

Data analytic approach for manipulation detection in stock market

Abstract

The term “price manipulation” is used to describe the actions of “rogue” traders who employ carefully designed trading tactics to incur equity prices up or down to make profit. Such activities damage the proper functioning, integrity, and stability of the financial markets. In response to that, the regulators proposed new regulatory guidance to prohibit such activities on the financial markets. However, due to the lack of existing research and the implementation complexity, the application of those regulatory guidance, i.e. MiFID II in EU, is postponed to 2018. The existing studies exploring this issue either focus on empirical analysis of such cases, or propose detection models based on certain assumptions. The effective methods, based on analysing trading behaviour data, are not yet studied. This paper seeks to address that gap, and provides two data analytics based models. The first one, static model, detects manipulative behaviours through identifying abnormal patterns of trading activities. The activities are represented by transformed limit orders, in which the transformation method is proposed for partially reducing the non-stationarity nature of the financial data. The second one is hidden Markov model based dynamic model, which identifies the sequential and contextual changes in trading behaviours. Both models are evaluated using real stock tick data, which demonstrate their effectiveness on identifying a range of price manipulation scenarios, and outperforming the selected benchmarks. Thus, both models are shown to make a substantial contribution to the literature, and to offer a practical and effective approach to the identification of market manipulation.

1. Introduction

When traders set out to manipulate market prices, they use deliberately chosen and designed buying and selling actions, in order to affect the bid or ask prices - in some cases even the market prices - of given equities. Such activities can take a variety of formats, all of which are extremely damaging to the proper functioning, integrity, and stability of the financial markets. This problem attracted much attentions of regulators in USA, EU, UK and even China, especially after the financial crisis in 2010 (EU, Market Abuse, Amended Proposal for a Directive, 2012) (SEC, 2013) (CFTC, May 20 2013) (CME, 2014). The trading activities with “intent to mislead the market and other participants” have been identified as manipulative actions in the EU regulation, Market Abuse Directive, MAD (EU, Market Abuse, Amended Proposal for a Directive, 2012). The US Commodity Future Trading Commission further defined the manipulative activity as bid or offer orders with intention of cancellation before execution with reckless disregard for the orderly execution or market integration (CFTC, May 20 2013). Chicago Mercantile Exchange (CME) released a “Rule 575” on August 28, 2014 (CME, 2014) to further define the manipulation as the orders with the purpose to cancel, or mislead other market participants, or to disrupt the exchange or other fair executions. In EU, the Markets in Financial Instruments Directive II (MiFID II) is proposed as the legislation regulating firms’ activities in financial markets (EU, MiFID II, 2014). In MiFID II, market manipulative practice is explained as a format of financial market fraud that includes all types of market abuse and spoofing activities. Due to the lack of existing research and the implementation difficulties, the MiFID II application date has been extended by European Commission to 3 January 2018.

In academia, the manipulative trading activities were firstly introduced by Allen (Allen & Gale, 1992) as three primary types, named: information based manipulation, action based manipulation and trade based manipulation. Information based

manipulation is where false or misleading information is released by the manipulators to affect the price of a stock. Action based manipulation is where a person's market squeeze actions change the supply or demand of an asset. Trade based manipulation does not contain any illegitimate actions such as financial rumour spreading and market resource squeezing but is carried out only by legal trading activities of selling and buying sequences for inducing market prices move to follow manipulators' expectations to make profit (Allen & Gale, 1992) (Aitken, Harris, & Ji, 2009), such as the manipulation tactics, spoof trading and quote stuffing discussed in (Lee, Eom, & Park, 2013). Aggarwal and Wu (Aggarwal & Wu, 2006) carried on from Allen and Gale's work and looked at the profitability and the market effects of such activities. They looked at the implication of manipulation on markets efficiency and what indices were affected by it, and also found that equities with low values and illiquidity were commonly targeted. They found that manipulators can curtail the effectiveness of arbitrage activities and that during the manipulation period, liquidity, returns and volatility increased. Huang and Cheng analysed the manipulation cases on stock market in Taiwan (Huang & Cheng, 2015) and find out the stocks of poorly managed companies are usually the manipulated target on financial market and they also find out the manipulation brings stronger volatility and worsens the financial market efficiency. Before their studies, Chow et al (Chow, Hung, Liu, & Shiu, 2013) also investigated the manipulation cases on Taiwan markets and found that expiration day effect partially contributes to the market close manipulation attempts. Based on the study, Huang et al (Huang & Cheng, 2014) studied the new closing mechanism on the market variables and the aggressiveness of the trading activity while Jawadi et al (Jawadi, Louhichi, Cheffou, & Randrianarivony, 2016) further studied the trading volumes and volatility across four international markets.

In contrast to the volume of theoretical and empirical work on the financial markets in market manipulation, there are fewer works related to the detection of the manipulation activities. Traditional data mining approaches have been studied for identifying the manipulative trading activities in (Öğüt, Doğana, & Aktaş, 2009), (Mongkolnavin & Tirapat, 2009), (Diaz, Theodoulidis, & Sampaio, 2011), and (Cao, Li, Coleman, Belatreche, & McGinnity, 2015). However, those studies have focused primarily on detecting manipulation based on assumptions drawn from empirical studies: the changes of certain market variables indicate manipulations. Those market variables include daily return, daily trading volume and daily volatilities (Öğüt, Doğana, & Aktaş, 2009), liquidity (Diaz, Theodoulidis, & Sampaio, 2011), and transaction price, volume and time (Mongkolnavin & Tirapat, 2009). However, there is no proof showing that the market change is a sufficient condition for the manipulation, which means that, although a manipulation may incur the changes of market variables, those changes might not be only results of manipulations, but could possibly be results of other events in the market, i.e., big news, economic cycle. Therefore, the existing work does not contribute to a theoretical foundation for detecting manipulations. In other words, there is a "knowledge gap" between features of the selected market variables and the detected target: manipulation. Ngai et al (Ngai, Hu, Wong, Chen, & Sun, February 2011) find out a distinct lack of research on securities and commodities fraud, which is another term of market manipulation. Consequently, study of manipulative trading and its detection is important and necessary due to literature gap on as well as the urgent requirements from financial decision maker, i.e. the regulators around the world. At the same time, data analytic approaches are widely used in finance area to enhance the analysis performance in the tasks from daily stock return forecasting (Oztekin, Kizilaslan, Freund, & Iseri, 2016) portfolio management (Mendes, Paiva, Peruchi, & Balestrassi, 2016). However, the data analytic approaches on trading behaviour analysis have not been studied much yet. Therefore, this paper addresses two gaps in existing works by 1) proposing a framework of detection models focusing on the static and dynamic features of manipulative behaviours rather than the market variables through a comprehensive analysis of the manipulative strategies and the existing regulations; 2) detecting the manipulation through the data-driven models that is completely trained based on real trading behaviours data from the markets. Therefore, the only assumption of the proposed

models is that the manipulative behaviours can be distinguished from the normal (legitimate) trading actions, explicitly or implicitly. From the view of machine learning algorithms, the objects, which are not explicitly separable, can be transformed and mapped to a “feature space”, where the extracted features are easily separable. Those transformation and mapping is usually termed as feature extraction. In this paper, we propose two different feature extraction methods for static and dynamic model respectively for mapping the original data to a separable feature space, where the manipulative behaviours can be easily separated from the normal actions. To the best of our knowledge, this is the first study of manipulation detection models purely based on manipulative behaviours.

Thus, in this paper, two primary strategies used to manipulate price, spoofing trading and quote stuffing, were thoroughly studied and summarised. Then, two models - one static, one dynamic - were proposed, for detecting various aspects of price manipulation. In development of the static model, the trading behaviours were represented as non-stationary, multi-dimensional data. To model such data, we proposed a simple but effective method of transformation, which partially reduces the non-stationary nature of the multi-dimensional trading data. Based on this, two popular machine learning models, one-class support vector machine (OCSVM) and k-nearest neighbour (k-NN), were applied to the transformed data, and used to discern patterns of manipulative behaviour amidst a large volume of legitimate trading cases. Since both the OCSVM and k-NN models treat each data vector as a single object (i.e. not as contextually-related sequences), neither of them encodes temporal information concerning sequential trading activities. Thus, they were defined as static models. This aspect of the study offers two key contributions. The first is the formulation of manipulative activities detection in terms of a trading behaviour modelling problem based on thorough analysis of the strategies used to manipulate market prices. The second is the proposal of a static detection model that comprises a feature transformation approach and a behaviour learning module. To the best our knowledge, this is the first approach that models the trading behaviour data vector and detects the manipulative patterns.

In contrast, the dynamic model proposed in this paper explored the contextual relationships of the sequential trading orders. This is mainly because the existence of particular manipulation strategies, such as quote stuffing, is discovered by discerning particular contextual relationships between orders in a sequence. To identify such cases, a model is needed to analyse sequential relationships between aggressive orders and the associated temporal information. We then proposed an adaptive Dynamic Model (ADM) to achieve this; this model isolates manipulative behaviour through examination of a sequence of bid/ask price changes, and determines whether these are the result of sequential aggressive spoofing orders. This element of the work makes significant contributions in terms of (1) its proposal of a wavelet transformation-based feature extraction approach, which can, specifically, extract the short-term oscillation features of quote stuffing strategy and (2) the proposal of a hidden Markov model (HMM) based ADM, which models and detects patterns of price manipulation based on sequences of price variation/oscillation, as opposed to using a single value that exists at a given point in time.

This paper takes the following form. Section 3 begins with a detailed explanation of the problems of price manipulation, and an analysis of typical cases. Consequently, in Section 3.1, a transformation procedure for mapping non-stationary trading activities to pseudo-stationary attributes is given. Section 3.2 provides a discussion of the proposed static detection model and the results of the experimental evaluations. Section 4 describes a dynamic model to analyse the features of some manipulative strategies and the patterns of price change that results from these. In Section 4.1, a wavelet transformation-based feature extraction approach is put forward, an approach that isolates short-term oscillations from original price. In Section 4.2, a hidden Markov model-based detection approach is suggested, which can be used alongside a feature extraction method to identify the price

oscillations generated by manipulative trading activity. Section 4.3 presents the thorough evaluation of the proposed approach. Section 5 concludes this paper on price manipulation.

2. Features of Manipulation Behaviours

2.1 Static features

In capital markets, investors submit limit orders to the electronic platform of the exchange market; these limit orders describe the intentions of the investor in terms of buying or selling volumes of a specific equity at a given price. When such eligible limit orders can be matched, they are executed. When orders are outstanding, the highest prices that investors are willing to pay for the equity is referred to as the best bid price, while the lowest price at which the holder is willing to sell, is known as the best ask price. The gap between bid and ask price is called the bid-ask spread. The second, third or fourth highest price at which investors will buy/sell is termed the second, third or fourth bid or ask price, as appropriate. When this is given in an order book, it is generally designated as the first, second, third or fourth level of bid or ask price.

Quote stuffing is a primary form of manipulation in several major exchange markets in Europe and the USA (Tse, Lin, & Vincent, 2012) (Nanex, Whac-A-Mole is Manipulation, 2012). It causes the market to be flooded with massive volumes of new orders, which are then quickly cancelled in rapid succession in order to generate successive new best ask or bid quotes, each of which is potential bait for an opposite order, which may in turn generate profit. Ramping and quote stuffing are very similar, the only real difference being that quote stuffing fills the order book with a greater number of spoofing orders, which are cancelled more quickly than is the case with ramping. In short, all of the price manipulation techniques mentioned above employs the same tactic, namely the submission of fraudulent spoofing orders to the market, and subsequent taking advantage of the changed prices (assuming these are in line with expectations). Profits are generated in various ways, as is shown in Figure 1.

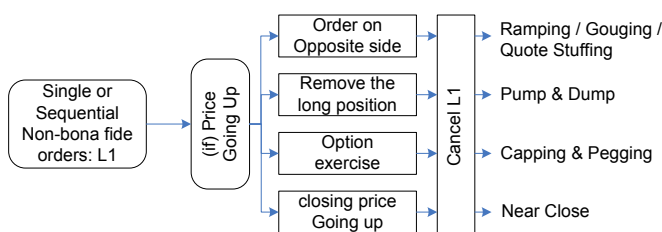


Figure 1 Types of Price Manipulation

According to the discussion, price manipulation strategies can be sub-divided into two main forms:

- (1) spoofing trading, which involves orders with large volumes and moderate prices and
- (2) quote stuffing, which uses aggressively-priced orders and a very short cancellation time.

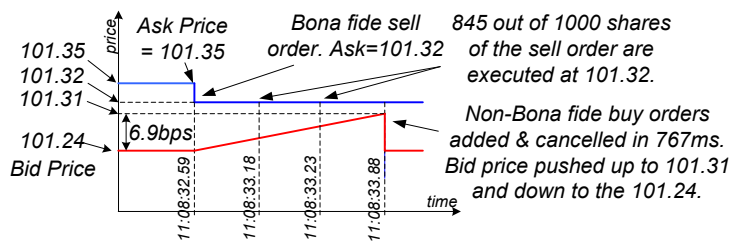
These features distinguish spoofing orders from their legitimate counterparts. Thus, regardless of the mechanism used to create profit, price manipulation is characterised by placement of the spoofing order, intended to mislead the market via the bid or ask price. Manipulators' deliberate construction of single or sequences of, trading orders alter the bid or ask price and thus reveal the traders' fraudulent intentions. The detection model targets the manipulation strategy, and that strategy is characterised by the nature of the spoofing orders. The tactics of manipulators are carefully designed in light of market impact theory (Hautsch & Huang, 2012), which suggests that the quotes and sizes of posted orders strongly correlate with subsequent market effects. A vector autoregressive model (VAR) has been used to provide a quantitative estimation of this effect (Hautsch & Huang, 2012), which revealed that notably large size or quote (compared with the current bid or ask) exerted a substantial effect on the market price. Clearly it is in the best interest of legitimate traders to eliminate those impacts, however for fraudulent traders, such effects are crucial to utilise in their strategies to make profit. Therefore it is reasonable to conclude that orders submitted with a purpose to manipulating prices should be either large sized or of a higher or lower price than the prevailing bid or ask, in order to maximise the effects on the market. However, in practice, obviously huge sized or aggressive priced trading can be easily suspicious as anomalies by the regulators (Hautsch & Huang, 2012) (Lee, Eom, & Park, 2013). Considering this, the manipulators usually carefully design the trading order sequences to make the profit as well as hide from the monitors. Thus the authors argue that any strategies for price manipulation require that such strategy (1) maximises the price change it induced and (2) minimises the chances of execution. Consequently, the detection model that targets on those features shall be able to effectively identify the manipulative activities that are documented in (Lee, Eom, & Park, 2013).

2.2 Dynamic features

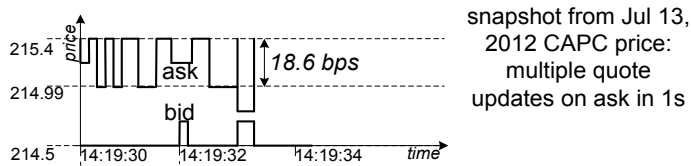
In addition to the features discussed in Section 2.1, quote stuffing, one of the mostly used manipulation in major exchange markets in both Europe and the USA (Tse, Lin, & Vincent, 2012), is also documented to flood the market with vast quantities of new orders, which are then cancelled in swift succession, thereby creating a succession of new best bid/ask quotes. Each of these quotes has the potential to lure an opposite order, which may generate a profit. When fraudulent traders manipulate prices, the intention is to create a significant price change using spoofing orders. This is achieved not incidentally by heuristic attempts of order submission, but by carefully designed process based on theories of market microstructure, referred to as market impact (i.e. the impact of the trader's activities upon the market) (Ferraris, 2008). For example, the quantitative research given in (Hautsch & Huang, 2012) suggests that a quote even slightly (5 bps) higher than the bid price can cause a further 5.95bps increase in the best bid price.

