	79.22	13.65	34.69	92.76
	SGD	54.19	81.90	40.49	40.49	81.90
	SVC	33.68	57.69	3.36	32.71	95.02
	LinearSVC	36.98	78.26	8.05	33.92	95.48
10	MultinomialNB	33.23	100.00	0.22	33.13	100.00
	BernoulliNB	67.07	67.02	100.00	100.00	0.45
	LogisticRegression	67.07	67.02	100.00	100.00	0.45
	SGD	67.07	67.02	100.00	100.00	0.45
	SVC	67.07	67.02	100.00	100.00	0.45
	LinearSVC	67.07	67.02	100.00	100.00	0.45
100	MultinomialNB	33.53	57.89	2.46	32.82	96.38
	BernoulliNB	33.53	57.89	2.46	32.82	96.38
	LogisticRegression	33.53	57.89	2.46	32.82	96.38
	SGD	33.53	57.89	2.46	32.82	96.38
	SVC	33.53	57.89	2.46	32.82	96.38
	LinearSVC	33.53	57.89	2.46	32.82	96.38
1000	MultinomialNB	36.98	96.43	6.04	34.38	99.55
	BernoulliNB	36.68	96.15	5.59	34.27	99.55
	LogisticRegression	34.28	83.33	2.24	33.38	99.10
	SGD	36.38	89.29	5.59	34.06	98.64
	SVC	32.63	45.16	3.13	32.03	92.31
	LinearSVC	35.63	90.48	4.25	33.85	99.10
10000	MultinomialNB	44.91	84.35	21.70	36.71	91.86
	BernoulliNB	46.11	77.02	27.74	36.29	83.26
	LogisticRegression	39.97	81.08	13.42	34.85	93.67
	SGD	44.61	81.82	22.15	36.38	90.05
	SVC	33.08	50.00	3.58	32.23	92.76
	LinearSVC	37.57	94.12	7.16	34.54	99.10
15000	MultinomialNB	42.37	79.25	18.79	35.41	90.05
	BernoulliNB	43.56	80.17	20.81	35.87	89.59
	LogisticRegression	40.42	83.56	13.65	35.13	94.57
	SGD	34.28	66.67	3.58	33.07	96.38
	SVC	33.08	50.00	3.58	32.23	92.76
	LinearSVC	35.63	81.48	4.92	33.70	97.74

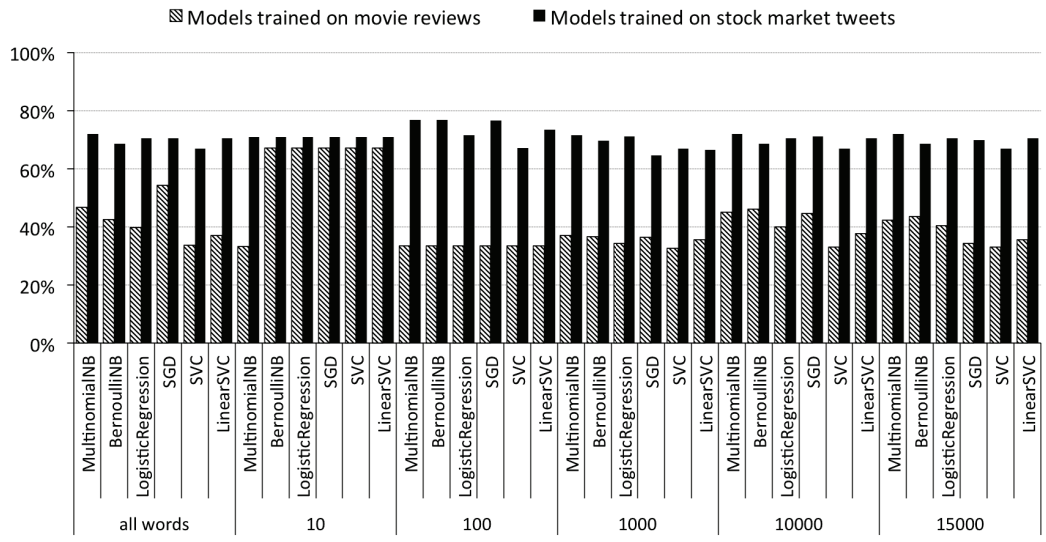


Figure 5.9: Accuracy of sentiment analysis models using training datasets in movie reviews and stock market

We developed a tool to extract labelled data from StockTwits [6] to address this issue (see Section 5.2.2 for more information on data). Table 5.4 shows performance results on models that are trained using data in the context of the stock market. Figure 5.9 illustrates that these models outperform models which are trained on movie review data consistently.

We observe that the number of features is an important parameter in the performance of sentiment analysis models. The results show using more features improves the performance results. However, performance of the models decay after hitting a threshold of about 10,000 features. This reiterates our hypothesis on utilizing feature selection to improve sentiment analysis on Twitter.

Table 5.4: Classification results using different classifiers

			Negative		Positive	
	Classifier	**Accuracy**	**Prec.**	**Rec.**	**Prec.**	**Rec.**
all words	MultinomialNB	72.01	71.96	95.30	72.37	24.89
	BernoulliNB	68.56	68.20	99.33	82.35	6.33
	LogisticRegression	70.51	69.59	99.33	90.00	12.22

[6] http://stocktwits.com/

	SGD	70.36	74.95	83.67	56.80	43.44
	SVC	66.92	66.92	100.00	NA	0.00
	LinearSVC	70.51	70.83	95.08	67.65	20.81
10	MultinomialNB	70.81	69.75	99.55	93.33	12.67
	BernoulliNB	70.81	69.75	99.55	93.33	12.67
	LogisticRegression	70.81	69.75	99.55	93.33	12.67
	SGD	70.81	69.75	99.55	93.33	12.67
	SVC	70.81	69.75	99.55	93.33	12.67
	LinearSVC	70.81	69.75	99.55	93.33	12.67
100	MultinomialNB	76.80	74.66	98.88	93.42	32.13
	BernoulliNB	76.80	74.66	98.88	93.42	32.13
	LogisticRegression	71.41	70.19	99.55	94.12	14.48
	SGD	76.50	75.89	95.08	79.63	38.91
	SVC	67.07	67.02	100.00	100.00	0.45
	LinearSVC	73.50	71.99	98.88	90.74	22.17
1000	MultinomialNB	71.41	71.19	96.20	73.44	21.27
	BernoulliNB	69.61	69.18	98.43	78.13	11.31
	LogisticRegression	71.11	70.22	98.66	85.00	15.38
	SGD	64.67	68.54	87.25	42.42	19.00
	SVC	66.92	66.92	100.00	NA	0.00
	LinearSVC	66.47	68.37	92.84	47.54	13.12
10000	MultinomialNB	72.01	71.96	95.30	72.37	24.89
	BernoulliNB	68.56	68.20	99.33	82.35	6.33
	LogisticRegression	70.51	69.59	99.33	90.00	12.22
	SGD	71.11	72.44	91.72	63.73	29.41
	SVC	66.92	66.92	100.00	NA	0.00
	LinearSVC	70.51	70.83	95.08	67.65	20.81
15000	MultinomialNB	72.01	71.96	95.30	72.37	24.89
	BernoulliNB	68.56	68.20	99.33	82.35	6.33
	LogisticRegression	70.51	69.59	99.33	90.00	12.22
	SGD	69.91	70.43	94.85	65.15	19.46
	SVC	66.92	66.92	100.00	NA	0.00
	LinearSVC	70.51	70.83	95.08	67.65	20.81

We studied impact of the proposed method in filtering false positives of CAD by first, running CAD on returns of Oil and Gas stocks during June 22 to July 27 of 2016 with no injected anomalies. The predicted anomalies would be false positives because the S&P 500 data is anomaly-free as we explained in Chapter 4.5. Then, using the proposed method we measured the number of false positives that are filtered.

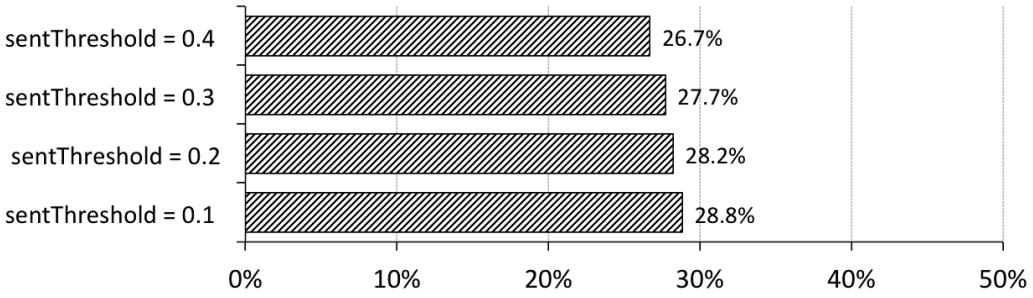


Figure 5.10: Filtering irrelevant anomalies using sentiment analysis on Oil and Gas industry sector

CAD predicts 261 data points as anomalous given stock market data for the case study (out of 1,092 data points). We used the proposed big data technique to determine how many of the 261 false positives could be filtered. Figure 5.10 shows experimental results on using the proposed method in the case study. *sentThreshold* is a parameter we use when comparing the aggregated polarity for a given stock per day as we explained section 5.2.3. Our experiments confirm that the proposed method is effective in improving CAD by filtering 28% of false positives.

Chapter 6

Future Work

This thesis can be extended in multiple ways and using various data mining techniques. We have identified three directions for future works on the thesis including exploring other stock time series features to improve anomaly detection, improving false positive filtering and extended experimental work.

1. The proposed anomaly detection can be further investigated by including other features in stock time series such as different ratios (e.g. price to book ratio, price to earning ratio, etc.) and stock volume (the number of stocks that are bought and sold at each time t).

