Stock Price Manipulation Detection using Generative Adversarial Networks

Teema Leangarun
Dept. of Control System and Instrumentation Engineering
King Mongkut's Univ. of Tech. Thonburi, Bangkok, Thailand
teema.lea@mail.kmutt.ac.th

Poj Tangamchit
Dept. of Control System and Instrumentation Engineering
King Mongkut's Univ. of Tech. Thonburi, Bangkok, Thailand
poj.tan@kmutt.ac.th

Suttipong Thajchayapong
National Electronics and Computer Technology Center
NSTDA Bangkok, Thailand
suttipong.tha@nectec.or.th

***Abstract*—We implemented Generative Adversarial Networks (GANs) for detecting abnormal trading behaviors caused by stock price manipulations. Long short-term memory (LSTM) was used as a base structure of our GANs, which learned normal market behaviors in an unsupervised way. After the training, the discriminator network of GANs was used as a detector to discriminate between normal and manipulative trading. Our work is different from the previous work in that we did not use manipulation cases to train the neural networks. Instead, we used normal data to train them, and simulated manipulation cases were only used for testing purposes. The detection system was tested with the trading data from the Stock Exchange of Thailand (SET). It can achieve 68.1% accuracy in detecting pump-and-dump manipulations in unseen market data.**

Keywords—stock price manipulation, generative adversarial networks, anomaly detection

I. INTRODUCTION

A stock market is a place in which participants are able to buy and sell ownership of companies. If all transactions in a stock market occur fairly among buyers and sellers, there are no problems. However, there are some bad players in a stock market who have irregular trading behaviors. The bad players attempt to gain personal profit from stock trading by deceiving other investors about artificial effects in the stock market. The act is called stock manipulations [1]. Stock manipulations are illegal in most cases, but it can be difficult to detect and prevent for market regulators [2]. The detection is even more difficult with the rising of automated transactions by high speed trading computer programs. There are million orders entering into a stock market every trading day. With this large amounts of trading data, it is impossible for human to monitor these transactions. Market regulators need a smart computer program to investigate these transactions, so that a countermeasure can be promptly executed.

Manipulation activities were initially introduced by Allen and Gale [3] in three types: action-based manipulations, information-based manipulations, and trade-based manipulations. Action-based manipulations are actions that manipulators act to change in demand or supply in a stock market. In information-based manipulations, manipulators aim to release false information or rumors to affect stock prices. Trade-based manipulations follow legal trading rules for buying and selling, but uses crafted order submissions and cancellations with an intent to control stock prices. This type is different from the two previously mentioned because the manipulating transactions are mixed with normal transactions. This can be clearly distinguished from regular trading orders. Our research only focuses on trade-based manipulations.

There are three challenges in this research field. The first challenge is the difficulty to distinguish manipulations from normal trading behaviors. Manipulations can have various patterns and they can be evolved over time, like the cat and mouse problem. Also, manipulation cases can get unnoticed easily when the amount of damage is small. We usually verify suspicious cases that can have high damage because small damage cannot influence on the stock price, trading volume, and investors' decision making in the stock market. The second challenge is that the data available for research are usually partially observable, because the market authority has to keep the identity of buyers and sellers as a secret. The effectiveness of stock manipulation detection depends on how much the information we have. The third challenge is that there are not many manipulation cases that get exposed. Therefore, the number of cases for validation is small.

In order to detect stock price manipulations, regulators usually use fixed rules for scanning stock market data. If there are new tactics to manipulate stock prices in the future, the rules may have to be continuously adjusted. With the recent advances of machine learning and artificial intelligence, our research attempts to implement a deep neural network model that can learn from actual market data to detect stock price manipulations. Our assumption is that manipulation cases are considered as irregular trading behaviors, which have characteristics that are different from normal trading behaviors. We chose an unsupervised learning approach to train the deep neural network because there are not many verified manipulation cases that we can use to label the data effectively. Unsupervised learning can learn without data labels. We propose LSTM-GANs for stock price manipulation detection. Our framework is shown in Fig. 1. We used market data from the Stock Exchange of Thailand (SET). The market data were used for order book reconstruction. The order book was normalized before being fed into the models for training (see red dash line in Fig. 1). In testing process, the models were evaluated on both unseen normal data and anomalous data (see yellow dash line in Fig. 1).

This research was supported by the National Science and Technology Department Agency (NSTDA), Thailand Graduate Institute of Science and Technology (Grant: TG-44-20-59-019D).