In September 2012, a real case of quote stuffing was reported and described by the Financial Industry Regulatory Authority (FINRA) of the USA (Ong & Condon, 2012) (Nanex, Whac-A-Mole is Manipulation, 2012). In this case, quote stuffing was used to create an impression of strong buying interest as represented in Figure 2(a) and is carried out as: when a large, completely genuine, selling order was placed at an expected price, it was followed by mass of sequential fraudulent spoofing buy orders with quotes successively higher than the best bid price. The spoofing orders were submitted to give the impression of active purchasing interest in that equity. The best bid price was elevated by 6.9 basis points (bps) as a result and thus, the bid-ask spread narrowed by 87.5%. An investor, encouraged by the (fake) bid price changes, thus responded to the genuine sell order. However, immediately after the execution, the spoofing buy orders were all cancelled, at which point the bid price plummeted back to its original level. In this case, spoofing benefited the manipulators through the genuine selling order because it was executed by other traders - who were fooled by the effects of the spoofing - at the manipulated price. The entire process of manipulation lasted just 819ms, and cancellation of the fraudulent orders took just 767ms.

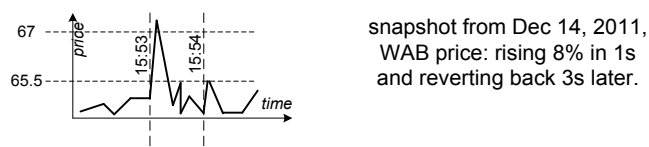
Such manipulative tactics are usually carried out extremely swiftly, in order to minimise the risk of the spoofing orders being picked up by genuine investors, and thus with its sequential quotes and speedy cancellations, the manipulative orders in Figure 2(a) made a sawtooth-shaped market impact upon the bid price. That is why this particular case can be shown as an instantaneous sawtooth pattern, on the bid price time series. That sawtooth pattern, which is also referred to in (Tse, Lin, & Vincent, 2012) and (Jan., 2012), has been identified as a characteristic pattern seen in bid/ask price time series when manipulation is on-going. Similarly, the market impact of ramping - another form of manipulation- can be visualised through characteristic patterns on the bid/ask price time series. In ramping, unlike quote stuffing, a single spoofing order is placed inside the bid-ask spread - i.e. at higher than the prevailing bid or below the prevailing ask. This allows small manipulations to the bid/ask price - a few percentage points and then a return to the original level - within very small time intervals. In Figure 2(b), small square wave fluctuations reveal the pattern of spoofing. However the spoofing pattern can, sometimes, take the form of more significant movement, as is seen in Figure 2(c). This is rare, since most exchange markets - including Euronext NSC - interrupt trading if prices change by more than 2% within a defined time period (such as 1s), however, (Jan., 2012) cites this as a typical case of manipulation.



(a) Sawtooth pattern



(b) Pulse pattern



(c) Square pattern

Figure 2 Typical patterns of price manipulation (reproduced from data in (Tse, Lin, & Vincent, 2012)).

The sawtooth, square wave and pulse patterns correspond to the dynamic features of quote stuffing. The study of genuine price manipulation cases has identified a mixture of those patterns (Tse, Lin, & Vincent, 2012) and consequently, the problem of identifying behaviours designed to manipulate market prices can be transformed into the problem of identifying unusual or unexpected patterns in the bid/ask time series. This links neatly with the stock volatility rules that were published in April 2013 by the SEC in the USA, which permit trading only within given price bands, in order to restrict volatile trading activities (SEC, 2013). Since manipulative tactics are generally enacted within a very brief timeframe and in a single trading day, the problem becomes detecting anomalous patterns in an intra-day bid/ask price time series. The bid/ask price update event determines the time scale, and so the adjustment speed as it is measured in physical time is dependent, ultimately, upon the underlying frequency of order activities in that area and thus varies across the market or markets.

The changes in price that are set off by manipulative tactics are unusual short term monetary oscillations which show small amplitudes around the equilibrium level of the price - for example, 6.9bps in Figure 2(a) and 18.6bps in Figure 2(c). It is common for such oscillations to be regarded as contamination of the financial data (Haven, Liu, & Shen, 2012). In fact it is crucial, when dealing with financial time series, to isolate the true signal from within the data series (Haven, Liu, & Shen, 2012), since that data series contains a mix of short, high-frequency noisy transients and long, low-frequency movement. The high-frequency noisy components are where manipulative behaviours are concealed, and so thoroughly examining those components is likely to be helpful in the development of an effective model to detect price manipulation.

3. Static Model

According to (Lee, Eom, & Park, 2013), and examination of genuine manipulation cases, spoofing trading is characterised by an order of a size at least twice that of the previous day's average order size, and a price at least 6bps away from the current bid or ask price. The cancellation time should exceed 30 minutes. These quantifications - twice, 6bps, 30 minutes - are a typical illustration for the authors' arguments, because they relate to maximisation of the induced price change (i.e. the impact of the actions) by a large order (at least twice the average of the previous day) that stays on the order book for a relatively long time (30 minutes) and to the minimisation of the risk of execution by passive quotes (6bps away from the bid or ask) (Lee, Eom, & Park, 2013). The quantifiable features of quote stuffing (Ong & Condon, 2012) arise from a sequence of aggressively-priced but quickly cancelled buy orders: in such cases the quoted prices are inside the spread at around 6.9bps and cancellation is at 819ms. These factors also conform to the authors' argument: the aggressive quotes maximise the fraudulently-achieved price changes and the immediate market sweeping minimises the risk of execution. Both of these are covered by two key conditions defined in the authors' argument.

The spoofing orders in these strategies are graphically shown in Figure 3. In this figure, a three-level order book is initiated at the best bid, $p_t^{b,1}$ and best ask $p_t^{a,1}$ - (b and a stand for bid and ask, respectively) and the dotted lines represent a quick sweep of the market, i.e. with an extremely small time to cancellation. The "sticks" outlined in solid lines, between the first and second level bid or ask prices, represent the large in volume, passive in price spoofing order, while the shorter "sticks" outlined in dotted lines, placed within the spread, represent the aggressive prices and small time periods used for spoofing orders in a quote stuffing strategy.

Both Figure 3 and the author's analysis shows that the combination of order price, volume and the time period between submission and execution/cancellation are illustrative of the legitimacy (or otherwise) of trading behaviour. Through construction of a feature vector that uses the features of price, volume and time, it is possible to quantitatively visualise the trading behaviour as a collection of such vectors. This form of presentation can be useful in modelling data regarding trading behaviour, using machine learning techniques. However, non-stationary nature of financial data makes such modelling far more

complex. In order to address this, this paper proposes a transformation method that partially reduces the non-stationarity of the original data, and the next section will explore that method in details.

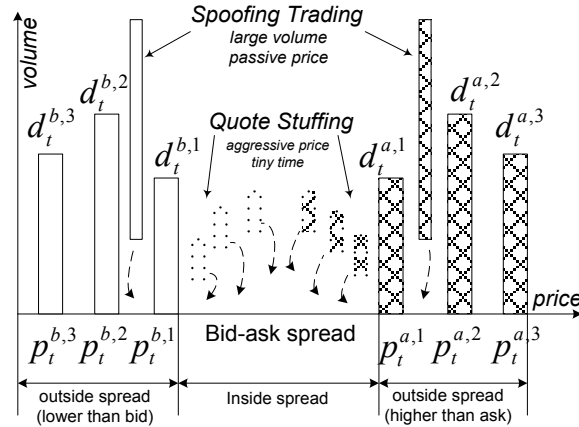


Figure 3 An illustration of spoofing trading and quote stuffing strategies in a three-level order book.

3.1 A transformation method to reduce the non-stationary nature of the data

As has been discussed, financial data are usually considered to be a non-stationary time series (Ghazali, Hussain, Nawi, & Mohamad, 2009) and the authors set out to transform such original, non-stationary, data into stationary data for reasons already given. Time series analysis, such as the autoregressive integrated moving average (ARIMA) model (which is usually used in cases where the stationary nature of data can be eliminated using an initial differencing step (Tsay, 2010)) echoes this situation. The log return is also regarded as a transformation from market data to price difference, which is believed not to be perfectly time invariant (Engle, 1982) (Bollerslev, 1986) but nonetheless has recently been analysed and determined to be stationary (Lee, Lee, & Lee, 2010). The authors, inspired by both the differencing step and log-return approaches, have defined a transformation procedure that converts the original order data into a new measure. Under this new measure, the data shows pseudo-stationary feature.

Taking i as a time index, which denotes all order book activities (namely submission, cancellation and execution), a buy (or sell) order o_i^B (or o_i^S) is shown as a three-dimensional vector, $[p_{o_i^B}, v_{o_i^B}, t_{o_i^B}]'$ (or $[p_{o_i^S}, v_{o_i^S}, t_{o_i^S}]'$), where p , v and t represent, respectively, price, volume and physical submission time and B and S are buy and sell, respectively. Further, p_i^b and p_i^a indicate the best bid and best ask prices instantaneously before the i^{th} order activity and τ is the length of a sliding window, used to calculate the moving average of the order volume. Given that manipulative behaviour usually occurs within a short time period, τ is set to one trading day. Thus, \bar{v}_τ^b and \bar{v}_τ^s define the moving average volume of the buy and sell orders in the previous period of time, τ , excluding the current data point, i . The cancellation and execution times in order o_i^B (or o_i^S) are given as $t_{o_i^B}^c$ and $t_{o_i^B}^e$ (or $t_{o_i^S}^c$ and $t_{o_i^S}^e$), and the lifecycle of an order $l_{o_i^{B/S}}$ can then be denoted as:

$$l_{o_i^{B/S}}^c = t_{o_i^{B/S}}^c - t_{o_i^{B/S}} \quad (1)$$

$$l_{o_i^{B/S}}^e = t_{o_i^{B/S}}^e - t_{o_i^{B/S}} \quad (2)$$

for the order that is cancelled or executed, respectively. Therefore, the average lifecycle of orders, in the prior period τ , are calculated as follows:

$$\bar{l}_{\tau,B/S}^c = \frac{1}{N} \sum_{i=1}^N l_{o_i}^{c,B/S} \quad (3)$$

$$\bar{l}_{\tau,B/S}^e = \frac{1}{N} \sum_{i=1}^N l_{o_i}^{e,B/S} \quad (4)$$

where N is the number of orders in the sliding window. Equations (5), (6) and (7), below, then define the transformation, with $p_{o_i}^n, v_{o_i}^n, t_{o_i}^n$ representing three transformed attributes.

$$p_{o_i}^n = \begin{cases} \ln\left(\frac{p_{o_i}^B}{p_i^b}\right) \\ \ln\left(\frac{p_{o_i}^S}{p_i^a}\right) \end{cases} \quad (5)$$

$$v_{o_i}^n = \begin{cases} \ln\left(\frac{v_{o_i}^B}{\bar{v}_\tau^b}\right) \\ \ln\left(\frac{v_{o_i}^S}{\bar{v}_\tau^s}\right) \end{cases} \quad (6)$$

$$t_{o_i}^n = \begin{cases} \ln\left(\frac{l_{o_i}^{c,B}}{\bar{l}_{\tau,b}^c}\right) \text{ or } \ln\left(\frac{l_{o_i}^{e,B}}{\bar{l}_{\tau,b}^e}\right) \\ \ln\left(\frac{l_{o_i}^{c,S}}{\bar{l}_{\tau,s}^c}\right) \text{ or } \ln\left(\frac{l_{o_i}^{e,S}}{\bar{l}_{\tau,s}^e}\right) \end{cases} \quad (7)$$

This transformation is developed as a testable hypothesis that the non-stationary nature of the (transformed) data is significantly less than that of the original. When a time series X , between time point t and s is represented, the stationary nature of the time series (X_t, X_s) is usually tested by the weak definition of stationarity (Priestley, 1982) (Van Bellegem, 2003) and thus the mean and variance of (X_t, X_s) are independent of t and/or s . In order to assess the degree to which the trading order data is stationary, the authors calculate both the mean and the variance for both original and transformed data pertaining to the trading orders. In doing this, each attribute is considered as a single time series, and so the original and the transformed data are presented as three time series respectively:

$$P = \{p_{o_i}^{B/S}\}, V = \{v_{o_i}^{B/S}\}, T = \{t_{o_i}^{B/S}\};$$

$$P_n = \{p_{o_i}^n\}, V_n = \{v_{o_i}^n\}, T_n = \{t_{o_i}^n\}.$$

Intel is one of the most popular NASDAQ stocks, when taken in terms of the total market capital, and is thus used to evaluate the difference in stationarity before and after transformation. Datasets from the LOBSTER project (Berlin, 2013) cover the messages exchanged over five trading days (11-15 June 2012) and contain more than 40,000 trading orders for each stock. We calculated both the rolling mean and the variance of the time series $(X_{t+\Delta t}, X_{s+\Delta t})$, with Δt from 0 to the length of the time series. Figure 4 and Figure 5 show the calculated rolling mean and variance of price, volume and time for the Intel dataset. It is important to note that only the first 200 lag values are shown in the figures to maintain the visual clarity, when comparing the original and transformed data. Figure 4(a)-(f) present the calculated values of mean and variance, for the original data. It is notable that the mean and variance values of price, volume and time are not constant with lag increase. The mean values of price, volume and time fall within the ranges of $2.755 \sim 2.756 \times 10^5$, $200 \sim 400$, and $1841 \sim 1846$, respectively and the variance of price, volume and time also fall within certain ranges of value, as opposed to holding constant values. The changes in mean and variance show the evolutions of price, volume and time on differing Δt , and thereby also show the non-stationary nature. Figure 5(a)-(f) shows the values of mean and variance of the transformed data. However, the Figures all show clearly

that the mean and variance of the transformed price, volume and time all fall within very tiny ranges of value, in other words they are almost constant. Such “nearly constant” values show features of being pseudo-stationary, because the mean and variance occur independently of the time points t and s .

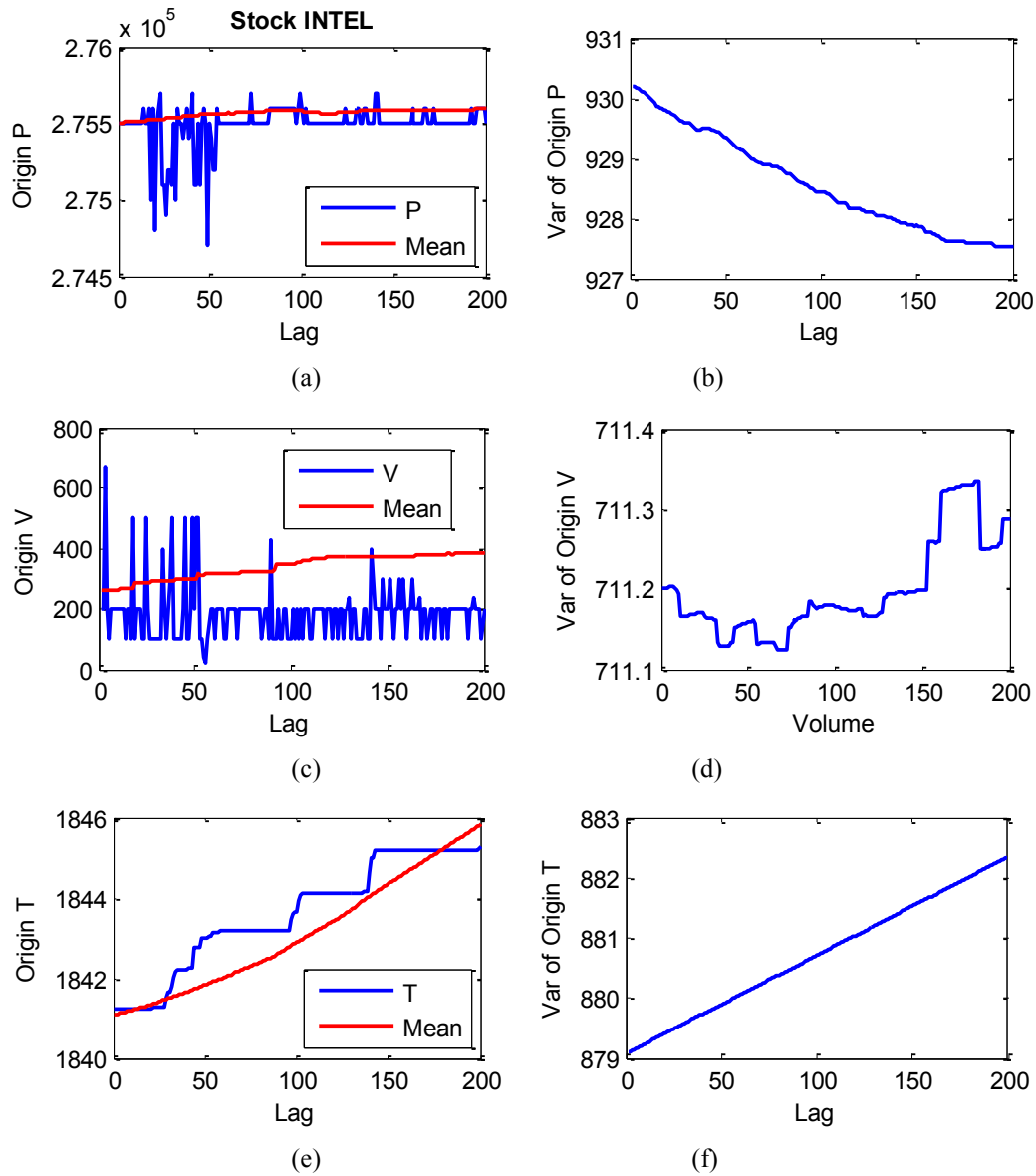


Figure 4 Mean and variance values for the original price, volume and time of Intel stock, measured across varying time lag Δt .

