Detecting Price Manipulation in the Financial Market

Yi Cao, Yuhua Li, Sonya Coleman, Ammar Belatreche, T.M.McGinnity
Intelligent Systems Research Centre
University of Ulster
Londonderry, United Kingdom
cao-y1@email.ulster.ac.uk, {y.li, sa.coleman, a.belatreche, tm.mcginnity}@ulster.ac.uk

Abstract— **Market abuse has attracted much attention from financial regulators around the world but it is difficult to fully prevent. One of the reasons is the lack of thoroughly studies of the market abuse strategies and the corresponding effective market abuse approaches. In this paper, the strategies of reported price manipulation cases are analysed as well as the related empirical studies. A transformation is then defined to convert the time-varying financial trading data into pseudo-stationary time series, where machine learning algorithms can be easily applied to the detection of the price manipulation. The evaluation experiments conducted on four stocks from NASDAQ show a promising improved performance for effectively detecting such manipulation cases.**

I. Introduction

Surveillance of the financial exchange market for monitoring market abuse activities has attracted much attention from financial regulators across different exchange markets in recent years especially since the flash crash in 2010. However, the lack of research in effective and efficient detection algorithms, in both industry and academia, causes challenges for regulators in their ability to monitor huge amounts of trading activities in real-time. A major concern to financial market abuse is price manipulation, where the manipulated target is the bid (or ask) price of certain financial instruments [1]. There is a large amount of literature regarding stock market manipulation theories [1] [2] [3] and a few empirical studies of real manipulation cases [4]. However, an effective detection model of price manipulation is yet to be developed due to the lack of understanding of strategic spoofing tactics.

In this paper, we summarize and further analyse the price manipulation strategies by examining actual reported cases as well as the empirical studies in existing literature. We define two key characteristics of price manipulation strategies, which enable us to propose a transformation procedure, converting the original market trading data to a comparable metric, where the non-stationary nature of the financial data is demonstrated to be "nearly removed" and the machine learning techniques can then be effectively applied as detection models. Our proposed detection approach is evaluated based on real trading records of selected stocks from NASDAQ.

The remainder of this paper is organized as follows: Section II provides a brief review of price manipulation and the corresponding detection methods. In Section III, the price manipulation tactics are thoroughly analysed. A data transform procedure is then proposed and illustrated with real trading data. Section IV evaluates the proposed machine learning based detection models and the obtained promising performance is reported. Finally Section V concludes the paper and discusses potential improvements and future work.

II. Review of Related Literature

Theoretical studies of the stock price manipulation were presented in a number of existing works. A model of transaction-based manipulation was developed, showing that price manipulation is profitable [1]. An "equilibrium model" was derived and proved that the existence of noise traders made it possible to manipulate the price, although theoretically, no profit should be expected according to the efficient market hypothesis [3]. A real price manipulation case conducted by large traders was examined and analysed in [5]. The real case proved that the manipulation tactic can make a risk-free profit, as a result of the significantly changing order flow. More empirical studies showed the increase of the volatility, liquidity, and returns of the underlying stock and an "up then down" process of the price during the manipulation period [1] [6] [2]. A comprehensive empirical study of the price manipulation strategy as well as the corresponding intention was carried out on real manipulation cases from Korea Exchange (KRX) [4]. One type of price manipulation strategy was formally defined according to its statistical features from the empirical study of the data from KRX; however, the thorough study did not lead to the design of a detection model.

Research regarding the detection of the stock price manipulation is comparably limited in both academia and the financial industry. The appropriateness of a sample entropy methodology as a measure for the detection was evaluated in [7]; however, the statistical results did not favour the properties of sample entropy as an indicator of price manipulation. Logistic regression with an artificial neural network and support vector machine has been studied and compared as a method of detecting trade based manipulation within the emerging Istanbul Stock Market [8]. The detection model was built based on the assumption of higher deviations of the statistical features of daily return, volume and volatilities from normal cases indicating manipulation. Similar work has been carried out by firstly studying the reported manipulated cases and constructing a dataset of manipulated cases, and then modelling the returns, liquidity and volatility as well as the news and events related to the stocks during the manipulation period by linear and logistic regression [9]. Evaluations and comparisons of different techniques were also presented in [8] and [9], yet both works lack a reliable, reasonable analysis of the link between the abnormalities of the stock features and the disclosure of price manipulation. Therefore, this leaves a knowledge gap between the data attribute deviations and the detection techniques. An Inverse Reinforcement Learning

This project is supported by the companies and organizations involved in the Northern Ireland Capital Markets Engineering Research Initiative.

(IRL) algorithm was applied to the learning and classifying of traders' behaviours. The experiments were conducted on a simulated S&P 500 futures market through a multi-agent approach [10] and achieved more than 90% classification accuracy. An empirical study of the relationship between the market efficiency and the market close price manipulation, defined as ramping, was carried out and showed a raise in execution costs of completing large trades when experiencing the market close ramping [11]. Ramping alert records generated by the detection algorithm from Smart Group International, a surveillance system provider, were analysed as a benchmark for this study. The algorithm detected market close ramping according to critical price changes where the threshold was set as the 99% histogram distribution cut-off of the historical price change during the benchmark period. A market close ramping alert was triggered if the changes of the closing price and price 15 minutes prior were greater than the chosen threshold [11].