The remaining of this paper is organized as follows. In section II, we give an overview of previous work on stock manipulation detection and present our assumptions of this research. Section III describes stock price manipulation patterns. Next, we give a review of anomaly detection using GANs in Section IV. Section V describes the mathematical background of the models. Data preprocessing and normalization, experimental design, and experimental results are also followed in section VI. Finally, we conclude the whole work in section VII.

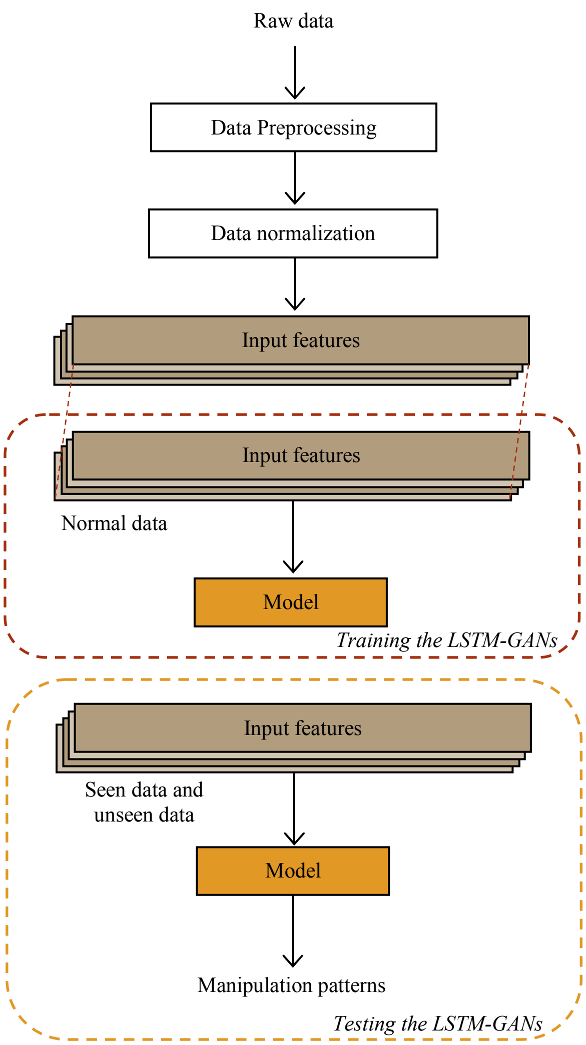


Fig. 1. Stock price manipulation detection framework.

II. RELATED WORK

Allen and Gale firstly define three categories of manipulation as mentioned above. There were a few researchers who studied in theoretical and empirical work on trade-based manipulations. Jarrow [6], Allen et al. [7], and Carhart and Reed [8] found that large investors have high probability to be a manipulator. An investor who has insider trading information of a firm also has high probability to be a manipulator [6], [7]. Bagnoli and Lipman [9] showed that a manipulator may gain his profit by making a takeover bid for stock price manipulation. So, the manipulator can sell the stock at his expected price. Aggarwal and Wu [10] extended Allen and Gale work. They studied stock manipulations in the US stock market from 1990 to 2011. They found that liquidity, returns, and volatility affected the stock price during manipulation periods, in which the stocks with low liquidity and value have high possibility to be manipulated. Comerton-Forde and Putniņš [11] studied stock price manipulations at the closing period. They found that day-end returns, spreads, trading activities, and price reversions were all increased in the stock manipulations.

Researchers have implemented stock manipulation detection using di□erent methods. Ögüt et al. [12] studied stock price manipulations in Istanbul stock market. The authors compared data mining techniques (artificial neural networks and support vector machines) with multivariate techniques (logistic regression and discriminant analysis). They found that the data mining techniques performed better. They examined variables that relates to stock price such as index's average daily return, average daily change in trading volume, and average daily volatility. Mongkolnavin and Tirapat [13] used association rules to detect mark-the-close in Thai Bond Market. The authors focused on trading behaviors and price variation in historical trading data. The system can analyze warning signals in real time. The results also showed a list of investors in the market who may be manipulators. Golmohammadi et al. [14] used many techniques of supervised learning algorithms based on market manipulation cases in the US Securities and Exchange Commission (SEC): decision tress, Naïve Bayes, artificial neural networks, support vector machines, and k-nearest neighbor. The authors divided market manipulation tactics into three groups: marking the close, wash trades, and cornering the market. Yang et al. [15] proposed a logistic regression model for stock manipulation detection in Shanghai and Shenzhen stock markets. The authors classified variables that may be caused by stock manipulations into four groups: variables in term of price (return and abnormal return), variables in term of trading volume and liquidity (value of daily traded stock and turnover rate), variables in term of equity scale and shareholding concentration (number of circulating stocks, number of shares per stockholder, and market value per stockholder), and other variables (price-earnings ratio and fluctuation). Zhai et al. [4] proposed a paper based on analytics analyzing trading behavior data for manipulation detection in the NASDAQ stock market. The authors divided the methods into two groups: static models (k-nearest neighbor and one-class support vector machine) and dynamic models (adaptive dynamic model and Gaussian mixture model). Stock price manipulation strategies were classified and analyzed into two forms: spoofing trading and quote stuffing. Both models were effective to detect manipulative behaviors.

