Stock Price Manipulation Detection using Variational Autoencoder and Recurrence Plots

***Abstract*— Stock price manipulation refers to deceptive traders' practices which aim to influence the normal market behaviour in order to make illicit profit at the expense of other genuine market participants. Such manipulation undermines investors' confidence in the financial market and damages its efficiency and integrity. There remains a growing need for developing robust anomaly detection methods capable of reliably identifying increasingly sophisticated manipulation attempts. Inspired by the recent success of deep learning in computer vision, we propose a novel approach for stock market manipulation detection that leverages the combined power of Recurrence Plots (RP) and beta Variational Autoencoders (beta-VAEs). The proposed approach first splits stock price time series data into overlapping temporal windows, each of which is transformed into a colour image using Recurrence Plots (RP). Then, a beta-VAE network composed of two Convolutional Neural Networks (CNNs) is trained on these images derived from normal market activity. The resulting model effectively learns the inherent characteristics of normal (i.e legitimate) trading behaviour. Finally, the mean square error between the original images and the reconstructed ones is used as the manipulation (i.e anomaly) detection score measure for flagging significant deviations indicative of potential manipulation. The efficacy of the proposed approach is rigorously validated on 1-level tick data obtained from the LOBSTER project. Evaluation results demonstrate superior manipulation detection performance which is evidenced by a highly promising area under the ROC curve (AUC). The robustness of this performance can be attributed to the beta-VAE's ability to extract pertinent features of normal market behavior from the proposed RP-generated 2D representations of stock price data. This novel framework paves the way for the development of more effective market abuse detection systems which contribute to a safer, fairer and more transparent financial ecosystem for both investors and regulators alike.**

Keywords*— *Stock market, stock price manipulation, market abuse, market surveillance, anomaly detection, Recurrence Plot, variational autoencoder network.

I. INTRODUCTION

Stock markets play a key role in the development of a domestic economy. They help stimulate investment and raise capital for both public and private sector companies, create personal wealth and serve as barometer of a state economy. Stocks are listed and traded on stock exchanges which provide real-time price information on all securities being traded on that exchange and enable a fair and orderly trading for all market participants. However, in some cases stock markets can be abused through manipulative practices whereby some market players deliberately interfere with the free and fair operation of the market in order to influence targeted stock prices and make illicit profits at the expense of other genuine market participants. Such illicit manipulations undermine investors' confidence in the financial markets and damage their efficiency and integrity. Financial regulatory authorities in most countries combat market abuse and constantly monitor trading activities for any potential forms of market manipulation. One of the most prominent cases of market manipulation which led to the establishment of existing systems and institutions is the complicity of groups of traders (such as investment banks, market makers etc.) to raise the stock prices on the New York market through spreading favorable rumors about a firm, which led to the adoption of the Securities of the Financial Markets act in 1933 and the creation of the United States Securities and Exchange Commission (SEC) in 1934 [1].

The illegal activities which come under market manipulation have been classified into three main forms by Allen and Gale[1]: information-based manipulation, action-based manipulation, and trade-based manipulation. Information-based manipulation occurs when the manipulators spread false rumors or release some inside information about a company or its stock, hence affecting their value expectations. Action-based manipulation occurs when the manipulators acquire shares in a particular company then announce a takeover offer. This causes a rise in the company stock and helps the manipulators to sell their holdings at a higher price [2]. Trade-based manipulation occurs when the manipulators perform transactions that have direct effects on the prices and quantities traded in the market, which influences the expectations of other traders about the value of the securities being traded. Unlike action-based and information-based manipulations, in trade-based manipulation the manipulators intentionally use fraudulent and disruptive trading practices by submitting (or cancelling) multiple manipulative trade orders (bids or offers) to create artificial price movements or to influence prices or other investors' trading. Furthermore, while action-based and information-based manipulations can be prevented through appropriate enforcement of relevant laws and regulations, trade-based manipulation of stock prices can be difficult to spot and remains harder to eradicate [3].

On August 28th, 2014 the Chicago Mercantile Exchange (CME) submitted the text of new Rule 575 [4] that prohibits several types of disruptive transactions and manipulative practices. This rule prohibits the spoofing activity, an example of trade-based manipulation, which includes placing non bona fide orders with the intention not to be executed but rather to create misleading market conditions or to artificially move market prices upwards or downwards then unfairly make profit by sending a bona fide order on the opposite side of the market. The CME Rule 575 also prohibits quote stuffing activity, another form of trade-based manipulation, where traders enter multiple orders and cancel them immediately for the purpose of overloading the quotation system and delaying the execution of other market participants' trades.

There remains a growing need for developing enhanced automatic methods that are capable of reliably detecting today's complex and sophisticated market manipulation attempts which pose a major challenge for the regulatory authorities. Traditional data mining approaches have been proposed to automatically identify various types of stock manipulation [5-12]. However, while some of these approaches need labelled datasets which is usually not available due to privacy and confidentiality issues in most markets, other approaches need very complex feature engineering that requires specialist domain knowledge to craft pertinent representations of the stock data. On the other hand, deep learning has become increasingly popular and has been successfully applied in various applications which is owed to the ability of deep learning models to learn complex feature representations from massive volumes of data through a hierarchical learning process.