2. The proposed big data method for reducing false positives in anomaly detection can be improved through:

    - improving sentiment classification by using more training data for sentiment analysis models. this would potentially introduce new features (i.e. words) that rank high in the feature selection and eventually improve the classification models. The principal idea is improving sentiment analysis in reflecting expected behaviour through tweets about stocks.
    - improving aggregated polarity of messages through application of Social Network Analysis (SNA). In this thesis tweets are considered

to be uniform meaning there is no weighting associated with a given tweet. SNA can be utilized to assign different weights to tweets. Social Network refers to the network of entities and patterns and their relations. More formally, a social network is defined as a set of actors that are connected through one or more type of relations [161]. Social Network Analysis is the study of this structure and relationships to provide insights about the underlying characteristics of the network (see Section 2.3 for more information about SNA). Twitter can be described as a network where nodes represent users and the edges are relationship of the users. SNA methods provide different tools to assign weights to the users, thus their tweets, based on the network structure. The proposed big data technique in this thesis can be extended using SNA to determine weight of each tweet based on the position and impact of its poster in the network.

3. The experiments in this thesis, although extensive, can be extended through:

    - running CAD on stock time series with lower granularity (e.g. hourly rate). It should be noted this may impose the risk of increasing noise substantially as volatility of stocks generally increase in a lower granularity (e.g. going from daily prices to hourly prices).
    - running the proposed big data technique on a larger set of stocks.
    - trying other classifiers in addition to the 6 classifiers that are used in this thesis for sentiment analysis.

Chapter 7

Conclusion

In this thesis we studied local anomaly detection for complex time series that are non-parametric, meaning it is difficult to fit a polynomial or deterministic function to the time series data. This is particularly a significant problem in fraud detection in the stock market as the time series are complex. Market manipulation periods have been shown to be associated with anomalies in the time series of assets [156] [209], yet the development of effective methods to detect such anomalies remains a challenging problem.

We proposed a Contextual Anomaly Detection (CAD) method for complex time series that is applicable to identifying stock market manipulation. The method considers not only the context of a time series in a time window but also the context of similar time series in a group of similar time series. First, a subset of time series is selected based on the window size parameter (we call this step chunking), Second, a centroid is calculated representing the expected behaviour of time series of the group within the window. The centroid values are used along with correlation of each time series X_i with the centroid to predict the value of the time series at time t (i.e. $\widehat{x_{it}}$). We studied different aggregate functions for determining the centroid time series including mean, median, mode and maximum probability. We designed and implemented a comprehensive set of experiments to evaluate CAD on 5 different sectors of

S&P 500 with daily and weekly frequencies including 636 time series over a period of 40 years. The results indicate that the proposed method improves recall from 7% to 33% compared to the comparable linear methods kNN and random walk without compromising precision.

Although CAD identifies many anomalies (i.e. relatively high recall), it flags false positives (i.e. low precision). Specifically in the stock market domain, this means that regulators would have to sift through the true and false positives. We developed a novel and formal method to improve time series anomaly detection using big data techniques. We utilized sentiment analysis on Twitter to filter out false positives in CAD. First, we extract tweets with respect to time series (i.e. extracting relevant tweets using Twitter Search API). Second, we preprocess tweets' texts to remove irrelevant text and extract features. Third, the sentiment of each tweet is determined using a classifier and the tweets' sentiments for each time series are aggregated per day. Finally, this additional information is used as a measure to confirm or reject detected outliers using CAD. For any given detected outlier at time t, we examine the stock sentiment at $t-1$. A stock sentiment that is in the same direction with the stock return at time t (e.g. positive sentiment before an increase in the return) implies that the detected data point is in fact not an anomaly because the market expected the change in that direction.

We developed a case study on Oil and Gas sector of S&P 500 to explore the proposed method for filtering irrelevant anomalies. We collected tweets about all of the 44 stocks in the sector for a 6-week period and used the proposed method to filter out false positives that CAD predicts during this period. Furthermore, we studied several hypotheses through these experiments including: i) efficacy of training data in the domain context in improving classifiers, ii) impact of feature selection in sentiment analysis models, and, iii) competence of different classifiers. Our studies confirm that training classifiers using stocks

tweets considerably improves sentiment analysis models compared to using the standard dataset for sentiment analysis, the movie reviews dataset. We also developed tools to automatically generate labelled data from StockTwits, a popular social media platform that is designed for investors and traders to share ideas. The results show that feature selection improves the performance of sentiment analysis regardless of the classification algorithm. Naive Bayes and SVM in most experiments outperformed other classifiers in our studies. Our experiments confirm that the proposed method is effective in improving CAD through removing irrelevant anomalies by correctly identifying 28% of false positives.

Bibliography

[1] Bovas Abraham and George EP Box. Bayesian analysis of some outlier problems in time series. *Biometrika*, 66(2):229–236, 1979.

[2] Bovas Abraham and Alice Chuang. Outlier detection and time series modeling. *Technometrics*, 31(2):241–248, 1989.

[3] Deepak Agarwal. Detecting anomalies in cross-classified streams: a bayesian approach. *Knowledge and information systems*, 11(1):29–44, 2007.

[4] Amrudin Agovic, Arindam Banerjee, Auroop R Ganguly, and Vladimir Protopopescu. Anomaly detection in transportation corridors using manifold embedding. *Knowledge Discovery from Sensor Data*, pages 81–105, 2008.

[5] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.

[6] Tarem Ahmed, Mark Coates, and Anukool Lakhina. Multivariate online anomaly detection using kernel recursive least squares. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 625–633. IEEE, 2007.

[7] R Aktas and M Doganay. Stock-price manipulation in the istanbul stock exchange. *Eurasian Review of Economics and Finance*, 2(1):21–8, 2006.

[8] E. Aleskerov, B. Freisleben, and B. Rao. *CARDWATCH: a neural network based database mining system for credit card fraud detection*, pages 220–226. IEEE, 1997.

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

[10] Shin Ando. Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 13–22. IEEE, 2007.

[11] Frank J Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, 1960.

[12] Dolan Antenucci, Michael Cafarella, Margaret Levenstein, Christopher Ré, and Matthew D Shapiro. Using social media to measure labor market flows. Technical report, National Bureau of Economic Research, 2014.

[13] Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan. A linear method for deviation detection in large databases. In *KDD*, pages 164–169, 1996.

[14] Sitaram Asur and Bernardo A Huberman. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 1, pages 492–499. IEEE, 2010.

[15] V Bamnett and T Lewis. Outliers in statistical data. 1994.

[16] Vic Barnett. The ordering of multivariate data. *Journal of the Royal Statistical Society. Series A (General)*, pages 318–355, 1976.

[17] Eli Bartov, Lucile Faurel, and Partha S Mohanram. Can twitter help predict firm-level earnings and stock returns? *Available at SSRN 2782236*, 2016.

[18] Roberto Basili, Alessandro Moschitti, and Maria Teresa Pazienza. Language sensitive text classification. In *Content-Based Multimedia Information Access-Volume 1*, pages 331–343. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 2000.

[19] Stephen D Bay and Mark Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 29–38. ACM, 2003.

[20] L Bing. Sentiment analysis: A fascinating problem. *Morgan and Claypool Publishers*, pages 7–143, 2012.

[21] C Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn, 2007.

[22] Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[23] Michael Blume, Christof Weinhardt, and Detlef Seese. Using network analysis for fraud detection in electronic markets. *Information Management and Market Engineering*, 4:101–112, 2006.

[24] Johan Bollen, Huina Mao, and Alberto Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. *ICWSM*, 11:450–453, 2011.

[25] Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[26] Richard J Bolton, David J Hand, et al. Unsupervised profiling methods for fraud detection. *Credit Scoring and Credit Control VII*, pages 235–255, 2001.

[27] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[28] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[29] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[30] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Optics-of: Identifying local outliers. In *Principles of data mining and knowledge discovery*, pages 262–270. Springer, 1999.

[31] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[32] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, June 1998.

[33] Simon Byers and Adrian E Raftery. Nearest-neighbor clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, 93(442):577–584, 1998.

[34] Fatih Camci and Ratna Babu Chinnam. General support vector representation machine for one-class classification of non-stationary classes. *Pattern Recognition*, 41(10):3021–3034, 2008.

[35] Rich Caruana and Alexandru Niculescu-Mizil. *An empirical comparison of supervised learning algorithms*, pages 161–168. ACM Press, June 2006.

[36] Jonnathan Carvalho, Adriana Prado, and Alexandre Plastino. A statistical and evolutionary approach to sentiment analysis. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02*, pages 110–117. IEEE Computer Society, 2014.

[37] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata, and A. Pulvirenti. *Similarity Measures and Dimensionality Reduction Techniques for Time Series Data Mining*. InTech, September 2012.

[38] Matthew V. Mahoney Chan. and Philip K. *Trajectory boundary modeling of time series for anomaly detection*. 2005.

[39] P.K. Chan and M.V. Mahoney. Modeling multiple time series for anomaly detection. *Fifth IEEE International Conference on Data Mining (ICDM05)*, pages 90–97, 2005.

[40] V. Chandola, a. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, May 2012.

[41] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[42] Varun Chandola, Deepthi Cheboli, and Vipin Kumar. Detecting anomalies in a time series database. *Department of Computer Science and Engineering, University of Minnesota, Technical Report*, pages 1–12, 2009.

[43] Chris Chatfield. *The Analysis of Time Series: An Introduction*. Chapman and Hall/CRC; 6 edition, 2003.

