

Adaptive Hidden Markov Model with Anomaly States for Price Manipulation Detection

Yi Cao, Yuhua Li, Sonya Coleman, Ammar Belatreche, T.M. McGinnity

Abstract—Price manipulation refers to the activities of those traders who utilise carefully designed trading behaviours to manually push up or down the underlying equity prices for making profits. With increasing volumes and frequency of trading, price manipulation can be extremely damaging to the proper functioning and integrity of capital markets. Existing work focuses on either empirical studies of market abuse cases or analysis of particular manipulation types based on certain assumptions. Effective approaches for analysing and detecting price manipulation in real-time are yet to be developed. This paper proposes a novel approach, called Adaptive Hidden Markov Model with anomaly States (AHMMAS) for modelling and detecting price manipulation activities. Together with wavelet transformations and gradients as the feature extraction methods, the AHMMAS model caters for price manipulation detection and basic manipulation type recognition. The evaluation experiments conducted on seven stock tick data from NASDAQ and the London Stock Exchange and ten simulated stock prices by stochastic differential equation show that the proposed AHMMAS model can effectively detect price manipulation patterns and outperforms the selected benchmark models.

Index Terms—Anomaly Detection, Price Manipulation, Capital Market Microstructure, Hidden Markov Model, Market Abuse, Feature Extraction.

I. INTRODUCTION

Risk management has been a major concern for banks, portfolio managers and companies concerned with stock exchange transactions for many years. However, new aspects of risk management have been identified by regulators after the financial crisis in 2008 and especially since the flash crash in 2010. One important aspect is the surveillance of financial exchange market for preventing market abuse activities. The abuse of financial markets can take on a variety of forms, all of which can be extremely damaging to the proper functioning and integrity of the market. The forms contain three primary categories, namely information based manipulation, where false information (financial rumour) is released to affect the equity price, action based manipulation, where the equity price is changed by actions of squeezing the supply/demand of the equity, and trade based manipulation, where the equity price is shifted only by simply buying and selling [1]. A major format of the trade based abuse is price manipulation, where

the manipulation tactic targets the equity *bid/ask* prices only [1], [2].

Compared with the information and action based manipulations, trade based manipulation usually does not contain any illegitimate actions such as financial rumour spreading and market resource squeezing but is carried out only by legal trading activities. With carefully designed selling and buying sequences, manipulators could make the market *bid/ask* price change following their expectation. In the price manipulation tactics, a series of trading actions are often linked in contextual relationships. The monitoring of any single action would not contribute to a thorough detection of the underlying problem. This is the main challenge of such price manipulation detection algorithms. Another challenge arises from the fact that the manipulation behaviours are often not obvious when mixed with a mass of normal trading records. The evolution of manipulation strategies over time is also a challenge for the detection algorithms. Most of the existing related literature empirically studies the manipulations and the corresponding market responses. Very few works analyse the features of different price manipulation strategies and the corresponding detection approaches.

A number of pioneering papers introduced by White in [3], such as trading the foreign exchange by the neuro-wavelet hybrid system [4] and forecasting the bond price by least-square support vector machine [5], opened the door for more advances in modelling and predicting financial markets using computationally intelligent techniques. These received fresh attention after the 2007-2009 financial crisis recede. More accurate results than the traditional models were reported [6], [7]. Inspired by finds in those literatures, we proposed an approach that considers the challenges of the price manipulation detection problem. In this approach, we formulate the problem based on the analysis of typical cases, extract the intrinsic features of the manipulation patterns and present a new model to detect price manipulation activities. The main contributions of our work are as follows.

The problem of price manipulation is discussed together with the analysis of three typical examples, from which the key feature patterns are extracted. This provides a clear problem formulation and explains the significance of exploring the price manipulation patterns. A Hidden Markov Model with Anomaly States (HMMAS) is proposed to model and detect the price manipulation patterns. The HMMAS makes a detection decision based on a sequence of price information rather than a single value at a given time instant. The price manipulation types are also quantified and designated by the HMMAS based on the features of each manipulation type. To the best of

The authors gratefully acknowledge the support of Dr P. McParland and B. Bland from Fidessa Group plc. This project is supported by the companies and organisations involved in the Northern Ireland Capital Markets Engineering Research Initiative.

Y. Cao, Y. Li, S. Coleman, A. Belatreche and T.M. McGinnity are with the Intelligent Systems Research Centre, University of Ulster, UK. Email: cao-y1@email.ulster.ac.uk; {y.li, sa.coleman, a.belatreche, tm.mcginny}@ulster.ac.uk

our knowledge, this is a novel application of traditional Hidden Markov Model customized to solving market abuse detection. To enhance the adaptability of the HMMAS, a traditional “re-training” mechanism is proposed for automatically tracking the changes in the statistical properties of the financial time series. Substantial experiments have been conducted on both real data from primary USA and UK markets for testing the practicability on real-life and simulated data for testing the robustness to non-stationarity. This verification mechanism provides a complete spectrum of workability testing from both business and theoretic perspectives.

The remainder of the paper is organised as follows: Section II provides a brief review of price manipulation and the corresponding detection methods as well as the analysis of the manipulation cases and the formulated detection logic. The features of basic price manipulation cases are characterised and extracted by the proposed approaches in Section III. Section IV presents the proposed AHMMAS approach for detecting patterns of price manipulation. Performance evaluation of the proposed approach is provided in Section V. Finally, Section VI concludes the paper and discusses potential improvements and future work.

II. PRICE MANIPULATION AND ITS DETECTION

A. Price Manipulation

In capital markets, limit orders, the instructions to buy or sell equities, are submitted by investors to the electronic trading platform of one exchange market. The limit orders indicate the trading intention of the investor to buy or sell volumes of a specific equity at a specific price. The trading occurs once there are eligible matched orders from the investors. Among those outstanding unmatched limit orders, the highest price investors are willing to pay for an equity determines the *bid* price and the lowest price at which investors are willing to sell an equity decides the *ask* price. The gap between the *bid* and *ask* price is defined as the *bid-ask spread* [8].

Price manipulation can be carried out in many different forms [9]. One primary form is termed *ramping* (or *gouging* [10] and *momentum ignition* [11]), where one investor enters a buy/sell order, usually called a “spoofing order” at a price which is higher/lower than the *bid/ask* to create the appearance of active interest in a security, followed by a “bona fide” order on the opposite side of the order book waiting to be executed, subsequently followed by the withdrawal of the first order when the “bona fide” order was mostly executed. Once the “spoofing order” is created, a price upward/downward movement is expected from the manipulator and latter actions are followed only when there is an enough potential profit for the investor. *Ramping* has roughly the same effect as another two forms of price manipulation, *pump&dump* and *capping&pegging* schemes [10], where the profit making approaches are different. In *pump&dump*, the exchange participants make profit by carrying out a quick flip at the manipulated price to remove the already held long positions (Holding of shares of a stock is said to be “has a long position” in the stock.). In *capping&pegging*, the held derivative (e.g., call option, defined as an agreement that gives an investor

the right to buy a stock at a specific price within a specific time period.) of the manipulated security is exercised if a price increase is generated. If the profit is exploited in another market, the form of price manipulation is then termed as cross order/market. If the closing price is manipulated, the manipulation is termed as *near close*.

In September 2012, an analogous type of *ramping* was reported and documented by the Financial Industry Regulatory Authority of the USA [12]. In this case, a liquidity-enhancing strategy, *quote stuffing*, was used for creating the fictitious wild impression of the buying interest on the *bid* side. As a result of the *quote stuffing*, where a mass of “non-bona fide orders” was submitted, the best *bid* price was pushed up 6.9 basis points (bps)¹ and hence the *bid-ask* spread was narrowed by 87.5%. The manipulators have benefited from the trading on their previous submitted order on the *ask* side at the pushed price. The manipulation process lasted for only 819ms and the cancellation time of the “non-bona fide orders” was only 767ms.

Quote stuffing is one of the dominant manipulation strategies in a number of major exchange markets in Europe and USA [11], [13]. It floods the market with massive numbers of new orders, which are then cancelled in rapid successions for creating a large number of successive new best *bid/ask* quoting, each potentially acting as a bait of an opposite order, upon which the potential profit might be realized. The only difference between *ramping* and *quote stuffing* is that the latter sweeps the order book with more spoofing orders and faster cancellation time.

As a summary, all of the above-mentioned forms of price manipulation refer to the same tactic: submitting “non-bona fide” orders to the market, taking advantage of the shifted prices (if expected changes occurred), where the profits are made by distinct ways in various profit-making scenarios as shown in Fig. 1 and Table I.

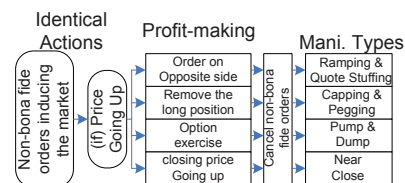


Fig. 1: Price manipulation.