Inspired by the recent success of deep learning in computer vision, we present a novel approach for detecting stock price manipulation. The proposed approach leverages the power of combining beta variational autoencoder and recurrence plots encoding method. This approach involves learning the normal behavior of bid/ask time series using beta variational autoencoder. Unknown manipulation schemes are then detected using an anomaly score that measures how much the corresponding data point deviates from the normal behavior. The main contribution of our work lies in in investigating the efficacy of using deep generative models in tandem with with recurrence plot encoding methods. this approach aims to detect both known known and unknown manipulation schemes solely by analysing the behavior of univariate stock bid/ask time series data. To achieve this, we first split the normal bid/ask time series data into segments of multiple timestamps using a fixed sliding window size. Next, we transform each segment into phase spaces while retaining the temporal correlation between data points in each segment. Then, we encode each transformed segment as a colour image using recurrence plot encoding methods. A deep generative model is then trained to learn the reconstruction of normal images obtained from normal stock price data. Finally, stock price encoded images that result in a high reconstruction error when fed to the trained model are identified as manipulations (anomalies). We validate our approach on 1-level tick data obtained from the LOBSTER project and demonstrate its ability to achieve significant manipulation detection performance in terms of the AUC. To the best of our knowledge, this is the first study that explores the viability of combining deep generative models with recurrence plots methods for detecting known and unknown types of stock price manipulation tactics. The remainder of this paper is organised as follows: Section II reviews existing work related to stock price manipulation detection. Section III presents the proposed approach for stock price manipulation based on beta-VAE and recurrence plots, including a discussion of variational autoencoders (VAEs). Section IV presents the experimental evaluation of the proposed approach, discusses the obtained performance results and their significance. Finally, section V concludes the paper and discusses the implications of our findings on potential directions for future work in this area.

II. RELATED WORK

The detection of stock market manipulation can broadly be classified into two categories: supervised and unsupervised techniques. Öğüt et al. [5] employed supervised learning techniques to identify market manipulation in stocks traded on the Istanbul Stock Exchange. They analysed statistical features of daily volatility, daily trading volume, and daily return, and employed support vector machines (SVM) and artificial neural networks (ANNs) to detect manipulation. Their method outperformed multivariate statistical techniques such as discriminant analysis and logistic regression. However, this approach is limited to detecting known manipulation strategies and may fail to discover previously unseen manipulation schemes. Diaz et al. [6] employed a combination of three decision tree algorithms, namely QUEST, C5.0, and C&RT, to detect intraday price manipulation. The proposed model was applied to a dataset constructed from the SEC (US) manipulation cases in 2009. A set of 10 features including different technical indicators that reflect changes in trading liquidity, volatility, returns, and abnormal returns was used. The statistical features selected by these methods were based on empirical research which assumed that changes in particular market variables indicate manipulation. However, there is no evidence that such changes necessarily condition manipulation, as other market events, such as news events, may also cause a change in a market variable. Therefore, the proposed methods may be ineffective in detecting real cases of manipulation.
Golmohammadi et al. [7] adopted different classification methods, including CART, conditional inference trees, C5.0, Random Forest, Naïve Bayes, Neural Networks, SVM, and KNN, to detect market manipulation. Unlike Diaz et al. [6], who only used the price of stocks for feature extraction, Golmohammadi et al. [7] used the percentage return price of a stock as a feature. They argued that the price of securities did not necessarily reflect the size of a company or its revenue. They also used the same dataset of manipulation cases employed by Diaz et al. [6]. Their study found that the Naïve Bayes classifier outperformed other classifiers, with an F2 measure of 53%. However, the statistical features used in this method make it suitable only for detecting individual manipulations, and it may fail to detect collective manipulations such as the quote stuffing strategy. Uslu et al. [8] developed a machine learning-based approach to detect trade-based manipulations in the Borsa Istanbul (BIST) capital market. The dataset used in their study included 22 cases of manipulation that occurred between 2010 and 2015, and the authors trained supervised machine learning classification models using daily data of manipulated stocks. The performance of the proposed model was assessed using the accuracy, sensitivity, and F1 score metrics, and the model was found to outperform LR (Logistic Regression), NB (Naive Bayes), and SVM (Support Vector Machine) in detecting trade-based manipulation in the stock market. Specifically, this model achieved an F1 score of 91%, a sensitivity of 95%, and an accuracy of 93%, which indicates its ability to accurately identify potential market manipulation with high precision and recall in the BIST capital market.

In the context of stock price manipulation detection, labelled instances of both genuine and manipulative trades are necessary for supervised methods to work. However, obtaining labelled manipulation data is a challenging problem due to confidentiality and the proprietary nature of trading data in most markets. As a result, the development of unsupervised learning algorithms is necessary to address this problem. Cao et al. [9] utilized machine learning techniques, specifically the One-Class Support Vector Machines (OCSVM) and K-Nearest Neighbours (KNN), to identify various forms of market manipulation, including spoofing trading and quote stuffing. To achieve this, they transformed the data into pseudo-stationary time series and constructed a feature vector comprising order price, volume, and submission time. This approach was assessed on a level 1 tick dataset of four stocks, namely Apple, Google, Intel, and Microsoft, obtained from the LOBSTER project, covering five trading days in June 2012. The obtained performance results demonstrated that the detection models using the transformation technique outperformed the models that used original market data in terms AUC. However, one-class classification methods such as OCSVM and KNN treat each data vector as a single point in a high-dimensional space with no temporal structure, which can be problematic when analysing time series or sequential data, such as stock market data. To address this limitation, the same authors, Cao et al. [10], proposed an adaptive Hidden Markov Model with Anomaly States (AHMMAS) that incorporates temporal dependencies. The AHMMAS approach employs continuous wavelet transformation and gradient information as features, extracted from trading instances. The reported results indicate that AHMMAS outperforms other benchmark models, including OCSVM, KNN, and Gaussian Mixture Models (GMM), in terms of AUC and F-measure. However, it is important to note that this method focuses solely on one specific form of market manipulation, namely quote stuffing, by analysing the most aggressive buy/sell orders, hence it overlooks other manipulation behaviours.

