Stock Price Manipulation Detection based on Autoencoder Learning of Stock Trades Affinity

Baqar Rizvi
Dept. of Computer & Information Sciences
Northumbria University
Newcastle, UK
baqar.rizvi@northumbria.ac.uk

Ammar Belatreche *Member, IEEE*
Dept. of Computer & Information Sciences
Northumbria University
Newcastle, UK
ammar.belatreche@northumbria.ac.uk

Ahmed Bouridane *Member, IEEE*
Dept. of Computer & Information Sciences
Northumbria University
Newcastle, UK
ahmed.bouridane@northumbria.ac.uk

Kamlesh Mistry *Member, IEEE*
Dept. of Computer & Information Sciences
Northumbria University
Newcastle, UK
k.mistry@northumbria.ac.uk

***Abstract*—Stock price manipulation, a major problem in capital markets surveillance, uses illegitimate means to influence the price of traded stocks in order to reap illicit profit. Most of the existing attempts to detect such manipulations have either relied upon annotated trading data, using supervised methods, or have been restricted to detecting a specific manipulation scheme. There have been a few unsupervised algorithms focusing on general detection yet none of them explored the innate affinity among the stock trades, be it normal or manipulative. This paper proposes a fully unsupervised model based on the idea of learning the relationship among stock prices in the form of an affinity matrix. The proposed affinity matrix based features are used to train an under-fitting autoencoder in order to learn an efficient representation of the normal stock prices. A kernel density estimate of the normal trading data is used as the reconstruction error of the autoencoder. During the detection phase, the normal dataset has been injected with synthetic manipulative trades. A kernel density estimation based clustering technique is then used to detect manipulative trades based on their autoencoder representation. The proposed approach is validated on benchmark stock price data from the LOBSTER project and the obtained results show dramatic improvements in the detection performance over existing price manipulation detection techniques.**

***Index Terms*—Market Abuse, Stock Price Manipulation, Affinity Matrix, Autoencoders, Kernel Density Estimate Clustering.**

I. Introduction

There is an increasing demand of analyzing stock price data at most of the stock exchanges around the world. One of the key objectives in doing so is the establishment of a detection model that can identify manipulative instances caused by the market manipulators or market abusers. Stock price manipulation can be explained as the illicit trade transactions made by the manipulator that represents falsifying market prices using illegal means [1]. This is due to the fact that it diminishes the investor confidence as it creates a false impression about the manipulated stock and eventually effects the stature of the market as well. To accomplish such an objective, the stock price data needs to be thoroughly studied, analysed and a optimum decision boundary needs to be established between normal and abnormal patterns. One of the key constraints here is the unavailability of the annotated datasets having both normal and manipulative trades required to train a given machine learning model. Due to which, it becomes difficult to analyse and provide specific parameters to a detection model. This leads us to propose a fully unsupervised model that can determine the exact location of the manipulative instances without much human intervention.

In order to make such a prediction, it is crucial to comprehend the problem from its basics. Stock price manipulation is an act of manipulating stock prices by using some predefined strategies like pump & dump [2] and spoof trading [3]. Pump & dump is a scheme where the manipulator deceives the investors by pumping the price of a given stock through the creation a false demand for the same stock which leads to several added investors who believe the demand to be genuine. However, the manipulator then sells its own investment bought at a cheaper price (bid) when the desired price is achieved. Figure 1 shows such an example from a real life case of pump & dump. Spoof trading on the other hand is related to fabricating a false impression of a given stock's increasing demand by adding huge non bona-fide orders for the purchase of the same. This again creates a fake impression of the stock which eventually increases its selling (ask) price and once the favourable ask price is achieved, the manipulator sells its position and cancels those non bona-fide orders. Figure 2 represents such a situation explaging the progress of a spoofing case in 2012. It should be kept in mind that unlike pump & dump, spoofing can occur at a deeper (although visible) level of the order book.

In this work, we aim to capture the above mentioned manipulations in a dataset, acquired from an open source database

This work is sponsored by Research & Development Fund, Northumbria University, Newcastle upon Tyne, UK.

by training autoencoders (AEs). Autoencoders are a neural network approach to learn the specifics of the underlying dataset in an unsupervised manner generally used for data denoising [4] or dimensionality reduction [5]. The goal here is to encode the input stock price data using an encoding function, to further reconstruct the input using a decoding function and to minimize the reconstruction error by optimizing the loss function. Although some of the recent researches [6]–[9] attempt to use the AE for anomaly detection making use of the fact that training an AE on the normal dataset and testing it on a dataset having both normal and abnormal instances will provide high irregularities in the output only over the abnormal instances. Although most of the existing research using AE for anomaly detection claimed substantial improvements in the results, very few of them explored the spatial aspect of the time-series dataset under consideration. It becomes extremely important for a robust model to learn the space-time representation of the dataset and its evolution with time whether it is normal or a combination of both normal and abnormal stock trades. Unlike past approaches, this proposed research will first envisage the spatio-temporal characteristics of the dataset and then train an AE further upon it. The validity of any model can be determined from its ability to detect the anomalies (market manipulations here) and minimum amount of human effort required in detecting them. Following are the key contributions made by the proposed approach;