[44] Nitesh V Chawla, Aleksandar Lazarevic, Lawrence O Hall, and Kevin W Bowyer. Smoteboost: Improving prediction of the minority class in boosting. In *Knowledge Discovery in Databases: PKDD 2003*, pages 107–119. Springer, 2003.

[45] Sanjay Chawla and Pei Sun. Slom: a new measure for local spatial outliers. *Knowledge and Information Systems*, 9(4):412–429, 2006.

[46] Anny Lai-mei Chiu and Ada Wai-chee Fu. Enhancements on local outlier detection. In *Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International*, pages 298–307. IEEE, 2003.

[47] William W Cohen. Fast effective rule induction. In *Proceedings of the twelfth international conference on machine learning*, pages 115–123, 1995.

[48] Carole Comerton-Forde and Tālis J Putniņš. Measuring closing price manipulation. *Journal of Financial Intermediation*, 20(2):135–158, 2011.

[49] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[50] Sanjiv Das and Mike Chen. Yahoo! for amazon: Extracting market sentiment from stock message boards. In *Proceedings of the Asia Pacific finance association annual conference (APFA)*, volume 35, page 43. Bangkok, Thailand, 2001.

[51] Hal Daumé III. Notes on cg and lm-bfgs optimization of logistic regression. *Paper available at https://www.umiacs.umd.edu/ hal/docs/daume04cg-bfgs.pdf*, pages 1–7, 2004.

[52] Dorothy E Denning. An intrusion-detection model. *Software Engineering, IEEE Transactions on*, (2):222–232, 1987.

[53] MJ Desforges, PJ Jacob, and JE Cooper. Applications of probability density estimation to the detection of abnormal conditions in engineering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 212(8):687–703, 1998.

[54] David Diaz, Babis Theodoulidis, and Pedro Sampaio. Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices. *Expert Systems with Applications*, 38(10):12757–12771, September 2011.

[55] Martin Dillon. Introduction to modern information retrieval: G. salton and m. mcgill, 1983.

[56] Xiaowen Ding, Bing Liu, and Philip S Yu. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240. ACM, 2008.

[57] Pedro Henriques dos Santos Teixeira and Ruy Luiz Milidiú. Data stream anomaly detection through principal subspace tracking. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1609–1616. ACM, 2010.

[58] Karl B Dyer and Robi Polikar. Semi-supervised learning in initially labeled non-stationary environments with gradual drift. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–9. IEEE, 2012.

[59] FY Edgeworth. Xli. on discordant observations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 23(143):364–375, 1887.

[60] Manzoor Elahi, Kun Li, Wasif Nisar, Xinjie Lv, and Hongan Wang. Efficient clustering-based outlier detection algorithm for dynamic data stream. In *Fuzzy Systems and Knowledge Discovery, 2008. FSKD'08. Fifth International Conference on*, volume 5, pages 298–304. IEEE, 2008.

[61] Ryan Elwell and Robi Polikar. Incremental learning of concept drift in nonstationary environments. *Neural Networks, IEEE Transactions on*, 22(10):1517–1531, 2011.

[62] David Enke and Suraphan Thawornwong. The use of data mining and neural networks for forecasting stock market returns. *Expert Systems with Applications*, 29(4):927–940, November 2005.

[63] Levent Ertoz, Eric Eilertson, Aleksandar Lazarevic, Pang-Ning Tan, Vipin Kumar, Jaideep Srivastava, and Paul Dokas. Minds-minnesota intrusion detection system. *Next generation data mining*, pages 199–218, 2004.

[64] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding topics in collections of documents: A shared nearest neighbor approach. In *Clustering and Information Retrieval*, pages 83–103. Springer, 2004.

[65] Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*, pages 255–262. Morgan Kaufmann, 2000.

[66] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pages 77–101. Springer, 2002.

[67] Eleazar Eskin, Wenke Lee, and Salvatore J Stolfo. Modeling system calls for intrusion detection with dynamic window sizes. In *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, volume 1, pages 165–175. IEEE, 2001.

[68] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[69] Angiulli Fabrizio and Pizzuti Clara. Fast outlier detection in high dimensional spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 15–26, 2012.

[70] Angiulli Fabrizio, Rachel Ben-eiiyahu Zohary, and Loc Feo. *Outlier Detection Using Default Logic*, pages 833–838. 2003.

[71] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. *ACM SIGMOD Record*, 23(2):419–429, June 1994.

[72] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan. Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, 6(5):507–527, April 2004.

[73] Wei Fan, Matthew Miller, Sal Stolfo, Wenke Lee, and Phil Chan. Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, 6(5):507–527, 2004.

[74] Sun Fang and Wei Zijie. Rolling bearing fault diagnosis based on wavelet packet and RBF neural network. In *Control Conference, 2007. CCC 2007. Chinese*, pages 451–455. IEEE, 2007.

[75] T Fawcett and F Provost. *Activity monitoring: Noticing interesting changes in behavior*, pages 53–62. 1999.

[76] Tom Fawcett and Foster Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 53–62. ACM, 1999.

[77] Ronen Feldman, Benjamin Rosenfeld, Roy Bar-Haim, and Moshe Fresko. The stock sonarsentiment analysis of stocks based on a hybrid approach. In *Twenty-Third IAAI Conference*, pages 1642–1647, 2011.

[78] Zakia Ferdousi and Akira Maeda. *Unsupervised Outlier Detection in Time Series Data*, page 121. IEEE, 2006.

[79] E Fersini, E Messina, and FA Pozzi. Expressive signals in social media languages to improve polarity detection. *Information Processing & Management*, 52(1):20–35, 2016.

[80] Anthony J Fox. Outliers in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 350–363, 1972.

[81] Pedro Galeano, Daniel Peña, and Ruey S Tsay. Outlier detection in multivariate time series by projection pursuit. *Journal of the American Statistical Association*, 101(474):654–669, 2006.

[82] Lise Getoor and Christopher P Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005.

[83] Rafael Giusti and Gustavo E.A.P.A. Batista. *An Empirical Comparison of Dissimilarity Measures for Time Series Classification*, pages 82–88. IEEE, October 2013.

[84] Ole Gjolberg and Berth-Arne Bengtsson. Forecasting quarterly hog prices: Simple autoregressive models vs. naive predictions. *Agribusiness*, 13(6):673–679, November 1997.

[85] Koosha Golmohammadi and Osmar R Zaiane. Data mining applications for fraud detection in securities market. In *Intelligence and Security Informatics Conference (EISIC), 2012 European*, pages 107–114. IEEE, 2012.

[86] Koosha Golmohammadi and Osmar R Zaiane. Time series contextual anomaly detection for detecting market manipulation in stock market. In *The 2015 Data Science and Advanced Analytics (DSAA'2015)*, pages 1–10. IEEE, 2015.

[87] Koosha Golmohammadi, Osmar R Zaiane, and David Diaz. Detecting stock market manipulation using supervised learning algorithms. In *The 2014 International Conference on Data Science and Advanced Analytics (DSAA'2014)*, pages 435–441. IEEE, 2014.

[88] Mark Graham, Scott A Hale, and Devin Gaffney. Where in the world are you? geolocation and language identification in twitter. *The Professional Geographer*, 66(4):568–578, 2014.

[89] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE, 1999.

[90] Ville Hautamäki, Ismo Kärkkäinen, and Pasi Fränti. Outlier detection using k-nearest neighbour graph. In *ICPR (3)*, pages 430–433, 2004.

[91] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[92] Zengyou He, Shengchun Deng, and Xiaofei Xu. An optimization model for outlier detection in categorical data. In *Advances in Intelligent Computing*, pages 400–409. Springer, 2005.

[93] Zengyou He, Shengchun Deng, Xiaofei Xu, and Joshua Zhexue Huang. A fast greedy algorithm for outlier mining. In *Advances in Knowledge Discovery and Data Mining*, pages 567–576. Springer, 2006.

[94] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9):1641–1650, 2003.

[95] Guy G Helmer, Johnny SK Wong, Vasant Honavar, and Les Miller. Intelligent agents for intrusion detection. In *Information Technology Conference, 1998. IEEE*, pages 121–124. IEEE, 1998.

[96] Heiko Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[97] Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska de Jong, and Uzay Kaymak. Exploiting emoticons in sentiment analysis. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 703–710. ACM, 2013.

[98] J Hong, I Mozetic, and RS Michalski. Aq15: Incremental learning of attribute-based descriptions from examples, the method and users guide. report isg 86-5. Technical report, UIUCDCS-F-86-949, Dept. of Computer Science, University of Illinois, Urbana, 1986.

[99] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. *A practical guide to support vector classification*, pages 1–16. 2010.

[100] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 537–546. ACM, 2013.

[101] Mao Lin Huang, Jie Liang, and Quang Vinh Nguyen. A visualization approach for frauds detection in financial market. *2009 13th International Conference Information Visualisation*, pages 197–202, July 2009.

[102] Wei Huang, Yoshiteru Nakamori, and Shou-Yang Wang. Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10):2513–2522, 2005.

[103] Peter J Huber. *Robust statistics*. Springer, 2011.

[104] Paul S Jacobs. Joining statistics with nlp for text categorization. In *Proceedings of the third conference on Applied natural language processing*, pages 178–185. Association for Computational Linguistics, 1992.

[105] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[106] Nathalie Japkowicz. *Concept-learning in the absence of counter-examples: an autoassociation-based approach to classification*. PhD thesis, Rutgers, The State University of New Jersey, 1999.

[107] Nathalie Japkowicz. Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning*, 42(1-2):97–122, 2001.

[108] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[109] Veselina Jecheva. *About Some Applications of Hidden Markov Model in Intrusion Detection Systems*. 2006.

[110] Wen Jin, Anthony KH Tung, and Jiawei Han. Mining top-n local outliers in large databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–298. ACM, 2001.