To date, existing research has mainly focused on either empirical studies of certain price manipulation cases or the detection techniques based on abnormalities of the market features during the manipulation period. An effective classification algorithm was shown in [10] but it was based only on simulated markets where the traders and their strategies were clearly defined.

In this paper, the manipulative strategies are analysed with no assumptions on unusual changes of market features. The proposed detection approaches are aimed at learning and modelling the trading behaviours and further identifying the manipulative actions by the learned model. Our approach is evaluated in a real data context.

III. CHARACTERISING PRICE MANIPULATION

Price manipulation activities affect price fluctuation in capital markets, where the returns, volatilities and liquidities, unexpectedly rise then decline during the manipulation period [1] [6] [2]. However, the occurrence of manipulation is hard to prove given the observed changes of the market attributes, which in most cases are the result of economic cycles, market (index) moves and even public events. The detection models based on the significant deviation of the market attributes are doomed to suffer from the error rate of the unusual but legitimate activities that are recognised as manipulation [7] [8] [9]. Instead of using the discrepancies of the financial market attributes, the manipulation strategic behaviour intrinsically offers a more accurate measure. Nevertheless, a model that is capable of directly monitoring behaviours is not available due to the lack of accurate definitions of manipulative behaviours. This is one of the major challenges faced when attempting to detect price manipulation. Recognise

A. Price manipulation strategy characteristics

A generic price manipulation tactic is defined as artificially pushing up (or down) the bid (or ask) price of a security and taking advantage of the shifted price so as to make a profit [12]. The deliberately constructed trading order sequences change the market bid (or ask) price and show the trader's manipulative intentions. The characteristics of those orders define the manipulation strategy, which is the target of the detection model. The strategy is not constructed as incidentally heuristic attempts of placing orders but as careful designs of every single attribute of the placed orders according to the market impact theory [13], which suggests that the market effects are correlated with the quotes and sizes of the posted orders. A quantitative estimation of this effect given by a Vector Autoregressive Model (VAR) [13] showed that either the larger size or the higher (or lower) quote (compared with the current bid (or ask)) induces stronger price impact on the market. For normal traders, measuring and eliminating the market impact is crucial; however, for the market manipulators, the market impact is simply utilized by them in strategies to make an economical profit. According to this, the price manipulation orders ought to be large-sized and of a higher (or lower) price than the bid (or ask) to maximize the market impact. However, none of the reported price manipulation will be completely conducted when following such format [13] due to another constraint: the placed orders for spoofing the market are expected to have little chance of being executed [4], (Execution refers to a failed manipulative action that is not accepted by the manipulators). Consequently, we argue that a price manipulative strategy is deemed to be fulfilling of two conditions: (1) maximising the induced price change; (2) minimising the execution risk.

The definition of the primary manipulation tactic, *spoofing trading*, summarized from the real manipulation cases, is given as: an order with a size at least twice the previous day's average order size, with a price at least 6 basis points (bps) [1] away from the current bid (or ask) price and with a cancellation time longer than 30 minutes [4]. Those numerical definitions, "6 bps", "2 times" and "30 minutes", show a typical case of our argument: maximizing the induced price change (impact) by large size (at least twice the previous day's average size) order staying at order book for a relatively longer time (30 minutes) and minimizing the risk by passive quotes (6 bps away from the bid (or ask)) [4].

In September 2012, a price manipulation case was reported and documented by the Financial Industry Regulatory Authority of the USA [14]. In this case, a sequence of spoofing buy orders was placed inside the spread, pushing up the bid price by 6.9 bps. After the manipulators had benefited from the transaction on their previous sell order at a higher price, the spoofing orders were cancelled. The complete manipulation process lasted for only 819ms and is known as *quote stuffing*. Another 17 analogous *quote stuffing* cases from 2011 - 2012 were then reported by Nanex [15]. The average time duration and the induced bid (or ask) changes of the cases were calculated as 6.2 seconds and 627 bps respectively. Obviously, the numerical features, 6.9 bps and 819ms, of *quote stuffing* also conform to our argument: the *aggressive* quotes maximising the fictitious wild price changes and the instantaneous market sweeping minimising the execution risk.

Spoofing trading and *quote stuffing* suggest two primary strategies of price manipulation. The former utilises a large volume and a passive quote for inducing the impact and reducing the risk while an aggressive quote and a tiny cancellation time are used in the latter, respectively. Both

[1] A basis point is a unit equal to one hundredth of a percentage point

formats can be depicted by two key conditions defined in our argument.

The two strategies are graphically illustrated in Fig. 1, where a three-level order book is initiated at the best bid, $p_t^{b,1}$, and best ask, $p_t^{a,1}$ and the dotted lines represent a quick sweep of the market (tiny cancellation time).

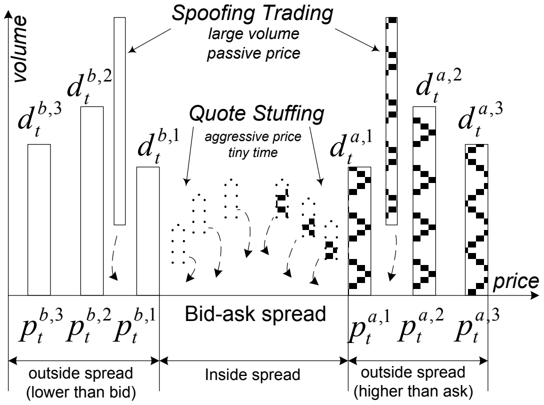


Fig. 1 *Spoofing trading* and *quote stuffing* strategies in a three-level order book.