- *Affinity matrix describing the relationship among data points* - A new dataset is generated describing the affinity among all the input stock price data instances (Size - N*d, d $\in \mathbb{R}^d$) given length, N and d, dimensions. Although a number of affinity matrix based clustering techniques exist [10]–[12], all of them asks for pre-defined parameters including the number of clusters. The research proposes to describe the innate relationship or affinity among stock prices through a graph laplacian representation [13]. Such a matrix is suitable for explaining the relationship among all the stock prices. The research proposes to describe the innate relationship among stock prices within a given dataset using a distance measure. It is also useful in describing the affinity of a normal data instance towards normal/abnormal data instance and vice-versa.
- *Optimization of under-fitting Autoencoder (AE) using kernel density estimates* - An under-fitting AE is well suited for reducing the dimensions of the input dataset while optimizing the loss function for a minimum reconstruction error. The aim is to extract most significant features that can represent the stock price data. The input dataset here is the affinity matrix, size $(N * N)$ for N data instances and a single hidden layer. Such an AE is optimized while fitting the inherent data distribution using kernel density estimate (KDE) as an objective/likelihood function [14]. This helps in preserving the inherent characteristics of the input stock price data in the extracted features from hidden layer.

The rest of the paper is organised into related work, reviewing some of the benchmark previous researches in section II, followed by the explanation and implementation of affinity matrix and AE. Thereafter, the proposed work plan along with the processing of the output from the AE using Multi-dimensional KDE (MKDE) clustering technique is described in section 5. Experimental results for price manipulation detection on the stocks used are presented and discussed in section 6. Finally, conclusions are drawn in section 7.

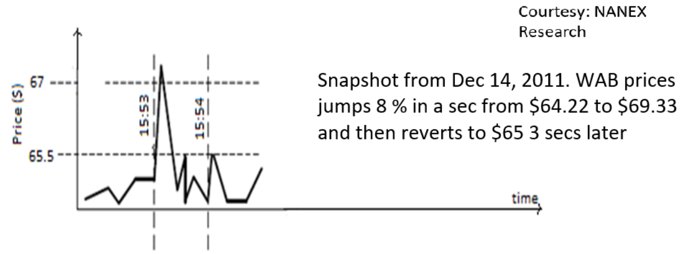

Fig. 1: Snapshot of the pump & dump (Spike waveform) manipulation activity from Dec 14, 2011 shows an 8% rise of WAB price within 1 sec and return to previous level 3 secs later. [19]

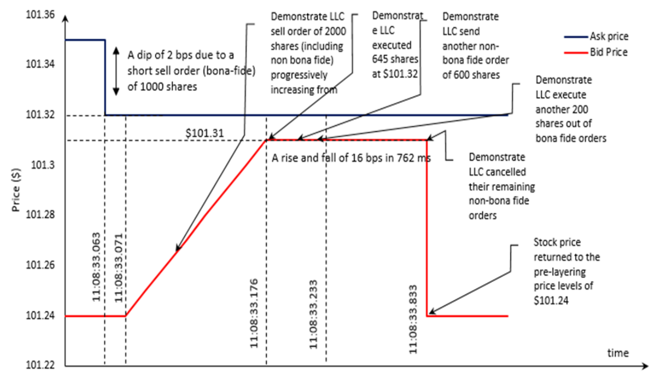

Fig. 2: Shows a reconstruction of the spoofing activity (Saw-tooth waveform) on 25th Sept, 2012 [15]