TABLE I: Price manipulation

Mani. Types	Typical Scenarios
ramping or Gouging or quote stuffing or capping & pegging	liquid stocks price manipulation and the profit from opposite orders or positions
pump & dump	manipulation cross stock or option markets.
near close	market closing price manipulation.

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

B. Price Manipulation Detection

The detection of price manipulation has however been less studied in contrast to the volume of theoretical and empirical work on the manipulation activities. Two computational approaches for detecting trade based manipulation within the emerging Istanbul Stock Market, logistic regression with an artificial neural network and support vector machine, have been studied and compared [14]. The detection was based on empirical studies of the statistical features of daily return, daily trading volume and daily volatilities. Higher deviations from the non-manipulated cases indicated manipulation. Similar work has been done by firstly studying the manipulated case pursued by the SEC (Securities and Exchange Commission) and constructing a manipulated cases dataset, 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 [15]. Evaluations and comparisons of different techniques were presented in [14], [15], yet both studies lack reliable quantitative analysis of the relationship between the manipulation tactic and the relevant features such as return, trading volume, liquidity and volatility, which in most of cases are the result of economic cycles, market (index) moves and even public events. The detection models based on the significant deviation of those market attributes are doomed to suffer from the error rate of the unusual but legitimate activities that are recognised as manipulation [14], [15]. Therefore there is a knowledge gap between the data features and the detection techniques.

A very simple technique, association rule, has been proposed for detecting the closing price manipulation in the Thai Bond Market [16]. The approach was based on the assumption that the trading time of a trader should be random regardless of the traders. Therefore, any association between a trader and the transaction orders may indicate price manipulation. This approach has been stated as a supplementary to the existing surveillance system within the Thai Bond Market for identifying a specific type of trade based manipulation [16] rather than a general detection approach.

A market close “Ramping” detection algorithm developed by Smarts Group International, a surveillance technique provider, and applied in NASDAQ OMX for assisting regulators and brokers in detecting trade-based manipulation was discussed in [17]. This work mainly focused on the empirical study of relationships between the market efficiency and the manipulations detected by the algorithm from Smarts Group rather than the analysis of detection algorithms. The algorithm detected market close “Ramping” according to the historical price change 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 the price 15 minutes prior were greater than the chosen threshold [17]. Though straight forward, the market proven detection approach from Smarts Group International provided an industry reference for the study presented in this paper.

To date research has mainly focused on the detection of manipulation in pre-recorded datasets according to the as-

sumptions from empirical studies. To the best of our knowledge, only limited efforts appear to have been made in deep analysis of the strategic behaviour of price manipulation tactic [18] and to a lesser extent on computational modelling of the manipulation strategies [17] which is the focus of this paper.

C. Problem Formulation

During the price manipulation period, a significant price change is the expected effect of the “spoofing” orders by the manipulators. The change is not realized incidentally by heuristic attempts of order submission but by careful designs based on the market microstructure theories, called market impact, which is defined as the impact of the trader’s own actions on the market [19]. The quantitative research in [20] estimated that a quote that is even slightly higher (5 bps) than the *bid* price can induce a further 5.95 bps increase of the market best *bid* price. For normal traders, measuring and eliminating the market impact is crucial for trading models [19]. However, for the market manipulators, the market impact is what they utilize in the strategies for making economical profit.

The documented manipulation case of quote stuffing shown in Fig 2(a), showed an actual market impact scenario. After placing a “bona fide” sell order at an expected price, a number of “non-bona fide” buy orders with quotes successively higher than the best *bid* price were submitted to make it appear that there was active buying interest on that equity. Somebody who was encouraged by the fictitious *bid* price changes responded to the “bona fide” sell order. Immediately after the sell order was nearly executed, the previous *bid* quoting orders were all cancelled, which steeply dropped the *bid* price to its initial level. To minimize the risk of the buy spoofing orders being unintentionally picked up by other investors, the manipulation process usually occurred within an extremely short time period. By the sequential quotes and quick cancellations, the “non-bona fide” orders in Fig 2(a) made a sawtooth shaped market impact on the *bid* price. Thus this price manipulation case can be illustrated as an instantaneous sawtooth pattern on the *bid* price time series. The “sawtooth pattern”, also discovered in [11] and [21], was identified as one of the highlighted patterns on *bid/ask* price time series during manipulation periods [13].

Similarly, the market impact triggered by another price manipulation type, “ramping”, can also be illustrated by special patterns on *bid/ask* price time series. Unlike the activities of “quote stuffing”, in “ramping”, one single “spoofing order” was usually placed inside the *bid ask* spread (higher than the current *bid* or lower than the current *ask*). By this activity, the manipulated *bid/ask* price can be moved a small percentage and reverted to its prior level in tiny time intervals. Small square wave fluctuations were then shown (Figure 2(c)) as the pattern of “ramping” [22], which can also occasionally be a significant movement as illustrated in Figure 2(b). Although this stunning pulse rising is rare since most exchange markets such as Euronext NSC suspend continuous trading if prices change by more than 2% within the defined interval (for example 1 sec) [20], it is highlighted as one of the typical manipulation cases in [21].

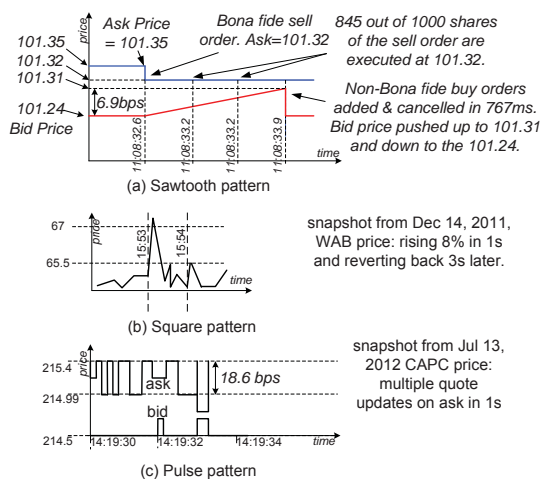


Fig. 2: Feature extraction module

The sawtooth, square wave and pulse (Fig. 2) represent the intrinsic patterns corresponding to the primary manipulation tactics. A mixture of such patterns have been found in real manipulation cases [11]. Consequently, the problem of detecting price manipulation activities can be transformed to the problem of detecting unusual/anomalous patterns in the *bid/ask* price time series. This transformation also corresponds with the “stock-volatility rules” released in April 2013 by SEC in the USA, which allows trading only within certain price bands to avoid unusually volatile trading behaviours [23]. Since the manipulation activities usually occur in tiny time intervals within one single trading day, the scope of our problem is to detect unusual/anomalous patterns in intra-day *bid/ask* price time series. The time scale is discretely measured in terms of *bid/ask* price update event. Hence, the adjustment speed measured in physical time ultimately depends on the underlying frequency of order activities and differs across the market.

In the financial area, it is often believed that the nature of the *bid/ask* price follows the mean-reverting feature, the tendency to randomly oscillate away from and, over time, back towards an “equilibrium” price level determined by the long-term mean of the equity [24]. Studies that support the mean-reversion features can be traced back to 1930s in the empirical studies in [25] and then furthered in [26], [27].

The price fluctuations triggered by the manipulation strategies are merely the unusual short-termed momentary oscillations with small amplitudes around the “equilibrium” level of the price (6.9 bps in Figure 2(a) and 18.6 bps in Figure 2(c)). Those tiny oscillations are usually considered as the “contamination” of the financial data [28], [29]. For studies of financial time series, it is very important to reveal the true signal from the financial data series [29], which contains short, high-frequency “noisy” transients as well as long, low-frequency movement. For price manipulation detection, the high-frequency “noisy” parts are where the manipulation patterns hide. Accordingly, retrieving and directly analysing those high-frequency components as well as the original price information might help develop an effective detection model.

III. CHARACTERISING PRICE MANIPULATION

The detection system proposed in this paper is comprised of a feature extraction module and a detection model, where the model is trained based on the features extracted from the equity *bid/ask* price by the feature extraction module.

A. Price Manipulation Features

The three patterns that characterise typical price manipulations can be defined in two ways: remarkable pulse (Fig. 2(b)) or short term small fluctuations (sawtooth and square in Fig. 2(a) and 2(c)). The intuitive feature of the three patterns can be extracted as the rate of price (*bid/ask*) change, which 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 the changes of the *ask/bid* prices.