Abbas et al.[11] presented an unsupervised method for detecting price manipulation using Kernel Density Estimation (KDE) with Empirical Mode Decomposition (EMD) as feature extraction. The proposed approach was evaluated on a level 1 tick dataset of five stocks (Apple, Amazon, Google, Microsoft, and Intel Corp) obtained from the LOBSTER project, and outperformed other unsupervised methods based on Principal Component Analysis (PCA), k-Means, and Dirichlet Process Gaussian Mixture Model (DPGMM) in terms of AUC. In a separate work, the same authors, Rizvi et al. [12], proposed a semi-supervised learning method based on a modified bio-inspired dendritic cell immune system based and KDE for detecting price manipulation, which was also tested on the same dataset used in [11]. The results indicated that this method performed better than other approaches such as K-Means, PCA, k-Nearest Neighbours (K-NN), and One-Class Support Vector Machines (OCSVM). However, these approaches, [11] and [12], relied on manual feature extraction, which required identifying and describing features relevant for a given type of manipulation scheme. For instance, the proposed approach used the sawtooth and spike patterns to detect manipulation, and the change between two consecutive data orders exceeding a given threshold as a feature. Therefore, these methods may not be suitable for detecting various types of manipulation tactics and are only appropriate for detecting well-known manipulations such as pump and dump and layering patterns.

In another work, Rizvi et al. [13] proposed another unsupervised learning approach using Kernel Principal Component Analysis (KPCA) to extract latent features, and then applied Multidimensional Kernel Density Estimation (MKDE) clustering on the selected components to identify abnormal patterns of manipulation in the data. This approach outperformed other methods, including the adaptive hidden Markov model with anomaly states, Naïve Bayes, Probabilistic Neural Network (PNN), and PGA, in terms of F-measure values and false alarm rate (FAR). The dataset used in this approach involved thirteen different stocks. However, the proposed MKDE clustering did not consider the time dependence between data points in the time series, which is an important factor in detecting various manipulation schemes. Furthermore, the approach treated the problem of detecting multi-order manipulation as a single-order manipulation detection by splitting the time series into different windows and treating each window as a single point in a high-dimensional feature space disregarding any temporal correlations. Such an approach may lead to unexpected manipulation detection as small changes in a single feature may result in different clusters.

Leangarun et al. [14] developed a technique based on Generative Adversarial Networks (GANs) to detect pump and dump manipulation. Their approach used a Long Short-Term Memory (LSTM) network in both the generator and the discriminator networks, with the discriminator network serving as an anomaly detector. The authors evaluated their method on data obtained from the Stock Exchange of Thailand (SET) and reported a detection accuracy of 68.1% when tested on unseen market data. However, this approach focused only on pump and dump manipulation strategy and did not consider generalisation to other various manipulative schemes. In another study, Leangarun et al. [15] proposed an unsupervised learning approach that used deep neural networks to detect stock price manipulation. Specifically, the models were trained to recognize normal trading behaviors expressed in a limit order book and identify anomalous trading actions that did not conform to the learned patterns as potentially manipulative. To evaluate the effectiveness of their approach, the authors applied two model architectures, autoencoder (AE) and generative adversarial networks (GANs), to six real cases of manipulation that occurred in the Stock Exchange of Thailand (SET) between 2004 and 2016. The results indicated that both models had a low false-positive rate and were able to detect five out of the six cases, with "MinManiMax" strategy introduced to optimise the decision boundary for practical application of the models. However, the approach had a low recall rate and limited ability to identify new manipulation patterns. To address these issues, Chullamonthon et al. [16] developed an ensemble approach that combined the unsupervised LSTM-based autoencoder model with a supervised long short-term memory (LSTM) network that incorporated knowledge of the popular pump-and-dump pattern. This combined model achieved a higher detection rate with low false alarm rates on the same six real manipulation cases from the SET, demonstrating the potential of integrating supervised and unsupervised learning methods to improve the accuracy of detecting stock price manipulation. In a related study, Wang et al. [17] proposed a recurrent neural network-based ensemble learning (RNN-EL) approach for stock price manipulation detection. The authors created a labeled dataset that included trade-based features, such as volatility, turnover rate, stock returns, and abnormal returns, as well as company-specific features, including market capitalisation, share proportion of large shareholders, stock circulation share ratio, fund holding, and bid-ask spread. They found that incorporating company-related features can lead to better results in detecting trade manipulation schemes. Their reported AUC performance surpassed that of other state-of-the-art approaches proposed by Öğüt et al. [5] , Diaz et al. [6] , and Cao et al. [10]. However, since the dataset used in this work, which is built based on the prosecuted manipulation cases reported by the China Securities Regulatory Commission (CSRC), was supervised in nature, this method may not be suitable for detecting other manipulative strategies. Furthermore, the use of recurrent neural networks in this method may not be effective for detecting manipulation in stocks where a long history is required, as such networks have limited memory to capture the temporal interdependency of the data. Most proposed approaches for detecting trade-based manipulation require a complex feature engineering step that demands domain knowledge from experts to create a pertinent representation of equity prices. Furthermore, these methods typically focus only on specific manipulation schemes and require manual identification and description of relevant features. Also, these approaches treat trade-based manipulation as a single point anomaly detection problem, disregarding the temporal interdependencies that usually characterise sequential trading activities.