II. Related Work

Several previous researches have attempted to improve the manipulation detection by claims of significant results. Although only a few of them were able to generalize their model and make it independent of different manipulation schemes. Li *et al.* [16] implemented basic classification algorithms on both daily trading and tick trading stock price data along with labels. The authors claimed better performance when processing daily trading data and poor performance for tick trading data in terms of accuracy and AUC scores. The dataset used are stock prices of 64 companies and is taken from China security regulation commission (CSRC). The fact that the annotations are assigned manually to the manipulated time instance by the authors (as no precise information about the time stamps was available) makes the algorithm biased and prone to errors. Recently, Wang *et al.* [17] claimed to outperform traditional manipulation detection methods by 29.5% in AUC using recurrent neural networks for trade based features rather than statistical features. Despite the substantial results claimed, the model is still biased towards the CSRC dataset considered being supervised in nature. Zhai *et al.* [18] proposed a two model theory for price manipulation detection implemented on dataset from the LOBSTER project. The authors first removed the stationarity of the features calculated and then applied OCSVM and kNN algorithms as part of the first supervised learning model. For the second model, an adaptive hidden markov model similar to [19] has been implemented using Gaussian mixtures model (GMM) to declare the decomposed components as normal and abnormal. Although, significant detection results were claimed using this model, the model is provided with the number of decomposed components from GMM which is misleading as calling any number of components as normal and the rest abnormal cannot be justified for all the feature sets without a significant criterion. Leangurun *et al.* [20] proposed the use of generative adversarial networks for the detection of pump-and-dump manipulation scheme and achieved 68.1% detection accuracy. The authors focused their work on Thailand stock market and trained their model using LSTM as the base structure upon normal data and later tested it using a test data having both normal and abnormal trades/transactions. Despite the simplicity and the capability of the model proposed, the authors focused only on one manipulation scheme rather than generalizing it over multiple manipulative strategies.

It is evident from the literature that most of the detection techniques used either implemented supervised learning algorithms upon financial stock price data where precise labelled information is rare and expensive or focused on specific manipulation schemes rather than making a general price manipulation detection model. The research proposed here aims to address these issues with an objective of making the detection model including its parameters completely data-driven rather than seeking for human input. The following sections shows an insight into the calculation of affinity matrix and the training of an under-fitting autoencoder used in the proposed research.

III. Distance based Affinity Matrix

Affinity matrix can be described as a technique that explores the relationship among data points. Also known as similarity matrix, it is also used to explore the similarity among data points by using distance as a measure. The idea is to compute affinity among stock price data points, apply feature selection and then group the extracted features using proposed clustering techniques. A number of approaches for calculating the affinity based clustering techniques have been proposed in the literature [21]–[23], although most of them require the number of clusters to be specified a priori. The process of creating an affinity matrix is taken from the fact that every stock price data instance within a similar group is strongly correlated to each other compared to the ones that are far apart. One can also understand this as the manifold creation within graphs, where the contiguous stock price data instances have similar labelling information and the distant stock price data instances differ. For a set of n stock price data instances under consideration $x = (x_1, x_2, ...x_n) : x_i \in \mathbb{R}^d$,and considering the affinity matrix to be non - negative matrix, $W : W \geq 0$ can be explained as follows,

$$dist = ||d(x_i, x_j)|| \tag{1}$$

$$W_{i,j} = exp\left(\frac{dist}{2 * \sigma^2}\right) \tag{2}$$

where $d(x_i, x_j)$ (1) is the $l2-norm$ distance metric between every stock price data instance x_i and x_j across multiple dimensions. Such a matrix can also be termed as adjacency matrix here as it calculates a correlation factor between all the stock prices within the dataset. The non-negative adjacency matrix, W_{ij} (2) is sufficient to make the resulting matrix graph Laplacian $L = D - W$ where $D \subset \mathbb{R}^{n*n}$ is a diagonal matrix whose entries being $D_{ii} = \sum W_{ij}$ positive semidefinte which makes the task computationally inexpensive [13]. Interpretation of W, in most of the previous researches, a sparse representation is preferred to avoid spurious connections between far away stock price data points (disjoints) [24]. Although, such a technique becomes insensitive to outliers and hence is avoided in this research.

IV. Under-fitting Auto-encoder

Out of a number of AEs available, standard under-fitting AEs were found suitable for detecting anomalies. This is due to its advantage over the other AEs that it minimizes the influence of small variations in the data during the learning of the model by avoiding any regularization/penalty terms as in Contractive, Sparse or Denoising AE [25]. The autoencoder is trained upon the dataset in a way that the inherent distribution of the dataset is efficiently learned. For this purpose, the dataset is modeled using kernel density estimates and to best fit the parameters of AE to the stock price data, the loss function here is selected as the kernel density estimation of the dataset under consideration.