Various machine learning techniques have been used in data analytics. Especially, in recent years, deep learning concepts have also been used with financial data. Most of academic papers in financial area proposed deep learning techniques for stock prediction [16], [17], [18]. Also, Deng et al. [19] introduced deep recurrent neural networks (DRNNs) for real-time financial signal representation and trading on both stock and commodity future markets. In this paper, LSTM-GANs with deep architecture have been introduced as a classifier of stock manipulation detection in SET. We did not use manipulation cases to train the models, because they are rarely compared to the huge amount of regular data. So, we synthesized and injected manipulation cases into the normal trading data. Using this method, we first checked whether there were known manipulation cases in the stock that we used for training. We selected only the stocks that have no record of such cases. Then, we made an assumption that these stocks had no abnormality and were free from manipulations. To the best of our knowledge, there are no previous work that used deep learning techniques with stock price manipulation detection.

III. Stock Manipulation Patterns

Our research focuses on trade-based manipulations which are difficult to be detected. This method involves manipulators' expectation to gain their profits by sending crafted orders. Orders can be a mix of genuine (matched) and artificial (with no intention for manipulators to be matched). We studied pump-and-dump which is the most popular manipulation case [5]. This is also similar to ramping or gouging [20]. The main concept of pump-and-dump is to pump a stock price up and then sell the stock at a higher price, which is profitable to the manipulators.

Pump-and-dump can be divided into two periods: pumping period and dumping period. Pump-and-dump diagram is shown in Fig. 2. In the pumping period, a manipulator enters bid orders (spoof orders) into the stock market. Both bid price and volume are significantly pushed up. In the same period, other investors think that the stock price is rising. So, most of them do not hesitate to enter their orders into the stock market. When the manipulator pushes the price up to his target (his expected price), the bid orders are then quickly cancelled. Also, the manipulator suddenly sells the stock at the higher price after cancelling and makes a profit in the dumping period. There are two variables that involve stock price manipulations: cancelled volumes and matched volumes. Order cancellations represent any specific prices and volumes that investors withdraw from the stock market. Order cancellations can happen from both manipulators and regular investors. However, irregular size of cancelled orders in a short period of time should be closely monitored, because this is an important indicator for stock manipulations. Moreover, irregular size of matched orders at the opposite side of the cancelled orders should be observed altogether.

IV. Anomaly Detection using GANs

There are many researchers who studied anomaly detection models within various application domains [21], [22], [23]. Stock manipulation detection is a subset of anomaly detection problems. This aims to identify irregular or unusual patterns in normal trading behaviors. Our work proposes GANs for stock manipulation detection. In recent years, GANs have emerged and been used in many tasks such as road detection [24], medical imaging [25], text to image synthesis [26], and natural language processing [27]. There were some researchers who used GANs in anomaly detection problems. In biomedical engineering, Schlegl et al. [28] proposed GANs to guide markers discovery. The authors hypothesized that the latent vector of GANs represents the distribution of data. This model was trained with healthy data, and was tested with unseen healthy data and anomalous data in order to identify anomalies. As reviewed in security, Kiran et al. [29] reviewed different adversarial trainings in video anomaly detection. Zenati et al. [30] explored GANs with anomaly detection problems. The authors applied their method with MNIST (image data) and KDD99 (non-image data). This study was the first GANs implemented for anomaly detection using the KDD99 dataset. Motivated by [28], we propose GANs training framework for stock manipulation detection. However, in contrast, our approach requires a sequential model to be used with GANs.