In this study, we propose a novel approach for effective detection of both seen and unseen trade-based manipulation schemes through encoding of stock price data into colour images and learning the normal behaviour of historical bid/ask time series data from the constructed colour images using a deep generative model such as a variational auto-encoder. The proposed approach is presented in detail in the following section.

III. The Proposed Approach

The presence of both known and unknown types of manipulation within the stock market inevitably influence the dynamics of bid/ask time series data. However, directly analysing raw time series data to extract hidden patterns for detecting various manipulation schemes poses a significant challenge. Such a challenges stems from the fact that time series data alone does not inherently reveal the underlying features of dynamic systems such as the stock market. Univariate bid/ask time series data can be viewed as a projection of the multi-dimensional stock market space onto a lower-dimensional subspace. Therefore, in order to extract useful patterns pertinent for identifying trade-based manipulations, it becomes necessary to transform bid/ask time series data into another space while retaining the temporal dependencies of the original univariate time series. Our proposed approach involves the transformation of bid/ask time series into phase space utilising recurrence plots (RP) method. Figure 1 illustrates the detailed process of our approach which consists of four main steps: (1) data pre-processing, (2) encoding time series into colour images using RP method, (3) learning normal behaviors, and (4) detecting manipulation.

First, sliding windows are used to partition the time series into different windows such that each window is normalised within the interval [0, 1]. Second, each window is transformed into a colour image using the RP encoding method. Finally, a variational autoencoder is used to learn the normal behaviors of these images by building a Deep Convolutional Neural Network for the encoder and the decoder network, after the training the network is able to better reconstruct the normal images than the anomalies, hence the reconstruction error of the network is used as an anomaly score. The sections below describe in detail each step of the proposed approach.

Step1: Data pre-processing

The data used in this work consisted of the bid/ask price of five stocks: Amazon, Apple, Google, Intel, and Microsoft based on the official NASDAQ Historical TotalView-ITCH [18]. ITCH is a direct data-feed protocol that allows traders to track the status of orders from the time they are submitted until they are executed or canceled.

A. *Sliding Window*

The original bid/ask times series of length L $x(t) = \{x_1, x_2, \dots x_L\}$ is first windowed into segments of a fixed size S (S>1) with a shift S/2, we obtain m time windows $x(t) = \{x(t_1), x(t_2), \dots x(t_m)\}$, where m=(2*$L$-$S$)/$S$ as shown in figure 2.

into

our.

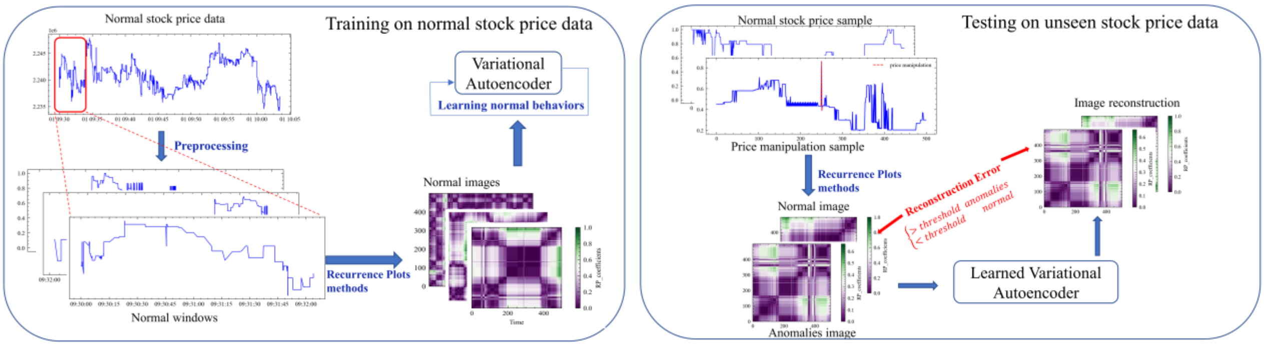

Fig. 1. Overview of the proposed approach. It consists of four main steps (1) data pre-processing, (2) Recurrence plots methods, (3) learning the normal trading behaviour using a beta variational autoencoder and finally the (4) manipulation detection.

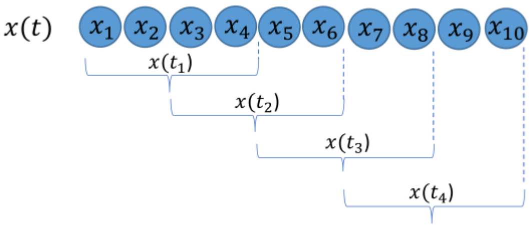

Fig. 2. sliding windows mechanism: a time series $x(t)$ is windowed into four segments $x(t_1), x(t_2)$, $x(t_3)$ and $x(t_4)$ each of them of size four.