Empirical mode decomposition (EMD), continuous wavelet transform (CWT) and synchrosqueezed transform (SST) are commonly used for signal decomposition. EMD is limited by its rather low frequency resolution [30] and the obtainable frequency resolution of EMD depends on a critical frequency limit [31]. CWT can map any signal to a set of base functions obtained through dilation and translation of a mother wavelet, and it is able to achieve decomposition of the signals in different frequency bands and at different time points [32]. Unlike Fourier transform, which gives a representation of frequency content of a signal without time information, the time-frequency localization feature of CWT describes the frequency content “locally” in time by “scale” and “shift” parameters [32]. SST extends the CWT by reallocating the wavelet coefficients through further calculating the first derivative (instantaneous frequency) of the coefficients, which is named as synchrosqueezing [33]. SST has been recently applied as a bandpass filter for removing high frequency components from a signal in [34]. In our work, to retrieve and analyse the high-frequency components of the *bid/ask* prices, wavelet transform is applied as a feature extraction approach due to its wide and successful application in finance [29], [35].

The wavelet decomposition of a signal results in levels of approximation coefficients and detailed coefficients. The approximation coefficient vector reflects the low-frequency features and the detail coefficient vector reflects the high-frequency component of the signal. To retrieve the high frequency components and remove the low-frequency “equilibrium” component, the detail coefficient “hard” thresholding method, which is usually applied in financial signal de-noising [29], is applied inversely, so that the wavelet coefficients outside the thresholds are set to zeros:

$$T_{m,n} = \begin{cases} =T_{m,n} & \text{if } |T_{m,n}| \leq \lambda \\ =0 & \text{if } |T_{m,n}| > \lambda \end{cases} \quad (1)$$

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 consist of three steps: (1) Calculate the wavelet coefficient vectors $\{T_{m,n}, T_{m-1,n}, \dots, T_{1,n}\}$ by discrete wavelet transform; (2) Select a threshold λ and filter the coefficient vectors by through the “hard” thresholding in Equation 1; (3)

Recompose the data using the filtered wavelet coefficients. In this paper, Symlets wavelet is chosen and applied for extracting the high frequency components (denoted by \hat{P}_t) of an equity price P_t . Since the short term small fluctuation is only associated with high frequency components, we also refer to \hat{P}_t as the short term small fluctuation.

B. Features Extraction

The first order derivatives of the original price P_t as well as the short term oscillation \hat{P}_t with respect to time are extracted as the features of the patterns of price manipulation. To capture this, the gradient of P_t and \hat{P}_t are all calculated. Among the methods for calculating gradients, finite-difference is chosen and its central difference form is used since it is widely used in the finance area [36]: $\frac{df(t)}{dt} \approx \frac{f(t+\frac{1}{2}\Delta t) - f(t-\frac{1}{2}\Delta t)}{\Delta t}$, where $f(t)$ represents P_t and \hat{P}_t and Δt is the time interval between the changes of *ask/bid* prices. As discussed in Section III-A, three representative manipulation patterns are defined by significant pulse (Fig. 2(b)) and short term small fluctuations, where the former is associated with original price change rate $\frac{dP_t}{dt}$ and the latter mainly corresponds to the high-frequency oscillations $\frac{d\hat{P}_t}{dt}$.

Financial data usually contain low-frequency trend component upon which a variety of different frequencies are superimposed [34]. From a financial perspective, $\frac{dP_t}{dt}$ and $\frac{d\hat{P}_t}{dt}$ represent the features of the original and the de-trended (low-frequency component removed) prices, which intrinsically correlated with some typical price manipulation activities as previously discussed. Those two patterns that emerged with price manipulation activities persist regardless of the price manipulation forms (Table. I). At the same time, P_t and \hat{P}_t are correlated with the unusual pulses and large square patterns.

Accordingly, all of the four values, P_t , \hat{P}_t , $\frac{dP_t}{dt}$ and $\frac{d\hat{P}_t}{dt}$, should be included as features to cover all the possible manipulation circumstances (sawtooth, pulse and square patterns). The structure of the feature extraction module comprises a wavelet filter and gradient calculator and is shown in Fig. 3.

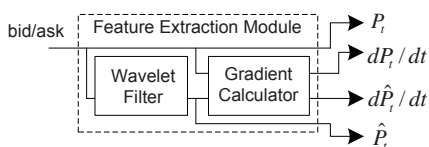


Fig. 3: Feature extraction module

An example of the output of the feature extraction module is shown in Figure 4. The *bid/ask* price P_t is filtered by the Symlets wavelet at level 8. The smoothed “equilibrium” values (illustrated by a dotted line in Figure 4(a)) are removed and the short term oscillation \hat{P}_t is extracted (Fig. 4(b)). The gradients of P_t and \hat{P}_t are both calculated and illustrated in Fig. 4(c) and 4(d) respectively.

IV. DETECTION MODEL

Financial data are considered non-stationary in nature, meaning that the statistical properties (e.g. mean and variance)

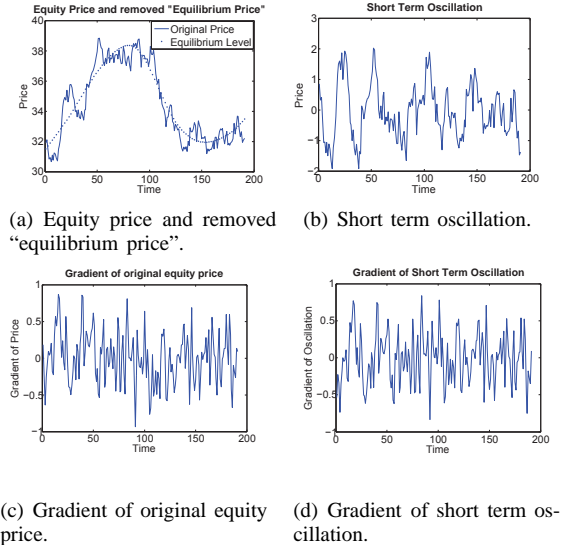


Fig. 4: Example of the equity price filtered by the wavelet.

of the data change over time [37]. These changes are caused by various business and economic cycles in the longer term and demand-supply microstructures in the short term [38]. Of particular interest is the tendency of any unusual activities to exhibit a degree of non-stationarity of the extracted features. That is, the distribution of the extracted features changes over its duration. This may be interpreted as resulting from some of the irregular trading behaviours shown in Fig. 2. Hence, the detection model must be adopted to first capture the distributions of the extracted features and then detect the shifts in them.

A. Gaussian Mixture Model and Hidden Markov Model

A common solution for modelling the probability density function (PDF) of an observed variable is to approximate its unknown density with a Gaussian Mixture Model (GMM) [39]. 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 the mixture weights, and $g(\mathbf{x}|\mu_i, \Sigma_i), i = 1, \dots, M$, are the component Gaussian densities. Each component density is a D -variate Gaussian function. Traditional GMM runs Expectation Maximization (EM) based on heuristic trials for the number of Gaussian components. To overcome this, a Dirichlet Process Gaussian Mixture Model (DPGMM) [40], which provides a statistically principled manner for generating the number of the GMM components, is applied in our approach. DPGMM is the same as a regular GMM, except the component parameters drawn from a Dirichlet process as a prior probability [40] and the Gibbs sampling from the posterior probabilities [40].

In the price manipulation detection problem, the PDFs of extracted features of an equity price are modelled with DPGMM. The observed changes of the features can be detected by testing which mixture component dominates the given value. This gives a simplified state view of the features. The temporal changes of distribution are probabilistically related to the

changes of the observed features and the observed features sequences instead of any single values decide the potential manipulation. Such cases can be modelled by a Hidden Markov Model (HMM) containing two sets of states, “observable” feature states and “hidden” mixture components states, which are assumed to depend only upon the previous states and modelled as a standard Markov process.

Usually, the application of a HMM in condition monitoring problems falls into two categories: signature modelling, where the detection model is learned from the activities to be recognised and the activities matching the model are reported as the detection results [41], [42], and anomaly detection, where a model of normality is learned from only the “normal activities” and the test cases “against” the model according to the pre-defined threshold are reported as anomalies [43], [44].

Due to the limited reports of market manipulation and regulatory rules prohibiting the disclosure of data which represents illegitimate trading behaviours, the availability of examples of market manipulation behaviours in capital market is far less than the availability of routine normal behaviour. Consequently, price manipulation detection can be considered as an anomaly detection problem, which is the identification of new or unknown data patterns, to which a learning system has not been exposed during training. In this approach, normal *bid/ask* price patterns are modelled and the system will trigger an alarm when the behaviour of the market action does not match with normal patterns. However, three problems arise in this approach when applied to price manipulation detection.

- 1) HMM is usually used to model 1-dimensional sequential data rather than multiple features.
- 2) Anomaly detection using HMM usually lacks the ability of recognizing the anomaly type and the probability density of being that type.
- 3) The PDFs of the equity price evolve due to the non-stationary feature of financial time series.

In order to address these challenges, an improved adaptive HMM with anomaly states is developed and is presented in the following two sections.