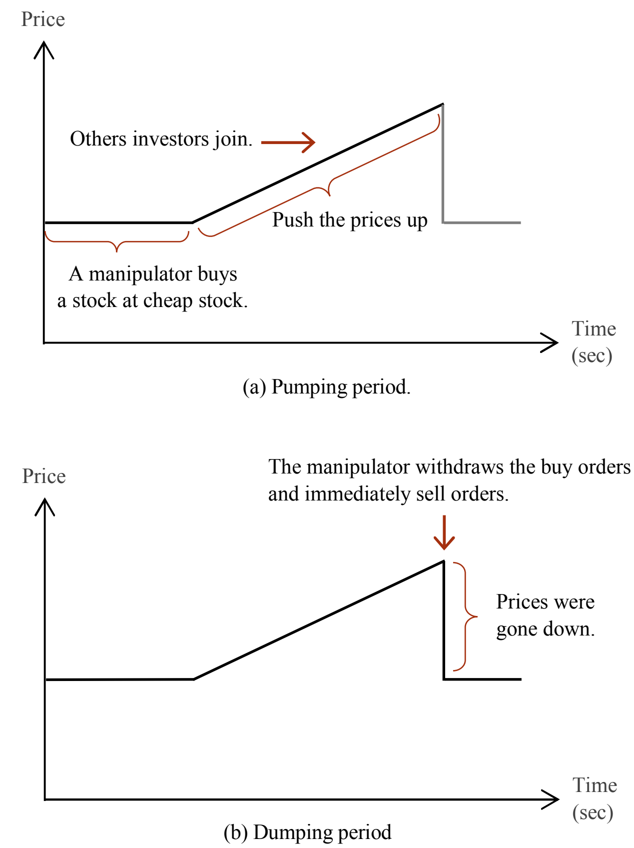


Fig. 2. Pump-and-dump diagram.

V. Models

There are not many known manipulation cases, because only the cases with severe damage were prosecuted. Compared to regular trading data, there is a big gap between the size of normal cases and manipulation cases. This is one of our challenges about data characteristics. So, we look for an unsupervised learning technique that can group and interpret data based only on input data (normal cases). Moreover, time series data are also concerned in this case. In the case of feedforward neural networks, input data are fed into the networks and pass through the hidden layer and the output layer. All decisions from the networks depend on the current inputs. There is no memory to remember the past and future points. So, this is the reason why recurrent neural network (RNN) was chosen to model our time-series data for stock price manipulation detection with GANs. GANs are a type of unsupervised models that can create original look-alike data. GANs can learn styles and characteristics of the source data. Therefore, GANs and RNN are merged to use together. This section describes the mathematical background of GANs and RNN that will be applied for stock manipulation detection.

A. GANs

GANs consist of two primary components, a generator G and a discriminator D. They were introduced by Goodfellow et al. [31]. Two networks try to compete with each other. The generator G synthesizes a realistic sample that is similar to the training set by learning the distribution of input data from the latent space. The discriminator D performs a classification task. The discriminator D differentiates real data from the training set (class 1) and generated data from the generator G (class 0). Generator's weights are frozen while the discriminator D is training and vice versa. Both machines try to fine-tune their parameters and become better in their capabilities. Although each player depends on each other, each player cannot control the other's parameters. The objective function of GANs is as follows:

$$\min_{\theta^{(G)}} \max_{\theta^{(D)}} \mathrm{E}_{x \sim p_{data}} [\log D_{\theta_d}(x)] + \mathrm{E}_{z \sim p_{(z)}} [\log(1 - D_{\theta_d}(G_{\theta_g}(z))], \qquad (1)$$

where z is vectors in the latent space. GANs solution involves minimization in the outer loop and maximization in the inner loop. The discriminator D aims to maximize those two terms. So, in the first term, the discriminator D should give an output '1' for real data. In the second term, the discriminator D should give an output '0' if the generator G cannot fool the discriminator D. On the other hand, the generator G aims to minimize only the second term. The discriminator D should give an output '1' for the generated data.

B. LSTM

An LSTM network [32] is a type of RNNs, which takes their outputs from the previous time step to be fed into inputs of the next time step. The main idea of LSTM is to learn what to forget and what to remember. So, it can learn patterns of stock movements. LSTM was used to predict the future by learning historical data [33], [34], [35]. In this paper, LSTM has been applied as a base structure of GANs to identify anomaly patterns. LSTM networks compose of forget gate f, input gate i, memory cell update C, and output gate o. The forget gate decides how much of the historical data are kept. The input gate decides which values will be updated. The memory cell update will update the old cell state into the new state. The output gate chooses the parts of the cell state that are going to be sent out. Then, the output h of LSTM units is calculated from the input $x(t)$, the previous output $h(t-1)$, and the current cell state $C(t)$. LSTM operations are calculated from the following equations:

$$f_t = \sigma\left(W_f \bullet [h_{t-1}, x_t] + b_f\right), \qquad (2)$$

$$i_t = \sigma\left(W_i \bullet [h_{t-1}, x_t] + b_i\right), \qquad (3)$$

$$\widetilde{C}_t = \tanh\left(W_C \bullet [h_{t-1}, x_t] + b_C\right), \qquad (4)$$