B. Market Data Transformation

Financial data is usually considered as a non-stationary time series [16]. From the detection model's perspective, the non-stationarity increases the difficulty of identifying the manipulation through the utilisation of one uniform model. One method that would compensate this is that proposed in [17], which adaptively updates the model by monitoring any deviations in the data. However, frequent updates increase the computational complexity of the model.

Here, we propose to transform the original data to a stationary domain while maintaining the desired features, which may bring an alternative computationally efficient solution. This idea is also reflected in a time series analysis, such as the autoregressive integrated moving average (ARIMA) model which is usually applied in cases where the non-stationary feature of the data can be removed by an initial *differencing step* [18]. The *log-return* is also considered to be a transformation from market data to price difference, which is not believed to be perfectly time invariant [19] [20] but has also been analysed and concluded as stationary in a recent investigation [21].

Inspired by the *differencing step* and *log-return* methods, we define a transformation procedure as converting the order data to a consistent and comparable metric. On one hand, this procedure transforms the original data to pseudo-stationary; on the other hand, it enables the evaluation of analytic relationships amongst stocks despite the original unequal values of the order series.

Denoting i as a time index, indicating all order book activities, i.e., submission, cancellation and execution, a buy (or sell) order o_i^b (or o_i^s) is described as a three dimensional vector, $\left[p_{o_i^b}, v_{o_i^b}, t_{o_i^b}\right]'$ (or $\left[p_{o_i^s}, v_{o_i^s}, t_{o_i^s}\right]'$), where p, v and t represent order price, volume and submission time (physical time) respectively. Furthermore, p_i^b and p_i^a denote the best bid and ask price instantaneously before the i^{th} order activity. τ is denoted as the length of a sliding window and is set to one trading day, corresponding with the *spoofing trading* definition. Thus, $\bar{v}_\tau^b$ and $\bar{v}_\tau^s$ define the moving average volume of the buy and sell orders in the previous τ period of time excluding the current data point i. The cancellation times and execution times of order o_i^b (or o_i^s) are denoted by $t^c_{o_i^b}$ and $t^e_{o_i^b}$ (or $t^c_{o_i^s}$ and $t^e_{o_i^s}$) and the lifecycle of an order $l_{o_i^{b/s}}$ can then be defined as

$$l^c_{o_i^{b/s}} = t_{o_i^{b/s}} - t^c_{o_i^{b/s}} \qquad (1)$$

$$l^e_{o_i^{b/s}} = t_{o_i^{b/s}} - t^e_{o_i^{b/s}} \qquad (2)$$

for the cancelled or executed order respectively. Thus the average lifecycle of orders in the prior τ period are calculated as

$$\bar{l}^c_{\tau,b/s} = \frac{1}{N}\sum_{i=1}^{N} l^c_{o_i^{b/s}} \qquad (3)$$

$$\bar{l}^e_{\tau,b/s} = \frac{1}{N}\sum_{i=1}^{N} l^e_{o_i^{b/s}} \qquad (4)$$

The transformation is then defined in Equations (5), (6) and (7), where $p_{o_i^n}, v_{o_i^n}, t_{o_i^n}$ represent three transformed attributes.

$$p^n_{o_i} = \begin{cases} \ln\left(\dfrac{p_{o_i^b}}{p_i^b}\right) \\ \ln\left(\dfrac{p_{o_i^s}}{p_i^a}\right) \end{cases} \qquad (5)$$

$$v^n_{o_i} = \begin{cases} \ln\left(\dfrac{v_{o_i^b}}{\bar{v}_\tau^b}\right) \\ \ln\left(\dfrac{v_{o_i^s}}{\bar{v}_\tau^s}\right) \end{cases} \qquad (6)$$

$$t^n_{o_i} = \begin{cases} \ln\left(\dfrac{l^c_{o_i^b}}{\bar{l}^c_{\tau,b}}\right) \; or \; \ln\left(\dfrac{l^e_{o_i^b}}{\bar{l}^e_{\tau,b}}\right) \\ \ln\left(\dfrac{l^c_{o_i^s}}{\bar{l}^c_{\tau,s}}\right) \; or \; \ln\left(\dfrac{l^e_{o_i^s}}{\bar{l}^e_{\tau,s}}\right) \end{cases} \qquad (7)$$

The stationary nature of a time series (X_t, X_s) is usually tested by the weak definition of stationarity [22] [23] meaning that the mean and variance of (X_t, X_s) do not depend on *t*, while the autocorrelation (*AutoCor*) between (X_t, X_s) and $(X_{t+\Delta t}, X_{s+\Delta t})$ depends only on the lag.

To evaluate the stationarity, we calculate the mean, variance and *AutoCor* for the original and transformed data of the trading orders. In the calculation, we consider each order attribute as a single time series thus the original and the transformed data are represented as three time series respectively:

$$\boldsymbol{P}=\left\{p_{o_i^{b/s}}\right\}, \boldsymbol{V}=\left\{v_{o_i^{b/s}}\right\}, \boldsymbol{T}=\left\{t_{o_i^{b/s}}\right\};$$

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

The top four stocks in NASDAQ in terms of the total market capital, Apple, Google, Intel and Microsoft, are selected for evaluation. The datasets, obtained from LOBSTER project [24], cover messages over five trading days, from the 11th to the 15th of June, 2012 and consist of more than 40,000 trading orders in total for each stock. We calculated the *mean* and *variance* of the time series $(\mathrm{X}_{t+\Delta t}, \mathrm{X}_{s+\Delta t})$ with Δt from 0 to the length of the time series. The autocorrelation, *AutoCor*, is calculated between the time series $(\mathrm{X}_t, \mathrm{X}_s)$ and $(\mathrm{X}_{t+\Delta t}, \mathrm{X}_{s+\Delta t})$ with the same Δt values.