An AE will learn the distribution pattern present for a given dataset and will try to maximize the log-likelihood $l(f(x))$ as shown in equation (10) to optimize the learning. For a given stock price dataset, $x = (x_1, x_2, ...x_n) : x_i \in \mathbb{R}^d$, a kernel density estimated function can be described as follows,

$$P(x; g) = \frac{1}{ng} \sum^{i} K\left(\frac{x - X_i}{g}\right) \tag{3}$$

at the location X_i, g is computed via the diffusion process [26], K is the Gaussian kernel shown below,

$$K(x) = \left(\frac{1}{2\pi}\right)^{-\frac{m}{2}} \exp \frac{x * x^T}{2} \tag{4}$$

The selection of such a function is based on the better adaptability of the AE to learn the underlying stock price data set [27]. The value of X_i is selected as a linear combination of the latent (hidden) layer output and the output bias, (the rationale for selecting a linear relationship proves to provide a better optimization of the parameter values while minimizing the reconstruction error [27])

$$X_i = b + W * h(x_j) \tag{5}$$

where $h(x_j)$ is the latent layer output for the j^{th} variable, W are the weights, assuming similar weights between input-latent and latent-output layers and b as the output bias. As explained in the details above, in order to make the AE learn and adapt to the dataset under consideration, it is proposed to select the loss function as the density estimate of the data obtained from (3).

$$f(x) = P(\hat{x}|h(x)) = P(\hat{x}; g) \tag{6}$$

Substituting the value from (5) in (3),

$$P(\hat{x}; g) = \frac{1}{ng}\sum^{i} K\left(\frac{\hat{x} - (b + W * h(x_j))}{g}\right) \tag{7}$$

Let $\hat{x}$ be the output of the decoder. Let also consider the latent-input layer relationship to be linear (8),

$$h(x_j) = a + W * x_i \tag{8}$$

From (7), (8) and (4), following conclusion is made for the log-likelihood,

$$l(f(x)) = -\log(\mathrm{P}(\hat{x}; g)) \tag{9}$$

$$l(f(x)) = \frac{1}{(2\pi)^{-\frac{m}{2}} * ng^2}\sum^{i}\frac{\left|\left|\hat{x} - \left((W)^2 x_i + C\right)\right|\right|^2}{2} \tag{10}$$

where

$$C = b + aW \tag{11}$$

As the added bias C and the weights $(W)^2$ are a linear transformation of the input x_i in (10), the loss function can be regarded similar to the $l2$-norm (sum of the Euclidean distances) as with a standard autoencoder for real inputs. Such an AE is first trained upon the dataset having normal trades. Once trained the same AE is then used upon the test stock price data, containing both normal and abnormal trades.

V. Proposed Method

As mentioned briefly in the introduction, the proposed research aims to create a clear description of data distribution for clustering algorithms to follow for manipulation detection. The approach allows statistics of the dataset to be processed in such a way that the separation between normal and abnormal trades becomes clearly distinguishable. For the purpose of achieving so, firstly a pre-processing step of removing artifacts such as periodicity [28] from the stock prices is applied as illustrated in Figure 3. As shown in Figure 4, the approach computes the features relevant in capturing the effect of anomalies from the pre-processed time series. As the high frequency elements in the stock price data is more prone to anomalies [29] wavelet transform is applied to analyse only the high

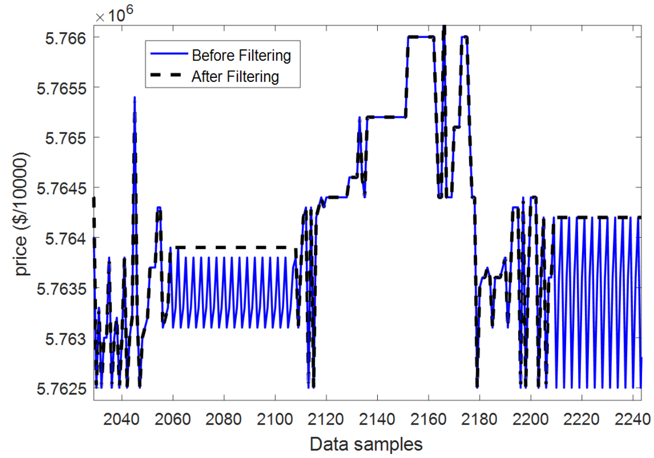

Fig. 3: Stock prices before and after periodicity removal [15]. The red ones after the pre-processing overlap original prices shown in blue.

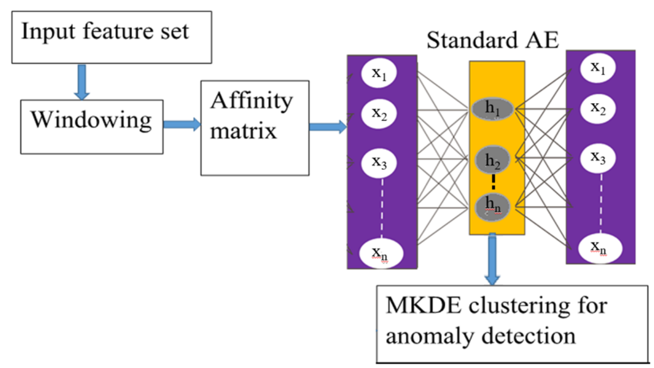

Fig. 4: Architecture of the work flow

frequency elements in the data and neglect the low frequency elements i.e. for input stock prices, $x(t) : t \in (t, t+n)$ for n number of data instances within the window, only its high frequencies portion $\hat{x}(t) : t \in (t, t+n)$ is selected. In order to do so, descrete wavelet transform (DWT) has been used to first extract the approximate and detail coefficients using a level-5 decomposition where approximate represents the low frequency components and the detail represents the high frequencies. Then a hard thresholding is applied upon the detail coefficients, $X_{a,b}$ where a and b are scaling and shifting parameters over the threshold λ for the given coefficient. The components exceeding this threshold are set to zero.