$$C_t = \left(f_t \times C_{t-1}\right) + \left(i_t \times \widetilde{C}_t\right), \qquad (5)$$

$$o_t = \sigma\left(W_o \bullet [h_{t-1}, x_t] + b_o\right), \qquad (6)$$

$$h_t = o_t \times \tanh\left(C_t\right). \qquad (7)$$

The stock trading data are time-series, and GANs can be applied for stock manipulation detection. Thus, we propose a combination of LSTM-GANs, which can learn the characteristics of the stock trading data. There are several ways to use LSTM [36]. In Fig. 3, the generator G takes a random noise from the latent space at each time step. Sample outputs are generated from the generator G, and toggled with the real samples to be inputs of the discriminator D. The discriminator D obtains the toggled samples, performs classification tasks, and gives a single output (real or fake).

VI. Experiments

This section describes data selection, data preprocessing, normalization, and experiments for training and testing of the models. For model evaluation, the models will be tested on both normal and manipulation cases. We had no real manipulation cases for testing the model, because the cases are limit and rare to obtain. Thus, we synthesized and injected manipulative patterns mixing with the normal cases into the models for evaluation. This is the best practice method for researchers in this field [37].

A. Datasets

Our training dataset consists of big market capitalization companies from SET. The companies are ranked as the top 50 on SET (SET50) [38]. This means that the companies are large market capitalization, high liquidity, and passing of requirements in the distribution of shares to minor shareholders. The dataset covers 22 full trading days (January 2-31, 2014). There were no reported manipulation cases. Therefore, we assumed that the data are regular trading activities. The dataset was divided into two groups: 70% for training as regular trading activities (seen normal data) and 30% for testing. The testing data was used as normal trading activities to measure the detection performance. They were also combined with manipulative patterns and used as manipulative trading activities.

B. Data preprocessing and normalization

We focus on the format of data called limit order book (LOB). LOB describes buy and sell orders at specific prices and volumes. The orders are arranged by time priority when traders enter different prices into the market system. The raw data contain event types from buyers and sellers such as insert orders, cancelled orders, and matched orders at specific timestamps. From the raw data, we performed order book reconstruction to monitor the order book step by step from all transactions. LOB tables were set in five depths with one second sampling period.

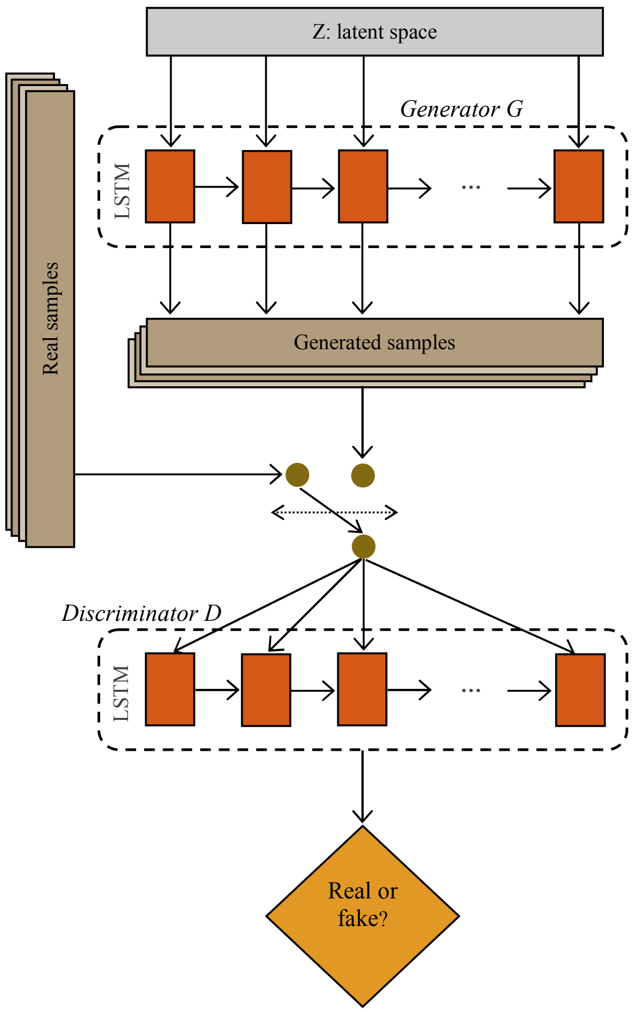


Fig. 3. Architecture of LSTM-GANs training.