The calculated *mean* and *variance* of three attributes, price, volume and time, for the Intel dataset are shown as an example in Fig. 2. It should be noted that only the first 200 lag values are illustrated in the figures for a clear comparison between the original and transformed data. As shown in Fig. 2(g)-(l), the transformed price, volume and time all fluctuate around a nearly constant mean value with an approximately constant variance, while the attributes of original data Fig. 2(a)-(f) move with volatility across the time.

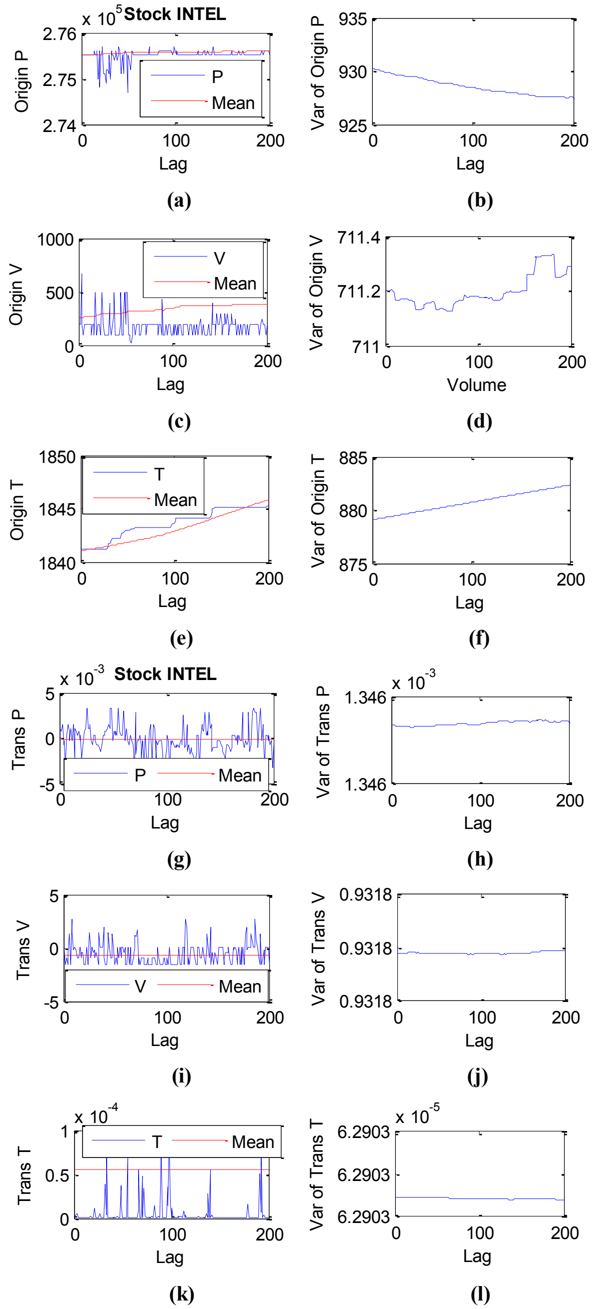
When further observing the dispersion of the sequences of calculated *mean* and *variance*, the measure of the dispersion of a given sequence, the coefficient of variation (CV), defined as the ratio of standard deviation to the mean of a data sequence, is further calculated. The CV of the *mean* and *variance* sequences under different lag values for three attributes, price, volume and time are calculated for both the original data and the transformed data across four datasets and shown in Table I.

The significantly smaller CV values of the *mean* and *variance* sequences of the transformed price, volume and time compared with the original data show far lower level of dispersion in the transformed data sequences, which indicates "nearly constant" *mean* and *variance* values. The "nearly constant" *mean* and *variance* partially conform to the weak definition of stationarity.

Table I COEFFICIENT OF VARIATION OF THE SEQUENCE OF MEAN & VARIANCE OF THE PRICE, VOLUME AND TIME FOR THE DATASET OF FOUR STOCKS

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

As a time series matches up perfectly with itself ('zero-lag'), the figures in Fig. 3 begin from Δt = 1 to avoid large value at Δt = 0. Fig. 3 clearly shows the autocorrelation, *AutoCor*, of three transformed time series $\boldsymbol{P_n}$, $\boldsymbol{V_n}$ and $\boldsymbol{T_n}$, decreasing with an increasing lag and tailing off to tiny values, which additionally suggests the decorrelation of the transformed time series. Meanwhile, *AutoCor* of the original data also declines with increasing lags. Although not given in this paper, the datasets of another three stocks (Google, Intel and Microsoft) show identical features as illustrated in both Fig. 2 and Fig. 3.