B. Linear interpolation and normalisation

Because every window is an irregularly spaced time series, we have applied a linear interpolation to transform this window into equally spaced time (we have used 10 milliseconds). Therefore, each window is represented by a row vector of 100 observations as illustrated in figure 3. In addition, each window is normalised within the interval [0..1] using Equation (1) below.

$$X\_normalized = \frac{X\_window - X\_min}{X\_max - X\_min} \quad (1)$$

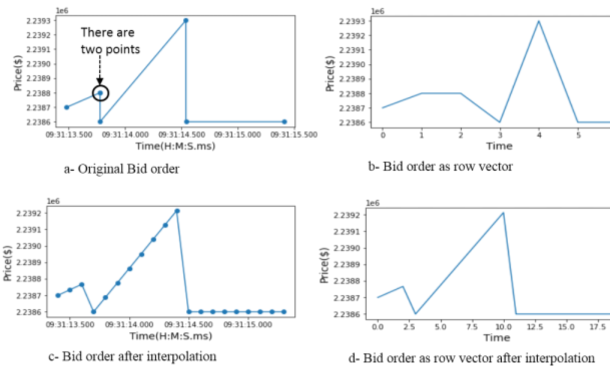

Fig. 3. The Bid order as a vector without time index (b) is very different to the original bid order with time index x-axis (a), the bid order as row vector after linear interpolation (d) is very similar to the original bid order with time index in x-axis (a).

Step 2: Time series to image encoding using recurrence plots

Recurrence plot (RP) is a visualisation tool, introduced by Eckmann et al [19], which is widely used for analysing recurrent behaviours of time series generated in dynamical systems [20]. By using RP, we transform the original bid/ask time series into a colour image format that captures the temporal interdependency patterns. This is formally described as follows:

Given an ask/bid time series $X = (x_1, x_2, \ldots x_n)$ of size n, we begin by extracting the phase space trajectories defined by

$$\vec{x}_i = (x_i, x_{i+\tau}, x_{i+2\tau}, \ldots, x_{i+(m-1)\tau}), \quad \forall i \in \{1, \ldots, n-(m-1)\tau\}$$

$$= \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & \vdots & & \vdots \\ x_{n-(m-1)\tau} & x_{n-(m-2)\tau} & \cdots & x_n \end{bmatrix} \quad (2)$$

where m is the embedding dimension of trajectories and τ is the time delay. Each row of the matrix in Equation (2) corresponds to a point in the phase space. Then the corresponding recurrence plot RP is extracted based on the following recurrence square matrix, denoted by R, which is the pairwise distance between each point in the phase space trajectories $\vec{x}_i$

$$R_{i,j} = \Theta(\epsilon - \|\vec{x}_i - \vec{x}_j\|), \quad \forall i,j \in \{1, \ldots, n-(m-1)\tau\}$$
$$= \begin{cases} 1 & if\ \|\vec{x}_i - \vec{x}_j\| \leq \epsilon \\ 0 & otherwise \end{cases} \quad (3)$$

where Θ is the Heaviside function used to binarise the distance , ϵ is the threshold and $\|\ .\ \|$ is an L2 norm

In this paper, instead of using the recurrences with a black point, represented by "1" in equation (3), we have used the unthresholded version of the recurrence plot [21], sometimes called distance plot, by giving each point a colour as a function of the value of the distance defined by

$$D_{ij} = \|S_i - S_j\|, \quad \forall i,j \in \{1, \ldots, n-(m-1)\tau\} \quad (4)$$

Figure 4 shows a state of a price manipulation represented as colour image by using the recurrence plots encoding method.

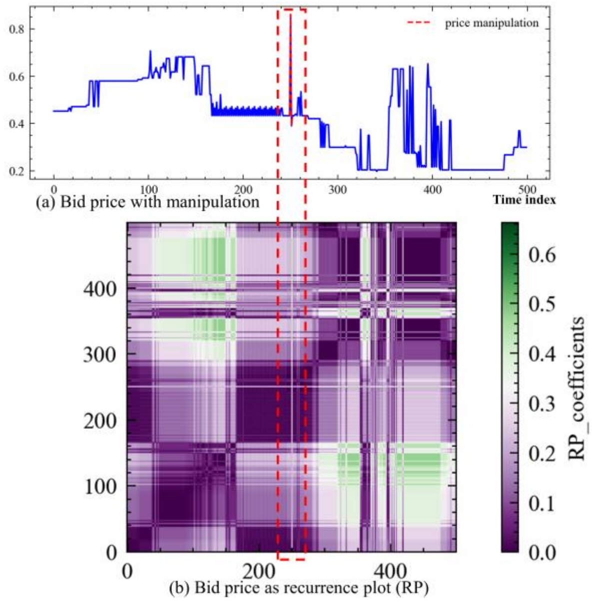

Fig. 4. (a) bid price with a price manipulation at time index 250. (b) a recurrence plots encoded as colour image of the bid price showing a vertical line on the manipulation period at time index 250.

The states of stock market typically change over time, especially during manipulations periods. There are three state of the stock market, state of stock in pre-manipulation periods, state of stock in manipulation periods and state of stock in post-manipulation periods in which the stock market return to its initial state. Figure 5 shows an example of pump and dump manipulation scheme represented by the flowing bid time series (1,2,1.5,5,0.5,2,1,2), by using equation (2), the phase space trajectories are extracted and represented by the following seven points {(1,2) , (2,1.5), (1.5,5), (5,0.5), (0.5,2), (2,1), (1,2) }, where the embedding dimension m is set to 2, and the time delay τ is set to 1. Figure 6 shows that the pre-manipulation and post manipulation periods are very close in the phase space. Therefore, utilising recurrence plots to analyse the states of the bid/ask time series can help in detecting different manipulation schemes.