Input features were arranged in three dimensions for RNN data format. The data's dimensions were the number of samples, the number of features, and the number of time steps (sequence length). There were 15 input features per time step. The input features are described in TABLE I. The process of data normalization is shown in Fig. 4. The raw data from SET were transformed into limit order book (LOB) format. Then, the missing data in the LOB table were filled. We defined d as the market depth where $d = \{1, 2, 3, 4, 5\}$. Market depth at $d = 1$ represents the best bid/ask price. Normally, the spread between prices at market depth d and $(d+1)$ is based on stock's price range from SET [39]. So, the LOB table has equal spread for every market depth. Therefore, there is no need to put all prices to be features, because they are equally spread apart. Therefore, we only put the best bid/ask prices into the input features format. After that, the best bid/ask prices and the volumes were normalized. The best bid/ask prices were normalized by transforming them to a price percentage change because each stock has different price and size. Stock price manipulations also involve changes in prices. The price percentage change is called rate of change (ROC), which is calculated by:

$$\Delta P = \frac{P_t - P_{(t-n)}}{P_{(t-n)}} \times 100 \tag{8}$$

The ROC compares the current price with the price at the previous n time steps. For example, P_t is the best bid price of a stock at 10:00:10 AM and $P_{(t-n)}$ is the best bid price of the same stock before one second (10:00:09 AM).

TABLE I. INPUT FEATURES

Column	Variable explanation
1	best bid volume
2	second best bid volume
3	third best bid volume
4	fourth best bid volume
5	fifth best bid volume
6	best ask volume
7	second best ask volume
8	third best ask volume
9	fourth best ask volume
10	fifth best ask volume
11	best bid price
12	best ask price
13	matched volume
14	Cancelled volume (bid side)
15	Cancelled volume (ask side)

The stock volumes were normalized by a common logarithm (base 10) and then with z-score. Volumes are high values positive numbers. The use of logarithm will help soften the discrepancy between different activities. The normalized z-score as shown in the following equation will help adjust the value to have a normal distribution.

$$z_i = \frac{\log_{10} V_i - \overline{\log_{10} V}}{\sigma} \tag{9}$$

C. Model structure and its parameters

Fig. 5 shows the training process of the LSTM-GANs. The LSTM-GANs were trained on sequential data with 15 input features. Variable Z was sampled from a normal distribution. We set Z dimension as 150. This came from 15 input features with 10 time steps. The random noise was fed to a dense layer and the LSTM layers respectively. After that, the outputs from the LSTM units were reshaped into three dimensions. Input features from the generator G and the discriminator D were set to have a batch size of 8192. Both of them were toggled as inputs for the discriminator D. The input features were entered into the LSTM layers and the dense layers of the discriminator D for classification task respectively. An output from the discriminator D should be '1' for real data and '0' for generated data.

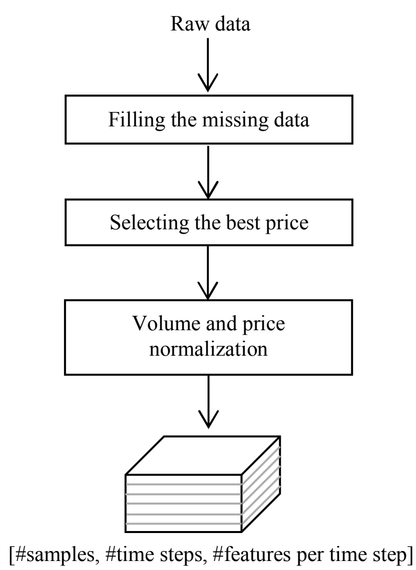


Fig. 4. Process of data normalization.

D. Model testing

The models were trained with normal sequential data cases. For performance evaluation of stock price manipulation detection, we tested the model under the assumption that our data have no abnormality from stock price manipulations. There are four inputs to be evaluated with the models as shown in TABLE II. As mentioned above, we injected a manipulative pattern into the input features. The pattern is referred from the pump-and-dump characteristics in section II. The manipulative pattern is illustrated in Fig. 6. We assumed a manipulation period as 5 seconds. The first four seconds were in the pumping period, and the rest were in the dumping period. The increases for both bid and ask prices were a constant rate of 0.1% per second in the pumping period. In this period, bid volumes were also injected with 20% of the normalized volumes in each time step. In the dumping period (the fifth second), the bid orders were then cancelled and the ask orders were quickly executed. So, the cancelled bid volumes and matched volumes were injected four times of the first four normalized volumes.