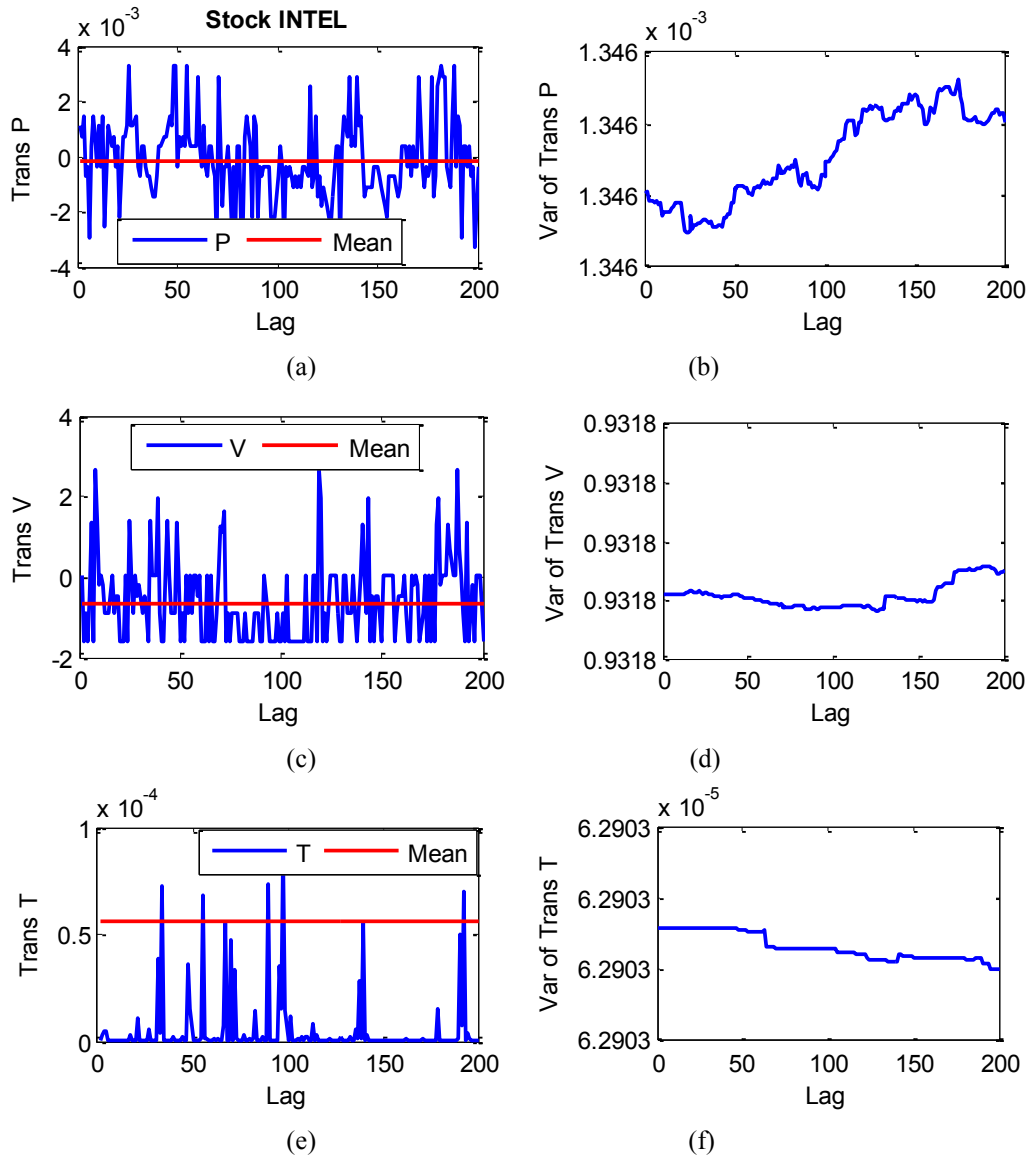


Figure 5 Mean and variance of the transformed price, volume and time data for Intel stock, shown across varying time lag Δt . Please note that the y-axis values of (b), (d) and (f) are so very small that the visual display does not show any differences.

Although the transformed data exhibits mean and variance values for price, volume and time that are almost entirely independent of Δt , a quantitative measurement of the dispersion of the sequences of the calculated mean and variance is needed, in order to permit further evaluation of the extent to which the proposed transformation approach can reduce the degree of non-stationarity. In order to achieve this, the coefficient of variation (CV) is calculated. The CV is defined as the ratio of standard deviation to the mean of a data sequence. Calculation of the CV is an established statistical method that provides a normalised measure of the dispersion of a distribution. The trading data's mean and variance are calculated as a new time series, namely mean and variance across different lags. To ascertain the dispersion of those new time series, the CV values of price, volume and time are calculated, both for the original and the transformed data. The results are compared using Intel data together with three other high-profile NASDAQ stocks, which are Microsoft (MSFT), Google (GOOG) and Apple (APPA). The results are presented in Table 1, with the CV values of the original data given in brackets. Comparing the CV values in Table 1 reveals that the CV values of the transformed price, volume and time are substantially smaller than those for the original data. The level of dispersion in the transformed data sequences is far lower than that found in the original data, which suggests that the

mean and variance values of the transformed data series are virtually constant. Thus, these values satisfy the weak definition of stationarity, and thus demonstrate the ability of the transformation approach to effectively reduce the non-stationary nature of original data. Given the ability of the transformation to induce stationarity, and the characteristics of manipulation strategies that have already been discussed, Figure 6 and Figure 7 show the transformed price $p_{o_i^n}$, volume $v_{o_i^n}$ and time $t_{o_i^n}$ of the trading orders as well as the original price $p_{o_i^{B/S}}$, volume $v_{o_i^{B/S}}$ and time $t_{o_i^{B/S}}$ in 3-dimensional space.

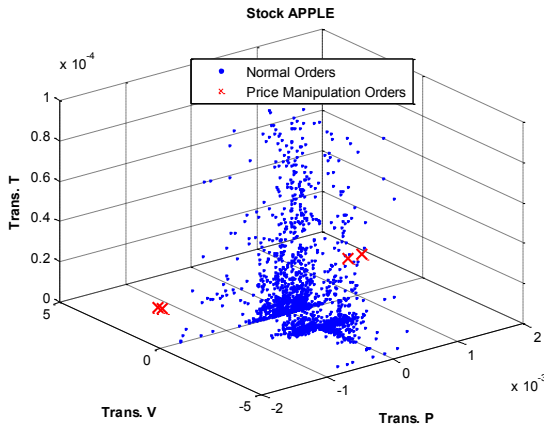
Table 1 Coefficient of variation for the sequence of mean and variance of the price, volume and time of four stocks, with CV of the original data given in brackets.

		Coefficient of variation				
		MSFT	INTC	GOOG	APPA	
Price	Mean	Trans. Data (original Data)	-1.08E-04 (3.31E-01)	-3.73E-06 (8.73E-02)	-2.63E-05 (1.20E-02)	1.19E-04 (1.01E-01)
	Var.	Trans. Data (original Data)	1.31E-10 (2.02E-01)	6.76E-11 (2.05E-01)	6.59E-10 (3.66E-01)	4.71E-10 (2.16E-01)
Volume	Mean	Trans. Data (original Data)	-1.11E-06 (4.26E-02)	-9.51E-07 (1.17E-01)	-1.33E-06 (5.74E-02)	-1.31E-06 (1.59E-02)
	Var.	Trans. Data (original Data)	1.68E-10 (1.88E-01)	1.08E-10 (2.08E-01)	1.83E-10 (9.44E-02)	1.64E-10 (1.40E-01)
Time	Mean	Trans. Data (original Data)	3.38E-06 (2.25E-01)	3.70E-06 (7.75E-02)	3.26E-06 (8.74E-01)	3.34E-06 (5.14E-01)
	Var.	Trans. Data (original Data)	2.37E-09 (2.08E-01)	4.63E-09 (1.92E-01)	1.76E-09 (1.66E-01)	1.00E-08 (1.83E-01)

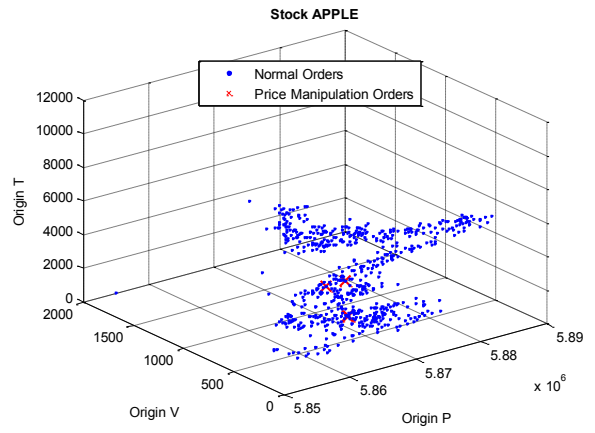
In this four stock dataset, cases for Microsoft, Google and Intel were captured on 15 June 2012 and for Apple were captured on 10 July 2013. Cases for Apple stock contain both legitimate and manipulated cases, which were reported by Nanex (Nanex, Incredible, Blatant Manipulation in Apple Stock, 2013) on the same trading day. Figure 6 shows the static features of the manipulative action:

- The cases of price manipulation, which are shown as red crosses, deviate from the mass legitimate trading orders in the transformed domain in Figure 6(a) and overlap with them in the original data as given in Figure 6(b).
- The transformed legitimate trading orders tend to be distributed along three axes as illustrated in Figure 6(a), whereas the original data exhibit exotic, and time variable, distributions as per Figure 6(b).

The legitimate data generated for three other stocks, Microsoft, Google and Intel, given in Figure 7-Figure 9, echoes the patterns seen in Figure 6 (The figures only show part of the data of legitimate and manipulated cases to maintain the visibility of the graph). It can be observed from the figures that the transformed legitimate orders congregate in the form of an agglomerative cluster, which occupies a given space with analogous, but varying, shapes due to the distinct trading behaviours. In contrast, the shapes generated by the original data are markedly more exotic. Since the cases of manipulation shown in Figure 6(a) seem to be located at some distance from the legitimate trading data cluster, the distribution of such clusters can be used to determine an effective decision boundary. However, it is not possible to declare such boundaries simply by establishing thresholds on three attributes, due to the unknown distribution of the 3-D vectors. These vectors may (or may not) have a convex surface, and in the detection of anomalies, inaccurate results are generated when thresholds are used as boundaries on concave surface distributions. Some sophistication is needed in order to describe cluster shapes accurately, using only the legitimate trading data.

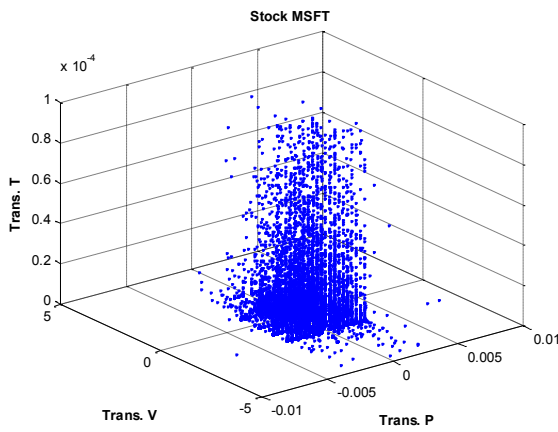


(a)

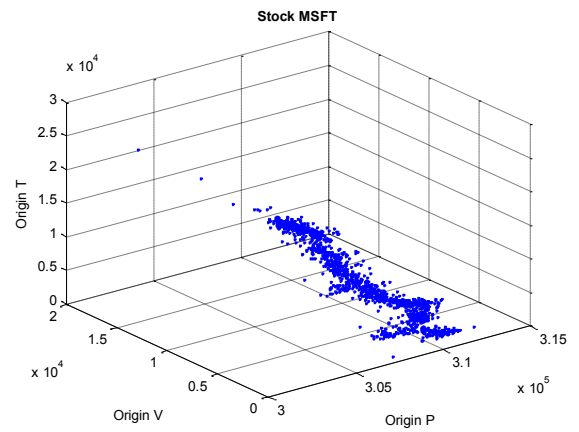


(b)

Figure 6 legitimate and manipulated trading cases in Apple stock, (a) transformed data, (b) original data.

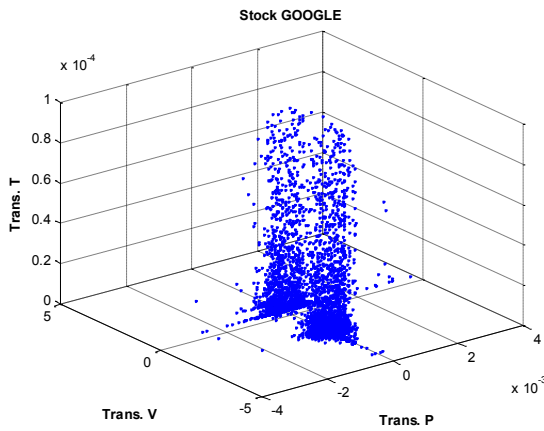


(a)

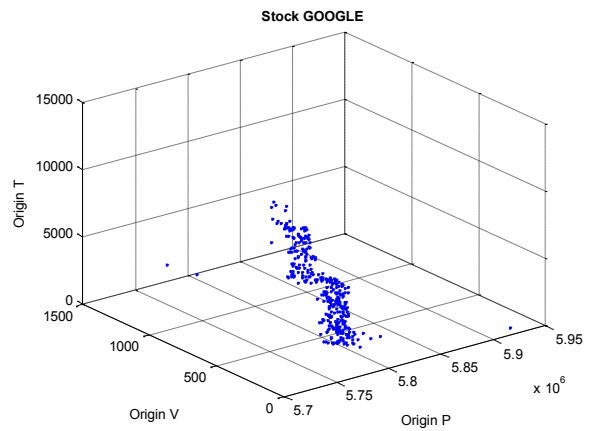


(b)

Figure 7 legitimate trading in Microsoft stock, (a) showing transformed data; (b) showing original data.



(a)



(b)

Figure 8 legitimate trading in Google stock, (a) transformed data; (b) original data.