[111] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[112] PK Kankar, Satish C Sharma, and SP Harsha. Fault diagnosis of ball bearings using machine learning methods. *Expert Systems with Applications*, 38(3):1876–1886, 2011.

[113] E. Keogh, J. Lin, and A. Fu. *HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence*, pages 226–233. Ieee, 2005.

[114] E Keogh, J Lin, SH Lee, and H Van Herle. Finding the most unusual time series subsequence: algorithms and applications. *Knowledge and Information Systems*, 11(1):1–27, 2007.

[115] Eamonn Keogh and Shruti Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Mining and Knowledge Discovery*, 7(4):349–371, October 2003.

[116] Eamonn Keogh, Stefano Lonardi, and Chotirat Ann Ratanamahatana. Towards parameter-free data mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 206–215. ACM, 2004.

[117] Gary King. Ensuring the data-rich future of the social sciences. *science*, 331(6018):719–721, 2011.

[118] J. Dale Kirkland, Ted E. Senator, James J. Hayden, Tomasz Dybala, Henry G. Goldberg, and Ping Shyr. The nasd regulation advanced-detection system (ads). *AI Magazine*, 20(1):55, March 1999.

[119] Genshiro Kitagawa. On the use of aic for the detection of outliers. *Technometrics*, 21(2):193–199, 1979.

[120] Florian Knorn and Douglas J. Leith. *Adaptive Kalman Filtering for anomaly detection in software appliances*, pages 1–6. IEEE, April 2008.

[121] Edwin M Knorr and Raymond T Ng. A unified approach for mining outliers. In *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*, page 11. IBM Press, 1997.

[122] Edwin M Knorr and Raymond T Ng. Finding intensional knowledge of distance-based outliers. In *VLDB*, volume 99, pages 211–222, 1999.

[123] Edwin M Knorr, Raymond T Ng, and Vladimir Tucakov. Distance-based outliers: algorithms and applications. *The VLDB JournalThe International Journal on Very Large Data Bases*, 8(3-4):237–253, 2000.

[124] Edwin M Knox and Raymond T Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the International Conference on Very Large Data Bases*, pages 392–403. Citeseer, 1998.

[125] Levente Kocsis and Andras Gyorgy. *Fraud Detection by Generating Positive Samples for Classification from Unlabeled Data*. 2010.

[126] Yufeng Kou, Chang-Tien Lu, and Dechang Chen. Spatial weighted outlier detection. In *SDM*, pages 614–618. SIAM, 2006.

[127] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11:538–541, 2011.

[128] Mark A Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, 1991.

[129] Ohbyung Kwon, Namyeon Lee, and Bongsik Shin. Data quality management, data usage experience and acquisition intention of big data analytics. *International Journal of Information Management*, 34(3):387–394, 2014.

[130] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D Kolaczyk, and Nina Taft. Structural analysis of network traffic flows. In *ACM SIGMETRICS Performance evaluation review*, volume 32, pages 61–72. ACM, 2004.

[131] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining*, pages 37–44, 2000.

[132] W Lee and D Xiang. *Information-theoretic measures for anomaly detection*, pages 130–143. 2001.

[133] Wenke Lee, Salvatore J Stolfo, and Philip K Chan. Learning patterns from unix process execution traces for intrusion detection. In *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, pages 50–56, 1997.

[134] Wenke Lee and Dong Xiang. Information-theoretic measures for anomaly detection. In *Security and Privacy, 2001. S&P 2001. Proceedings. 2001 IEEE Symposium on*, pages 130–143. IEEE, 2001.

[135] Ming Li and Paul Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer Science & Business Media, 2013.

[136] Xiaolei Li and Jiawei Han. Mining approximate top-k subspace anomalies in multi-dimensional time-series data. In *Proceedings of the 33rd international conference on Very large data bases*, pages 447–458. VLDB Endowment, 2007.

[137] Yihua Liao and V Rao Vemuri. Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security*, 21(5):439–448, 2002.

[138] J. Lin, E. Keogh, A. Fu, and H. Herle. *Approximations to Magic: Finding Unusual Medical Time Series*, pages 329–334. IEEE, 2005.

[139] Jessica Lin, Eamonn Keogh, Ada Fu, and Helga Van Herle. Approximations to magic: Finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 329–334. IEEE, 2005.

[140] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, April 2007.

[141] Song Lin and Donald E Brown. An outlier-based data association method for linking criminal incidents. *Decision Support Systems*, 41(3):604–615, 2006.

[142] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[143] Kun-Lin Liu, Wu-Jun Li, and Minyi Guo. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI*, pages 1678–1684, 2012.

[144] Zheng Liu, Jeffrey Xu Yu, and Lei Chen. *Detection of Shape Anomalies: A Probabilistic Approach Using Hidden Markov Models*, pages 1325–1327. IEEE, April 2008.

[145] Thomas Lotze, Galit Shmueli, Sean Murphy, and Howard Burkom. A wavelet-based anomaly detector for early detection of disease outbreaks. *Workshop on Machine Learning Algorithms for Surveillance and Event Detection, 23rd Intl Conference on Machine Learning*, 2006.

[146] J. Ma and S. Perkins. *Time-series novelty detection using one-class support vector machines*, volume 3, pages 1741–1745. IEEE, 2003.

[147] Junshui Ma and Simon Perkins. Online novelty detection on temporal sequences. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–618. ACM, 2003.

[148] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[149] M. V. Mahoney and P. K. Chan. Learning rules for anomaly detection of hostile network traffic. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 601–604, Nov 2003.

[150] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schutze. Introduction to information retrieval cambridge university press, 2008. *Ch*, 20:405–416.

[151] Huina Mao, Scott Counts, and Johan Bollen. Predicting financial markets: Comparing survey, news, twitter and search engine data. *arXiv preprint arXiv:1112.1051*, 2011.

[152] Yuexin Mao, Wei Wei, Bing Wang, and Benyuan Liu. Correlating s&p 500 stocks with twitter data. In *Proceedings of the first ACM international workshop on hot topics on interdisciplinary social networks research*, pages 69–72. ACM, 2012.

[153] David Martínez-Rego, Oscar Fontenla-Romero, and Amparo Alonso-Betanzos. Power wind mill fault detection via one-class ν-SVM vibration signal analysis. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 511–518. IEEE, 2011.

[154] Li Meng, Wang Miao, and Wang Chunguang. Research on SVM classification performance in rolling bearing diagnosis. In *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, volume 3, pages 132–135. IEEE, 2010.

[155] C.C. Michael and A. Ghosh. *Two state-based approaches to program-based anomaly detection*, pages 21–30. IEEE Comput. Soc, 2000.

[156] Marcello Minenna. The detection of market abuse on financial markets: a quantitative approach. *Quaderni di finanza*, (54):1–53, 2003.

[157] Tom M. Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, March 1982.

[158] H Zare Moayedi and MA Masnadi-Shirazi. Arima model for network traffic prediction and anomaly detection. In *Information Technology, 2008. ITSim 2008. International Symposium on*, volume 4, pages 1–6. IEEE, 2008.

[159] David Moore. *Introduction to the Practice of Statistics*. 2004.

[160] Satoshi Morinaga, Kenji Yamanishi, Kenji Tateishi, and Toshikazu Fukushima. Mining product reputations on the web. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 341–349. ACM, 2002.

[161] Dip Nandi, Margaret Hamilton, and James Harland. Evaluating the quality of interaction in asynchronous discussion forums in fully online courses. *Distance Education*, 33(1):5–30, 2012.

[162] Vu Dung Nguyen, Blesson Varghese, and Adam Barker. The royal birth of 2013: Analysing and visualising public sentiment in the uk using twitter. In *Big Data, 2013 IEEE International Conference on*, pages 46–54. IEEE, 2013.

[163] Caleb C Noble and Diane J Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2003.

[164] Hulisi Öğüt, M Mete Doğanay, and Ramazan Aktaş. Detecting stock-price manipulation in an emerging market: The case of Turkey. *Expert Systems with Applications*, 36(9):11944–11949, 2009.

[165] Miho Ohsaki, Shinya Kitaguchi, Hideto Yokoi, and Takahira Yamaguchi. Investigation of rule interestingness in medical data mining. In *Active Mining*, pages 174–189. Springer, 2005.

[166] M. Ohshima. Peculiarity oriented multidatabase mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):952–960, July 2003.

[167] M Otey, Srinivasan Parthasarathy, Amol Ghoting, G Li, Sundeep Narravula, and D Panda. Towards nic-based intrusion detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 723–728. ACM, 2003.

[168] Matthew Eric Otey, Amol Ghoting, and Srinivasan Parthasarathy. Fast distributed outlier detection in mixed-attribute data sets. *Data Mining and Knowledge Discovery*, 12(2-3):203–228, 2006.

[169] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[170] Girish Keshav Palshikar. Distance-based outliers in sequences. In *Distributed Computing and Internet Technology*, pages 547–552. Springer, 2005.

[171] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[172] Spiros Papadimitriou, Hiroyuki Kitagawa, Philip B Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326. IEEE, 2003.

[173] A Papoulis and SU Pillai. *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[174] Lucas Parra, Gustavo Deco, and Stefan Miesbach. Statistical independence and novelty detection with information preserving nonlinear maps. *Neural Computation*, 8(2):260–269, 1996.

[175] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[176] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*, pages 1–14, 2005.

[177] Brandon Pincombe. Anomaly detection in time series of graphs using ARMA processes. 24(4):2–10, 2005.

[178] A Pires and Carla Santos-Pereira. Using clustering and robust estimators to detect outliers in multivariate data. In *Proceedings of the International Conference on Robust Statistics*, 2005.