B. Hidden Markov Model with Anomaly States (HMMAS)

For a *bid/ask* price of an equity with T points, four features are extracted by the module illustrated in Fig. 3. 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 , $\mathbf{F}_t = [F_t^o, F_t^{og}, F_t^s, F_t^{sg}]$, are described as follows:

- 1) original price: F_t^o ;
- 2) gradient of the of original price: F_t^{og} ;
- 3) short-term fluctuation: F_t^s ;
- 4) gradient of the short-term fluctuation: F_t^{sg} ;

The PDFs of the four features are learned separately by the Gaussian Mixture Model. Since the PDF of each feature might have multiple Gaussian components, the changes of the observation distribution can be detected by testing which mixture component in the PDF was most likely to have generated a given value. Those mixture components are then designated as hidden states of the observed equity price. This gives a simplified “state view” of time series data.

For the learned PDFs, anomaly thresholds separating the normal and anomaly are set according to the minimum acceptable data likelihood value, which is adopted from the industry reference detection algorithm from Smart Group [17]: the 99% cumulative distribution cut-off. That means, the data are accepted as “normal” for which $P(\mathbf{F}_t) \geq 99\%$. This is to identify the highest and lowest 0.5% frequent values for each feature. This heuristic method is also applied in a one-class Support Vector Machine [45], where the detection boundary is usually set to include most (for example 99%), but not all, training data to avoid high miss detection rate. By doing this, the 1% extreme values are not simply assumed to be “abnormalities” but generate the “dummy anomaly states” for the Hidden Markov Model.

Figure 5 shows an example of the PDFs of four features of the time series data. The PDFs of the features are learned by the GMM. Two thresholds (shown by dotted lines) are set to 0.5% and 99.5% cumulative probability of each PDF, separating the “normal” and “anomaly” regions. Figure 5(a) shows a PDF of an equity price (illustrated by the small circles) with three Gaussian components (illustrated by the solid line). The thresholds (illustrated by the dotted lines) are set such that the 99% cumulative distribution of the equity prices are accepted as being “normal” leaving the maximal and minimal 0.5% as being “anomalies”. By this, the PDF of the original equity price can be partitioned into four parts: three parts corresponding to three Gaussian components in the “normal” region and one corresponding to the regions with “anomalies” values (both maximal and minimal). The four parts can then be designated as four hidden states respectively for the equity price feature as illustrated in Fig. 6. Defining those hidden states as $\{f_{j_o}^o, j_o = 1, \dots, 4\}$, we have that the extracted feature $F_t^o \in \{f_{j_o}^o, j_o = 1, \dots, 4\}$.

Similarly, two hidden states (one “anomaly” and one “normal”) are generated for another three 1-component PDFs (illustrated by Figure 5(b)-5(d)). We can also define $\{f_{j_{og}}^{og}, j_{og} = 1, 2\}$, $\{f_{j_s}^s, j_s = 1, 2\}$ and $\{f_{j_{sg}}^{sg}, j_{sg} = 1, 2\}$ as the hidden states of price gradient, short term oscillation and oscillation gradients respectively. We also have $F_t^{og} \in \{f_{j_{og}}^{og}\}$, $F_t^s \in \{f_{j_s}^s\}$ and $F_t^{sg} \in \{f_{j_{sg}}^{sg}\}$, where $j_{og}, j_s, j_{sg} = \{1, 2\}$.

The hidden states of all four features are then combined as the quantized hidden states of the HMMAS model so that $4 * 2 * 2 * 2 = 32$ hidden states are generated.

The quantized 32-state view of the observed *bid/ask* price can be represented 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 a unique combination of the feature status. The state having all features within the “normal” region is considered as the completely “normal” hidden state and others having at least one feature within the “anomaly” region are considered “anomaly” hidden states. Thus only three among 32 states are “normal” while 29 are “anomaly”.

According to the official definitions of the price manipulation in Market Abuse Directive [9] and the discussion in Section II, the typical price manipulation types are associated

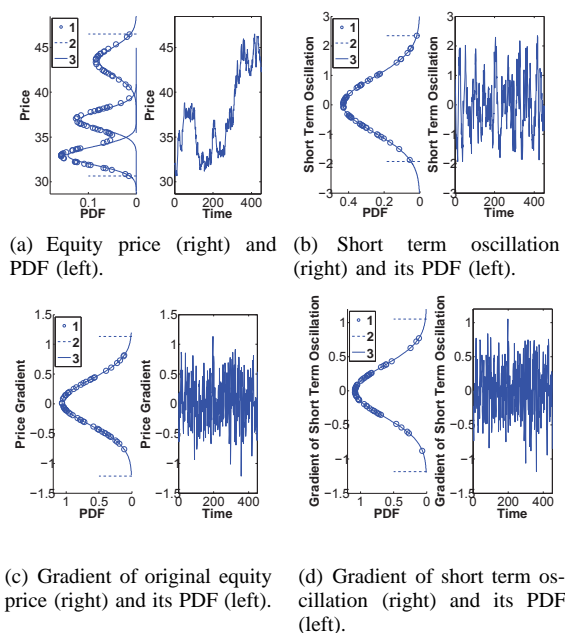


Fig. 5: Example of the features learned by GMM. The PDF of Original Equity Price, the Short Term Oscillation, the Gradient of Original Equity Price and the Gradient of Oscillation are plotted. (Legend: 1: PDF; 2: Components of PDF; 3: Thresholds)

with the combinations of different features status. The types, *quote Stuffing*, *momentum ignition* and *gouging*, which are featured by instantaneous sweeping of order books, are essentially associated with the significant changes of gradients of short term oscillations and the original price ($f_{j_{sg}}^{sg}$ and $f_{j_{og}}^{og}$) rather than the large price fluctuation. Besides, the types, *ramping*, *pump&dump*, *capping&pegging* and *near close* are primarily associated with significant changes of the original price or short term fluctuations ($f_{j_o}^o$ and $f_{j_s}^s$) but not necessarily related to the gradients.

Consequently, according to the features status, the 29 “anomaly” states can be combined into three manipulation states, namely: *quote Stuffing*, *ramping* and *other anomalies*. The states with only anomalous gradient features fall into the *quote Stuffing* while the states with only the anomalous price features (original and the oscillation) are *ramping*. The manipulation state named *other anomalies* is associated with the ones in which all of the features are anomalous. By this merging, the 32 states are further simplified as a six-state view of the observed *bid/ask* price with three normal states and three anomaly states referring to primary manipulation types. Hence the six hidden states provide a complete spectrum as well as an efficient way for representing the *bid/ask* price shifting triggered by the trading behaviours, where, for brevity, only the transfers in hidden states are shown rather than the observed temporal dynamics on each feature. In this way, the traditional HMM is extended to comprise “anomaly” states that can not be otherwise achieved, hence the name Hidden Markov Model with Anomaly States (HMMAS).

The basic property of HMMAS, inherited from the traditional HMM, is the probabilities yielded by the Viterbi algorithms, which determines from an observation sequence the most likely sequence of underlying hidden states that might have generated it. Together with the probability of an observed sequence, HMMAS provides a measure of specific price manipulation type identification according to the states and the features.

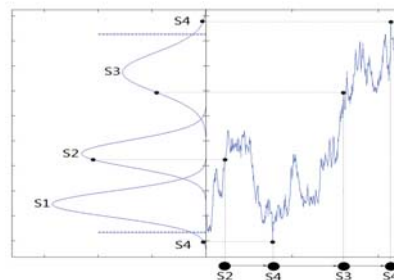


Fig. 6: A hidden Markov model with anomaly states generated from a sequence of observation of variable. S1, S2 and S3 are normal states and S4 is the abnormal state.

C. Adaptive HMMAS

Due to the non-stationary feature of financial time series, the mean and the variance of the PDFs of the equity price might vary over time. In order to deal with the non-stationarity of the time series of *bid/ask* price, the HMMAS is improved by an adaptive mechanism (and renamed the Adaptive HMMAS (AHMMAS)), where the model is trained using the data in previous specific time periods, defined as a sliding window with length w . During the course of detecting the price manipulation activities, the window is slid forward to maintain the closest w data points and AHMMAS is updated if significant discrepancies between the w data points in the current sliding window and the prior training data are detected.

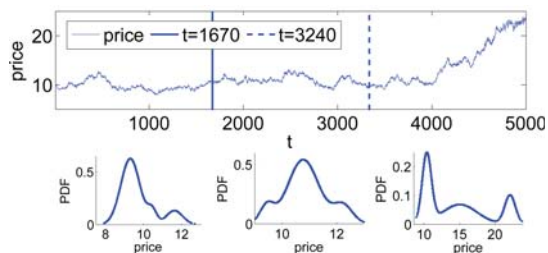


Fig. 7: Non-stationary changes shown by different PDF.