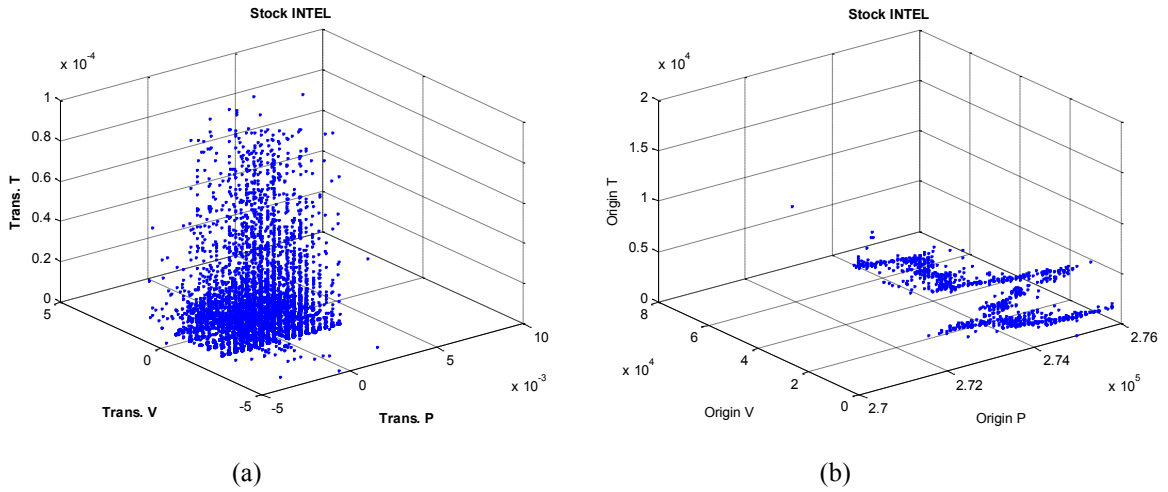


Figure 9 legitimate trading in Intel stock, (a) transformed data; (b) original data.

3.2 Static detection model and experimental evaluation

Describing the shape of a cluster of normalities is widely referred as novelty detection (Ding, Li, Belatreche, & Maguire, 2014), where abnormalities are identified through comparison with the model of normality. The abundance of legitimate trading data and relative lack of manipulated cases (which occurs because regulations forbid the publication of actual illegitimate cases) mean that the novelty detection paradigm is particularly suited to the identification of market manipulation. Therefore, compared with the existing studies, the model in this study has no assumption. The anomaly detection model is directly applied on the trading data rather than any market variables.

Novelty detection based on k-Nearest Neighbour (kNN) is one of the most straightforward non-parametric approaches. It assumes that normal data cluster in k nearest neighbourhoods, while anomalies are found at a distance from these. Another approach to novelty detection lies in one-class support vector machine (OCSVM), which generates a description of the boundary of normality, via the support vectors (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson., 2001) (Hayton, et al., 2007). When applied to the detection of price manipulation, OCSVM can be used to measure the novelty in trading activity by learning a representation of legitimate orders. In this paper, the authors examine the problem of detecting price manipulation, using these two learning machine models on the transformed time series. This application is defined as a static model and its structure is shown in Figure 10. The machine learning models are also applied directly to the original market data, to compare performance with the static model.

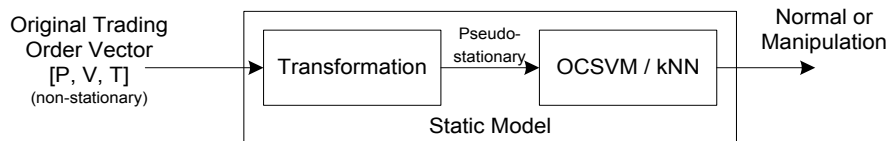


Figure 10 A static model for the detection of price manipulation.

When the novelty detection methods are applied to the detection of price manipulation, a set of normal data vectors $\mathbf{S} = \{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_t\}$ is collected as the training dataset. The vector \mathbf{o}_t is from either (a) the vector of the original market data $[p_{o_i}, v_{o_i}, t_{o_i}]'$ (o_i : a buy or sell order), or (b) the transformed data $p_{o_i}^n, v_{o_i}^n, t_{o_i}^n$, as calculated by equations (1) to (10). The illustration of the original and the transformed data are shown previously from Figure 6 to Figure 9. For the purposes of

experimental evaluation, the OCSVM and kNN are applied to the above-mentioned datasets of four stocks, Apple, Google, Intel and Microsoft, with additional three more liquid stocks, ARM, Barclay, and Vodafone. The seven datasets include five trading days' tick data from 11 to 15 June 2012 with one more day of 10 July 2013 of Apple stock to include the reported real manipulative case. Therefore the datasets comprise more than 2,000,000 total data points for each stock.

The five-fold cross-validation (Ron, 1995) is used to generalize the model parameters for avoiding over-fitting in training the detection models. In the validation, the datasets are partitioned into five folders, four of which are used as training and one is used as testing in each round of the validation and the validation results are averaged over the rounds.

Usually, benchmarks of both normality and abnormality (or novelty) are needed to evaluate a detection model. However it is relevant to this study that, due to the lack of genuine market cases of manipulation, the financial industry has collectively accepted an approach whereby reported cases of manipulation are studied and their characteristic patterns reported in the context of other financial data, so that exploratory manipulation cases may be synthesised (Palshikar & Apte, 2008). Academia also accepts the use of synthesised exploratory financial data for evaluation purposes, when genuine data is difficult to obtain. Thus, in this study, the experimental evaluations are designed in two groups: group 1 contains the evaluation on the real manipulation cases and group 2 contains the evaluation based on synthesized manipulation cases. Since the only manipulative cases we can collect are the cases on Apple stock reported by Nanex (Nanex, Incredible, Blatant Manipulation in Apple Stock, 2013), we test the models only using data of Apple stock in group 1 evaluation, which therefore shows us specific result of detecting real manipulation cases. The manipulation data obtained from Nanex contains 104 manipulative orders occurred on 10 July 2013. In group 2, however, the manipulation cases are synthesised and injected into a financial time series data, while the normal statistical features (such as the mean, variance and volatility) are maintained. All of the experimental evaluations in the remaining sections follow this synthetic case injection method. The generation of synthetic data is based on the (few) reported cases of genuine market price manipulation, which may be summarised as representing the two primary forms of price manipulation discussed in Section 3, namely spoofing trading and quote stuffing, the characteristics of which may be defined as:

- Spoofing trading: orders are of a size at least double that of the previous day's average order size, and prices are at least 6bps outside the current bid-ask spread. Cancellation time is around 30 minutes.
- Quote stuffing: orders are of a regular size, with quotes being 627bps higher or lower than the current bid or ask price. Approximately 6.2 seconds to cancellation (discussion in Section 3 refers).

In the group 2 evaluations, the synthesized cases of price manipulation are randomly injected into the corresponding order records, which create a mixture of both "normal" (i.e. in this context, legitimate) and "abnormal" patterns in the testing datasets. To ensure comprehensive assessment, 6000 synthesised cases are added to each dataset, these comprising 3000 examples of each type of manipulative strategy. For the Apple stock, the reported real cases are also used in the evaluation process. Therefore,

In the experiments, two open source libraries, LIBSVM (Chang & Lin, 2011) and DDTTool (D.M.J., 2013), are used as the implementation of the OCSVM and kNN, respectively. To assure stable and optimised results, the model parameters, the Gaussian kernel and the k value for k-NN, are determined by five-fold cross-validation (Ron, 1995). The five-fold cross-validation is to partition the training dataset into five equal sized subsets, of which each single subset is retained as validation data and the remaining four subsets are used as training data. The cross-validation process is then repeated five times (the

fold), with each of the five subsets used exactly once as the validation set. The five results from the folds are then averaged to produce a single estimation.

Since group 1 only contains a few real manipulative cases, the basic accuracy measures recall (sensitivity) and precision are used as performance measures for a clear comparison of the selected models. The performance measure in group 2 is based upon the Receiver Operating Characteristic (ROC). The recall (sensitivity), precision and ROC are calculated according to the confusion matrix, using the expressions given in Table 2.

Table 2 Confusion Matrix

	Normal cases	Manipulation cases
Test Outcome Normal	true positive (TP)	false positive (FP)
Test Outcome Manipulation	false negative (FN)	true negative (TN)

The precision is defined as $\left(\frac{TP}{TP+FP}\right)$ while the recall (sensitivity) is defined as the true positive rate $\left(\frac{TP}{TP+FN}\right)$. The ROC curve is popular for use in the evaluation and comparison of binary classifiers, and plots the true positive rate $\left(\frac{TP}{TP+FN}\right)$ against the false positive rate $\left(\frac{FP}{FP+TN}\right)$, with the discrimination threshold of the binary classifier being varied. To assess the overall performance of a novelty detector, the area under the ROC curve should be measured - this area is known as the AUC. The maximum AUC value is 1, and larger AUC values suggest better quality classification.

Table 3 shows the testing results of group 1. It's clearly that after data transformation, even the simple algorithms, OCSVM and kNN, can achieve significant better performance. OCSVM achieving recall of 0.9933 shows the low false negative, which means almost all normal cases are classified as normal, while achieving the precision of 1, which shows that all manipulated cases are detected as the manipulation. This results show that genuine cases of manipulation in the transformed data for the Apple dataset can be successfully detected by both models.

Table 3 Group 1 test results based on real manipulation cases only

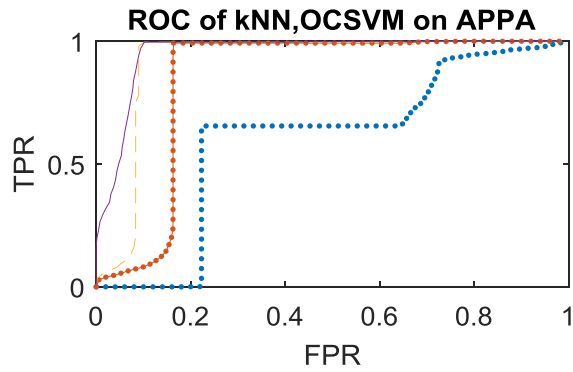
	Data	Original dataset		Transformed dataset	
		Recall	Precision	Recall	Precision
OCSVM	APPA	0.9117	0.8749	0.9933	1.0000
KNN		0.8902	0.8528	0.9904	0.9910

Figure 11 shows the ROC curves for the two models of the seven stock datasets (both original and transformed), with 6000 novelties injected into each dataset. Table 4 shows the summarised calculated AUC values. Both OCSVM and kNN models obtained higher AUC on all transformed datasets than the original data, and even the smallest of these (an AUC value of 0.870426 to kNN on the transformed Vodafone dataset) can be considered a good performance. OCSVM and kNN achieve the highest performance improvement on the Apple and Intel dataset after transformation. In contrast, there are relatively small improvements using kNN in the Google (4.22%) and OCSVM in the ARM (8.52%) datasets. Clearly, both detection methods achieved significantly better results when the transformation method was applied than they did with the original data, in terms of the AUC values in Table 4. However, the improvements do not achieve a unique value, and vary across datasets. The success results from the ability of the OCSVM and kNN models to effectively model the boundaries of normal behaviour clusters; as Figure 6(a), Figure 7(a), Figure 8(a), and Figure 9(a) show, the cluster shapes of the transformed data are

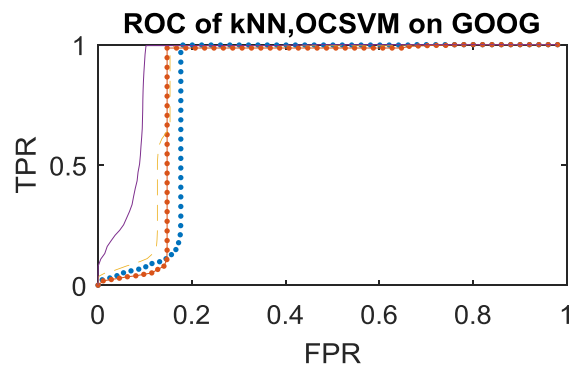
considerably more regular than those of the initial data as shown in Figure 6(b), Figure 7(b), Figure 8(b), and Figure 9(b). The authors believe that this results at least in part from the proposed transformation procedure, which helps to generate the “pseudo stationary” cluster shapes. This transformation, at least from a machine learning perspective, pre-processes the original data in such a way that features, which help to separate data into the classes of normal and abnormal (manipulative) behaviour, are extracted. Interestingly, the OCSVM performs at a higher level due to its ability to provide a better description of the clusters of normal behaviour through support vectors that delineate the boundary more accurately.

Table 4 AUC values for OCSVM and kNN models on original and transformed dataset of seven stocks

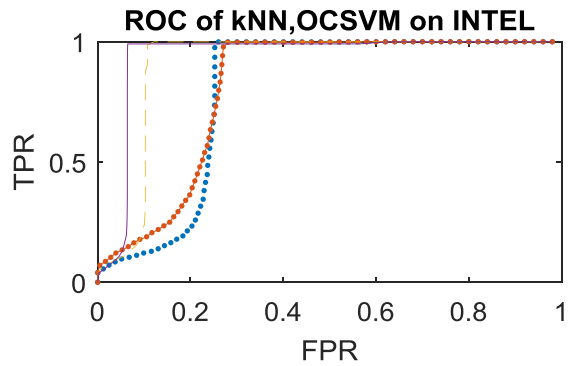
AUC value	Original dataset		Transformed dataset		Improvement	
	kNN	OCSVM	k-NN	OCSVM	kNN	OCSVM
Google	0.836166	0.851331	0.871484	0.928321	4.22%	9.04%
Apple	0.60055	0.846419	0.920497	0.954983	53.28%	12.83%
Intel	0.793418	0.809736	0.908378	0.937078	14.49%	15.73%
Microsoft	0.838143	0.869631	0.954004	0.990927	13.82%	13.95%
ARM	0.755739	0.84861	0.894391	0.920933	18.35%	8.52%
Barcelay	0.806637	0.840949	0.903739	0.924911	12.04%	9.98%
Vodafone	0.763688	0.812543	0.870426	0.906201	13.98%	11.53%



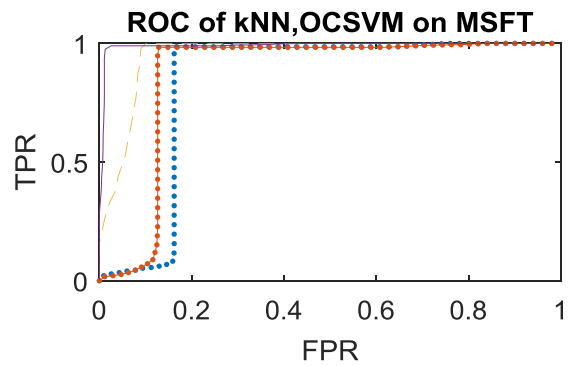
(a)



(b)



(c)



(d)

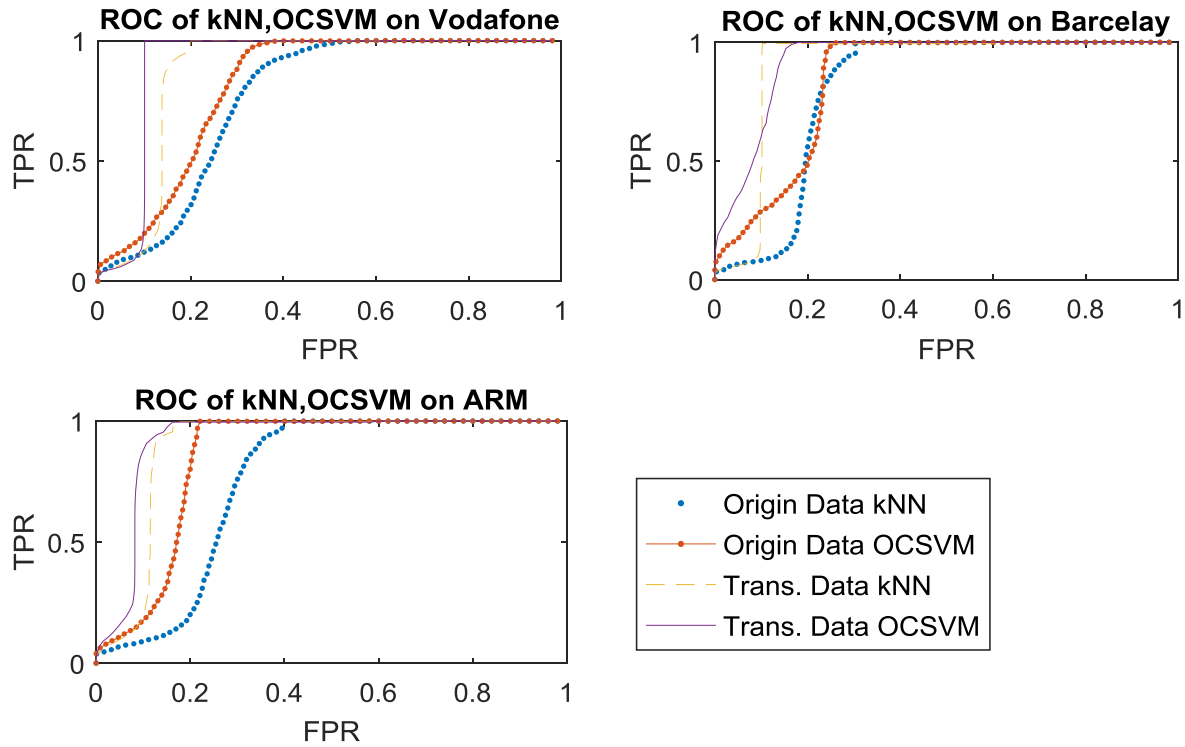


Figure 11 ROC curves for two models, on seven stock datasets.