[179] Federico Alberto Pozzi, Elisabetta Fersini, Enza Messina, and Daniele Blanc. Enhance polarity classification on social media through sentiment-based feature expansion. *WOA@ AI* IA*, 1099:78–84, 2013.

[180] Federico Alberto Pozzi, Daniele Maccagnola, Elisabetta Fersini, and Enza Messina. Enhance user-level sentiment analysis on microblogs with approval relations. In *Congress of the Italian Association for Artificial Intelligence*, pages 133–144. Springer, 2013.

[181] P. Protopapas, J. M. Giammarco, L. Faccioli, M. F. Struble, R. Dave, and C. Alcock. Finding outlier light curves in catalogues of periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, 369(2):677–696, June 2006.

[182] P. Protopapas, J. M. Giammarco, L. Faccioli, M. F. Struble, R. Dave, and C. Alcock. Finding outlier light curves in catalogues of periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, 369(2):677–696, June 2006.

[183] Y. Qiao, X.W. Xin, Y. Bin, and S. Ge. Anomaly intrusion detection method based on HMM. *Electronics Letters*, 38(13):663–664, June 2002.

[184] JR Quinlan. *C4. 5: programs for machine learning*. 1993.

[185] L. Rabiner and B. Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[186] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM SIGMOD Record*, volume 29, pages 427–438. ACM, 2000.

[187] Umaa Rebbapragada, Pavlos Protopapas, Carla E. Brodley, and Charles Alcock. Finding anomalous periodic time series. *Machine Learning*, 74(3):281–313, December 2008.

[188] Stephen J Roberts. Novelty detection using extreme value statistics. In *Vision, Image and Signal Processing, IEE Proceedings-*, volume 146, pages 124–129. IET, 1999.

[189] Stephen J Roberts. Extreme value statistics for novelty detection in biomedical data processing. In *Science, Measurement and Technology, IEE Proceedings-*, volume 147, pages 363–367. IET, 2000.

[190] Volker Roth. Kernel fisher discriminants for outlier detection. *Neural computation*, 18(4):942–960, 2006.

[191] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[192] Eduardo J Ruiz, Vagelis Hristidis, Carlos Castillo, Aristides Gionis, and Alejandro Jaimes. Correlating financial time series with micro-blogging activity. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 513–522. ACM, 2012.

[193] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[194] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In *International Semantic Web Conference*, pages 508–524. Springer, 2012.

[195] Stan Salvador and Philip Chan. Learning states and rules for detecting anomalies in time series. *Applied Intelligence*, 23(3):241–255, December 2005.

[196] Stan Salvador, Philip Chan, and John Brodie. Learning states and rules for time series anomaly detection. In *FLAIRS Conference*, pages 306–311, 2004.

[197] Steven L. Salzberg. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240, September 1994.

[198] Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.

[199] Robert E Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine learning*, 39(2-3):135–168, 2000.

[200] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[201] Robert P Schumaker and Hsinchun Chen. Textual analysis of stock market prediction using breaking financial news: The azfin text system. *ACM Transactions on Information Systems (TOIS)*, 27(2):12, 2009.

[202] Steven Salzberg Scott Cost. A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78, 1993.

[203] Ted E. Senator. *Ongoing management and application of discovered knowledge in a large regulatory organization*, pages 44–53. ACM Press, August 2000.

[204] Songwon Seo. *A Review and Comparison of Methods for Detecting Outliers in Univariate Data Sets*. August 2006.

[205] Gholamhosein Sheikholeslami, Surojit Chatterjee, and Aidong Zhang. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In *VLDB*, volume 98, pages 428–439, 1998.

[206] Mei-ling Shyu, Shu-ching Chen, Kanoksri Sarinnapakorn, and Liwu Chang. A novel anomaly detection scheme based on principal component classifier. In *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM03*. Citeseer, 2003.

[207] Rasheda Smith, Alan Bivens, Mark Embrechts, Chandrika Palagiri, and Boleslaw Szymanski. Clustering approaches for anomaly based intrusion detection. *Proceedings of intelligent engineering systems through artificial neural networks*, pages 579–584, 2002.

[208] Panu Somervuo and Teuvo Kohonen. Self-organizing maps and learning vector quantization for feature sequences. *Neural Processing Letters*, 10(2):151–159, 1999.

[209] Yin Song, Longbing Cao, Xindong Wu, Gang Wei, Wu Ye, and Wei Ding. Coupled behavior analysis for capturing coupling relationships in group-based market manipulations. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 976–984. ACM, 2012.

[210] Clay Spence, Lucas Parra, and Paul Sajda. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. page 3, December 2001.

[211] Ashok Sriastava et al. Discovering system health anomalies using data mining techniques. pages 1–7, 2005.

[212] Pei Sun and Sanjay Chawla. On local spatial outliers. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 209–216. IEEE, 2004.

[213] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[214] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, et al. *Introduction to data mining*, volume 1. Pearson Addison Wesley Boston, 2006.

[215] PN Tan, V Kumar, and J Srivastava. Selecting the right interestingness measure for association patterns. *Proceedings of the eighth ACM SIGKDD* , 2002.

[216] Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. Fast anomaly detection for streaming data. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1511, 2011.

[217] Jian Tang, Zhixiang Chen, Ada Wai-Chee Fu, and David W Cheung. Enhancing effectiveness of outlier detections for low density patterns. In *Advances in Knowledge Discovery and Data Mining*, pages 535–548. Springer, 2002.

[218] Yufei Tao, Xiaokui Xiao, and Shuigeng Zhou. Mining distance-based outliers from large databases in any metric space. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 394–403. ACM, 2006.

[219] David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1):45–66, 2004.

[220] DM Tax. J. one-class classification: concept-learning in the absence of counter-examples. *Delft University of Technology*, 2001.

[221] Henry S Teng, Kaihu Chen, and Stephen C Lu. Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*, pages 278–284. IEEE, 1990.

[222] Richard M Tong. An operational system for detecting and tracking opinions in on-line discussion. In *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification*, volume 1, page 6, 2001.

[223] Philip HS Torr and David W Murray. Outlier detection and motion segmentation. In *Optical Tools for Manufacturing and Advanced Automation*, pages 432–443. International Society for Optics and Photonics, 1993.

[224] Ruey S Tsay, Daniel Peña, and Alan E Pankratz. Outliers in multivariate time series. *Biometrika*, 87(4):789–804, 2000.

[225] John Wilder Tukey. *Exploratory Data Analysis*. 1977.

[226] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[227] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[228] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[229] Alessandro Vespignani. Predicting the behavior of techno-social systems. *Science*, 325(5939):425–428, 2009.

[230] S. Viaene, R.A. Derrig, and G. Dedene. A case study of applying boosting naive bayes to claim fraud diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 16(5):612–620, May 2004.

[231] Michail Vlachos, Kun-Lung Wu, Shyh-Kwei Chen, and Philip S. Yu. Correlating burst events on streaming stock market data. *Data Mining and Knowledge Discovery*, 16(1):109–133, March 2007.

[232] Gang Wang, Jianshan Sun, Jian Ma, Kaiquan Xu, and Jibao Gu. Sentiment classification: The contribution of ensemble learning. *Decision support systems*, 57:77–93, 2014.

[233] Haixun Wang, Wei Fan, Philip S Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. ACM, 2003.

[234] Y Wang. Mining stock price using fuzzy rough set system. *Expert Systems with Applications*, 24(1):13–23, January 2003.

[235] Y Wang. Mining stock price using fuzzy rough set system. *Expert Systems with Applications*, 24(1):13–23, January 2003.

[236] Li Wei, Eamonn Keogh, and Xiaopeng Xi. Saxually explicit images: finding unusual shapes. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 711–720. IEEE, 2006.

[237] Li Wei, Nitin Kumar, Venkata Lolla, Eamonn J. Keogh, Stefano Lonardi, and Chotirat Ratanamahatana. Assumption-free anomaly detection in time series. pages 237–240, June 2005.

[238] Li Wei, Weining Qian, Aoying Zhou, Wen Jin, and X Yu Jeffrey. Hot: Hypergraph-based outlier test for categorical data. In *Advances in Knowledge Discovery and Data Mining*, pages 399–410. Springer, 2003.

[239] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, June 2009.

[240] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

[241] Janyce Wiebe. Learning subjective adjectives from corpora. In *AAAI/IAAI*, pages 735–740, 2000.

[242] Carl Edward Rasmussen Williams and Christopher K. I. *Gaussian Processes for Machine Learning*. Cambridge: MIT Press., 2006.

[243] Kenji Yamanishi, Jun-ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, May 2004.

[244] Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

[245] Dragomir Yankov, Eamonn Keogh, and Umaa Rebbapragada. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. *Knowledge and Information Systems*, 17(2):241–262, March 2008.

[246] Dantong Yu, Gholamhosein Sheikholeslami, and Aidong Zhang. Findout: finding outliers in very large datasets. *Knowledge and Information Systems*, 4(4):387–412, 2002.

[247] Yang Yu, Cheng Junsheng, et al. A roller bearing fault diagnosis method based on emd energy entropy and ann. *Journal of sound and vibration*, 294(1):269–277, 2006.

[248] Achim Zeileis, Torsten Hothorn, and Kurt Hornik. Model-based recursive partitioning. *Journal of Computational and Graphical Statistics*, 17(2):492–514, June 2008.

[249] Ji Zhang and Hai Wang. Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. *Knowledge and information systems*, 10(3):333–355, 2006.

[250] Jun Zhang, Fu Chiang Tsui, Michael M Wagner, and William R Hogan. *Detection of outbreaks from time series data using wavelet transform.*, pages 748–752. January 2003.

[251] Wenbin Zhang and Steven Skiena. Trading strategies to exploit blog and news sentiment. In *ICWSM*, 2010.