(a) (b)

(c) (d)

(e) (f)

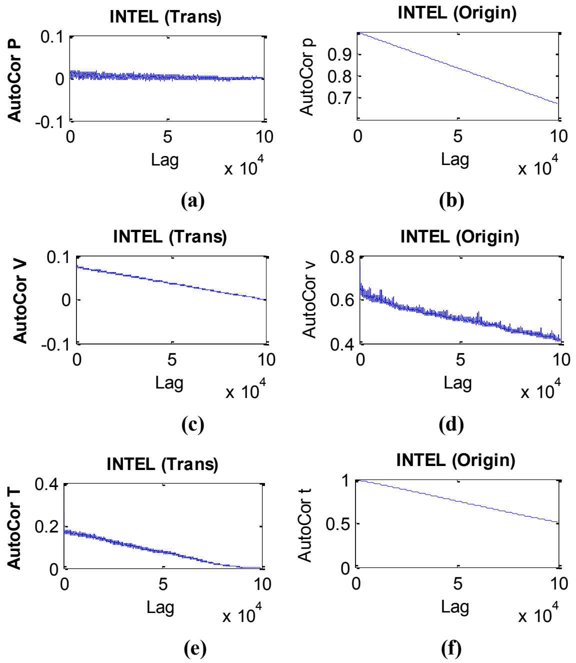
(g) (h)

(i) (j)

(k) (l)

Fig. 2 Mean & Variance of original and transformed Price, Volume and Time for Intel stock.

The comparison between these figures shows that the non-stationary features of the original data are "nearly removed" by the transformation; pseudo-stationary data is then generated, compensating for the time-varying features.



(a) (b)

(c) (d)

(e) (f)

Fig. 3 *AutoCor* of original and transformed Price, Volume and Time for Intel stock.

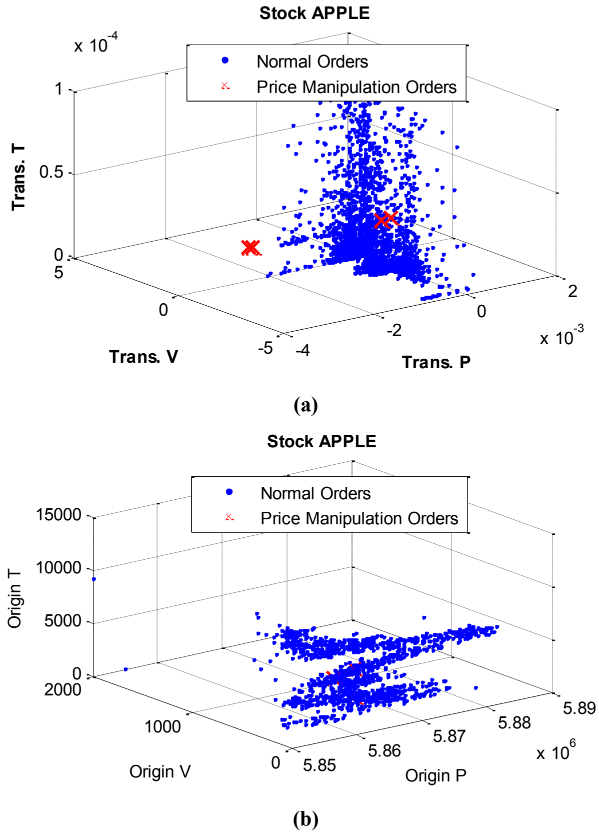
C. *Strategic behaviour illustration*

Given the "non-stationarity" removing transformation and the characteristics of the manipulation strategy previously discussed, we illustrate the converted data by denoting the *x*, *y* and *z*-axis by transformed price $p_{o_t^n}$, volume $v_{o_t^n}$ and time $t_{o_t^n}$ respectively. The trading orders are then represented in a new 3-dimensional domain as shown in the example in Fig. 4(a). Similarly, the original orders are also illustrated in the domain where *x*, *y* and *z*-axis is represented by original price $p_{o_i^{b/s}}$, volume $v_{o_i^{b/s}}$ and time $t_{o_i^{b/s}}$ respectively in Fig. 4(b). The example is formed from the Apple stock dataset captured on the 10th July 2013 with both "normal" and real manipulation cases, which were reported by Nanex [25] later on the same day.

Fig. 4 clearly shows that: (1) the price manipulation cases (shown as the crosses) deviate from the mass normal trading orders in the transformed domain in Fig. 4(a) while overlapping with the normal orders in the domain of the original data in Fig. 4(b); (2) the normal trading orders tend to be distributed along the three axes in the transformed domain in Fig. 4(a) while the original data shows exotic (and time varying) distributions in Fig. 4(b). The reported manipulation cases in the Apple stock trading also follow the proposed arguments: maximising the manipulated effects by aggressive quotes and minimising the execution risks by a tiny cancellation time.

Similarly, the normal orders data of other selected stocks shows a similar pattern to Fig. 4. In the transformed domain, the normal orders congest together as an agglomerative cluster occupying a certain space with analogous but different shapes due to the naturally distinct trading behaviours across financial instruments while the original data show the exotic shapes.

Since the manipulation cases are apparently located apart from the cluster (as Fig. 4(a)), the boundaries of such clusters provide an effective decision threshold. However, such boundaries cannot be described by simply setting up thresholds on three attributes; this is due to the unknown convexity feature of the 3-dimensional spherical surface. Precisely describing the cluster shapes by only the normal data requires some sophistication.



(a)

(b)

Fig. 4 Normal and Manipulation cases illustration. (a) The transformed data; (b) The original data.