4. The dynamic model

4.1 Feature extraction

The detection system proposed in this section possesses both a “feature extraction module” and a “detection model”, and the model is trained through the features extracted (by the “feature extraction module”) from the equity bid/ask price. The three patterns shown in Figure 2, which characterise typical price manipulation activity, fall into two categories, namely remarkable pulse (Figure 2(b)) and short-term minor fluctuations (sawtooth and square as shown in Figure 2(a) and Figure 2(c)). It is possible to extract the intuitive feature of these patterns as the rate of price (bid/ask) change - this is defined as the first order derivative of $P(t)$ with respect to t ,

$$\frac{dP(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{P(t+\Delta t) - P(t)}{\Delta t},$$

where Δt is the time interval between changes in the ask/bid prices.

Signal composition is often achieved using empirical mode decomposition (EMD), continuous wavelet transforms (CWT) and synchrosqueezed transform (SST). However, there are some limitations involved: EMD has a low frequency resolution (Wu & Huang, 2004) and its obtainable frequency resolution depends on a critical frequency limit (Feldman, 2009). Any signal can be mapped, using CWT, to a set of base functions that are gathered using dilation and translation of a mother wavelet, and CWT can achieve decomposition of the signals in different frequency bands and at different points in time (Daubechies, Ten Lectures on Wavelets, 1992). The time-frequency localisation feature of CWT differentiates it from Fournier transform, which represents the frequency content of a signal without any time information. In CWT, the frequency content is described locally in time, by scale and shift parameters (Daubechies, Ten Lectures on Wavelets, 1992), and SST extends CWT by re-allocating

the wavelet coefficients. It does this by further calculating the first derivative (instantaneous frequency) of those coefficients - this is referred to as synchrosqueezing (Daubechies, Lu, & Wu, Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool, 2011). Synchrosqueezed transform has been used as a band pass filter for removing high frequency components from a signal (Ahrabian, Took, & Mandic, 2012). In this study, we apply wavelet transform as a feature extraction approach to retrieve and analyse the high-frequency elements of the bid/ask prices. This approach is used because wavelet transform has been frequently and successfully used in financial contexts (Daubechies, Ten Lectures on Wavelets, 1992) (Caetano & Yoneyama, 2009).

Wavelet decomposition of a signal generates levels of approximation and detailed coefficients; the approximation coefficient vector illustrates the low-frequency features of a signal, while the detail coefficient vector represents the high frequency component. In this study, the detail coefficient thresholding method, which is usually used to remove noise from financial signals (Haven, Liu, & Shen, 2012) is applied inversely to set those wavelet coefficients that fall outside the threshold to zero. This facilitates retrieval of the high frequency components, while the low-frequency equilibrium component is removed, thus:

$$T_{m,n} = \begin{cases} T_{m,n} & \text{if } |T_{m,n}| \leq \lambda \\ 0 & \text{if } |T_{m,n}| > \lambda \end{cases} \quad (8)$$

Where $T_{m,n}$ is the detail coefficient, m, n are scale and shifting parameters of the wavelet function, respectively, and λ is the fixed threshold. The wavelet and inverse thresholding procedure occurs in three distinct steps:

- 1) Calculation of the wavelet coefficient vectors $\{T_{m,n}, T_{m-1,n}, \dots, T_{1,n}\}$, using discrete wavelet transform.
- 2) Selection of a threshold λ and filtration of the coefficient vectors through the hard threshold as given in equation (8).
- 3) Recomposition of the data using the filtered wavelet coefficients.

In this study, the Symlet wavelet is applied in order to extract the high frequency components, which are denoted as $\hat{P}(t)$, of an equity's price, denoted as $P(t)$. Short-term, small fluctuation is associated only with the high frequency elements, therefore $\hat{P}(t)$ is also used to denote that small, short-term oscillation. Both the first-order derivatives of the original price, given as $P(t)$, as well as the short-term fluctuation $\hat{P}(t)$ regarding time, are extracted as being characteristic of patterns of price manipulation. This is done through calculation of the gradient of $P(t)$ and $\hat{P}(t)$ - the method chosen to achieve this is finite difference, and specifically its central difference form, since this is of established value in financial contexts (Hull, 2011), thus:

$$\frac{df(t)}{dt} \approx \frac{f(t+\frac{\Delta t}{2}) - f(t-\frac{\Delta t}{2})}{\Delta t},$$

with $f(t)$ representing $P(t)$ and $\hat{P}(t)$, and Δt the time interval elapsing between changes to ask/bid prices. As has been discussed in Section **Error! Reference source not found.**, there are three recognised patterns of manipulation that can be visualised via significant pulse and short-term, minor, oscillations, with the former being associated with the original price changed rate, $\frac{dP(t)}{dt}$, and the latter chiefly linked to high-frequency fluctuations, $\frac{d\hat{P}(t)}{dt}$. It is generally the case that financial data consist chiefly of a low-frequency trend component, over which are superimposed various other frequencies [33]. From a financial point of view, $\frac{dP(t)}{dt}$ and $\frac{d\hat{P}(t)}{dt}$ are representative of features occurring in the original and de-trended prices (i.e. those with the low-frequency component removed), and thus associated with various known manipulation tactics, as has been discussed. Such patterns emerge in the presence of manipulative activities regardless of the price manipulation strategy employed; at the same time, $P(t)$ and $\hat{P}(t)$ are associated with unusual pulses and large, and square patterns. Therefore, all four values - $P(t)$,

$\hat{P}(t), \frac{dP(t)}{dt}$ and $\frac{d\hat{P}(t)}{dt}$ - must be included as features, in order to assure coverage of all possible manipulation contexts, including sawtooth, pulse and square patterns. The feature extraction module includes a wavelet filter and gradient calculator, and its structure is illustrated in Figure 12.

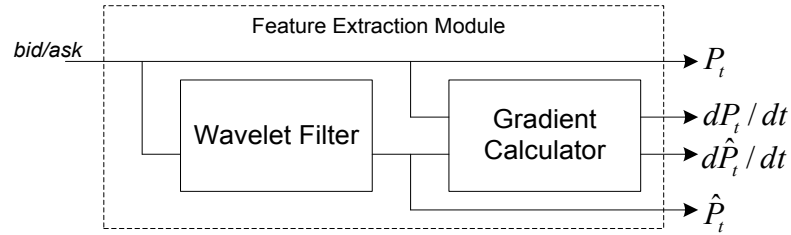


Figure 12 Structure of the feature extraction module.

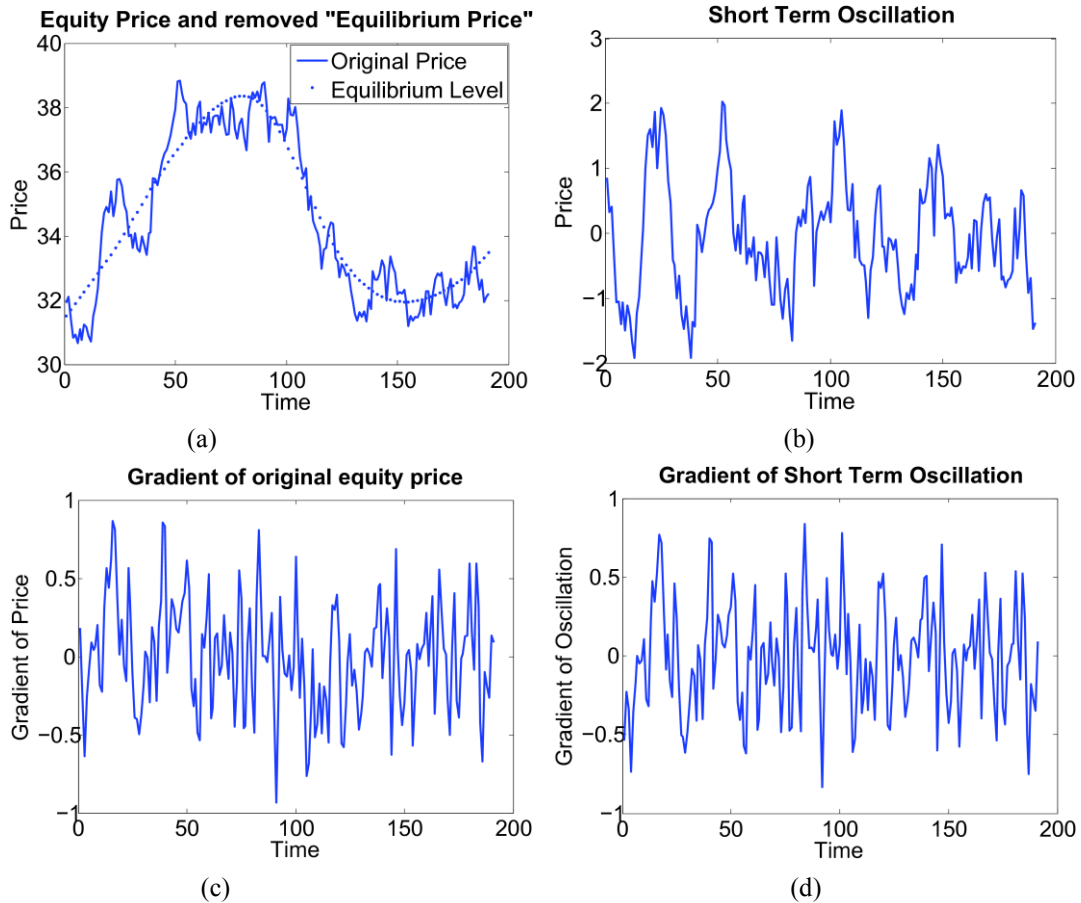


Figure 13 An example of the equity price filtered by the wavelet: (a) equity price and the equilibrium price to be removed, (b) short-term oscillation, (c) gradient of the original equity price (d) gradient of short-term oscillation.

Figure 13 presents an example of the feature extraction module's output: the bid/ask price $P(t)$ is filtered by the Symlet wavelet at level 8, while the smoothed equilibrium values are shown as a dotted line in Figure 13(a), and the extracted short-term oscillation $\hat{P}(t)$ is represented in Figure 13(b). Figure 13(c) and Figure 13(d) show the gradients of $P(t)$ and $\hat{P}(t)$, respectively.

4.2 The detection model

Financial data are generally considered to be inherently non-stationary, and so it is accepted that the statistical properties of such data, for example mean and variance, vary over time. The dynamic model is designed to track the trading activities in a contextual relationship since the sequence of the trading action may lead to a manipulative behaviour. Therefore the transformation method, proposed in Section 3.1 to remove the direct contextual link of sequential actions, is not applicable in dynamic model. These data variations result from a range of factors, including business and economic cycles (which apply chiefly over the long term) and demand/supply microstructures (which apply largely over the short term). That makes it particularly interesting when unusual activities are associated with a lack of stationarity in the extracted features, i.e. when the distribution of those extracted features alters during its existence. This can be seen as an outcome of various irregular trading activities shown in Figure 2, thus the detection model must have the capacity not only to capture the distributions of the extracted features, but also to track shifts in them. Therefore, distribution, followed by sequential data modelling methods, is suggested for the detection model in this section.

4.2.1 The Gaussian mixture model and hidden Markov model

An established method for modelling the probability density function (PDF) of a variable is to approximate its (unknown) density using a Gaussian mixture model (GMM) (Bishop, 2006). A GMM is a weighted sum of M component Gaussian densities as:

$$p(\mathbf{x}) = \sum_{i=1}^M w_i g(\mathbf{x}|\mu_i, \Sigma_i),$$

where \mathbf{x} is a D -dimension continuous-valued data vector, $w_i, i = 1, \dots, M$, are mixture weights, and $g(\mathbf{x}|\mu_i, \Sigma_i), i = 1, \dots, M$, are component Gaussian densities, wherein μ_i and Σ_i are the mean and covariance matrix for each component respectively. Each component density is a D -variate Gaussian function. Under traditional GMM, expectation maximisation is run according to heuristic trials, for the number of Gaussian components. In this study, a Dirichlet process GMM (DPGMM) (Neal, 2000) is applied, providing a method, with statistical integrity, that generates the number of GMM components. In most respects DPGMM resembles traditional GMM, however in the Dirichlet process the component parameters are given as a prior probability and the Gibbs sampling from the posterior probabilities (Neal, 2000). In the interests of detecting price manipulation, the PDFs of the extracted features of an equity price are modelled using DPGMM. Testing can be used to detect the observed changes in the features, in particular discerning which component of the mixture generates a given value. The outcome is a simplified view of the state of the features. The time-related changes in distribution are related, in terms of probability, to the changes in observed features and it is the sequences of observed features - rather than any single value - that suggests (or not) attempts to manipulate. A hidden Markov Model (HMM), containing two sets of states, observable feature states and hidden mixture component states, can be used to model such cases. These HMM contents are modelled on a standard Markov process, on the assumption that they depend solely on previous states. Usually, application of an HMM can be categorised as belonging to one of two classes; either (a) signature modelling, where the detection model is learned from the activities to be recognised and the activities matching the model are reported as detection outcomes. Examples of this include the coupled behaviour detection in (Cao, Ou, & Yu, 2012), industry process monitoring in (Wang, Mehrabi, & Kannatey-Asibu, 2002) and (Stephen, 2009), the intrusion detected in (Yeung & Ding, 2003) and video surveillance in (Guo, Miao, Zhang, Shen, & Wang, 2012). The other category of HMM application is (b) anomaly detection, whereby a model of normality is learned solely from the normal activity, and cases tested against the model, with a pre-determined threshold, are declared to be

anomalies (Yeung & Ding, 2003). In such cases, it is very important that the hidden states be identified and assigned; thereby fulfilling the Markov assumption and allowing the parameter matrices to generate different observed outputs.

As has been discussed in the previous section, the detection of behaviour intended to manipulate market prices can be framed as an anomaly detection problem, which is to identify new or unknown data patterns that have not previously been clarified during training process (manipulation data has not been disclosed in accordance with industry regulations). Taking the anomaly detection approach requires that the normal bid/ask price patterns are modelled and that an “alarm” be triggered when market activity deviates from those normal patterns. However, when the approach is applied to the detection of price manipulative activity, some problems arise. These include:

- Anomaly detection with HMM generally lacks the ability to recognise the type of anomaly, and the probability density related to that type.
- The PDFs of equity price evolve as a result of the financial time series being inherently non-stationary.

In order to overcome these obstacles, an improved and adapted HMM, with anomaly states, has been developed and is presented in the following sections.

4.2.2 An HMM-based dynamic model (DM)

For a bid/ask price of an equity with T points, the module described in Figure 12 is used to extract four features. Letting \mathbf{F}_t , $t = 1, \dots, T$, be the feature vector at the t -th point in time, the pre-processed four features at time t , $\mathbf{F}_t = [\mathbf{F}_t^o, \mathbf{F}_t^{og}, \mathbf{F}_t^s, \mathbf{F}_t^{sg}]$, are described as:

- 1) Original price: \mathbf{F}_t^o .
- 2) The gradient of the of original price: \mathbf{F}_t^{og} .
- 3) Short-term fluctuation: \mathbf{F}_t^s .
- 4) The gradient of the short-term fluctuation: \mathbf{F}_t^{sg} .