Step 3: Learning normal behaviour

Once the images are computed, a variational autoencoder

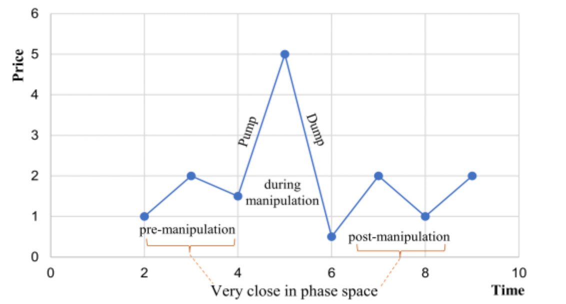

Fig. 5 The state of stock market before, during and after manipulation, in phase space.

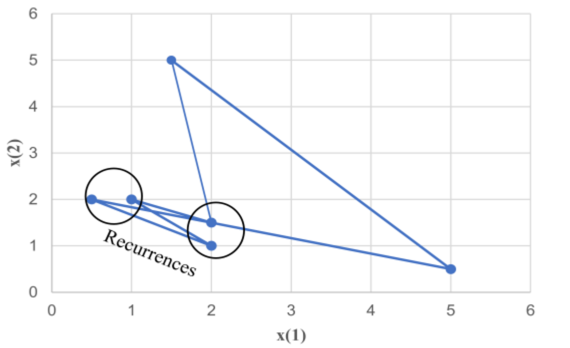

Fig. 6 The phase space trajectories of the pump and dump manipulation scheme, by using the embedding dimension $m = 2$ and the time delay $\tau = 1$, recurrences occur between pre-manipulation and post-manipulation state as they are very close unlike manipulation period.

(VAE) is used to learn the normal behaviour of these images. Our architecture is composed of two Networks: the Encoder network and the Decoder network. The Encoder network is composed of three convolutional layers where the number of filters is double the number of previous layer with 3x3 kernel size. The decoder network is composed of three deconvolution layers where the number of filters is half the number of previous layer with 3x3 kernel size. Adam optimiser is used with the learning rate set to 0.001.

A variational auto-encoder (VAE) is a probabilistic graphical model composed of an encoder network f_θ , a decoder network g_φ and a loss function. The encoder maps the input data X into a distribution with a mean μ and a standard deviation Σ, then the latent variable Z is sampled from this mean and standard deviation and the decoder learn to reconstructs the data given this hidden representation Z by back-propagating the loss function errors [22]. The learning objective of VAE is to minimise the following Loss function:

$$L_{VAE}(\theta,\varphi) = L_{REC} + L_{KL}$$
where: $L_{REC} == -\mathbb{E}_{z\sim q_\varphi(z|x)}\, log\ p_\theta(x|z)$
$$L_{KL} = D_{KL}(q_\varphi(z|x)\ ||\ p_\theta(z)) \qquad (5)$$

The loss function in equation (5) has two components, a reconstruction loss L_{REC} which takes into account the quality of the reconstruction and a Regulariser L_{KL} that ensures that the learned distribution $q_\varphi(z|x)$ does not deviate from the prior distribution $p_\theta(z)$ [22], [23]. In our work, we have used the reconstruction loss L_{REC} as anomaly detection measure. We add a hyper-parameter $\beta < 1$ to the regulariser L_{KL} to reduce penalty for KL-divergence in the objective function (5), and to focus more on capturing useful features to reconstruct the input data x with small errors. A high β value can lead to a poor reconstruction and therefore poor anomaly detection. In this case, the Variational autoencoder is called β-VAE [24], [25]. The final loss function is defined as follow:

$$L_{\beta-VAE}(\theta,\varphi) = -\mathbb{E}_{z\sim q_\varphi(z|x)}\, log\ p_\theta(x|z) + \beta . D_{KL}(q_\varphi(z|x)\ ||\ p_\theta(z)) \qquad (6)$$

In this work, β was set to 0.7.

Step 4: Anomaly detection using β-VAE

Once the variational autoencoder network has learned the normal images behavior, these normal images are well reconstructed by the network, while anomalies are not well constructed given they have not been seen by the network during training. First, test images fed into the encoder network and the parameters μ and Σ of the latent distribution are obtained. Then, L samples are obtained by sampling from a Gaussian distribution with parameters μ and Σ. Each sample Z_L is fed to the decoder network to produce the reconstructed image. Finally, the average of squared differences between the real image and the reconstructed images is computed. If this score exceeds a pre-defined threshold, the test image is classified as an anomaly. The anomaly detection algorithm is described below.

Algorithm : Anomaly detection

Input : x_{test}
Output : normal or anomaly
1: $squared\_error \leftarrow 0$
2: $(\mu,\Sigma) \leftarrow Encode(x)$
3 : $for\ l = 1\ to\ L\ \ do$
4: $z_l \sim Normal(\mu,\Sigma)$
5 : $\hat{x}_l \leftarrow decoder(z_l)$
6 : $squared\_error\ \leftarrow squared\_error + \|x_{test} - \hat{x}_l\|^2$
7 : end for
8 : $ReconstructionScore\ = \frac{1}{L}\ .\ squared\_error$
9: $if\ ReconstructionScore > threshold \qquad x_{test} \leftarrow anomaly$
$else : \qquad x_{test} \leftarrow normal$

IV. Results and Discussion

A. Datasets

The data used in this work includes five stocks, namely Amazon, Apple, Google, Intel and Microsoft. The stock data is based on the official NASDAQ Historical TotalView-ITCH. This dataset was made available from Lobster project [18]. We have used the 1-level tick data for 21th June 2012.