E. Experimental Results and Performance Discussion

We reported the detection performance of our models based on the accuracy which is calculated by the number of correct predictions per the total number of data. We divided the inputs into four groups: (1) seen normal cases with target '1' (real), (2) unseen normal cases with target '1', (3) seen normal cases plus a manipulative pattern with target '0', and (4) unseen normal cases plus a manipulative pattern with target '0'. The model accuracy for each case is shown in Table II. In the cases of normal trading, the model performed better on the seen data than the unseen one. This is typical for most machine learning problems. For the cases of manipulative trading, the result was the opposite. The seen normal data with an added manipulative pattern got lower accuracy, because the models may partly remember the training data too well. When the manipulative pattern was added, it did not sufficiently change the data characteristics. Therefore, the discriminator thought that this was genuine data. The unseen data did not have this effect, and therefore achieved a higher accuracy of 68.10%

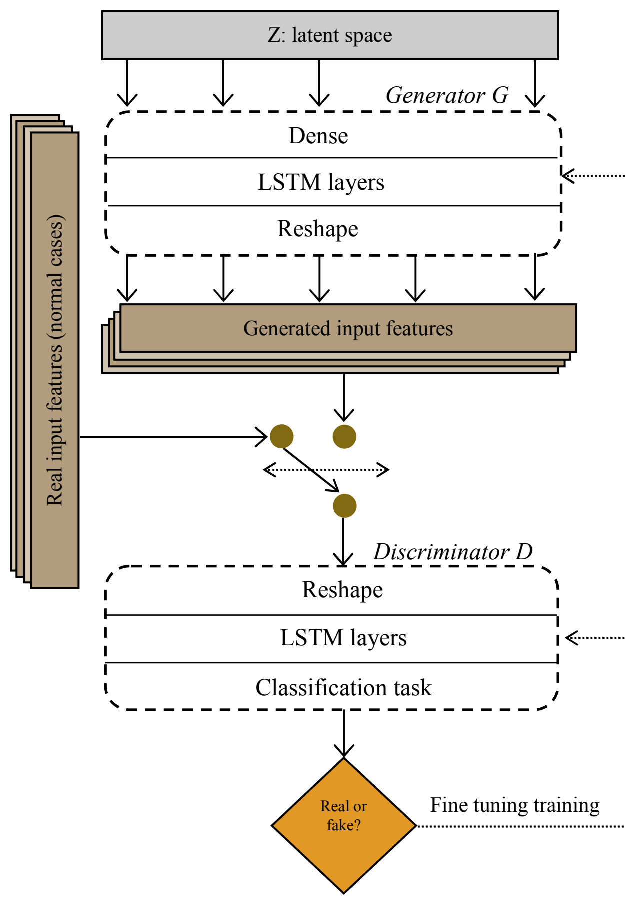
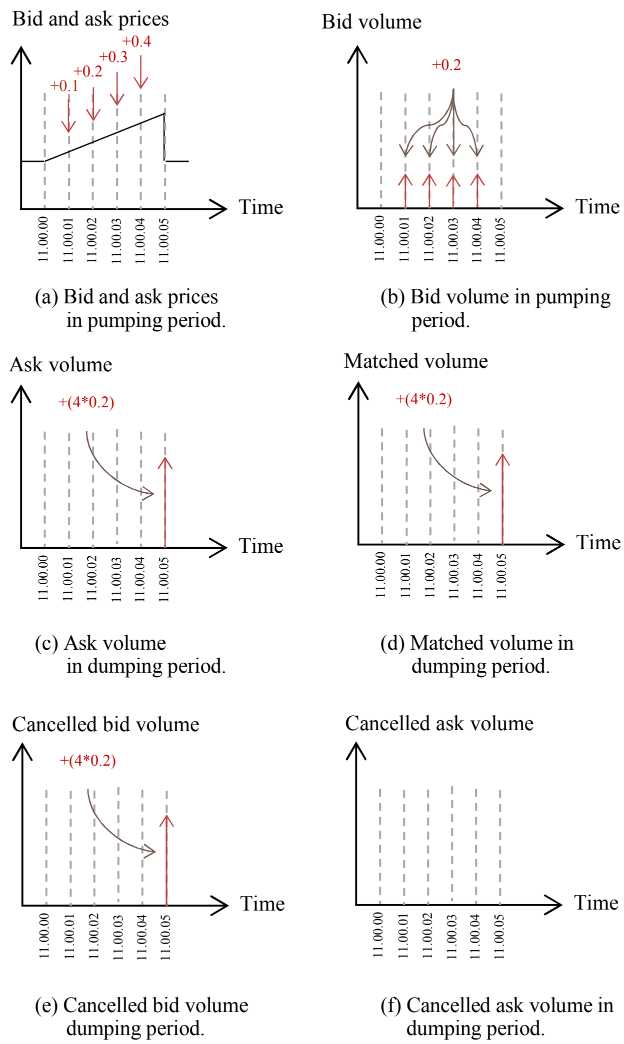