The deviation between the two data sequences is detected using the *t-test*. A simple example of the non-stationarity of equity price is illustrated in Fig. 7. Recall that a process is strongly stationary whenever its distribution is invariant under time shifts [46]. Part of the Apple stock bid price on 12 June 2012 $P_t, t = 1, \dots, 5000$ is illustrated in Fig. 7. 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 $P_t, t = 3241, \dots, 5000$ (illustrated as the three parts in Fig. 7) are detected as significant changes

of the statistical features of the time series by *t-test* under the significance level 1%. A *t-test* module is then designed as a post-processing block of AHMMAS for detecting the statistical difference between the updated data sequence in the sliding window and the previous training data sequence and a popular significance level, 1%, is chosen for balancing the trade-off between the computation load and the performance.

The adaptive mechanism of AHMMAS is as shown in Figure 8. The model is initially trained by w data points in the *Training Window* ($F_t, t \in [t_{s1}, t_{e1}]$, where t_{s1} and t_{e1} are starting and ending time points respectively). A new incoming data point is then detected by the model. If detected as normal, it will be updated into the *Update Window* ($F_t, t \in [t_{s2}, t_{e2}]$) for testing the statistical difference. As illustrated in Fig. 8, the data in the sliding window are tested every time the sliding window is updated, but the model might be re-trained after working time period t . Since the scope of this paper is to detect the intra-day price manipulation activities, the *Training Window* is set to one day.

D. The Detection Algorithm

The algorithm for detecting the price manipulation activities by AHMMAS can then be summarized as the following steps.

Algorithm. 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^E using wavelet and gradient approaches described above; Obtain the features vector $F_{t_i} = [F_{t_i}^o, F_{t_i}^{og}, F_{t_i}^s, F_{t_i}^{sg}]$.

Step 3: Construct the AHMMAS detection model:

- Calculate the PDF separately for the features, $F_{t_i}^o, F_{t_i}^{og}, F_{t_i}^s, F_{t_i}^{sg}$, using the Gaussian Mixture Model and set the corresponding anomaly thresholds;
- Construct the sub-states for each attributes and then construct the hidden states by combining four attributes according to discussion in Section IV-B and the illustration shown in Fig.6.

Step 4: Train the AHMMAS 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 $P_{t_i}^E, t_i \in [t_w, t_{w+L-1}]$.

- calculate the four features of the sequence using the wavelet and gradient blocks,
- feed the features into AHMMAS model for calculating the probability of being specific states.

Step 6: If the state is identified as any “anomaly” states, the alert is triggered with the identified possible manipulation type and the probability of being the type.

Step 7: If the state is among the “normal” states, the *bid/ask* price in sliding window t_L to t_{w+L-1} are fed to the adaptive mechanism for model updating check. If model updating is needed, the algorithm flow goes to Step 1.

Step 8: If the 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

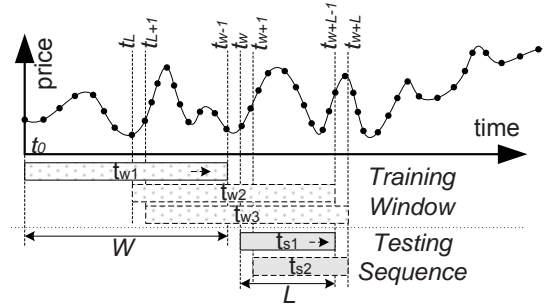


Fig. 8: Training and testing mechanism of price manipulation detection system.

the algorithm flow goes to Step 5.

Fig. 8 shows the initial states of the training window and the testing sequence and the sliding mechanism of each during the price manipulation detection process. In Fig. 8, the AHMMAS model was trained with the data in $tw_1 = P_{t_i}^E, t_i \in [0, w - 1]$. The first testing sequence was constructed as $ts_1 = P_{t_i}^E, t_i \in [t_w, t_{w+L-1}]$. If the detection indicate normal behaviours, the adaptive mechanism then checks for the significant changes between the data in the updated training window $tw_2 = P_{t_i}^E, t_i \in [t_L, t_{w+L-1}]$ and the original training window tw_1 . If no re-training is required, the training window will slide forward as tw_3 and a new testing sequence will be constructed as ts_2 to include the new data point at t_{w+L} and exclude the point at t_w . Similarly, if no anomaly is detected, the data in the updated training window $tw_3 = P_{t_i}^E, t_i \in [t_{L+1}, t_{w+L}]$ and in the original training window tw_1 are detected for adaptation. This detection and updating process continues for all incoming data streams.

V. EXPERIMENTS AND EVALUATION

Evaluating a detection model usually relies on the labelled benchmarks of both “normal” and “anomaly” cases. Since the absence of the real market “anomaly” cases, to evaluate the proposed detection model, it is acceptable to the financial industry business that all the characteristic patterns of reported manipulation examples are studied and then reproduced in other financial data context to synthesize exploratory manipulation cases [47]. Synthetic exploratory financial data are also accepted in academia for evaluating the proposed model when real market data are hard to collect [48], [49], [50]. Thus in this approach, the manipulation cases are synthetically generated and injected in any financial time series data while maintaining the normal statistical features such as the mean, variance and volatility.

A. Experimental Setup

The experimental data used in this work involves two categories. The data examples in the first category involve real market data of seven representative stocks: Google, Microsoft, Intel and Apple from NASDAQ (obtained from the LOBSTER project [51]) and ARM, BARCLAYS and Vodafone from London Stock Exchange (LSE). The selection of these datasets is

due to their relatively high trading volumes and price volatility, the factors that might increase the likelihood of manipulation across the exchanges [18], [52]. This dataset covers tick data over five trading days (11th-15th June 2012), and consists of more than 2,000,000 data points in total for each stock. The data sets have been examined by our financial partners to guarantee that no data are related to any reported manipulation cases by the regulators. Additionally, a statistical test widely used in trading firms [11] is performed on those datasets targeting intra-day volatility anomalies. The testing is defined as: 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 whole dataset. According to the manual examination and testing, no manipulation as well as volatility spikes are involved in those datasets.

The data examples in the second category are simulated stock prices generated using the stochastic differential equation (SDE): $dS_t = A(L - S_t)dt + \text{stochastic term}$, where A is the rate of the mean reversion, L is the value around which the process S_t tends to oscillate and the stochastic term is a Brownian motion process. The end result is that the stock price tends to be a modulated non-stationary process of periodic waves and stochastic oscillations [53]. It should be noted that the price is not assumed to be 100% following the SDE but is merely to test the validity and robustness of the detection model under any non-stationary randomness of equity processes. In this category, there are 10 simulated prices paths, each containing 2,000,000 time points as illustrated in Figure 9.

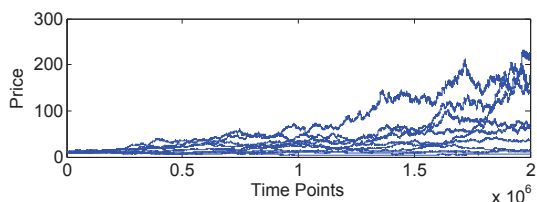


Fig. 9: 10 simulated prices paths.

Three typical manipulation patterns, shown in Fig. 2, are reproduced within each equity following the original characteristics: 6.9 bps sawtooth patterns in 819 ms, 18.6 bps square wave in 0.1 s and 800 bps pulse in 1 s. Those anomalies are then injected into the corresponding time series making the test data a mixture of both “normal” and “anomaly” patterns. The trained models are deployed on the mixed test data to detect anomaly patterns. This is practical and acceptable to business people [54], especially when it is very costly to obtain real manipulation cases.

Since the length of the sliding window for the adaptive mechanism is set to be one day, for the real market data in the first testcase category, the data on 11 June 2012, the first day among the five, are chosen as the training dataset, which contains around 400,000 points. The 5-fold cross-validation is used on the 400,000 data points for training the detection model. The data in days 12-15 June 2012 are set as the testing dataset. Similarly, for the simulated stock price in the second

category, the data are equally partitioned into five sets, each of which contains exactly 400,000 points. The training dataset consists of the first set and the testing dataset consists of the other four sets. The 5-folder cross-validation is used on the training dataset for constructing the detection model. Since the manipulation activities usually occur in an extremely short time period, the testing sequence length L (Fig. 8) is set to 1 minute, which is sufficient for one or two manipulation patterns.

To ensure comprehensive assessment of the approach, multiple testing datasets are built on both real market data and simulated data. For real stock prices, 300 synthesized patterns are injected to each dataset with each type having 100 examples (One pattern in Fig. 2 is considered as one anomaly example.). For the simulated stock prices, five groups of test-cases $n = 1, \dots, 5$ are generated and injected to each dataset, where group $n = 1$ contains $10 (\text{examples/type}) * 3 (\text{types}) = 30$ total numbers of injected anomaly examples; group $n = 2$ has $20 * 3 = 60$ numbers and group $n = 3, 4, 5$ has $40 * 3 = 120$, $80 * 3 = 240$, and $100 * 3 = 300$ total numbers of injected anomaly examples respectively.