The GMM learns the PDFs of these four features separately, and because the PDF of each feature may have multiple Gaussian components, testing is used to identify changes in the observation distribution, and to determine which element of the PDF mixture is most likely to have resulted in a given value. The various elements of that mixture are designated hidden states of the observed equity price, and this provides a simplified state view of the time series data.

Anomaly thresholds, which separate and distinguish the normal from the novel, are set for the learned PDFs according to the minimum data-likelihood value. That value is taken from the Smart Group's industry reference detection algorithm (Aitken, Harris, & Ji, 2009), and, in particular, is the 99% cumulative distribution cut-off. In other words, data are accepted as normal when $P(\mathbf{F}_t) \geq 99\%$; this is in order to identify the 0.5% most and least frequent values for each feature. The same heuristic method is applied in a one-class support vector machine (OCSVM) (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson., 2001). Here, the detection boundary is generally set to include most (for example, 99%) but not all training data; this is done to avoid a high mis-detection rate. In this context, the 1% outlying values are not taken as abnormalities, rather they are used to generate the dummy anomaly states for the HMM.

Figure 14 illustrates the PDFs of four features pertaining to the time series data - these are learned by the GMM. Two thresholds, which are shown as dotted lines, are set to 0.5% and 99.5% cumulative probability for each PDF, thus separating the areas for normal and anomalous events. Figure 14(a) illustrates the PDF of an equity price, shown as small circles, with

three Gaussian components - these are shown as a solid line. The dotted lines show the thresholds, which are set in such a way that the 99% cumulative distribution of the equity prices is accepted as the area of normality, leaving the maximal and minimal 0.5% to represent anomalies. Thus the PDF of the original price can be divided into four sections, three of these correspond to three Gaussian components in the area of normality and one to the area of anomaly (i.e. to both minimal and maximal values). The four sections can then be seen as four hidden states for the equity price feature, and this is illustrated in Figure 15. Defining those hidden states as $\{f_{j_o}^o, j_o = 1, \dots, 4\}$, leads to the extracted feature $F_t^o \in \{f_{j_o}^o, j_o = 1, \dots, 4\}$.

In a similar way, two hidden states, one anomalous and one normal, are generated for the other three one-component PDFs, as can be seen in Figure 14(b)-14(d). If $\{f_{j_{og}}^{og}, j_{og} = 1, 2\}$, $\{f_{j_s}^s, j_s = 1, 2\}$ and $\{f_{j_{sg}}^{sg}, j_{sg} = 1, 2\}$ are defined as the hidden states of price gradient, short-term oscillation and oscillation gradients, respectively, then $F_t^{og} \in \{f_{j_{og}}^{og}\}$, $F_t^s \in \{f_{j_s}^s\}$ and $F_t^{sg} \in \{f_{j_{sg}}^{sg}\}$, where $j_{og}, j_s, j_{sg} = \{1, 2\}$.

The next step is to combine the hidden states of all four features, as the quantised hidden states of the dynamic model, which gives $4 \times 2 \times 2 \times 2 = 32$ hidden states. Thus quantised, the 32-state rendering of the observed bid/ask price and be shown as $\mathbf{S} = \{S_i\}$; where $S_i = \{f_{j_o}^o, f_{j_{og}}^{og}, f_{j_s}^s, f_{j_{sg}}^{sg}\}$; \mathbf{S} is a collection of the feature states $S_i, i \in \{1, \dots, 32\}$, $j_o \in \{1, \dots, 4\}$ and $j_{og}, j_s, j_{sg} \in \{1, 2\}$. Each hidden state contains its own unique combination of the feature status. That state which has all features quantified as falling within the area of normality is considered to be a completely normal hidden state, while others - which by definition have at least one feature that falls outside that region and is thus defined as anomalous - are considered to be hidden anomaly states. This means that just three of the 32 states are normal - the remaining 29 are all anomalous.

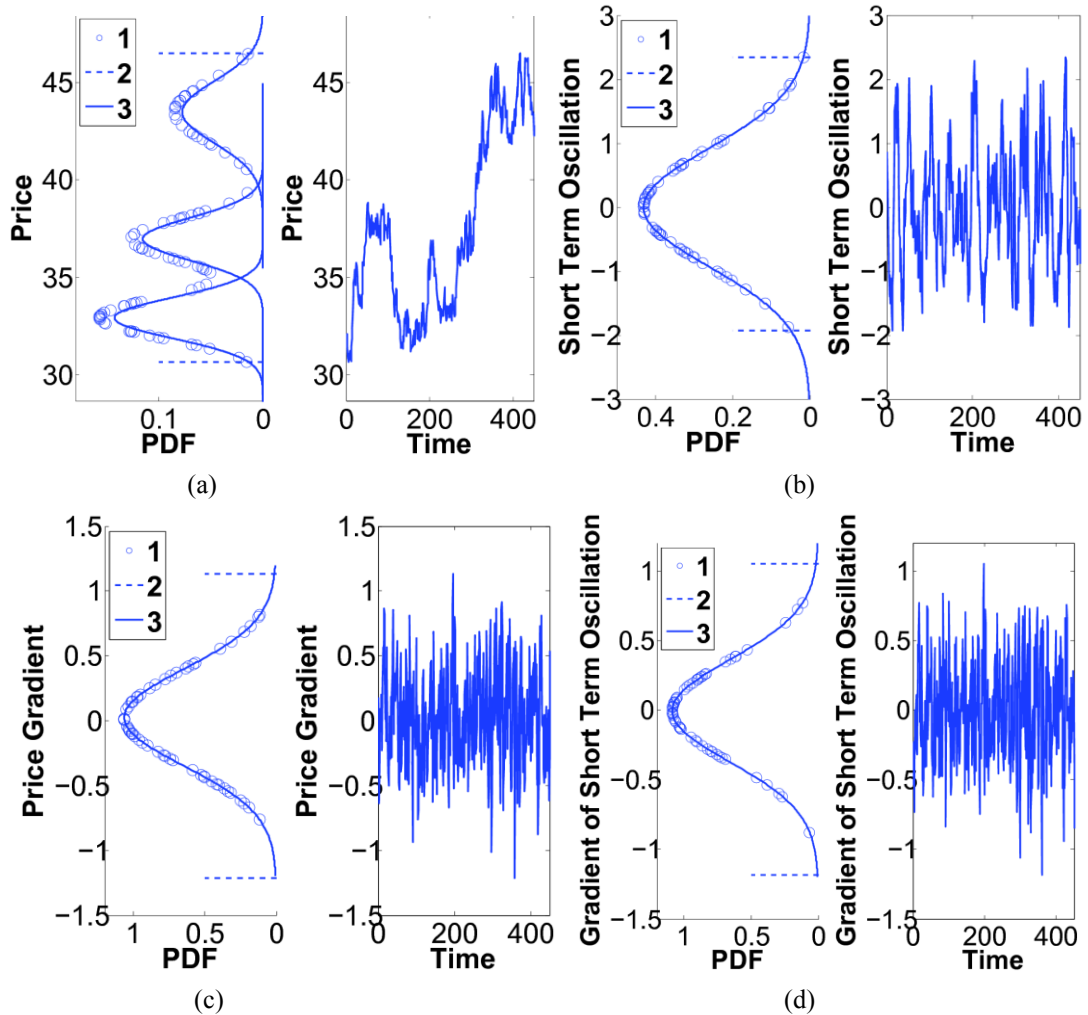


Figure 14 Examples of features learned by GMM. The PDF of the original equity price, the short-term fluctuation, gradient of the original equity price and the gradient of oscillation are all plotted. (Legends in the figure: 1: pdf. 2: Components of pdf. 3: thresholds) (a) equity price on the right and its PDF on the left; (b) short-term oscillation on the right and its PDF on the left; (c) the gradient of the original equity price on the right and its PDF on the left; (d) the gradient of short-term oscillation on the right and its PDF on the left.

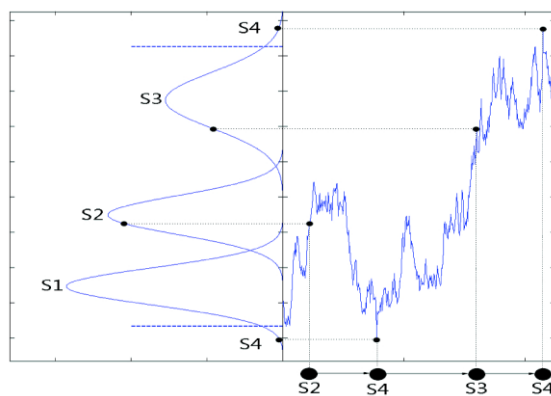


Figure 15 HMM with anomalies generated by a sequence of observations of variables. The state of normality is seen in S1, S2 and S3. The anomalous/abnormal state is S4.

The definitions given in the price manipulation market abuse directive (EU, Market Abuse, Amended Proposal for a Directive, 2012) and discussion in Section 2 of this paper, note that typical price manipulation tactics are associated with distinctive combinations of feature status. The various forms of price manipulation, such as quote stuffing (whereby order books are instantaneously swept) are accompanied by significant changes to the gradients of both short-term oscillations and the original price (f_{jsg}^{sg} and f_{jog}^{og}). In contrast, another form such manipulation, spoofing trading, is linked with significant changes to the original price or short-term oscillations (f_{jo}^o and f_{js}^s), but these do not necessarily exhibit any relationship with the gradients. All of this means that the 29 anomalous states can be combined into three manipulation categories, according to the characteristics of their feature status. The manipulation categories are quote stuffing, ramping and other anomalies. Those states in which the only anomaly lies in the gradient features are categorised under quote stuffing, while those with anomalous price features alone (in both the original data and the oscillation) are classified under ramping. The state termed “other anomalies” comprises all of the cases in which all features are anomalous. Thus when classification is complete, the initial 32 states are simplified into a six-state view of the observed bid/ask price, comprising three normal states and three anomalous states, all corresponding to primary price manipulation tactics. This allows the six hidden states to provide a complete spectrum, and to present an efficient means of modelling the bid/ask price alterations that are triggered by trading actions. In this context, the observed temporal dynamics on each feature are not recounted; rather just the transfers in hidden states are shown.

The fundamental property of this dynamic model (DM), which is a result of its descent from traditional HMM, comprises the probabilities given by the Viterbi algorithms, which determines the most likely sequence of underlying hidden states capable of generating the observations sequences. Taken alongside the probability of an observed sequence, the DM is thus able to give a measure of the identification of a specific manipulation approach, according to the states and features.

4.2.3 *The adaptive dynamic model (ADM)*

The financial time series is non-stationary, which means that the mean and variance of the PDFs of the equity price may be subject to variation over time. In order to mitigate the non-stationarity, the DM is augmented with an adaptive mechanism, thus it becomes (and is henceforth referred to as) an adaptive dynamic model (ADM). In the ADM, the model is trained using data generated at previous, specified time points, which are defined as a sliding window of length w . During the process of manipulation detection, the window slides forward, in order to maintain the closest w data points, and the ADM is updated if any meaningful differences between the w data points in the current sliding window, and prior training data, are found.

The t-test is used to identify deviation between the two data sequences, and a clear example of the non-stationary nature of the equity price can be seen in Figure 16. As has been discussed, a process is deemed strongly stationary if its distribution remains unchanged under time shifts (Grimmett & D. Stirzaker, 2001). If the sliding window length is set to $w = 1670$, the variant PDF of the time series in three windows $P_t, t = 1, \dots, 1670$, $P_t, t = 1671, \dots, 3240$ and $P_t, t = 3241, \dots, 5000$ (which are shown as the three parts of Figure 16) are detected as significant changes in the statistical features of the time series, by t-test under the significance level of 1%. Further, a t-test module is used as a post-processing block of the ADM, to identify the statistical differences between the updated data sequence in the sliding window, and the (prior) training data sequence. A widely-used significance level, 1%, is used to balance the trade-off between computational load and performance.

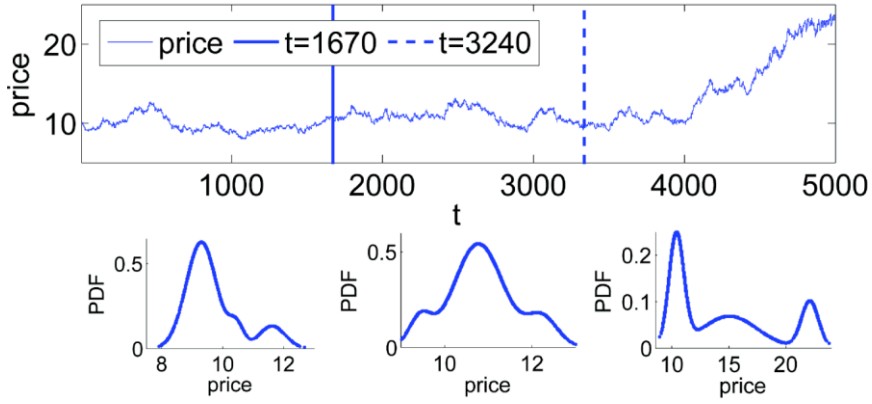


Figure 16 Non-stationary changes illustrated with different PDFs.

Figure 17 describes the adaptive mechanism for this ADM. At first, the model is trained by w data points in the training window, $F_t, t \in [t_{s_1}, t_{e_1}]$, where t_{s_1} and t_{e_1} are the starting and ending time points, respectively. Then, the model detects a new incoming data point - if this is assessed as being normal, it is updated into the update window $F_t, t \in [t_{s_2}, t_{e_2}]$ for statistical difference testing. Figure 17 shows how the data in the sliding window are tested every time that window is updated, however the model itself may be retrained after working time period t . The training window is set to a single day, because the intention is to detect intra-day price manipulation behaviours.

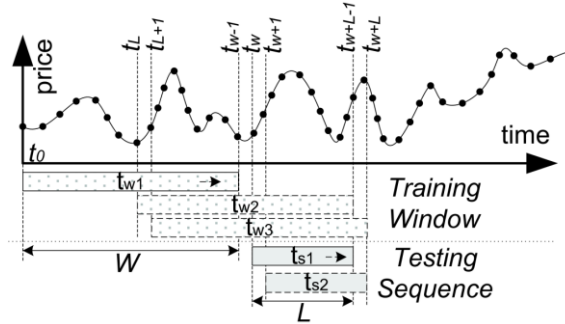


Figure 17 The training and testing mechanism used in price manipulation detection.

4.2.4 The detection algorithm

The steps given in Algorithm 1 summarise the steps taken when using the ADM to detect price manipulation. The initial states of the training window and testing sequence are given in Figure 17, which also shows the sliding mechanism of each as the detection of price manipulation is carried out. In the case shown in Figure 17, the ADM model was trained with data in $t_{w_1} = P_{t_i}^E, t_i \in [0, w - 1]$ and the first testing sequence was denoted as $t_{s_1} = P_{t_i}^E, t_i \in [t_w, t_{w+L-1}]$. If detection assesses behaviours as being normal, the adaptive mechanism then examines the significant changes between data in the updated training window, $t_{w_2} = P_{t_i}^E, t_i \in [t_L, t_{w+L-1}]$, and that of the original training window, t_{w_1} . If there is no need for retraining, then the training window slides forward as t_{w_3} and a new testing sequence is formed as t_{s_2} to include the new data point at t_{w+L} but exclude the point at t_w . If the model does not detect any anomaly, the data in the updated training window, $t_{w_3} = P_{t_i}^E, t_i \in [t_{L+1}, t_{w+L}]$ and

the original training window t_{w_1} are examined for adaption. This process of examination/detection and updating is carried out on all incoming data streams.

4.3 Experimental evaluation

Labelled benchmarks of both normal and anomalous cases are usually used in the evaluation of a detection model, however given the lack of genuine market data of manipulative cases, a slightly adapted approach is taken in this case. In order to evaluate the proposed detection model, as per the approach described in Section 3, the characteristics of manipulated cases have been studied and synthetically generated, then injected into the financial time series data. Meanwhile, statistical features such as mean, variance and volatility, are all maintained.