$$X_{a,b} = \begin{cases} X_{a,b} & X_{a,b} \leq \lambda \\ 0 & X_{a,b} > \lambda \end{cases} \tag{12}$$

The value of the threshold λ is selected using the universal threshold algorithm [30]. Along with the above mentioned feature, another feature vector also known as Wilson's amplitude is also used. It measures the difference between two consecutive samples and amplifies the sample value if the difference exceeds a given threshold m , typically a threshold of 3 bps (basis points, one basis point = 0.01% of the price) is selected [15].

$$q(t) = x(t) - x(t-1) \tag{13}$$

$$w(t) = \begin{cases} 3 * q(t) & q(t) > m \\ q(t) & q(t) \leq m \end{cases} \tag{14}$$

Furthermore, stock prices $x(t)$, the slope of the stock prices $\delta x(t)/\delta t$ measuring the rate of change of stock prices, the gradient of wavelet high frequency component feature $\delta \hat{x}(t)/\delta t$, stock traded volume information $v(t)$ and the slope of traded stock volume $\delta v(t)/\delta t$ are also considered as features. This makes a total of seven feature vectors including the original stock prices for the proposed model, $X = \left[x(t), \hat{x}(t), w(t), \delta x(t)/\delta t, \delta \hat{x}(t)/\delta t, v(t), \delta v(t)/\delta t\right]$

The architecture of the proposed work allows such time specific features as the input to being divided into windows of fixed length. Windowing the whole dataset into smaller set of samples reduces the number of computations for the affinity matrix to be calculated next. Once windowed, each set of features are now transformed into an affinity matrix using the proposed method explained before. The output is now processed through an under-fitting single layered autoencoder pre-trained upon the normal dataset. Following which, the 6 encoded features are extracted from the AE are then passed to the MKDE based clustering approach without stipulating the amount of clusters required up front [31]. It should be noted here that to estimate the distribution of a non-stationary and volatile stock price dataset where the mean and variance regularly varies with time, it is not reasonable to assume the stock price data to be normally distributed. In order to extract meaningful information, a data driven population distribution estimate needs to be created. Hence the density estimate of the extracted features from the AE is created by fitting a kernel based distribution prior to be processed by the proposed MKDE clustering approach [15].

The MKDE based clustering is summarised in the Algorithm 1 mentioned below;

The results obtained after the implementation of the above mentioned proposed research are presented and discussed in the following section along with the dataset used.

VI. Results & Discussion

The datasets used in this approach are tick data for level 1 orderbook taken from the LOBSTER project, an open source and include stocks like Apple, Amazon, Google, Microsoft and Intel corporation for June 12, 2012 operating on NASDAQ, USA [32]. The dataset provides stock prices and volume information versus time. The rationale behind selecting such stocks is the popularity of each of them with the amount of influence they have on the market and are reported to have no manipulative trades [33]. The fact that acquiring labelled dataset is extremely difficult because of data confidentiality regulations and the hefty sum one has to pay annually, artificial manipulation of two different types as shown in Figure 1 and 2 is preferred to test the robustness of the detection model.

Algorithm 1 MKDE Clustering

$X = X_1, X_2 \ldots X_t$; where X$\in \mathbb{R}^d$ is the feature sample
€ $= \phi$, t = length(X); where € is a cluster
k = 0; Cluster iterations
while length(X) $\neq 0$ **do**
  $j = j + 1$
  define the bandwidth g
  for i = 1,2,...t **do**
    if $\bar{X} - X_i < g$; where $\bar{X}$ is the mean(s) location in the distribution of the dataset **then**
      €$_j$ = €$_j \cup X_i$; Affiliate X_i data instance to the cluster €$_j$
      $X = X/X_i$; remove the clustered data samples from the dataset
    end if
  end for
end while

A saw-tooth like waveform having a rise of 7 bps in 95 msecs creates the impression of a real life example of trading activity by Demonstrate LLC condemned for spoof trading on 25th Sept, 2012 [34]. Type 2 is an example of pump and dump manipulation strategy for WAB prices having a rise and fall of 30 bps in a duration of 0.1 sec on 14th Dec, 2011 [19]. As the number of data instances varies among the stocks, the amount of manipulative instances injected is also varied. The number of manipulations injected in Apple, Amazon and Google stocks are 100 anomalies/type and for Microsoft and Intel corp stocks, 200 anomalies/type. To ensure the effectiveness of the detection model, a random injection of the manipulation in the original stock price dataset is practiced making a combination of both normal and abnormal trading patterns.