As discussed in Section II-B, some generic computational models were used for manipulation detection such as SVM, neural network and logistic regression in [14], [15] as well as the rule based algorithms in [16], [17]. Since the rule based algorithms were specific for the special manipulation scenarios discussed in [16], [17], they are not suitable for general manipulation detection problems. According to those related work, we choose three popular computational models as the benchmarks of our model: one-class support vector machine (OCSVM) [45], k Nearest Neighbour (kNN) [39] and Gaussian Mixture Model (GMM) [39]. Those benchmark models are applied on the same feature vector $F_t = [F_t^o, F_t^{og}, F_t^s, F_t^{sg}]$. In our experiments, LIBSVM [55] and DDTool [56], are used as the implementation of the benchmark models. All the model parameters (i.e., kernel width for OCSVM, k value for kNN and the number of components for GMM) are fine-tuned through 5-folder cross-validation on each different dataset for stable and optimized testing results.

The performance evaluation of the proposed model is based on two types of metrics, the Receiver Operating Characteristic (ROC) and F-measure. Both of them are based on the confusion matrix, where *false positive* (FP), is defined as manipulation cases detected as normal and *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 [57]. The ROC curve plots the true positive rate ($\frac{TP}{TP+FN}$) against the false positive rate ($\frac{FP}{FP+TN}$) while the discrimination threshold of the binary classifier is varied. In order to assess the overall performance of a binary classifier, one can measure the area under the ROC curve (AUC). The maximum value of AUC is 1 and therefore larger AUC values indicate generally better classification performance. The ROC curve and AUC are used as the performance measure on seven real stock datasets. The F-measure integrates both precision and sensitivity into one

single metric and widely used in anomaly/novelty detection area [58]. The F-measure is used for evaluating the testing performance on simulated datasets.

B. Experimental Results

The ROC curves of four models on seven real-stock datasets with $3 \times 100 = 300$ numbers of injected novelties in each dataset are illustrated in Figure 10. To compute such a curve, the discrimination thresholds are set from 0.1 to 0.9 with increment, 0.01, for each model. The calculated AUC values for four models are summarized in Table II.

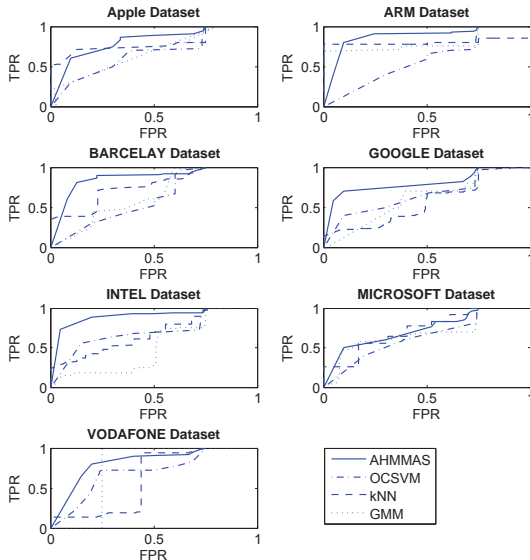


Fig. 10: ROC of four models on 7 stock datasets. (TPR: True Positive Rate; FPR: False Positive Rate.)

TABLE II: AUC of four detection models on seven real stock datasets

AUC	AHMMAS	OCSVM	kNN	GMM
AAPL	0.8142	0.6603	0.7926	0.6695
ARM	0.8270	0.5830	0.7982	0.7918
BARC	0.8710	0.6125	0.7627	0.6466
GOOG	0.8025	0.6593	0.5612	0.6163
INTC	0.8971	0.6970	0.6280	0.5200
MSFT	0.7336	0.6419	0.6250	0.6802
VOD	0.8775	0.7044	0.7278	0.7495

The AHMMAS model achieved the highest AUC on all of the seven real stock price datasets and it outperforms the benchmark models. The best AUC value for AHMMAS model appears on the Intel dataset (0.8971), which is 22%, 29% and 42% higher than OCSVM (0.6970), kNN (0.6280) and GMM (0.5200) models respectively. The second best AUC value for AHMMAS model is on the Vodafone dataset (0.8775), which is around 15-20% better than other three models. The lowest AUC value for AHMMAS model is on the Microsoft dataset (0.7336) and is still 12%, 14% and 7% higher than OCSVM (0.6419), kNN (0.6250) and GMM (0.6802) models respectively. The performance difference between the AHMMAS model and other three models in terms of the AUC

values are calculated on seven stock datasets as $(AUC_{AHMMAS} - AUC_{others})/AUC_{AHMMAS}$ and are illustrated in Fig. 11, where the legends, OCSVM, kNN and GMM, represent the performance differences between AHMMAS and OCSVM, kNN and GMM model respectively.

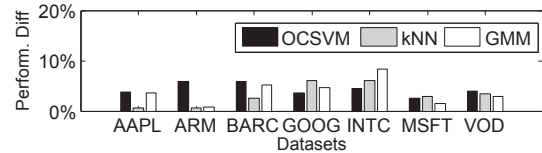


Fig. 11: Performance Difference between AHMMAS and other models on 7 datasets.

The lowest value of the difference is between the AHMMAS and kNN models on Apple dataset, where the AHMMAS model (0.8142) performed 2.6% better than the kNN model (0.7926). The minimum value indicates the roughly identical performance of AHMMAS and kNN models on Apple dataset. The largest value of the difference is between the AHMMAS and GMM models on Intel dataset, where AHMMAS model (0.8971) performed 42% better than GMM model (0.5200). This is where AHMMAS performs the best among all the testcases. From Fig. 11, it is also obviously that the performance difference between AHMMAS and OCSVM models is relatively stable in range from 12% to 30% over different datasets while the other two performance differences are both volatile in range from 2% to 46%.

The performance comparison shows that the AHMMAS model retains its advantage over all other models on seven real market datasets. It also indicates the performance stabilities of the three benchmark models, among which, OCSVM model performed relatively steady while the kNN and GMM model performed volatile across the different datasets.

The performance of the four models on ten simulated prices datasets is evaluated based on F-measure metrics as shown in Fig. 12 where the horizontal axis represents the number of injected anomaly examples and the vertical axis represents the value of the F-measure. As discussed before, the models are tested on ten simulated datasets, each of which contains five groups of injected anomaly examples. Thus 50 different experiments are carried out as a robust evaluation plan for each model.

From the results in Fig. 12, the AHMMAS model achieved the highest F-measure among the four models on most of the 10 datasets. In the case of simulated price #6 with 10 injected anomaly examples, the initial F-measure for the AHMMAS model (0.8081) is lower than that of OCSVM (0.8791). But the performance of AHMMAS increased on this testcase as the number of anomaly examples increased. Another case is on simulated price #7 with 10 injected anomaly examples, the F-measure for AHMMAS model (0.8535) is very close to kNN model (0.8506) but increased on the testcases with 20 injected anomaly examples.

C. Discussion

The experiments on real and simulated price datasets have shown that the AHMMAS model outperforms the other three

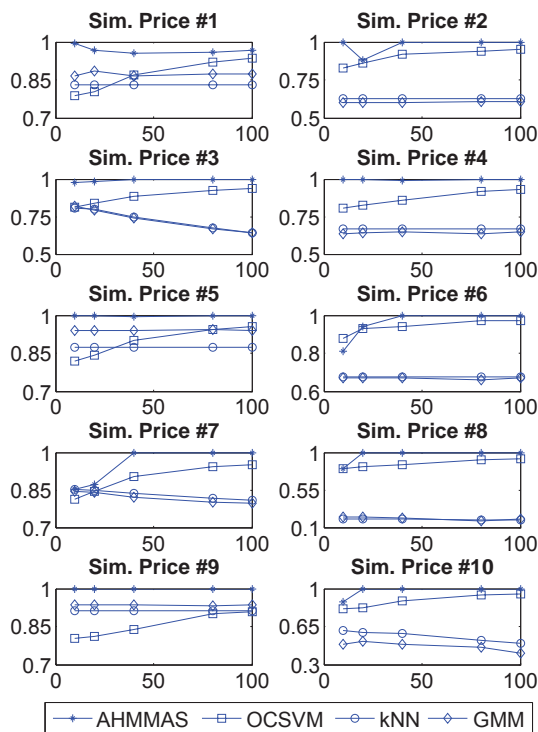


Fig. 12: F-measure of four models on simulated prices. (X-axis: Number of Anomalies; Y-axis: F-measure.)

benchmark models: OCSVM, kNN and GMM. The robustness can be explained by the inherited nature from Hidden Markov Model on time series data modelling, where the algorithm does not simple-mindedly accept the most likely state for a given time instant, but takes a decision based on the whole sequence. Thus, if there are some particularly normal events midway through a sequence of anomalies, this will not matter provided the whole context of what is seen is reasonable. This is particularly valuable in price manipulation detection where normal and manipulation trading behaviours may be interleaved, but the overall sense of the events may be detectable. However, other detection models are trained for building tightly enclosed boundaries, which consider the testing examples as a set of separate points.