IV. Detecting Price Manipulation

The concept of describing the shape of a cluster of normalities is often referred to as novelty detection, where the abnormalities are subsequently identified by testing for novelty against the model of the normality. The mass of normal trading data and the scarceness of the manipulation cases, due to the regulatory rules prohibiting the disclosure of illegitimate market cases, are another reason that the novelty detection paradigm suits the detection of such manipulation.

K-Nearest Neighbour (kNN) based novelty detection is one of the simplest non-parameter methods. It simply computes the distances between the test data and all training samples and uses the lowest distance score plus a threshold (a radius around that example) to make the novelty decision. In general, the Euclidean distance is used between two samples [26].

One-class support vector machine (OCSVM) is another ideal approach for novelty detection, as it provides a direct description of the boundary of normality (the support vectors) [27] [28]. OCSVM applied to price manipulation detection provides a measure of unusualness in trading activity by learning a representation of normal orders.

In this paper, we examine the price manipulation detection problem using the above-mentioned two machine learning models on the transformed time series as well as the original market data. We argue that both models work effectively on the underlying detection problem and the proposed transformation procedure significantly improves the detection performance.

A. Application of OCSVM and kNN to price manipulation detection

When applying the novelty detection approaches to the price manipulation detection problem, a set of normal data vectors $\boldsymbol{S} = \{\boldsymbol{o_1}, \boldsymbol{o_2}, \ldots, \boldsymbol{o_t}\}$ is collected as the training dataset. The vector $\boldsymbol{o_t}$ is from either the vector of the original market data $[p_{o_i}, v_{o_i}, t_{o_i}]'$ (o_i: a buy or sell order), or the transformed data $p_{o_i^n}, v_{o_i^n}, t_{o_i^n}$, calculated by Equations (5)-(7). In the experimental evaluation, the OCSVM and kNN are applied to the four datasets Apple, Google, Intel and Microsoft discussed in Section III.B. The selection of these datasets is according to their relatively high trading volumes and more volatile price fluctuation, factors that may increase the likelihood of manipulation across the exchanges [4] [29]. Each dataset is divided into five subsets according to the trading date. One subset is chosen as the training dataset, where the 5-fold cross-validation is used for training the models; the remaining four subsets are used in the testing.

The evaluation of a detection model is usually reliant upon the labelled benchmarks of both normality and abnormality. Due to a few real manipulation cases being reported, we needed to synthesize a number of abnormal cases based on our study of the characteristics of the manipulation strategy. Synthetic exploratory financial data is accepted in academia for evaluating the proposed model when real market data is hard to collect [30] [31].

Two primary formats of price manipulation, *spoofing trading* and *quote stuffing*, are reproduced in the context of the datasets of each stock following the original characteristics discussed in Section III.A:

- *spoofing trading*: orders with sizes of at least twice the previous day's average order size, with prices of at least 6 basis points outside the current bid-ask spread and with a cancellation time of 30 minutes;
- *quote stuffing*: orders with regular size, with quotes 627 bps higher (or lower) than the current bid (or ask) price and with approximately 6.2 seconds of cancellation time.

The generated manipulation cases are then randomly injected into the corresponding order records, creating a mixture of both "normal" and "abnormal" patterns in the testing datasets. In order to ensure comprehensive assessment of the approach, 5000 synthesized cases are injected to each dataset with each type containing 2500 examples. For the Apple stock dataset, the reported real cases are also injected for evaluation.

In our experiments, LIBSVM [32] and DDTool [33], two open source libraries, are used as the implementation of the OCSVM and kNN respectively. The model parameters, namely the Gaussian kernel and the k value for kNN are determined by 5-folder cross-validation for stable and optimised results.

Performance evaluation is based on the Receiver Operating Characteristic (ROC), which is calculated according to the confusion matrix, where false positive (*FP*), is defined as manipulation cases detected as normal, false negative (*FN*) is defined as normal cases detected as manipulation, true positive (*TP*) is defined as normal cases detected as normal and true negative (*TN*) is defined as manipulation cases detected as manipulation. The ROC curve is a widely used metric for evaluating and comparing binary classifiers. The ROC curve plots the true positive rate $\left(\frac{\mathrm{TP}}{\mathrm{TP+FN}}\right)$ against the false positive rate $\left(\frac{\mathrm{FP}}{\mathrm{FP+TN}}\right)$ while the discrimination threshold of the binary classifier is varied. In order to assess the overall performance of a novelty detector, one can measure the area under the ROC curve (AUC). Larger AUC values are generally an indication of better classification performance.