Initially, a total of seven time specific features are extracted from the synthetic dataset. An affinity matrix $L_{500*500}$ is then generated by considering the window length of 500 data instances for the input feature set. Following which, a pre-trained AE (upon normal dataset) is used to process the affinity matrix and extract six encoded features before being processed by the MKDE clustering approach to cluster normal and manipulative trades separately. The input to the MKDE clustering is a dataset of size 500 by 6 using a Gaussian kernel without specifying the number of clusters up front. The proposed approach is evaluated by using area under the receiver operating curve (AUC) as the performance measure along with false positive ratio and F-measure. Table I shows the comparative assessment of stocks with k-means based approach [35], PCA based [36], K nearest neighbour based [37] and OCSVM based manipulation detection techniques [37] in terms of AUC. Such techniques are selected for comparison being some of the commonly used methods in both unsupervised and supervised learning for manipulation detection and an optimum selection of the parameters is carefully carried out to assure a fair comparison. Similarly, table II,

& III shows the comparative assessment of the proposed stock price manipulation detection method against k-NN, PCA, k-Means and OCSVM approaches in terms of FAR and F-score. Finally, the proposed model is also assessed on the basis of its comparison in terms of AUC values with existing benchmark research in stock price manipulation detection [19], [28] as shown in table IV. Given the fact that only methods that aim to generalize their detection model towards different manipulation schemes with unsupervised learning and have used the similar datasets are selected.

It can be easily observed that the proposed approach outperforms the a selection of existing manipulation detection techniques (both supervised and unsupervised) along with existing research in stock price manipulation detection. It is also important to notice the significant enhancements in terms of false alarm rates as most of the values calculated are two decimal places below zero. To assess the performance in Table I, the AUC value of the proposed approach in stocks surpassed existing manipulation detection methods by 25.92% for Apple, 10.92% for Amazon, 2.77% for Google, 11.76% for Intel corp and 15.41% for Microsoft stocks. It is also worth analysing the low AUC value for Google stock (comparatively to other stocks) which can be attributed to the volatility and also the overlapping of normal and manipulative trading behaviours. However, this can be improved by a thorough analysis of the time series in Google stock price and carefully selecting the injection locations of the manipulative data. In addition, from Table II & Table III, the performance comparison of the proposed approach in terms of FAR and F scores, shows a dramatic improvement of no less than 95.76% and 90.07% respectively, for all the stocks.

As is evident from table IV, the results also outperforms the existing research in stock price manipulation detection in terms of AUC values by a maximum of 22.3% over the same stocks. It can also be observed from table IV that higher AUC values are obtained for stocks like Apple, Amazon, Intel corp and Microsoft, however it decreases slightly for Google stock. The comparatively lower AUC value for Google stock still justifies the effectiveness of the proposed model as it shows an improvement over existing researches namely, 0.7896 for EMD-KDE approach [28] (authors previous research), 0.8025 for AHMMAS [19]. The rationale behind the effectiveness of the proposed approach can be explained by making the autoencoder learn the relationships among stock prices captured by the affinity matrix. In addition, the optimization of the autoencoder parameters while selecting the kernel density estimate of the dataset as the loss function leads to an improvement in the learning of the model. Moreover, the automatic selection of the KDE clustering based parameters including independent selection of the number of clusters adds to the robustness of the proposed model.

VII. Conclusion

A novel approach for detecting stock price manipulation was proposed based on the blend of affinity matrix and KDE clustering independent of the annotated data. A concise

TABLE I: Comparison of Stocks in terms of AUC values with a selection of existing manipulation detection techniques

Dataset	**Proposed Approach**	**kNN** [37]	**PCA** [36]	**K-means** [35]	**OCSVM** [37]
Apple	**0.9981**	0.7926	0.6902	0.5819	0.6603
Amazon	**0.9998**	0.7982	0.9013	0.5799	0.8933
Google	**0.8215**	0.5612	0.7993	0.6328	0.5911
Intel	**0.9701**	0.5469	0.8680	0.5077	0.6970
Microsoft	**0.9989**	0.5509	0.8655	0.5047	0.6419

TABLE II: Comparison of Stocks in terms of FAR values with a selection of existing manipulation detection techniques