Another reason for the robust test results is the adaptive mechanism. Due to the non-stationary feature, the distribution of financial time series data x_{t_1}, \dots, x_{t_l} may deviate from $x_{t_{1+\tau}}, \dots, x_{t_{l+\tau}}$. Detecting the statistically deviated data by a constant model increases the risk of obtaining incorrect detections. The mechanism of capturing the significant difference and updating the detection model provides a reasonable compensation for the non-stationarity.

The AHMMAS model performs distinctly on different datasets. In Table II, most of the AUC values for AHMMAS model go beyond 0.8, which is considered as a threshold of good performance for a classifier [57]. However, the result for Microsoft dataset is lower than others (0.7336) and is even 8.5% lower than the 2nd lowest AUC (0.8025) on Google dataset. Further investigation showed that the small AUC on Microsoft dataset is due to the relatively high *false negative*

numbers (normal cases detected as anomaly) decreasing the *true positive rate* on each discrimination threshold. After examining the *false negative* misclassification cases and comparing the training and testing datasets, it was discovered that the normal patterns in the testing dataset that failed to be detected as normal were in fact identical to the anomaly patterns shown in Fig. 2 and were never seen in the training dataset. In the example of the normal Microsoft testing dataset (no anomaly pattern injected) shown in Fig. 13, the *bid* price from 8:00:00am to 8:01:00am on 12 June 2012 oscillated unusually and showed mixed patterns of sawtooth and square wave, which are very easily identified as the anomalies by the detection model.

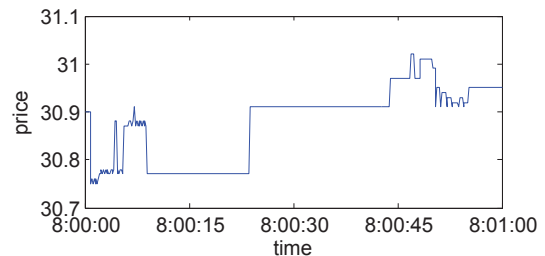


Fig. 13: Microsoft stock *bid* price from 8:00:00am to 8:01:40am on 12 Jun 2012: Normal data showed anomaly patterns.

Consultation with financial experts from industry suggested that although rarely, such “normal” but wild oscillation sometimes happen especially when experiencing high frequency trading (HFT). Although HFT is out of the scope of this work due to its hyper-fast trading speed (e.g., 1000 updates in 1 millisecond), the wild oscillations in Fig. 13 show analogous patterns as the manipulation algorithm triggered by HFT. The oscillation also conforms with the recent report from Nanex [59] that stock of Paychex showed the wild sawtooth oscillation when being hit by a high frequency trading algorithm. The wild pattern is identical to the manipulation case documented in 2012 (Fig. 2(a)) although not reported by regulators yet. Therefore, although the *false negative* cases (normal cases mis-classified as anomaly) seem to decrease the detection performance, most of them referred to suspicious trading activities which are identical to the reported manipulation cases.

From the experimental results and the analysis of the misclassification, it can be concluded that the AHMMAS model is better suited for the price manipulation detection problem based on a *bid/ask* price time series and its performance is consistently superior to OCSVM, kNN and GMM models under most of the randomness of the non-stationarity of the underlying equity price.

Based on consultations with financial experts, “real-time” surveillance in practice is required by the trading floor to recognise any unusual patterns, which are used as a reference to adapt their trading behaviours accordingly. Therefore, they do not conclusively determine the intention behind every anomaly or the contextual relation between those patterns but mitigate any possible negative impacts on their portfolios from the obvious problems in the market (e.g. the flash crash).

Instead, the proposed method determines the manipulation by considering the anomalies as well as their contextual relations (through the Markov chain). The detection increases the accuracy while sacrificing the efficiency for “real-time” detection, which essentially may not be the first priority for regulators. Therefore, AHMMAS best suits the “over-night” detection in real world considering the computational complexity of the proposed method.

VI. CONCLUSION AND FUTURE WORK

This paper presented a computational intelligence approach for price manipulation detection. Typical price manipulation cases were studied, where special patterns of the *bid* and *ask* price during a manipulation period were analysed for extracting reliable features. Based on the extracted features, an adaptive Hidden Markov Model with anomaly States (AHMMAS) was proposed for detecting the anomalies in the *bid* and *ask* prices. The AHMMAS considers the anomaly states according to the thresholds of four extracted features set by the probability density functions (PDF) of the features. A system has been developed and intensively tested on both real-life stock prices and simulated prices. The comparison of proposed approach with other benchmark models, OCSVM, kNN and GMM, has shown that the AHMMAS performs better in terms of the area under ROC curve and the F-measure, respectively. Finally, the experimental results were analysed and misclassification cases were discussed and further explained with examples.

In the AHMMAS model, the probability density functions (PDF) of the four extracted features are learned individually using Dirichlet Process Gaussian Mixture Model (DPGMM). However, the use of a joint-PDF may offer a more accurate approach for future studies since there is no empirical study showing the independence relationship of the four features and this will be the focus of our future work. Furthermore, modelling the *bid/ask* time series with the corresponding order information may provide a mechanism for decreasing the false negative rate mentioned before. SST may also provide an enhanced signal decomposition technique by extending the traditional wavelet. Meanwhile, the exploration of coupled behaviours of the *bid* and *ask* prices of one stock might help to enhance the detection results of potential price manipulation strategies that target both *bid* and *ask* price for maximising profits.

Additionally, the adaptive mechanism compensated the non-stationary features of the financial time series. However, the re-training processes increased the computational complexity. The rapidly growing trading frequency increased the non-stationarity of the time series and at the same time decreased the tolerance of latency for model re-training. But an increased non-stationarity requires more re-training. To solve this contradictory problem, additional to the adaptive mechanism, one strand of the future works might be building a method that removes or partially removes the non-stationarity while maintaining the necessary statistical features.

REFERENCES

- [1] F. Allen and D. Gale, “Stock price manipulation,” *The Review of Financial Studies*, vol. 5, no. 3, pp. 503–529, 1992.
- [2] F. Allen, L. Litov, and J. Mei, “Large investors, price manipulation, and limits to arbitrage: An anatomy of market corners,” *Review of Finance*, vol. 10, no. 4, pp. 645–693, 2006.
- [3] Y. S. Abu-Mostafa, A. F. Atiya, M. Magdon-Ismael, and H. White, “Introduction to the special issue on neural networks in financial engineering,” *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 653–656, 2001.
- [4] B.-L. Zhang, R. Coggins, M. Jabri, D. Dersch, and B. Flower, “Multiresolution forecasting for futures trading using wavelet decompositions,” *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 765–775, 2001.
- [5] T. Van Gestel, J. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, “Financial time series prediction using least squares support vector machines within the evidence framework,” *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 809–821, 2001.
- [6] C. Bergmeir, I. Triguero, D. Molina, J. Aznarte, and J. Benitez, “Time series modeling and forecasting using memetic algorithms for regime-switching models,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1841–1847, 2012.
- [7] A. Miranian and M. Abdollahzade, “Developing a local least-squares support vector machines-based neuro-fuzzy model for nonlinear and chaotic time series prediction,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 2, pp. 207–218, 2013.
- [8] K. Menyah and K. Paudyal, “The components of bid-ask spreads on the london stock exchange,” *Journal of Banking & Finance*, vol. 24, no. 11, pp. 1767 – 1785, 2000.
- [9] “Market abuse, amended proposal for a directive,” EUROPEAN COMMISSION, 2012.
- [10] D. Cumming, S. Johan, and D. Li, “Exchange trading rules and stock market liquidity,” *Journal of Financial Economics*, vol. 99, no. 3, pp. 651–671, 2011.
- [11] J. Tse, X. Lin, and D. Vincent, “High frequency trading - measurement, detection and response,” Credit Suisse, Tech. Rep., 2012.
- [12] FINRA, “Finra joins exchanges and the sec in fining hold brothers more than \$5.9 million for manipulative trading, anti-money laundering, and other violations,” <http://www.finra.org/Newsroom/NewsReleases/2012/P178687>, 2012.
- [13] Nanex, “Whac-a-mole is manipulation,” www.nanex.net/aqck2/3598.html, 2012.
- [14] H. Ögüt, M. Mete Doğanay, and R. Aktaş, “Detecting stock-price manipulation in an emerging market: The case of Turkey,” *Expert Systems with Applications*, vol. 36, no. 9, pp. 11 944–11 949, 2009.
- [15] 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. 12 757 – 12 771, 2011.
- [16] J. Mongkolnarin and S. Tirapat, “Marking the close analysis in Thai Bond Market surveillance using association rules,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8523 – 8527, 2009.
- [17] M. J. Aitken, F. H. deB. Harris, and S. Ji, “Trade-based manipulation and market efficiency: A cross-market comparison,” in *22nd Australasian Finance and Banking Conference*, 2009.
- [18] E. J. Lee, K. S. Eom, and K. S. Park, “Microstructure-based manipulation: Strategic behavior and performance of spoofing traders,” *Journal of Financial Markets*, vol. 16, no. 2, pp. 227 – 252, 2013.
- [19] A. Ferraris, “Equity market impact models: Mathematics at the interface between business and research,” *Stifterverband für die Deutsche Wissenschaft, Berlin*, 2008.
- [20] 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.
- [21] D. Ian, “Market abuse and surveillance,” www.bis.gov.uk/assets/foresight/docs/computer-trading/12-1076-eia17-market-abuse-and-surveillance, 2012.
- [22] J. Tse, X. Lin, and D. Vincent, “High frequency trading - the good, the bad and the regulation,” Credit Suisse, Tech. Rep., 2012.
- [23] SEC, “Securities and exchange commission release no. 34-69287,” www.sec.gov/rules/sro/nms/2013/34-69287.pdf, 2013.
- [24] H. Zhang and Q. Zhang, “Trading a mean-reverting asset: Buy low and sell high,” *Automatica*, vol. 44, no. 6, pp. 1511–1518, 2008.
- [25] A. Cowles and H. E. Jones, “Some a posteriori probabilities in stock market action,” *Econometrica, Journal of the Econometric Society*, pp. 280–294, 1937.