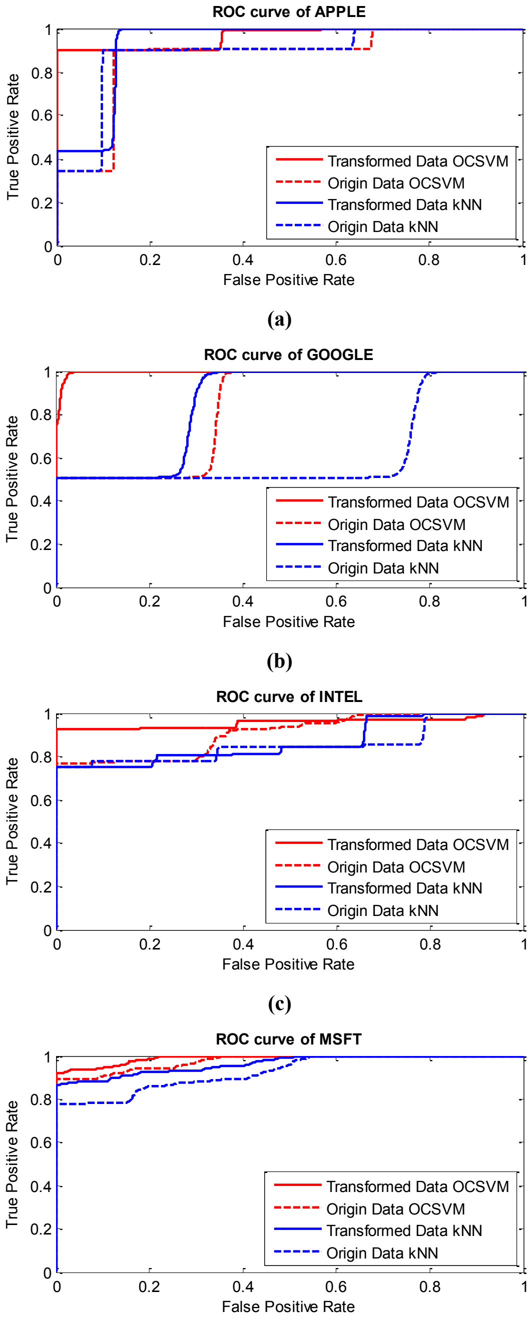
The ROC curves of two models on four stock datasets (original and transformed) with 5000 injected novelties in each dataset are illustrated in Fig. 5. The calculated AUC values are summarised in TABLE II.

TABLE II AUC OF TWO MODELS ON FOUR STOCK DATASETS

Model	**AUC**	**Data**		***Improvement***
		Original	***Transformed***	
OCSVM	APPLE	0.866	0.963	11.201%
	GOOGLE	0.831	0.997	19.976%
	INTEL	0.906	0.958	5.751%
	MSFT	0.976	0.990	1.379%
	AUC	**Data**		***Improvement***
		Original	***Transformed***	
kNN	APPLE	0.884	0.928	5.057%
	GOOGLE	0.625	0.857	37.279%
	INTEL	0.854	0.866	1.433%
	MSFT	0. 926	0.964	4.117%

It is clear that both detection models with the transformation procedure achieved a significantly better performance than with the original market data in terms of the AUC values in TABLE II, where the *Improvement* column is calculated by $\frac{(AUC_{tran.} - AUC_{origin})}{AUC_{origin}}$ as the improvement percentage. On the transformed data, both models achieved high AUC on all four of the datasets. Even the smallest AUC value, 0.857 of kNN on the Google dataset, can also be considered as a good performance [34]. After checking the testing results of the Apple stock, the injected real manipulation cases are successfully discovered by both models. The good performance can be explained by the OCSVM and kNN models effectively modelling the boundaries of the normal behaviour clusters.

Meanwhile, as shown in Fig. 4(a), the transformed data show a relatively regular cluster shapes compared with the exotically distributed initial data in Fig. 4(b). We argue that the proposed transformation procedure particularly contributes to establishing the "pseudo stationary" regular cluster shapes. From the machine learning perspective, the proposed transformation procedure pre-processes the data and sufficiently extracts the required features. The data points in the feature domain can be easily and relatively effectively modelled by both kNN and OCSVM.



(a)

(b)

(c)

(d)

Fig. 5 ROC of two models on four stock datasets.

As discussed before, to compensate the non-stationarity of the data, one approach is to adaptively updating the model by monitoring any deviations in the data distribution. The pseudo-stationarity feature of the transformed data will effectively reduce the necessary updates and consequently provides a computationally efficient approach.

It is also noted that the OCSVM outperforms the kNN across all four different datasets. The higher performance of OCSVM may be due to a better description of the clusters of normal cases through a more accurate description of the boundary by support vectors.

V. Conclusion and Future Work

This paper analysed the price manipulation in the financial market and defined its key features. A transformation procedure is proposed for mapping non-stationary market data to pseudo stationary attributes, to which the machine learning algorithms can easily be applied as a detection model. The evaluation, which has been conducted on top four stocks from NASDAQ in terms of the market capital, shows promising performance in terms of area under the ROC curve.

In the proposed method, the stationary nature of the data is tested separately on three attributes, which however, have been modelled by OCSVM and kNN as a feature vector. The study of the stationarity of the order vector and the corresponding detection model updating (re-training) will be the focus of our future work. Furthermore, in recent years, the market manipulation tends to be carried out in more than one exchange market by some tricky manipulators. Detection within any single market hardly achieves a complete and accurate result. This requires a cross-market detection model, which is also one of our primary future works.

References

[1] F. Allen and D. Gale, "Stock-price manipulation," *Review of Financial Studies,* vol. 5, no. 3, pp. 503-529, 1992.

[2] R. K. Aggarwal and G. Wu, "Stock Market Manipulations," *The Journal of Business,* vol. 79, no. 4, pp. 1915-1953, 2006.

[3] F. Allen and G. Gorton, "Stock Price Manipulation, Market Microstructure and Asymmetric Information," *European Economic Review,* vol. 36, pp. 624-630, 1992.