Dataset	**Proposed Approach**	**kNN** [37]	**PCA** [36]	**K-means** [35]	**OCSVM** [37]
Apple	**0.0054**	0.45	6.64	1.26	67.8
Amazon	**4.9197E-04**	0.14	3.9	7.33	49.54
Google	**0.0288**	0.68	7.22	9.95	75.2
Intel	**0**	0.23	57.29	0.02	59.08
Microsoft	**0**	0.08	49.89	0.02	77.48

TABLE III: Comparison of Stocks in terms of F score values with a selection of existing manipulation detection techniques

Dataset	**Proposed Approach**	**kNN** [37]	**PCA** [36]	**K-means** [35]	**OCSVM** [37]
Apple	**0.3689**	0.1344	0.1457	0.1708	0.0450
Amazon	**0.4704**	0.1714	0.1568	0.1484	0.0284
Google	**0.2566**	0.135	0.1806	0.1513	0.0196
Intel	**0.5836**	0.1014	0.2085	0.1119	0.0126
Microsoft	**0.5934**	0.1148	0.2077	0.2141	0.0920

TABLE IV: Comparison of Stocks in terms of AUC values with existing benchmark stock price manipulation detection methods

Dataset	**Proposed Approach**	**AHMMAS** [19]	**EMD-KDE** [28]
Apple	**0.9981**	0.8142	0.7946
Amazon	**0.9998**	Not Reported	0.9226
Google	**0.8215**	0.8025	0.7896
Microsoft	**0.9989**	0.8971	0.8805
Intel	**0.9701**	0.7336	0.8903

review of past researches in market manipulation detection was presented. The research aimed at the detection of two different types of manipulation schemes using fully unsupervised learning. For this purpose, a standard dataset (reportedly free from manipulation) is considered and to evaluate the effectiveness of the approach, it was injected with significant number of manipulative trade instances. Such a dataset having a combination of both normal and abnormal trades was further processed to compute an affinity matrix from a small set of features extracted. An autoencoder pre-trained using the density distribution of the normal dataset was used to extract the encoded features while providing the affinity matrix as an input. The encoded data was then subjected to a proposed KDE approach for clustering and the data instances left un-clustered are treated as manipulation. Finally, the obtained results were compared with a selection of existing manipulation detection techniques like kNN based, PCA based, OCSVM based and K-means based. In order to check the robustness of the proposed approach, it was evaluated in terms of AUC, F-Score and FAR. The approach was also compared with some existing benchmark research in stock price manipulation detection.

It was observed that the proposed approach clearly outperformed the existing methods in terms of AUC values, improved the F-score and reduced the false positives while avoiding the annotated data. The significant improvement in results can be attributed to improved learning of the AE using the information captured by the affinity matrix and preserving the information in the encoded features. However, there is possibility to further improve the proposed detection approach by identifying the type of manipulation being detected. In addition, an independent selection of some parameters like window length and threshold values for feature extraction is also a matter of future research.

REFERENCES

[1] Franklin Allen and Douglas Gale, "Stock Price Manipulation," The Review of Financial Studies, vol. 5, no. 3, pp. 503-529, 1992.

[2] Securities and Exchange Commission, March 12, 2001, Available online:: https://www.sec.gov/fastanswers/answerspumpdumphtm.html

[3] Eun Jung Lee, Kyong Shik Eom, and Kyung Suh Park, "Microstructure-based manipulation: Strategic behavior and performance of spoofing traders," Journal of Financial Markets, vol. 16, no. 2, pp. 227 - 252, 2013.

[4] Varghese Alex et al., "Semi-supervised learning using denoising autoencoders for brain lesion detection and segmentation," Journal of Medical Imaging, vol. 4, pp. 4 – 16, 2017.

[5] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in Proc. ACM Workshop Mach. Learn. Sensory Data Anal. (MLSDA), pp. 4-11, 2014.

[6] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min., pp. 665–674, 2017.

[7] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. AbuMallouh, "Deep and Machine Learning Approaches for Anomaly-Based Intrusion Detection of Imbalanced Network Traffic," IEEE sensors letters, vol. 3, pp. 1-4, 2019.

[8] D. Singh and C. K. Mohan, "Deep Spatio-Temporal Representation for Detection of Road Accidents Using Stacked Autoencoder," IEEE Trans. Intelligent Transportation Systems, pp. 1–9, 2018.

[9] T. Amarbayasgalan, B. Jargalsaikhan, and K. H. Ryu, "Unsupervised novelty detection using deep autoencoders with density based clustering," Appl. Sci., vol. 8, no. 9, pp. 1468-1486, 2018.

[10] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[11] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," Proc. Seventh IEEE Int'l Conf. Computer Vision, pp. 975-982, 1999.