To assess the effectiveness of the proposed method, we randomly injected a total of 800 manipulation instances into the financial time series data of each of the five stocks. These manipulations consist of three distinct types:

(type 1: layering, type 2: spoofing, and type 3: quote stuffing). Type 1(layering manipulation scheme) involves the insertion of 6.9-bps (basis point) saw-tooth patterns within an 819-millisecond window. Type 2 (spoofing manipulation scheme) consists of an 800-bps pulse occurring over a one-second interval. Type 3 (quote stuffing) featured an 18.6 bps square pattern lasting a mere 1 second. The normal and the anomalies time series were divided into 1000 millisecond sliding windows encoded into colour images using the recurrence plots method. The obtained images were divided into 70% normal training and 30% normal testing set, and 60% anomalies training and 40% anomalies testing set as summarised in Table I.

TABLE I.
STOCK DATA SPLITTING INTO TRAINING AND TESTING SETS

Stock	Training		Testing	
	Normal	Anomaly	Normal	Anomaly
Amazon	4416	480	1893	320
Apple	9760	480	4183	320
Google	3028	480	1299	320
Intel	1053	480	452	320
Microsoft	1300	480	558	320

B. *Performance evaluation*

The AUC and False Alarm Rate (FAR) are used as the performance measure on five real stock data sets on 21st June 2012 (Intel, Apple, Amazon, Google and Microsoft). The FAR is computed using Equation (7):

$$FAR = \frac{FP}{FP + TN} \quad (7)$$

Where:

- TP (True Positives) is defined as the total number of manipulation cases correctly detected as manipulation.
- TN (True Negatives) is defined as the total number of normal cases correctly detected as normal cases.
- FP (False Positives) represent the total number of normal cases detected as manipulations cases.
- FN (False Negatives) represent the total number of manipulation cases detected as normal cases.

TABLE II
AUC PERFORMANCE ON FIVE STOCKS

	Proposed approach	AHMMAS	KPCA-MKDE
APPLE	0.8500	0.5344	**0.9206**
AMZN	0.8721	0.5152	**0.9602**
MSFT	**0.9145**	0.6711	0.9143
INTC	**0.9376**	0.5169	0.8732
GOOG	0.8805	0.5119	**0.8996**

TABLE III
FAR (%) ON FIVE STOCKS

	Proposed approach	AHMMAS	KPCA-MKDE
APPLE	1.24	7.83	1.07
AMZN	1.58	9.22	1.22
MSFT	2.5	0.52	0.71
INTC	1.32	1,15	0.54
GOOG	2.16	0.5	1.62

A comparison between the obtained AUC and the FAR for the proposed approach and those obtained by the other AHMMAS and KPCA-MKDE methods are reported in table II and table III below.

The results demonstrate that the proposed approach consistently produces robust and significant performance across all five stocks. It achieves high AUC while maintaining a low false alarm rate. The promising performance of this approach can be attributed to the ability of deep convolutional neural networks, employed in both the encoder and the decoder networks, to automatically learn relevant features from recurrence plots which are encoded as colour images. The viability of encoding stock price data into colour images is evident in the success of the proposed approach. On one hand, the representation of RP plots as colour images effectively captures intricate patterns and relationships within the stock price time series data. The RP based encoding method allows for the visualisation of complex temporal dynamics of the stock price data in a comprehensible and intuitive manner. On the other hand, the use of colour images enables deep convolutional neural networks to automatically extract relevant features from the stock raw price data time series data, hence facilitating robust and accurate detection of different stock price manipulation schemes. The rich visual representation provided by the RP generated colour images helps highlighting subtle variations and anomalies in the stock price data that may not be apparent in traditional numerical representations.

V. Conclusion

We introduced a novel method for detecting stock price manipulation effectively. The proposed approach combines the strengths of beta variational autoencoder with recurrence plots image encoding techniques. The proposed approach initially transforms stock price time series data into 2D colour images using the RP method, followed by employing a β-VAE to learn the reconstruction and generation of these images. Anomalies are then classified based on mean squared errors calculated between the original images and the reconstructed ones. To evaluate its efficacy, we injected three types of manipulation randomly into the financial time series data of five prominent stocks, namely Amazon, Apple, Google, Intel, and Microsoft. The experimental results demonstrated successful detection of the injected manipulation cases. The proposed approach effectively captures pertinent patterns useful for identifying various manipulation schemes by transforming univariate bid/ask time series into colour images.

The proposed approach can be further enhanced by incorporating additional order book information such as volume, cancellation, and execution orders, alongside bid/ask orders. Furthermore, our future work will focus on refining the approach by incorporating feature learning techniques for similarity measurement and exploring more efficient methods for transforming stock market data into colour images.

References

[1] F. Allen and D. Gale, "Stock-Price Manipulation," *Rev Financ Stud*, vol. 5, no. 3, pp. 503–529, Jul. 1992, doi: 10.1093/rfs/5.3.503.