4.3.1 The experimental setup

In this paper, experimental evaluation is conducted using genuine market data from seven representative stocks, namely Google, Microsoft, Intel and Apple from the NASDAQ (with data being obtained via the LOBSTER project (Berlin, 2013)) and ARM, Barclays and Vodafone from the London Stock Exchange. These datasets were chosen because they have relatively high trading volumes and price volatility, which are elements likely to induce manipulation across exchanges (Cumming, F. Zhan, & Aitken, 2013) - thus they are at greater risk of manipulation than many others. The dataset covers tick data from five trading days, namely 11-15 June 2012 inclusive, and comprises more than 2,000,000 total data records (including order submission, modification, execution, and cancellation) for each stock. In addition, the dataset of Apple stock also include the tick data on 10 July 2013 to include the reported manipulative case. The datasets have been examined by the authors' expert financial partners to ensure that none of them are related to any cases of manipulation that have been reported to the regulatory authorities. Furthermore, a popular and established statistical test (Tse, Lin, & Vincent, 2012) has been applied to the datasets to target intra-day volatility anomalies, and according to both manual examination and testing: there is no discernible manipulation or volatility spikes in any of the data. The testing is defined as seeking an anomalous pattern, starting at time t if $\sigma_{t-s,t} \leq 2\bar{\sigma}$ and $\sigma_{t,t+s} > 2\bar{\sigma}$, where s is defined as one hour, σ_{t_1,t_2} is the volatility over the period (t_1, t_2) , and $\bar{\sigma}$ is the average intra-day volatility across the entire dataset. Figure 2 shows three typical manipulation patterns, which are reproduced in the datasets of equities using the original characteristics, which are 6.9-bps sawtooth patterns in 819 ms, 18.6-bps square wave in 0.1s and 800-bps pulse in 1s. These anomalies are then injected into the corresponding time series, which means that the test data

Algorithm 1 Price Manipulation Detection

- Step 1:** For the specific equity E , select a sliding window length w ; construct the *bid/ask* price time series P_t^E from t_0 to the current time point t_{w-1} : $P_{t_i}^E, t_i \in [t_0, \dots, t_{w-1}]$.
- Step 2:** Calculate the four feature sequences of $P_{t_i}^E$ using the wavelet and gradient approaches described above; obtain the features vector $\mathbf{F}_t = [\mathbf{F}_t^o, \mathbf{F}_t^{og}, \mathbf{F}_t^s, \mathbf{F}_t^{sg}]$.
- Step 3:** Construct the ADM detection model:
- 1) Calculate the PDF separately for the features, $F_t^o, F_t^{og}, F_t^s, F_t^{sg}$, using the GMM. Set the corresponding anomaly thresholds.
 - 2) Construct the sub-states for each of the attributes and then construct the hidden states by combining four attributes according to discussion in Section 4.2.2 and the illustration in Figure 17.
- Step 4:** Train the ADM model using the observed features and the constructed hidden states.
- Step 5:** If the testing sequence is not constructed: select a detection sequence length L ; the first testing sequence is then constructed as $P_{t_i}^E, t_i \in [t_w, t_{w+L-1}]$.
- 1) Calculate the four features of the sequence using the wavelet and gradient blocks.
 - 2) Feed the features into the ADM model for calculating the probability of specific states.
- Step 6:** If the state is identified as any anomaly state(s), the alert is triggered, along with identification of the possible manipulation type and the probability of that state being the type.
- Step 7:** If the state is declared to be among the normal states, the bid/ask price in sliding window t_L to t_{w+L-1} are fed into the adaptive mechanism for a model updating check. If model updating is needed, the algorithm flow goes to Step 1.
- Step 8:** If model updating is not needed, the testing sequence is afterwards updated as $P_{t_i}^E, t_i \in [t_{w+1}, t_{w+L}]$ and the algorithm flow goes to Step 5.
-

becomes a mix of both normal and anomalous patterns. The trained models are then applied to that mixed data, in order to identify patterns associated with anomaly or abnormality. This approach is of acknowledged value to the industry, particularly given the huge expense involved in isolating genuine cases of price manipulation. The adaptive mechanism has a sliding window set to a length of one day, and so for the genuine market data (that comprises the first test category), the data from 11 June 2012, which is the first day of trading for which data was collected, becomes the training dataset. This training dataset contains around 400000 points, which are subjected to five-fold cross-validation in order to train the detection model. The remaining data, from 12-15 June 2012 inclusive, are designated the dataset to be tested. For Apple stock It is important to ensure that the evaluation is comprehensive, and therefore multiple datasets are constructed using genuine market data. 6000 synthesised patterns are injected into each of the real stock data sets, with each type of anomaly having 2000 examples. In Figure 2, a single pattern is an example of one anomaly.

As is shown in the literature review of this paper, some generic computational models have already been applied to the problem of detecting manipulation; these include SVM, neural networks and logistic regression approaches (Aitken, Harris, & Ji, 2009) (Ögüt, Doğana, & Aktaş, 2009) (Diaz, Theodoulidis, & Sampaio, 2011). This work has informed the authors' decision to select three popular computational models as benchmarks for this model; those benchmarks being the OCSVM (Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson., 2001), k-nearest neighbour (k-NN) (Bishop, 2006) and GMM (Bishop, 2006). The benchmark models are applied on the same feature vector, namely $\mathbf{F}_t = [\mathbf{F}_t^o, \mathbf{F}_t^{og}, \mathbf{F}_t^s, \mathbf{F}_t^{sg}]$. In the authors' experiments, LIBSVM (Chang & Lin, 2011) and DDTTool (Tax, 2012) are used to implement the benchmark models and all the model parameters (specifically, kernel width for OCSVM, k value for k-NN and the number of components for GMM) are fine-tuned

through the application of five-fold cross-validation on each different dataset, to ensure the stability and quality of the results. Section 3 shows a confusion matrix; this underpins the performance evaluation of the proposed model. The ROC curve and AUC are used as performance measures for seven genuine stock datasets.

4.3.2 Results of experimental assessment

Figure 18 shows the ROC curves of four models, applied to seven datasets of genuine stock, with $3 \times 2000 = 6000$ manipulative data injected into each. To compute such a curve, the discrimination thresholds are set from 0.1 to 0.9, with an increment of 0.01 for each model. Table 5 shows the calculated AUC values for four models, summarised.

Table 5 The AUC of four detection models, applied to seven genuine stock datasets.

AUC value	ADM	OCSVM	k-NN	GMM
GOOG	0.9445	0.8317	0.8622	0.7692
MSFT	0.9329	0.8985	0.8183	0.7620
INTL	0.9497	0.8775	0.8648	0.8477
APPA	0.9189	0.8979	0.8193	0.7401
Barclay	0.9272	0.8373	0.8855	0.7492
ARM	0.9592	0.8792	0.8628	0.7334
Vodafone	0.9499	0.8469	0.8841	0.7949

The highest AUC of all seven real stock datasets was achieved using the ADM, the performance of which exceeded that of the benchmark models. The ADM's best AUC value appeared on the ARM data set (0.9592); that is 9.11%, 11.17% and 30.79% higher than the OCSVM (0.8792), k-NN (0.8628) and GMM (0.7334) models, respectively. The next best ADM value for AUC occurs on the Intel dataset (0.9497), which is in the region of 8-15% better than the results yielded by the other three models. The ADM's lowest AUC value was achieved on the Apple dataset (0.9189), yet even this is substantially higher than the performance of the other models, being 2.34%, 12.16% and 24.16% higher than the OCSVM (0.8979), kNN (0.8193) and GMM (0.7401) results, respectively. The difference in performance between the ADM and three other models in terms of AUC values are calculated on seven genuine stock datasets:

$$\frac{AUC_{ADM} - AUC_{others}}{AUC_{ADM}}$$

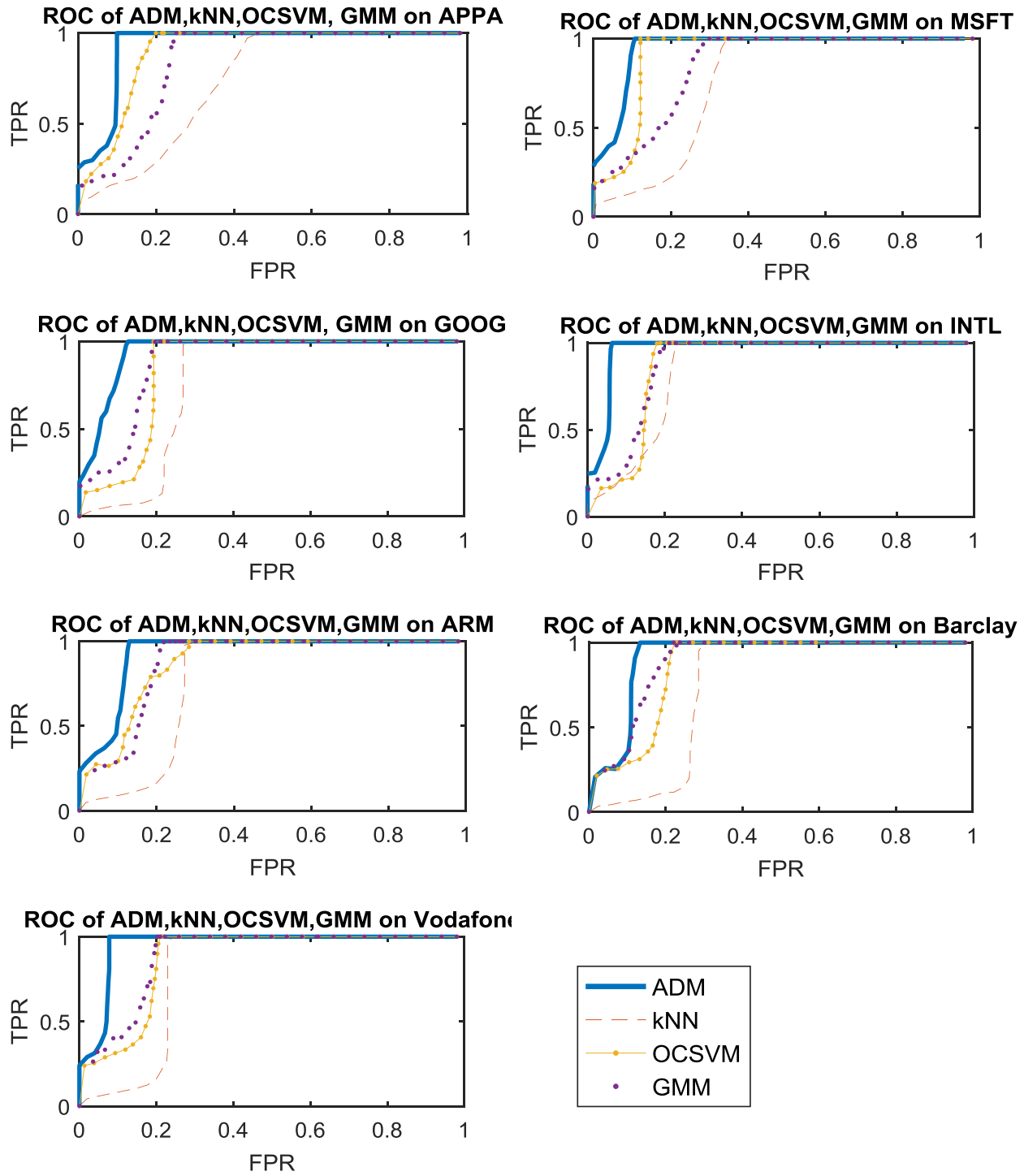


Figure 18 The ROC of four models using seven stock datasets (TPR and FPR).

Outcomes are given in Figure 19, where the legends OCSVM, kNN and GMM show the differences in outcome between ADM and OCSVM, kNN and GMM models, respectively.

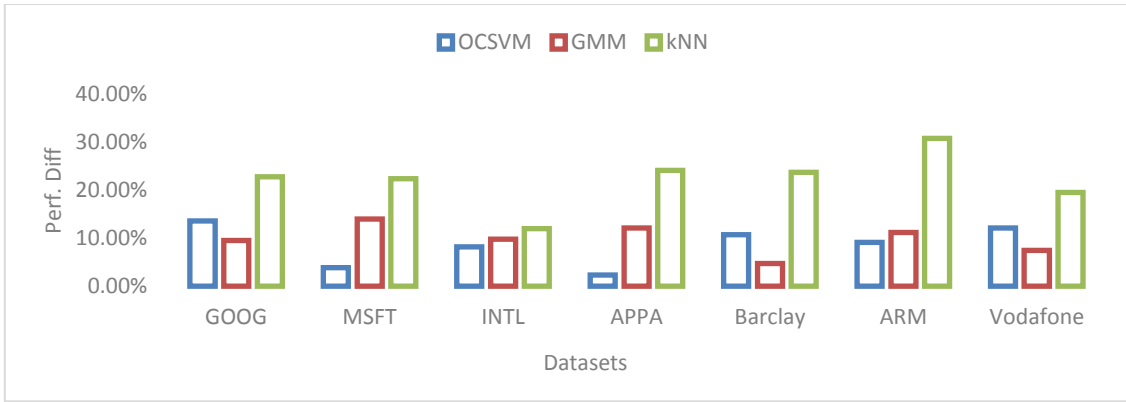


Figure 19 Differences in performance between ADM and three other models as applied to seven datasets.

The least degree of difference between the ADM and another model is seen in the Apple dataset, where the ADM (0.9592) outperformed the KNN model (0.7334) by 30.79%. The minimum value shows that the performance of ADM and k-NN on this dataset was almost identical; in contrast, the greatest degree of difference is between the ADM and GMM models as applied to the Microsoft dataset. Here, the ADM (0.9329) has outperformed the GMM (0.8183) by 14.01%, and it is here that ADM performs best amongst all test examples. Figure 19 shows that the difference in performance between ADM and OCSVM remains quite stable in range, from 5% to 30% over various datasets, while the differences in performance between ADM and the other two models are markedly more volatile, being in the range of 2% to 46%. A comparison of the models' performance reveals that the ADM sustains a superior performance to that of all other models over seven market datasets. These results also shed light on the performance characteristics of the benchmark models, and in particular their performance stabilities. For example, the OCSVM model performed relatively steadily, whereas the k-NN and GMM models showed some volatility of performance across the various datasets.

4.3.3 Performance discussion

These experiments demonstrate that the ADM outperforms its three benchmarks, namely the OCSVM, k-NN and GMM. This quality of performance can be attributed at least in part to elements inherited from HMM, specifically the aspects of time series data modelling in which the algorithm does not merely accept the most likely state for a given point in time, but takes the whole sequence into account when coming to a decision. Thus the presence of normal events part-way through an otherwise abnormal sequence will not alter the fact that the model detects the sequence as a whole as being abnormal. This quality is particularly helpful in the detection of price manipulation, because in the market, both legitimate and manipulative behaviours will inevitably blend and overlap, yet this will not (given the quality described) prevent the model from detecting anomalies. In contrast, other models are trained upon tightly marked boundaries, which take the examples tested as a series of separate points, rather than a cohesive sequence.

The adaptive mechanism also confers advantages upon the ADM. Owing to its non-stationary nature, the distribution of financial time series data x_{t_1}, \dots, x_{t_1} may deviate from $x_{t_1+\tau}, \dots, x_{t_1+\tau}$ and attempting to detect the anomalous data using a constant model raises the risk of error. The act of isolating the significant difference and updating the model used for detection gives reasonable compensation for the non-stationarity.

There are differences in the ADM's performance, depending on the dataset involved. In Table 5, the majority of AUC values for the ADM exceed 0.8, which is considered a threshold of good performance for a classifier. However, the outcome for the other data (for example Microsoft) is lower than the best (at 0.9329) and slightly lower than the best AUC (0.9497) on the Intel dataset. Further exploration revealed that the small AUC on the Microsoft dataset is caused by the relatively high number of FN cases (that is, normal cases incorrectly detected as anomalous), which decreases the TPR on each discrimination threshold. When the FN data was examined and comparison made between training and testing data, it was found that the patterns of normal cases that were picked up as being abnormal were identical to the patterns created by the actually abnormal cases shown in Figure 2, and were absent from the training dataset. In the non-anomalous Microsoft testing dataset (i.e. the set without injected anomaly patterns) the bid price oscillated unusually between 8:00:00 and 8:01:00 am on 12 June 2012 and showed mixed patterns of sawtooth and square wave, which are easily identified as abnormal by the detection model. Figure 20 illustrates this.