[4] E. J. Lee, K. S. Eom and K. S. Park, "Microstructure-based manipulation: Strategic behavior and performance of spoofing traders," *Journal of Financial Markets,* vol. 16, no. 2, p. 227–252, 2013.

[5] R. A. Jarrow, "Market manipulation, bubbles, corners, and short squeezes.," *Journal of financial and Quantitative Analysis,* vol. 3, p. 27, 1992.

[6] M. Jianping, G. Wu and C. Zhou, "Behavior based manipulation: theory and prosecution evidence.," New York University, 2004.

[7] M. Slama and E. Strömma, "Trade-Based Stock Price Manipulation and Sample Entropy," Stockholm School of Economics, 2008.

[8] H. Öğüt, M. M. Doğanay and R. Aktaş, "Detecting stock-price manipulation in an emerging market: The case of Turkey," *Expert Systems with Applications,* vol. 36, no. 9, p. 11944–11949, 2009.

[9] D. Diaz, B. Theodoulidis and P. Sampaio, "Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices.," *Expert Systems with Applications,* vol. 38, no. 10, pp. 12757-12771., 2011.

[10] S. Yang, M. Paddrik, R. Hayes, A. Todd, A. Kirilenko, P. Beling and W.

Scherer, “Behavior based learning in identifying High Frequency Trading strategies,” in *IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, New York, 2012.

[11] M. Aitken, F. R. Harris and S. Ji, “Trade-based manipulation and market efficiency: a cross-market comparison,” in *22nd Australasian Finance and Banking Conference*, 2009.

[12] Y. Cao, Y. Li, S. Coleman, A. Belatreche and T.M.McGinnity, “A Hidden Markov Model with Abnormal States for Detecting Stock Price Manipulation,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Manchester, pp.3014-3019, Oct 2013.

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

[14] M. Ong and N. Condon, “FINRA Joins Exchanges and the SEC in Fining Hold Brothers More Than $5.9 Million for Manipulative Trading, Anti-Money Laundering, and Other Violations,” 25 September 2012. [Online]. Available: http://www.finra.org/Newsroom/NewsReleases/2012/P178687.

[15] NANEX, “Whac-A-Mole is Manipulation,” 25 September 2012. [Online]. Available: http://www.nanex.net/aqck2/3598.html.

[16] R. Ghazali, A. J. Hussain, N. M. Nawi and B. Mohamad, “Non-stationary and stationary prediction of financial time series using dynamic ridge polynomial neural network,” *Neurocomputing,* vol. 72, no. 10-12, p. 2359–2367, 2009.

[17] L. Cao, Y. Ou and P. Yu, “Coupled Behavior Analysis with Applications,” *IEEE Transaction on Knowledge and Data Engeering,* vol. 24, no. 8, pp. 1378-1392, 2012.

[18] R. S. Tsay, Analysis of Financial Time Series, Wiley, 2010.

[19] R. F. Engle, “Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation,” *Econometrica,* vol. 50, no. 4, pp. 987-1008, 1982.

[20] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity,” *Journal of Econometrics,* vol. 31, no. 3, p. 307–327, 1986.

[21] C.-C. Lee, J.-D. Lee and C.-C. Lee, “Stock prices and the efficient market hypothesis: Evidence from a panel stationary test with structural breaks,” *Japan and the World Economy,* vol. 22, no. 1, p. 49–58, 2010.

[22] M. B. Priestley, Spectral Analysis and Time Series,, Academic Press, 1982 .

[23] S. Van Bellegem, “Adaptive methods for modelling, estimating and forecasting locally stationary processes,” Université catholique de Louvain, Louvain, 2003.

[24] LOBSTER, “LOBSTER,” Humboldt Universität zu Berlin, 2013. [Online]. Available: https://lobster.wiwi.hu-berlin.de/index.php.

[25] Nanex, “Incredible, Blatant Manipulation in Apple Stock,” 10 July 2013. [Online]. Available: http://www.nanex.net/aqck2/4352.html.

[26] C. M. Bishop, Pattern Recognition and Machine Learning, Springer , 2007 .

[27] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola and R. C. Williamson., “Estimating the support of a high-dimensional distribution,” *Neural computation,* vol. 13, no. 7, pp. 1443-1471, 2001.

[28] P. Hayton, S. Utete, D. King, S. King, P. Anuzis and L. Tarassenko., “Static and dynamic novelty detection methods for jet engine health monitoring,” *Philosophical Transactions of the Royal Society A: Mathematical,Physical and Engineering Sciences,* vol. 365, no. 1851, pp. 493-514, 2007.

[29] D. J. Cumming, F. Zhan and M. J. Aitken, “High Frequency Trading and End-of-Day Manipulation,” Social Science Research Network, 2012.

[30] G. K. Palshikar and M. M. Apte, “Collusion set detection using graph clustering,” *Data Mining and Knowledge Discovery,* vol. 16, no. 2, pp. 135-164, 2008.

[31] M. Franke, B. Hoser and J. Schröder, “On the Analysis of Irregular Stock Market Trading Behavior,” *Studies in Classification, Data Analysis, and Knowledge Organization,* pp. 355-362, 2008.

[32] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology,* vol. 2, p. 27:1–27:27, 2011.

[33] D. Tax, *DDtools, the Data Description Toolbox for Matlab version 2.0.1,* Delft University of Technology, 2013.

[34] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters,* vol. 27, p. 74, 2006.