[12] S. Yu and J. Shi, "Multiclass Spectral Clustering," Proc. Ninth IEEE Int'l Conf. Computer Vision, pp. 313-319, 2003.

[13] F. R. K. Chung. "Spectral Graph Theory," Number 92 in CBMS Regional Conference Series in Mathematics. AMS, 1997.

[14] G. Alain and Y. Bengio, "What Regularized Auto-Encoders Learn from the Data Generating Distribution," Technical Report, Arxiv report 1211.4246, Univ. de Montre´al, 2012.

[15] B. Rizvi, A. Belatreche and A. Bouridane, "A Dendritic Cell Immune System Inspired Approach for Stock Market Manipulation Detection," 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, pp. 3325-3332, 2019.

[16] Aihua Li, Jiede Wua and Zhidong Liua, "Market Manipulation Detection Based on Classification Methods," Elsevier Procedia Computer Science, Volume 122, Pages 788-795, 2017.

[17] Qili Wang, Wei Xu, Xinting Huang and Kunlin Yang, "Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning," Neurocomputing, Volume 347, Pages 46-58, 2019.

[18] J. Zhai, Y. Cao and X. Ding, "Data analytic approach for manipulation detection in stock market," Review of Quantitative Finance and Accounting, vol. 50, no. 3, pp. 897-932, 2018.

[19] 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, Feb. 2015.

[20] T. Leangarun, P. Tangamchit, and S. Thajchayapong, "Stock Price Manipulation Detection using Generative Adversarial Networks," In 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2104-2111, 2018.

[21] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," Proc. Seventh IEEE Int'l Conf. Computer Vision, pp. 975-982, 1999.

[22] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[23] S. Yu and J. Shi, "Multiclass Spectral Clustering," Proc. Ninth IEEE Int'l Conf. Computer Vision, pp. 313-319, 2003.

[24] X. Zhu, "Semi-supervied learning literature survey," Technical report, University of Wisconsin Madison, 2008.

[25] K. Konda, R. Memisevic and D. Krueger, "Zero-bias autoencoders and the benefits of co-adapting features," Feb. 2014, [online] Available: https://arxiv.org/abs/1402.3337.

[26] Z. Botev, J. Grotowski, and D. Kroese, "Kernel density estimation via diffusion," The Annals of Statistics, vol. 38, no. 5, pp. 2916–2957, 2010.

[27] N. Kourentzes, D.K. Barrow and S.F. Crone, "Neural network ensemble operators for time series forecasting," Expert Systems with Applications, vol. 41, pp. 4235-4244, 2014.

[28] Baqar Abbas, Ammar Belatreche and Ahmed Bouridane, "Stock Price Manipulation Detection Using Empirical Mode Decomposition Based Kernel Density Estimation Clustering Method," in Conference proceedings Intelligent Systems and Applications, Intellisys, Advances in Intelligent Systems and Computing, vol 869. Springer, Cham, pp. 851-866, 2018.

[29] Suman Neupane, S. Ghon Rhee, Kulunu Vithanage and Madhu Veeraraghavan, "Trade-based manipulation: Beyond the prosecuted cases," Journal of Corporate Finance, vol. 42, pp. 115-130, February 2017.

[30] N E Huang, Z Shen, S R Long, M C Wu, E H Shih, Q Zheng, C C Tung and H H Liu, "The empirical mode decomposition method and the Hilbert spectrum for non-stationary time series analysis," Proc. R. Soc., London vol. 454, pp. 903–995, 1998.

[31] L C Matioli, S R Santos, M Kleina and E A Leite, "A new algorithm for clustering based on kernel density estimation," Journal of Applied Statistics, vol. 44, no. 1, pp. 1-20, 2016.

[32] LOBSTER project, Atlanta, GA, USA. Limit Order Book System, 2012 [Online]. Available: http://www.lobster.wiwi.hu-berlin.de

[33] J Tse, X. Lin, and D Vincent, "High frequency trading—Measurement, detection and response," Credit Suisse, Zürich, Switzerland, Tech. Rep., Dec. 2012.

[34] Nanex, Denver, CO, USA. (2012). Whac-a-Mole is Manipulation [Online]. Available: http://www.nanex.net/aqck2/3598.html

[35] S. Chawla and A. Gionis, "K-means–: A unified approach to clustering and outlier detection," in Proc. SIAM SDM, pp. 187–197, 2013.
[36] M L Shyu, S C Chen, K Sarinnapakorn and L W Chang, "A Novel Anomaly Detection Scheme using Principal Component Classifier," In Proceedings of the 3rd IEEE International Conference on Data Mining, pp. 353–365, 2003.
[37] Christopher M Bishop, Pattern Recognition and Machine Learning.: springer New York, 2006.