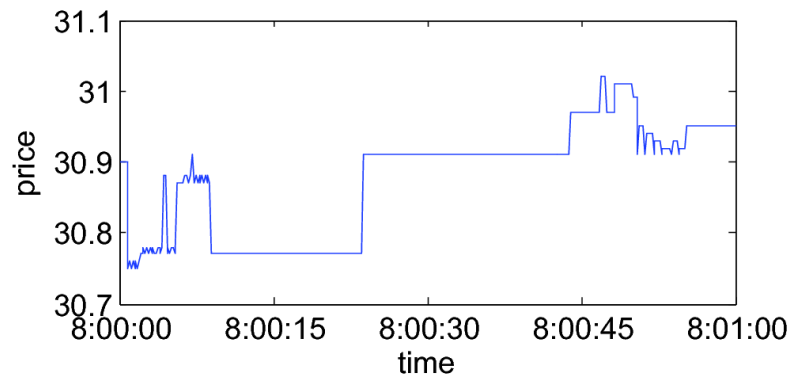


Figure 20 Activity of Microsoft stock bid price between 8.00.00 and 8.01.00 am on 12 June 2012, in which normal (legitimate) data exhibited patterns associated with anomaly.

The authors have raised this point with industry experts, who suggest that, although rare, such marked oscillation can occur with legitimate data, especially during periods of high frequency trading (HFT). While HFT falls outside the scope of this paper, due to its characteristics of hyper-fast trading speed, such as 1000 updates in 1ms, the wild fluctuations illustrated in Figure 20 do mimic patterns analogous with triggering of the manipulation algorithm by HFT. The wild pattern is also identical to that occurring in a genuine manipulation case from 2012, which is shown in Figure 2(a) but as yet unreported by the regulatory authorities. Thus although the FN cases (legitimate cases falsely identified as abnormal) appear to represent a compromise in detection performance, in fact they mostly refer to cases which present in forms identical to genuine manipulation cases.

Thus, based on the outcome of the experimental evaluation and analysis of the misclassifications, it is reasonable to conclude that the ADM gives a performance consistently superior to that of OCSVM, k-NN and GMM under most of the randomness of the non-stationary nature of the underlying equity price, and that ADM is better suited to detecting price manipulation in markets based on a bid/ask price than these other benchmark approaches. Further to consultation with financial experts, it is apparent that there is a need for real-time surveillance on trading floors, and that the ability to identify unusual patterns of trading would be a great advantage, and allows traders to adapt their behaviour accordingly. Traders do not necessarily need to determine the driver of every anomaly, nor understand all of the contextual relationships between patterns, but they do want to mitigate potentially damaging effects to their portfolios when obvious problems in the market, such as a “flash crash”, threaten to occur. The proposed method identifies potential manipulation by taking both anomalies and contextual relationships (through the Markov chain) into account, and such detection increases accuracy while sacrificing efficiency for real-time

detection, which may not be the first priority for regulators. Thus, given its computational complexity, the ADM may be best suited to overnight detection in real world settings.

5. Conclusion

This paper presents two models - one static, one dynamic- for analysis and detection of price manipulation. Examples of such price manipulation are spoofing trading and quote stuffing. In the former, a large volume and passive quote are used to induce the desired effects on the market while minimising risk to the (rogue) traders, while in the quote stuffing, the trader uses aggressive quotes and a very tiny time to cancellation.

With a view to detecting such manipulative action, the static model considers each trading order as a multi-dimensional object, and identifies the abnormal, behaviours by comparison with the normality. Static model is based on the concept of a safety-critical monitoring system, whereby the system's normal operations follow a stationary, or at least pseudo-stationary, process and anomalous events happen without warning or preamble, and are rare. Training data, which is known to only comprise normal activity, is used to train a model of normality against which new data is compared to recognise abnormalities.

In contrast the dynamic model, or ADM, looks more widely at the temporal contextual relationships between trading sequences, such as a sequence of quote stuffing orders. To create ADM, wavelet transformation is suggested for use as a feature extraction method, to eliminate the long-term low frequency trend from the original bid/ask prices. This leaves just the short-term fluctuations, amongst which manipulative behaviour is likely to be hidden. A model based on a hidden Markov model (HMM) is proposed for learning behaviours represented by the extracted features. Thresholds are established on those features to allow the extreme values to be used as dummy anomaly hidden states for the HMM, thus the traditional HMM is extended to comprise anomaly hidden states that would not otherwise be attained. In order to strengthen the model with regard to the non-stationary changes to the financial data, an adaptive mechanism is provided for use with the DM, thus rendering it an adaptive DM (or ADM).

The experimental evaluations show clearly that both the static and dynamic models are effective in the detection of particular manipulative behaviours and outperform the selected benchmark models.

Despite the proposed approaches, there is sufficient scope for future work. The static model proposed for detecting the single unusual spoofing order works with a transformation method that converts the trading behaviours, illustrated as a multi-dimensional vector, to pseudo stationary. The anomaly detection algorithms, OCSVM and k-NN, can accurately learn the transformed stationary trading behaviours. In the transformation method, each variable of the trading behaviours vector was converted separately. The stationarity of the transformed vector was also tested along each single dimension independently. Usually, a jointly multi-dimensional stationary process implies that each uni-dimensional process is stationary. However, the converse is not necessarily true. Therefore, future work can be focussed on a transformation method that considers all the dimensions as well as the correlations among them and could approximately convert the original non-stationary vector to be a pseudo-stationary multi-dimensional vector. In the dynamic model, an adaptive mechanism compensated for the non-stationary features of the extracted features. However, the retraining processes increased the computational complexity. In practice in a financial market, the rapidly growing trading frequency increased the non-stationarity of the financial time series and at the same time decreased the tolerance of latency for model retraining. However, an increased non-stationarity requires more retraining. To solve this contradictory problem, additional to the adaptive mechanism, a vector transformation method that removes or partially removes the non-stationarity while maintaining the necessary statistical features might be one strand of the future work. Finally, in real financial markets, the trading event may evolve over time very quickly. It would undoubtedly be a

great contribution to the field if future research into detection algorithms could deal with manipulation behaviours in real-time through very computational efficient algorithms or more complex algorithms that can tackle non-stationary features of the data flow.

6. Bibliography

- Aggarwal, & Wu, G. (2006). Stock Market Manipulations. *The Journal of Business*, 79(4), 1915-1953.
- Ahrabian, A., Took, C. C., & Mandic, D. P. (2012). Algorithmic trading using phase synchronization. *IEEE Journal of Selected Topics in Signal Processing*, 6(4), 399-404.
- Aitken, M., Harris, F., & Ji, S. (2009). Trade-based manipulation and market efficiency: A cross-market comparison. *22nd Australasian Finance and Banking Conference*, (p. 18). Sydney.
- Allen, F., & Gale, D. (1992). Stock Price Manipulation. *The Review of Financial Studies*, 5(3), 503-529.
- Berlin, H. U. (2013). (Humboldt Universität zu Berlin) Retrieved from LOBSTER: <https://lobster.wiwi.hu-berlin.de/index.php>
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer New York.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327.
- Caetano, M. A., & Yoneyama, T. (2009). A new indicator of imminent occurrence of drawdown in the stock market. *Physica A: Statistical Mechanics and its Applications*, 388(17), 3563-3571.
- Cao, L., Ou, Y., & Yu, P. (2012, August). Coupled Behavior Analysis with Applications. *IEEE Transactions on Knowledge and Data Engineering*, 24(8), 1378 - 1392.
- Cao, Y., Li, Y., Coleman, S., Belatreche, A., & McGinnity, M. (2015). Adaptive Hidden Markov Model With Anomaly States for Price Manipulation Detection. *IEEE Transactions on Neural Networks and Learning Systems*, 318 - 330.
- CFTC. (May 20 2013). *Antidisruptive Practices Authority No. 3038-AD96*. US Commodity Future Trading Commission.
- Chang, & Lin. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2, 27:1–27:27.
- Chow, E. H.-Y., Hung, C.-W., Liu, C. S.-H., & Shiu, C.-Y. (2013). Expiration day effects and market manipulation: evidence from Taiwan. *Review of Quantitative Finance and Accounting*, 441–462.
- CME. (2014, August). *Rule 575*. Retrieved from U.S. Commodity Futures Trading Commission: <http://www.cftc.gov/filings/orgrules/rule082814cmecm001.pdf>
- Cumming, D. J., F. Zhan, & Aitken, M. J. (2013). *High frequency trading and end-of-day manipulation*. Tech Report, York University, Toronto.
- Cumming, D., Johan, S., & Li, D. (2011, March). Exchange trading rules and stock market liquidity. *Journal of Financial Economics*, 99(3), 651–671.
- D.M.J., T. (2013). *DDtools, the Data Description Toolbox for Matlab version 2.0.1*.
- Daubechies. (1992). *Ten Lectures on Wavelets*. SIAM.
- Daubechies, Lu, & Wu. (2011). Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool. *Applied and computational harmonic analysis*, 30(2), 243-261.

- Diaz, D., Theodoulidis, B., & Sampaio, P. (2011, September). Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices. *Expert Systems with Applications*, 38(15), 12757–12771.
- Ding, X., Li, Y., Belatreche, A., & Maguire, L. P. (2014). An experimental evaluation of novelty detection methods. *Neurocomputing*, 135, 313-327.
- Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4), 987-1008.
- EU. (2012). *Market Abuse, Amended Proposal for a Directive*. European Commission, Brussels.
- EU. (2014, 6). *MiFID II*. Retrieved from European Commission Banking and Finance: http://ec.europa.eu/finance/securities/isd/mifid2/index_en.htm
- Feldman, M. (2009). Analytical basics of the EMD: two harmonics decomposition. *Mechanical Systems and Signal Processing*, 23(7), 2059-2071.
- Ferraris, A. (2008). *Equity market impact models: Mathematics at the inter- face between business and research*. Tech. Rep., Stifterverband fur die Deutsche Wissenschaft, Berlin, Germany.
- Ghazali, R., Hussain, A. J., Nawi, N. M., & Mohamad, B. (2009). Non-stationary and stationary prediction of financial time series using dynamic ridge polynomial neural network. *Neurocomputing*, 72(10-12), 2359–2367.
- Grimmett, G., & D. Stirzaker. (2001). *Probability and Random Processes*. New York, NY, USA: Oxford University Press.
- Guo, P., Miao, Z., Zhang, X.-P., Shen, Y., & Wang, S. (2012, September). Coupled Observation Decomposed Hidden Markov Model for Multiperson Activity Recognition. *IEEE Transaction on Circuits and Systems for Video Technology*, 22(9), 1306-1320.
- Hautsch, N., & Huang, R. (2012). The market impact of a limit order. *Journal of Economic Dynamics and Control*, 36(4), 501 - 522.
- Haven, E., Liu, X., & Shen, L. (2012, October). De-noising option prices with the wavelet method. *European Journal of Operational Research*, 222(1), 104–112.
- Hayton, P., Utete, S., King, D., King, S., Anuzis, P., & Tarassenko., L. (2007). Static and dynamic novelty detection methods for jet engine health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), 493-514.
- Huang, Y. C., & Cheng, Y. J. (2014). The Trading Behavior of Attention Securities with Different Closing Mechanisms: Evidence from Taiwan. *Review of Pacific Basin Financial Markets and Policies*, 1450026.
- Huang, Y. C., & Cheng, Y. J. (2015). Stock manipulation and its effects: pump and dump versus stabilization. *Review of Quantitative Finance and Accounting*, 791–815.
- Hull, J. (2011). *Options, Futures and Other Derivatives* (8th ed.). Upper Saddle River, NJ, USA: Pearson Education.
- Ian., D. (2012). *Market Abuse and Surveillance*. Economic impact assessment on market abuse and surveillance legislation, Government Office for Science, London.
- Jawadi, F., Louhichi, W., Cheffou, A. I., & Randrianarivony, R. (2016). Intraday jumps and trading volume: a nonlinear Tobit specification. *Review of Quantitative Finance and Accounting*, 1167–1186.
- Lee, C.-C., Lee, J.-D., & Lee, C.-C. (2010). Stock prices and the efficient market hypothesis: Evidence from a panel stationary test with structural breaks. *Japan and the World Economy*, 22(1), 49–58.

- Lee, E. J., Eom, K. S., & Park, K. S. (2013). Microstructure-based manipulation: Strategic behavior and performance of spoofing traders. *Journal of Financial Markets*, 16(2), 227–252.
- Mendes, R., Paiva, A., Peruchi, R., & Balestrassi, P. (2016). Multiobjective portfolio optimization of ARMA-GARCH time series based on experimental designs. *Computers & Operations Research*, 434–444.
- Mongkolnavin, J., & Tirapat, S. (2009, May). Marking the Close Analysis in Thai Bond Market Surveillance using Association Rules. *Expert Systems with Applications*, 36(4), 8523–8527.
- Nanex. (2012, September 25). *Whac-A-Mole is Manipulation*. Retrieved from <http://www.nanex.net/aqck2/3598.html>
- Nanex. (2013, July 10). *Incredible, Blatant Manipulation in Apple Stock*. Retrieved from <http://www.nanex.net/aqck2/4352.html>
- Neal, R. M. (2000). Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2), 249-265.
- Ngai, E., Hu, Y., Wong, Y., Chen, Y., & Sun, X. (February 2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 559–569.
- Öğüt, Doğana, & Aktaş. (2009, November). Detecting stock-price manipulation in an emerging market: The case of Turkey. *Expert Systems with Applications*, 36(9), 11944–11949.
- Ong, M., & Condon, N. (2012, September 25). *FINRA Joins Exchanges and the SEC in Fining Hold Brothers More Than \$5.9 Million for Manipulative Trading, Anti-Money Laundering, and Other Violations*. New York: SEC. Retrieved from SEC.
- Oztekin, A., Kizilaslan, R., Freund, S., & Iseri, A. (2016). A data analytic approach to forecasting daily stock returns in an emerging market. *European Journal of Operational Research*, 697–710.
- Palshikar, G. K., & Apte, M. M. (2008). Collusion set detection using graph clustering. *Data Mining and Knowledge Discovery*, 16(2), 135-164.
- Priestley, M. B. (1982). *Spectral Analysis and Time Series*. Academic Press.
- Ron, K. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (pp. 1137–1143). San Mateo, CA: Morgan Kaufmann.
- Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson., R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7), 1443-1471.
- SEC. (2013). *Securities and Exchange Commission Release*. SECURITIES AND EXCHANGE COMMISSION, Atlanta, GA, USA.
- Stephen, B. (2009, April). The Use of Hidden Markov Models for Anomaly Detection in Nuclear Core Condition Monitoring. *IEEE Transactions on Nuclear Science*, 56(2), 453-461.
- Tax, D. (2012, July). *DDtools, the Data Description Toolbox for MATLAB*. Retrieved from DDtools, the Data Description Toolbox for MATLAB: http://prlab.tudelft.nl/david-tax/dd_tools.html
- Tsay, R. S. (2010). *Analysis of Financial Time Series*. Wiley.
- Tse, J., Lin, X., & Vincent, D. (2012). *High frequency trading—Measurement, detection and response*. Tech. Rep., Credit Suisse, Zürich, Switzerland.

- Van Belleghem, S. (2003). *Adaptive methods for modelling, estimating and forecasting locally stationary processes*. Louvain: Université catholique de Louvain.
- Wang, L., Mehrabi, M. G., & Kannatey-Asibu, E. (2002). Hidden Markov model- based tool wear monitoring in turning. *Journal of Manufacturing Science and Engineering*, 124 (3), 651-658.
- Wu, Z., & Huang, N. E. (2004). A study of the characteristics of white noise using the empirical mode decomposition method. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 460(2046), 1597-1611.
- Yeung, D.-Y., & Ding, Y. (2003). Host-based intrusion detection using dynamic and static behavioral models. *Pattern Recognition*, 36, 229-243.