[252] Xiaoqiang Zhang, Pingzhi Fan, and Zhongliang Zhu. *A new anomaly detection method based on hierarchical HMM*, pages 249–252. IEEE, 2003.

[253] Jichang Zhao, Li Dong, Junjie Wu, and Ke Xu. Moodlens: an emoticon-based sentiment analysis system for chinese tweets. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1528–1531. ACM, 2012.

[254] Linhong Zhu, Aram Galstyan, James Cheng, and Kristina Lerman. Tripartite graph clustering for dynamic sentiment analysis on social media. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1531–1542. ACM, 2014.

Appendix A

Contextual Anomaly Detection results

Below are the performance results of the proposed Contextual Anomaly Detection (CAD) method, kNN and Random Walk in predicting anomalies on all datasets with daily frequency. The results indicate that CAD outperforms the recall of comparable methods from less than 7% to over 31% without compromising the precision.

Table A.1: Comparison of CAD performance results with kNN and Random Walk using weekly S&P 500 data (in percentage)

Win. size	**Dataset**	**Algorithm**	**Prec.**	**Rec.**	**F2**	**F4**
15	Consumer Staples	CAD-maxP	0.32	36.84	1.53	4.74
		CAD-mean	0.33	34.70	1.59	4.86
		CAD-median	0.31	32.70	1.47	4.52
		CAD-mode	0.35	31.39	1.70	5.11
		kNN	0.28	6.02	1.17	2.71
		RandomWalk	0.24	1.65	0.75	1.22
15	Consumer Dis.	CAD-maxP	0.35	37.99	1.68	5.17
		CAD-mean	0.33	34.49	1.58	4.83
		CAD-median	0.33	32.29	1.59	4.84
		CAD-mode	0.38	31.83	1.79	5.37
		kNN	0.29	6.26	1.24	2.86
		RandomWalk	0.25	1.72	0.79	1.28
15	Energy	CAD-maxP	0.27	33.85	1.32	4.10
		CAD-mean	0.33	34.70	1.59	4.86
		CAD-median	0.30	31.39	1.47	4.49
		CAD-mode	0.31	29.66	1.51	4.57

		kNN	0.29	6.36	1.23	2.86
		RandomWalk	0.34	2.39	1.09	1.77
15	IT	CAD-maxP	0.33	41.03	1.60	4.96
		CAD-mean	0.34	33.69	1.63	4.98
		CAD-median	0.34	32.95	1.61	4.91
		CAD-mode	0.37	32.28	1.75	5.27
		kNN	0.33	6.83	1.40	3.19
		RandomWalk	0.32	2.14	1.00	1.60
15	Financials	CAD-maxP	0.32	33.25	1.55	4.73
		CAD-mean	0.34	35.47	1.65	5.05
		CAD-median	0.36	33.94	1.74	5.27
		CAD-mode	0.34	31.02	1.63	4.92
		kNN	0.34	7.18	1.42	3.27
		RandomWalk	0.38	2.62	1.20	1.94
20	Consumer Staples	CAD-maxP	0.34	32.74	1.61	4.90
		CAD-mean	0.33	34.02	1.60	4.88
		CAD-median	0.32	33.31	1.53	4.69
		CAD-mode	0.35	32.10	1.68	5.08
		kNN	0.25	5.42	1.07	2.46
		RandomWalk	0.31	2.12	0.98	1.58
20	Consumer Dis.	CAD-maxP	0.34	36.04	1.65	5.05
		CAD-mean	0.32	34.16	1.53	4.69
		CAD-median	0.31	32.74	1.51	4.61
		CAD-mode	0.36	32.33	1.73	5.21
		kNN	0.31	6.94	1.31	3.06
		RandomWalk	0.35	2.53	1.12	1.85
20	Energy	CAD-maxP	0.27	34.74	1.31	4.08
		CAD-mean	0.31	32.30	1.48	4.53
		CAD-median	0.33	32.88	1.60	4.88
		CAD-mode	0.36	31.22	1.72	5.16
		kNN	0.31	6.77	1.30	3.03
		RandomWalk	0.33	2.29	1.04	1.69
20	IT	CAD-maxP	0.31	34.46	1.51	4.63
		CAD-mean	0.34	34.01	1.63	4.97
		CAD-median	0.34	31.87	1.65	5.00
		CAD-mode	0.36	29.90	1.73	5.18
		kNN	0.32	6.67	1.34	3.07
		RandomWalk	0.37	2.52	1.17	1.88
20	Financials	CAD-maxP	0.35	33.95	1.70	5.16
		CAD-mean	0.34	34.62	1.62	4.95
		CAD-median	0.37	34.71	1.77	5.37
		CAD-mode	0.36	30.52	1.70	5.11
		kNN	0.31	6.61	1.30	3.00
		RandomWalk	0.28	1.96	0.89	1.45

24	Consumer Staples	CAD-maxP	0.36	33.91	1.72	5.21
		CAD-mean	0.36	35.01	1.72	5.23
		CAD-median	0.32	33.70	1.52	4.67
		CAD-mode	0.39	32.35	1.84	5.51
		kNN	0.31	6.42	1.31	2.99
		RandomWalk	0.30	1.93	0.92	1.45
24	Consumer Dis.	CAD-maxP	0.39	35.78	1.87	5.65
		CAD-mean	0.36	34.19	1.71	5.18
		CAD-median	0.31	32.12	1.49	4.55
		CAD-mode	0.40	32.97	1.91	5.71
		kNN	0.34	6.84	1.41	3.21
		RandomWalk	0.40	2.63	1.25	1.99
24	Energy	CAD-maxP	0.40	43.36	1.93	5.94
		CAD-mean	0.36	35.77	1.72	5.23
		CAD-median	0.35	33.42	1.66	5.05
		CAD-mode	0.38	33.01	1.83	5.50
		kNN	0.35	7.35	1.47	3.37
		RandomWalk	0.33	2.19	1.03	1.65
24	IT	CAD-maxP	0.30	34.24	1.44	4.45
		CAD-mean	0.34	33.96	1.63	4.97
		CAD-median	0.33	31.42	1.60	4.84
		CAD-mode	0.31	28.29	1.50	4.51
		kNN	0.23	4.94	0.99	2.27
		RandomWalk	0.35	2.36	1.10	1.76
24	Financials	CAD-maxP	0.45	38.45	2.13	6.40
		CAD-mean	0.33	34.46	1.61	4.93
		CAD-median	0.34	33.64	1.64	5.00
		CAD-mode	0.33	28.74	1.58	4.74
		kNN	0.31	6.70	1.31	3.03
		RandomWalk	0.33	2.25	1.04	1.68
30	Consumer Staples	CAD-maxP	0.43	39.57	2.06	6.22
		CAD-mean	0.35	34.21	1.68	5.11
		CAD-median	0.34	34.53	1.63	4.97
		CAD-mode	0.36	30.69	1.72	5.14
		kNN	0.33	6.80	1.38	3.15
		RandomWalk	0.34	2.27	1.06	1.70
30	Consumer Dis.	CAD-maxP	0.36	36.87	1.73	5.29
		CAD-mean	0.35	33.54	1.70	5.15
		CAD-median	0.32	32.58	1.52	4.65
		CAD-mode	0.32	29.65	1.53	4.64
		kNN	0.32	6.43	1.34	3.03
		RandomWalk	0.33	2.09	1.01	1.59
30	Energy	CAD-maxP	0.31	37.44	1.52	4.70
		CAD-mean	0.35	34.57	1.70	5.18

		CAD-median	0.35	33.09	1.67	5.05
		CAD-mode	0.34	29.49	1.62	4.86
		kNN	0.35	7.20	1.47	3.35
		RandomWalk	0.29	1.94	0.91	1.45
30	IT	CAD-maxP	0.30	39.04	1.47	4.59
		CAD-mean	0.32	31.83	1.54	4.68
		CAD-median	0.32	32.94	1.55	4.74
		CAD-mode	0.34	28.92	1.63	4.89
		kNN	0.28	5.98	1.20	2.75
		RandomWalk	0.34	2.31	1.07	1.72
30	Financials	CAD-maxP	0.31	31.82	1.49	4.55
		CAD-mean	0.36	34.13	1.74	5.27
		CAD-median	0.35	32.85	1.69	5.10
		CAD-mode	0.42	32.56	2.01	5.95
		kNN	0.32	6.38	1.34	3.02
		RandomWalk	0.34	2.18	1.05	1.66
35	Consumer Staples	CAD-maxP	0.39	36.68	1.86	5.64
		CAD-mean	0.36	34.13	1.71	5.19
		CAD-median	0.34	35.22	1.65	5.05
		CAD-mode	0.36	31.24	1.73	5.18
		kNN	0.29	5.86	1.20	2.73
		RandomWalk	0.33	2.18	1.04	1.65
35	Consumer Dis.	CAD-maxP	0.30	33.62	1.47	4.53
		CAD-mean	0.39	35.29	1.85	5.60
		CAD-median	0.38	36.05	1.83	5.53
		CAD-mode	0.36	29.81	1.70	5.08
		kNN	0.35	6.85	1.46	3.28
		RandomWalk	0.37	2.28	1.12	1.75
35	Energy	CAD-maxP	0.32	40.93	1.56	4.87
		CAD-mean	0.32	32.55	1.55	4.73
		CAD-median	0.36	35.11	1.72	5.23
		CAD-mode	0.37	31.35	1.76	5.26
		kNN	0.32	6.97	1.36	3.16
		RandomWalk	0.27	1.85	0.85	1.38
35	IT	CAD-maxP	0.36	39.18	1.71	5.28
		CAD-mean	0.35	33.07	1.67	5.07
		CAD-median	0.36	33.64	1.74	5.26
		CAD-mode	0.35	30.96	1.68	5.05
		kNN	0.31	6.23	1.29	2.92
		RandomWalk	0.34	2.22	1.06	1.68
35	Financials	CAD-maxP	0.39	37.89	1.86	5.65
		CAD-mean	0.36	32.68	1.73	5.21
		CAD-median	0.36	32.50	1.74	5.25
		CAD-mode	0.40	32.05	1.91	5.68

	kNN	0.29	5.65	1.21	2.72
	RandomWalk	0.37	2.32	1.13	1.77

Table A.2: Comparison of CAD performance results with kNN and Random Walk using daily S&P 500 data (in percentage)