[2] T. J. Putniņ, "An Overview of Market Manipulation," *Corporate Law: Practitioner eJournal (Forthcoming)*, 2018.

[3] D. Kong and M. Wang, “The Manipulator’s Poker: Order-Based Manipulation in the Chinese Stock Market,” *Emerging Markets Finance & Trade*, vol. 50, no. 2, pp. 73–98, 2014, [Online]. Available: http://www.jstor.org/stable/24475662

[4] “Rule 575. Retrieved from U.S. Commodity Futures Trading Commission.” [Online]. Available: http://www.cftc.gov/filings/orgrules/rule082814cmedcm001.pdf

[5] H. Ögüt, M. M. Doganay, and R. Aktas, “Detecting stock-price manipulation in an emerging market: The case of Turkey,” *Expert Syst. Appl.*, vol. 36, pp. 11944–11949, 2009.

[6] D. Diaz, B. Theodoulidis, and P. R. F. Sampaio, “Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices,” *Expert Syst. Appl.*, vol. 38, pp. 12757–12771, 2011.

[7] K. Golmohammadi, O. Zaïane, and D. Diaz, “Detecting stock market manipulation using supervised learning algorithms,” *DSAA 2014 - Proceedings of the 2014 IEEE International Conference on Data Science and Advanced Analytics*, pp. 435–441, Oct. 2015, doi: 10.1109/DSAA.2014.7058109.

[8] N. C. Uslu and F. Akal, “A Machine Learning Approach to Detection of Trade-Based Manipulations in Borsa Istanbul,” *Comput Econ*, vol. 60, no. 1, pp. 25–45, 2022, doi: 10.1007/s10614-021-10131-8.

[9] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, “Detecting price manipulation in the financial market,” in *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, 2014, pp. 77–84. doi: 10.1109/CIFEr.2014.6924057.

[10] Y. Cao, Y. Li, S. Coleman, A. Belatreche, and T. M. McGinnity, “Adaptive Hidden Markov Model With Anomaly States for Price Manipulation Detection,” *IEEE Trans Neural Netw Learn Syst*, vol. 26, no. 2, pp. 318–330, 2015, doi: 10.1109/TNNLS.2014.2315042.

[11] A. and B. A. Abbas Baqar and Belatreche, “Stock Price Manipulation Detection Using Empirical Mode Decomposition Based Kernel Density Estimation Clustering Method,” in *Intelligent Systems and Applications*, S. and B. R. Arai Kohei and Kapoor, Ed., Cham: Springer International Publishing, 2019, pp. 851–866.

[12] B. Rizvi, A. Belatreche, and A. Bouridane, “A Dendritic Cell Immune System Inspired Approach for Stock Market Manipulation Detection,” in *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 3325–3332. doi: 10.1109/CEC.2019.8789938.

[13] B. Rizvi, A. Belatreche, A. Bouridane, and I. Watson, “Detection of Stock Price Manipulation Using Kernel Based Principal Component Analysis and Multivariate Density Estimation,” *IEEE Access*, vol. 8, pp. 135989–136003, 2020, doi: 10.1109/ACCESS.2020.3011590.

[14] T. Leangarun, P. Tangamchit, and S. Thajchayapong, “Stock Price Manipulation Detection using Generative Adversarial Networks,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 2104–2111. doi: 10.1109/SSCI.2018.8628777.

[15] T. Leangarun, P. Tangamchit, and S. Thajchayapong, “Stock Price Manipulation Detection Using Deep Unsupervised Learning: The Case of Thailand,” *IEEE Access*, vol. 9, pp. 106824–106838, 2021, doi: 10.1109/ACCESS.2021.3100359.

[16] P. Chullamonthon and P. Tangamchit, “Ensemble of supervised and unsupervised deep neural networks for stock price manipulation detection,” *Expert Syst Appl*, vol. 220, p. 119698, 2023, doi: https://doi.org/10.1016/j.eswa.2023.119698.

[17] Q. Wang, W. Xu, X. Huang, and K. Yang, “Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning,” *Neurocomputing*, vol. 347, pp. 46–58, 2019, doi: https://doi.org/10.1016/j.neucom.2019.03.006.

[18] “LOBSTER project, Atlanta, GA, USA. Limit order book.” [Online]. Available: https://lobsterdata.com/info/DataSamples.php

[19] J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle, “Recurrence Plots of Dynamical Systems,” *EPL*, vol. 4, pp. 973–977, 1987.

[20] N. Marwan, M. Carmen Romano, M. Thiel, and J. Kurths, “Recurrence plots for the analysis of complex systems,” *Phys Rep*, vol. 438, no. 5, pp. 237–329, 2007, doi: https://doi.org/10.1016/j.physrep.2006.11.001.

[21] J. S. Iwanski and E. Bradley, “Recurrence plots of experimental data: To embed or not to embed?,” *Chaos*, vol. 8 4, pp. 861–871, 1998.

[22] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” *CoRR*, vol. abs/1312.6114, 2013.

[23] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. J. Black, and B. Schölkopf, “From Variational to Deterministic Autoencoders,” *ArXiv*, vol. abs/1903.12436, 2019.

[24] I. Higgins *et al.*, “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework,” in *International Conference on Learning Representations*, 2016.

[25] C. P. Burgess *et al.*, “Understanding disentangling in β-VAE.” 2018.