- [26] C. M. Hafner and H. Herwartz, "Option pricing under linear autoregressive dynamics, heteroskedasticity, and conditional leptokurtosis," *Journal of Empirical Finance*, vol. 8, no. 1, pp. 1–34, 2001.
- [27] L. P. Bos, A. F. Ware, and B. S. Pavlov, "On a semi-spectral method for pricing an option on a mean-reverting asset," *Quantitative Finance*, vol. 2, no. 5, pp. 337–345, 2002.
- [28] R. R. Bliss and N. Panigirtzoglou, "Testing the stability of implied probability density functions," *Journal of Banking & Finance*, vol. 26, no. 2-3, pp. 381 – 422, 2002.
- [29] E. Haven, X. Liu, and L. Shen, "De-noising option prices with the wavelet method," *European Journal of Operational Research*, vol. 222, no. 1, pp. 104 – 112, 2012.
- [30] Z. Wu and N. E. Huang, "A study of the characteristics of white noise using the empirical mode decomposition method," *Proceedings of The Royal Society*, vol. 460, no. 2046, pp. 1597–1611, 2004.
- [31] M. Feldman, "Analytical basics of the emd: Two harmonics decomposition," *Mechanical Systems and Signal Processing*, vol. 23, no. 7, pp. 2059 – 2071, 2009.
- [32] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992, vol. 61.
- [33] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 243 – 261, 2011.
- [34] A. Ahrabian, C. Took, and D. Mandic, "Algorithmic trading using phase synchronization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 4, pp. 399–404, 2012.
- [35] M. A. L. Caetano and T. Yoneyama, "A new indicator of imminent occurrence of drawdown in the stock market," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 17, pp. 3563 – 3571, 2009.
- [36] J. C. Hull, *Options, Futures and Other Derivatives*, 8th ed. Pearson Education, 2011.
- [37] 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, pp. 2359 – 2367, 2009.
- [38] F. Allen and G. Gorton, "Stock price manipulation, market microstructure and asymmetric information," National Bureau of Economic Research, Tech. Rep., 1991.
- [39] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.
- [40] R. M. Neal, "Markov chain sampling methods for dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 249 – 265, 2000.
- [41] P. Guo, Z. Miao, X.-P. Zhang, Y. Shen, and S. Wang, "Coupled observation decomposed hidden markov model for multiperson activity recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 9, pp. 1306 –1320, Sep 2012.
- [42] L. Wang, M. Mehrabi, and E. Kannatey-Asibu, "Hidden markov model-based tool wear monitoring in turning," *Journal of Manufacturing Science and Engineering*, vol. 124, no. 3, pp. 651–658, 2002.
- [43] D. Y. Yeung and Y. Ding, "Host-based intrusion detection using dynamic and static behavioral models," *Pattern Recognition*, vol. 36, no. 1, pp. 229 – 243, 2003.
- [44] D. A. Clifton, L. Tarassenko, C. Sage, and S. Sundaram, "Condition monitoring of manufacturing processes," *Proceedings of condition monitoring 2008*, pp. 273–279, 2008.
- [45] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.
- [46] G. Grimmett and D. Stirzaker, *Probability and random processes*. Oxford University Press, USA, 2001.
- [47] NANEX, "Exploratory trading - Top 8 HFT take liquidity 59% of the time," www.nanex.net/aqck2/4136.html, 2013.
- [48] G. K. Palshikar and M. M. Apte, "Collusion set detection using graph clustering," *Data Min. Knowl. Discov.*, vol. 16, no. 2, pp. 135–164, Apr. 2008.
- [49] Y. Ou, L. Cao, C. Luo, and C. Zhang, "Domain-driven local exceptional pattern mining for detecting stock price manipulation," in *Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, ser. PRICAI '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 849–858.
- [50] M. Franke, B. Hoser, and J. Schröder, "On the analysis of irregular stock market trading behavior," in *Data Analysis, Machine Learning and Applications*. Springer, 2008, pp. 355–362.
- [51] LOBSTER, "Limit order book system," <http://www.lobster.wiwi.hu-berlin.de>, 2012.
- [52] D. J. Cumming, F. Zhan, and M. J. Aitken, "High frequency trading and end-of-day manipulation," York University, Tech. Rep., 2012.
- [53] B. Avanzi and B. Wong, "On a mean reverting dividend strategy with brownian motion," *Insurance: Mathematics and Economics*, vol. 51, no. 2, pp. 229 – 238, 2012.
- [54] L. Cao, Y. Ou, and P. Yu, "Coupled behavior analysis with applications," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1378–1392, Aug.2012.
- [55] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
- [56] D. Tax, "DDTools, the Data Description Toolbox for MATLAB," May 2012, version 1.9.1.
- [57] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006.
- [58] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, "SVMs Modeling for Highly Imbalanced Classification," *IEEE Trans. Syst., Man, Cybern. B*, vol. 39, no. 1, pp. 281 –288, 2009.
- [59] Nanex, "There is no stock market regulator," www.nanex.net/aqck2/4183.html, 2013.



Yi Cao received the B.Eng. in navigation and control in aeronautics from Beijing University of Aeronautics and Astronautics in 2002 and the M.S. degree in computer science from Florida International University, USA, in 2005. He is currently pursuing the Ph.D. degree in computational finance at University of Ulster. From 2005 to 2011, he was an ASIC and a senior system engineer in Vimicro, Conexant and ERICSSON, Beijing respectively.



Yuhua Li received the PhD degree in general engineering from the University of Leicester. He worked at the Manchester Metropolitan University and then the University of Manchester from June 2000 to September 2005 as a Senior Research Fellow and a Research Associate, respectively. Since October 2005, he has been a lecturer at the School of Computing and Intelligent Systems, the University of Ulster. His research interests include pattern recognition, machine learning, knowledge-based systems, signal processing, and condition monitoring and

fault diagnosis.



Sonya Coleman is a Reader in ISRC. She has a first class honours degree in Mathematics, Statistics and Computing, and a doctorate from the University of Ulster, and is a member of the IEEE. She is the author or co-author of over 80 research papers in image processing, robotics and computational neuroscience. She has experience of managing research grants (with respect to technical aspects and personnel) as both a principal and co-investigator. Additionally she is co-investigator on the EU FP7 funded projects RUBICON and VISUALISE.



PLACE
PHOTO
HERE

A. Belatreche received a PhD in Computer Science from the University of Ulster, UK. He worked as a Research Assistant in the Intelligent Systems Engineering Laboratory and is currently a lecturer at the School of Computing and Intelligent Systems, University of Ulster. His research interests include bio-inspired adaptive systems, machine learning, pattern recognition and image processing and understanding. He is a Fellow of the Higher Education Academy, a member of IEEE CIS and the Northern Ireland representative of the IEEE CIS. He is an

associate editor of *Neurocomputing* and has served as a PC member and reviewer for several international conferences and journals.



PLACE
PHOTO
HERE

T. Martin McGinnity received a First Class (Hons.) degree in Physics in 1975, and a Ph.D. degree from the University of Durham, Durham, UK in 1979. He is Professor of Intelligent Systems Engineering within the Faculty of Computing and Engineering, University of Ulster, Derry, Northern Ireland. He is currently the Director of the Intelligent Systems Research Centre, which encompasses the research activities of approximately 100 researchers. Formerly, he was Associate Dean of the Faculty and Director of the University's technology transfer company,

Innovation Ulster, and a spin out company Flex Language Services. He is the author or coauthor of approximately 300 research papers and has attracted over £24 million in research funding to the University. Prof. McGinnity is a Fellow of the IET, SMIEEE and a Chartered Engineer.