Win. size	**Dataset**	**Algorithm**	**Prec.**	**Rec.**	**F2**	**F4**
15	Consumer Staples	CAD-maxP	0.30	32.11	1.43	4.39
		CAD-mean	0.34	33.95	1.62	4.95
		CAD-median	0.34	34.06	1.63	4.97
		CAD-mode	0.37	30.66	1.74	5.22
		kNN	0.30	6.29	1.27	2.91
		RandomWalk	0.36	2.39	1.11	1.79
15	Consumer Dis.	CAD-maxP	0.32	33.49	1.52	4.66
		CAD-mean	0.34	34.91	1.65	5.03
		CAD-median	0.34	34.47	1.63	4.98
		CAD-mode	0.34	28.76	1.64	4.91
		kNN	0.31	6.48	1.29	2.97
		RandomWalk	0.28	1.92	0.89	1.43
15	Energy	CAD-maxP	0.30	35.56	1.45	4.50
		CAD-mean	0.32	32.42	1.54	4.70
		CAD-median	0.32	32.21	1.53	4.67
		CAD-mode	0.33	30.08	1.57	4.73
		kNN	0.29	6.17	1.24	2.84
		RandomWalk	0.40	2.73	1.27	2.04
15	IT	CAD-maxP	0.31	35.12	1.50	4.63
		CAD-mean	0.31	32.58	1.50	4.60
		CAD-median	0.32	32.98	1.52	4.66
		CAD-mode	0.33	30.37	1.58	4.76
		kNN	0.31	6.70	1.31	3.03
		RandomWalk	0.34	2.31	1.06	1.72
15	Financials	CAD-maxP	0.34	33.87	1.65	5.01
		CAD-mean	0.31	33.96	1.50	4.61
		CAD-median	0.31	33.73	1.49	4.58
		CAD-mode	0.36	31.31	1.74	5.23
		kNN	0.28	6.42	1.21	2.83
		RandomWalk	0.29	2.06	0.93	1.52
20	Consumer Staples	CAD-maxP	0.32	31.00	1.52	4.62
		CAD-mean	0.32	32.89	1.54	4.71
		CAD-median	0.32	33.11	1.55	4.75
		CAD-mode	0.34	31.98	1.65	4.99

		kNN	0.32	6.88	1.35	3.13
		RandomWalk	0.34	2.35	1.08	1.74
20	Consumer Dis.	CAD-maxP	0.31	30.90	1.47	4.48
		CAD-mean	0.35	35.69	1.68	5.14
		CAD-median	0.35	35.41	1.67	5.10
		CAD-mode	0.35	30.06	1.66	4.99
		kNN	0.30	6.28	1.24	2.86
		RandomWalk	0.29	2.00	0.93	1.49
20	Energy	CAD-maxP	0.33	36.60	1.60	4.93
		CAD-mean	0.36	34.02	1.73	5.25
		CAD-median	0.36	33.71	1.72	5.20
		CAD-mode	0.34	29.75	1.64	4.93
		kNN	0.31	6.00	1.27	2.86
		RandomWalk	0.43	2.72	1.32	2.07
20	IT	CAD-maxP	0.34	35.96	1.63	5.00
		CAD-mean	0.34	34.56	1.62	4.96
		CAD-median	0.34	34.39	1.62	4.94
		CAD-mode	0.32	29.60	1.55	4.67
		kNN	0.31	6.70	1.32	3.05
		RandomWalk	0.32	2.18	1.00	1.62
20	Financials	CAD-maxP	0.36	34.33	1.75	5.29
		CAD-mean	0.31	31.99	1.49	4.57
		CAD-median	0.31	31.76	1.49	4.54
		CAD-mode	0.36	29.99	1.71	5.13
		kNN	0.27	5.89	1.16	2.67
		RandomWalk	0.41	2.86	1.30	2.12
24	Consumer Staples	CAD-maxP	0.35	31.41	1.67	5.03
		CAD-mean	0.34	32.91	1.62	4.93
		CAD-median	0.34	32.91	1.63	4.94
		CAD-mode	0.34	31.90	1.63	4.95
		kNN	0.32	6.45	1.32	3.01
		RandomWalk	0.33	2.19	1.03	1.64
24	Consumer Dis.	CAD-maxP	0.30	30.75	1.43	4.36
		CAD-mean	0.33	32.48	1.58	4.80
		CAD-median	0.33	32.59	1.58	4.82
		CAD-mode	0.36	31.10	1.72	5.15
		kNN	0.27	5.64	1.14	2.61
		RandomWalk	0.32	2.17	1.02	1.63
24	Energy	CAD-maxP	0.33	36.88	1.60	4.94
		CAD-mean	0.37	35.15	1.79	5.43
		CAD-median	0.38	35.36	1.81	5.47
		CAD-mode	0.35	30.82	1.67	5.02
		kNN	0.31	6.18	1.30	2.94
		RandomWalk	0.36	2.32	1.11	1.76

24	IT	CAD-maxP	0.33	33.09	1.59	4.84
		CAD-mean	0.37	33.33	1.76	5.32
		CAD-median	0.37	33.28	1.76	5.32
		CAD-mode	0.36	31.06	1.71	5.14
		kNN	0.35	6.74	1.47	3.27
		RandomWalk	0.36	2.18	1.08	1.68
24	Financials	CAD-maxP	0.35	33.37	1.66	5.05
		CAD-mean	0.33	32.19	1.60	4.85
		CAD-median	0.33	32.03	1.59	4.84
		CAD-mode	0.34	31.46	1.64	4.96
		kNN	0.32	6.46	1.33	3.03
		RandomWalk	0.40	2.65	1.25	2.00
30	Consumer Staples	CAD-maxP	0.37	32.72	1.79	5.38
		CAD-mean	0.36	33.87	1.74	5.26
		CAD-median	0.36	33.81	1.74	5.26
		CAD-mode	0.36	31.21	1.70	5.11
		kNN	0.32	6.29	1.32	2.99
		RandomWalk	0.35	2.22	1.08	1.69
30	Consumer Dis.	CAD-maxP	0.34	32.15	1.62	4.91
		CAD-mean	0.33	30.96	1.58	4.78
		CAD-median	0.33	31.38	1.60	4.85
		CAD-mode	0.36	30.71	1.73	5.18
		kNN	0.31	6.18	1.29	2.93
		RandomWalk	0.28	1.77	0.85	1.34
30	Energy	CAD-maxP	0.34	35.97	1.62	4.96
		CAD-mean	0.32	32.08	1.54	4.70
		CAD-median	0.32	31.75	1.53	4.66
		CAD-mode	0.38	30.96	1.79	5.34
		kNN	0.30	6.28	1.25	2.88
		RandomWalk	0.34	2.32	1.08	1.73
30	IT	CAD-maxP	0.32	34.48	1.56	4.78
		CAD-mean	0.34	33.01	1.62	4.92
		CAD-median	0.34	33.17	1.63	4.95
		CAD-mode	0.36	31.00	1.73	5.19
		kNN	0.32	6.71	1.36	3.11
		RandomWalk	0.34	2.27	1.07	1.71
30	Financials	CAD-maxP	0.42	35.26	1.99	5.95
		CAD-mean	0.36	33.35	1.71	5.17
		CAD-median	0.36	33.20	1.70	5.16
		CAD-mode	0.36	32.24	1.73	5.22
		kNN	0.31	6.14	1.29	2.92
		RandomWalk	0.39	2.43	1.18	1.85
35	Consumer Staples	CAD-maxP	0.37	30.37	1.74	5.21
		CAD-mean	0.36	32.44	1.75	5.26

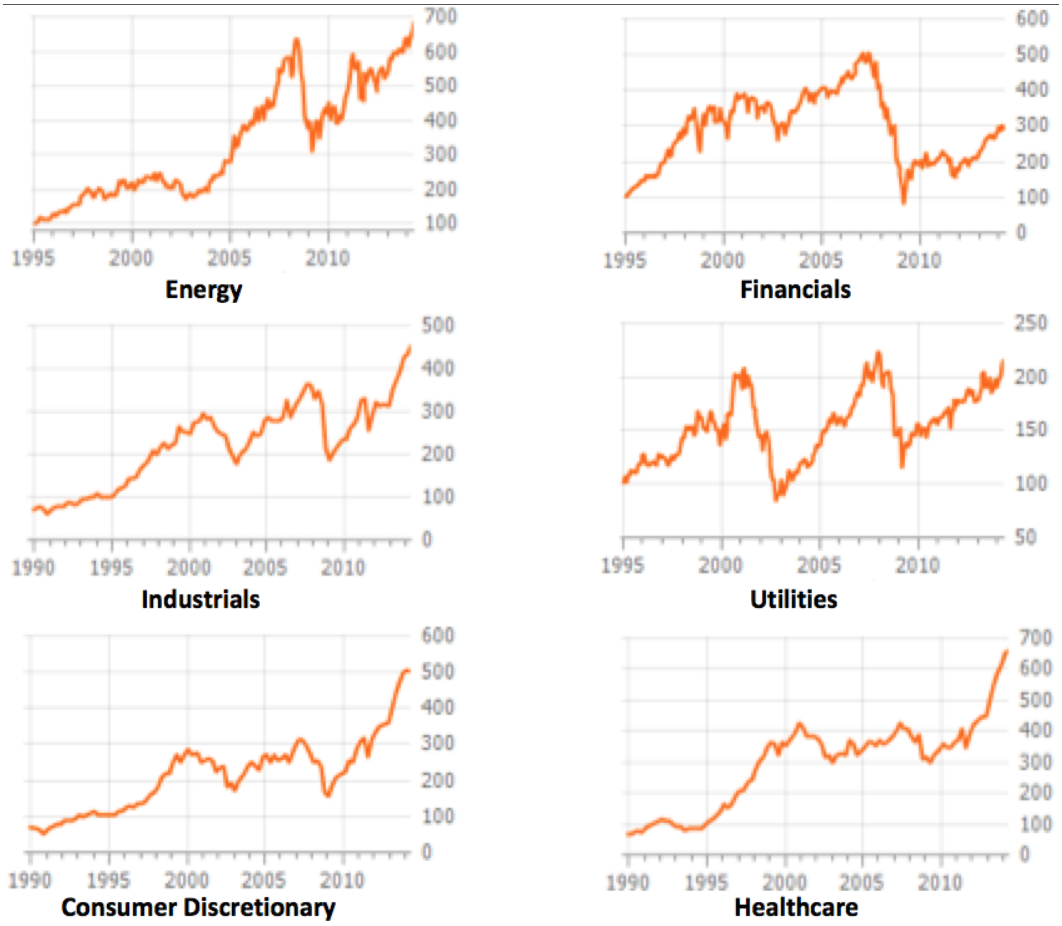
		CAD-median	0.37	32.59	1.76	5.29
		CAD-mode	0.34	30.75	1.65	4.97
		kNN	0.36	6.82	1.49	3.32
		RandomWalk	0.26	1.58	0.79	1.22
35	Consumer Dis.	CAD-maxP	0.33	32.66	1.60	4.87
		CAD-mean	0.34	33.82	1.62	4.93
		CAD-median	0.34	33.99	1.63	4.96
		CAD-mode	0.36	30.98	1.72	5.15
		kNN	0.27	5.71	1.12	2.59
		RandomWalk	0.29	1.96	0.91	1.46
35	Energy	CAD-maxP	0.37	37.86	1.78	5.44
		CAD-mean	0.35	33.46	1.68	5.10
		CAD-median	0.35	33.62	1.69	5.13
		CAD-mode	0.39	31.51	1.85	5.52
		kNN	0.32	6.44	1.32	3.01
		RandomWalk	0.28	1.81	0.86	1.36
35	IT	CAD-maxP	0.35	35.12	1.67	5.10
		CAD-mean	0.35	32.43	1.66	5.04
		CAD-median	0.35	32.80	1.69	5.10
		CAD-mode	0.35	30.25	1.69	5.06
		kNN	0.31	6.13	1.28	2.90
		RandomWalk	0.32	2.03	0.98	1.54
35	Financials	CAD-maxP	0.36	33.69	1.74	5.26
		CAD-mean	0.35	33.78	1.69	5.12
		CAD-median	0.35	33.73	1.69	5.12
		CAD-mode	0.32	29.01	1.51	4.57
		kNN	0.32	6.57	1.34	3.06
		RandomWalk	0.36	2.35	1.11	1.77

Appendix B

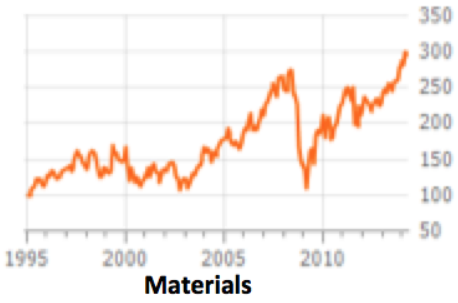
S&P industry Sector Trends

Table B.1 illustrates the price index of each industry sector in S&P during the past 20 years (the source of data is Thompson Reuters).

Table B.1: S&P 500 industry sector returns

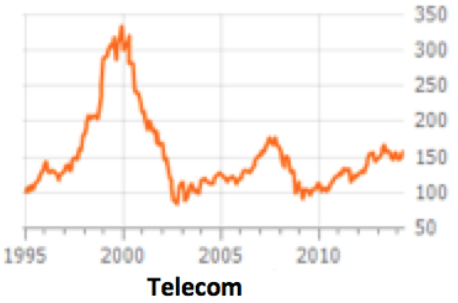


Information Technology



Materials

Consumer Staples



Telecom

Appendix C

Distance Measures

Let two time series $T = \{t_1, \ldots, t_n\}$, $S = \{s_1, \ldots, s_n\}$, the most widely used time series distance measures include [37]:

- **Lock-step Measure (one-to-one):**
  - Minkowski Distance: where p is called the order of Minkowski distance (for Manhattan distance we have $p = 1$, for the Euclidean, $p = 2$, and for the Maximum distance $p = \infty$).
  - Manhattan Distance ($L_1 norm$): $d(T, S) = \sum_{i=1}^{n} |T_i - S_i|$
  - Euclidean Distance ($L_2 norm$): $d(T, S) = \sqrt{\sum_{i=1}^{n} (|T_i - S_i|)^2}$
  - Maximum Distance ($L_\infty norm$): $d(T, S) = \max_{0<i\leq n} |T_i - S_i|$
  - Mahalanobis Distance: $d(T, S) = \sqrt{(T - S) \sum^{-1} (|T - S|)^T}$ where $\sum$ is the covariance matrix.
- **Elastic Measure (one-to-many/one-to-none):**
  - Dynamic Time Warping (DTW): let T and S be time series with lengths n and m, first a distance matrix of size $m \times n$ is calculated representing the distance of ith point of T with jth point of S ($1 \leq i \leq n$ and $1 \leq j \leq m$). Then the objective function $DTW(T, S) = \min(\sqrt{\sum_{k=1}^{K} w_k})$ is used to build the warping path $W = \{w_1, w_2, \ldots, w_k\}$

Appendix D

Extended Tables and Figures for Chapter 5

Table D.1: Statistics on the Oil and Gas sector of S&P 500 stocks during June 22 to July 27

	mean	**std**	**min**	**25%**	**50%**	**75%**	**max**
APA	54.93	1.67	51.76	54.06	54.89	56.26	57.90
APC	54.41	1.60	49.50	53.81	54.69	55.44	57.42
BHI	44.82	1.21	41.91	43.93	44.96	45.66	47.00
CHK	4.57	0.36	4.06	4.31	4.50	4.64	5.39
CNX	16.33	1.14	13.63	15.84	16.49	16.98	18.89
COG	25.03	0.75	23.54	24.54	25.28	25.55	25.97
COP	42.23	1.33	40.07	41.31	42.12	43.17	45.36
CVX	104.29	1.82	100.36	103.06	104.51	105.64	107.03
DNR	3.51	0.50	2.90	3.15	3.30	3.76	4.68
DO	24.46	1.03	21.95	23.99	24.37	25.25	26.11
DVN	37.32	1.27	33.75	36.36	37.52	38.07	40.01
EOG	82.41	1.80	78.43	81.46	82.60	83.74	84.97
EQT	76.55	1.68	73.48	75.29	76.76	77.62	79.33
ESV	9.91	0.55	8.78	9.65	9.89	10.34	10.80
FTI	26.52	1.00	24.42	25.87	26.52	27.34	28.11
HAL	44.36	1.11	41.88	43.54	44.54	45.27	46.03
HES	56.60	2.34	51.60	55.31	56.85	58.02	60.15
HP	66.19	2.03	62.66	64.91	66.19	67.93	69.77
KMI	19.48	1.41	17.29	18.40	18.94	20.76	21.95
MPC	36.86	1.34	32.93	36.41	37.01	37.73	39.27
MRO	14.73	0.63	13.13	14.61	14.81	15.17	15.68
MUR	30.65	1.43	27.80	29.81	30.69	31.89	32.66
NBL	35.82	0.85	33.89	35.03	35.97	36.52	37.01
NBR	9.76	0.46	9.12	9.44	9.71	10.00	10.79
NE	8.27	0.53	7.32	8.02	8.21	8.40	9.50

NFX	43.46	1.19	40.41	42.86	43.75	44.43	45.16
NOV	33.08	1.39	31.27	32.03	32.94	33.74	36.50
OKE	46.14	1.07	43.37	45.49	46.55	46.81	47.87
OXY	75.86	1.22	73.12	75.23	75.70	76.82	78.31
PSX	76.91	1.94	74.27	75.39	76.31	78.72	80.87
PXD	153.12	3.33	146.61	151.14	152.96	155.35	160.59
QEP	17.69	0.60	16.75	17.28	17.64	17.95	19.32
RDC	17.45	0.95	15.34	17.04	17.59	17.82	19.55
RIG	11.79	0.59	10.60	11.38	11.93	12.15	12.84
RRC	43.22	1.51	40.48	42.31	43.23	44.12	46.45
SE	36.11	0.84	34.37	35.89	36.42	36.73	37.05
SLB	78.89	1.45	75.07	78.03	79.09	79.78	81.61
SWN	13.33	0.76	11.66	12.88	13.40	13.94	14.47
TSO	75.38	2.09	70.01	74.44	75.73	76.85	78.73
VLO	50.66	1.62	47.24	49.85	50.60	51.90	53.71
WMB	22.20	1.70	20.00	20.74	21.73	23.59	25.13
XOM	92.95	1.75	88.86	91.60	93.64	94.07	95.12