Fig. 5. Training of GANs for both Generator and Discriminator

TABLE II. MODEL ACCURACY

Inputs	Model accuracy
1. seen normal cases	80.05%
2. unseen normal cases	73.09%
3. seen normal cases + manipulative pattern	61.46%
4. unseen normal cases + manipulative pattern	68.10%



(a) Bid and ask prices in pumping period.

(b) Bid volume in pumping period.

(c) Ask volume in dumping period.

(d) Matched volume in dumping period.

(e) Cancelled bid volume dumping period.

(f) Cancelled ask volume in dumping period.

Fig. 6. Manipulated pattern injection (Pump-and-Dump)

VII. CONCLUSION

Stock price manipulation detection is a challenging research for machine learning when there are not many manipulation cases available for training. To work around this problem, unsupervised learning techniques had to be used in order to learn the trading data with no labels. This paper proposes a combination of LSTM and GANs for stock price manipulation detection. For the training part, normal trading activities were used for the models to learn the normal data characteristics. Our experiment showed that the LSTM-GANs were effective for detecting stock price manipulations even though a set of stock manipulative patterns is limited. The model performed well with both seen and unseen data. To the best of our knowledge, this is the first research to apply deep generative adversarial networks on the problem of detecting stock price manipulations. Future directions for our research are to use a more extensive dataset with the models. We also plan to evaluate the models with other stock price manipulation tactics.

ACKNOWLEDGMENT

The authors would like to thank the Stock Exchange of Thailand (SET) for supporting this research. This research is also supported by the National Science and Technology Department Agency (NSTDA), Thailand Graduate Institute of Science and Technology (Grant: TG-44-20-59-019D).

REFERENCES

[1] "Manipulation," U.S. Securities and Exchange Commission, 28 March 2008. [Online]. Available: https://www.sec.gov/fast-answers/answerstmanipulhtm.html. [Accessed 2 April 2017].

[2] Y. Cao, Y. Li, S. Coleman, A. Belatreche and T. M. McGinnity, "Detecting Price Manipulation in the Financial Market," in Computational Intelligence for Financial Engineering & Economics (CIFEr), London, 2014.

[3] A. Franklin and G. Douglas, "Stock-price manipulation," The Review of Financial Studies, vol. 5, no. 3, pp. 503-529, 1992.

[4] 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.

[5] Y. C. Huang and Y. J. Cheng, "Stock manipulation and its effects: pump and dump versus stabilization," Review of Quantitative Finance and Accounting, vol. 44, no. 4, pp. 791-815, 2015.

[6] R. A. Jarrow, "Market manipulation, bubbles, corners, and short squeezes," Journal of financial and Quantitative Analysis, vol. 27, no. 3, pp. 311-336, 1992.

[7] 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.

[8] M. M. Carhart, R. Kaniel, D. K. Musto and A. V. Reed, "Leaning for the tape: Evidence of gaming behavior in equity mutual funds," The Journal of Finance, vol. 57, no. 2, pp. 661-693, 2002.

[9] M. Bagnoli and B. L. Lipman, "Stock price manipulation through takeover bids," The RAND Journal of Economics, pp. 124-147, 1996.

[10] R. Aggarwal and G. Wu, "Stock Market Manipulation-Theory and Evidence," Ssrn Electronic Journal, pp. 20-28, 2003.

[11] C. Comerton-Forde and T. J. Putniņš, "Measuring closing price manipulation," Journal of Financial Intermediation, vol. 20, no. 2, pp. 135-158, 2011.

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

[13] J. Mongkolnavin 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.

[14] K. Golmohammadi, Z. R. Osmar and D. Diaz, "Detecting stock market manipulation using supervised learning algorithms," in Data Science and Advanced Analytics (DSAA), 2014.

[15] F. Yang, H. Yang and M. Yang, "Discrimination of China's stock price manipulation based on primary component analysis," in International Conference on Behavioral, Economic, and Socio-Cultural Computing, 2014.

[16] R. Singh and S. Srivastava, "Stock prediction using deep learning," Multimedia Tools and Applications, vol. 76, no. 18, pp. 18569-18584, 2017.

[17] X. Ding, Y. Zhang, T. Liu and J. Duan, "Deep learning for event-driven stock prediction," in Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[18] R. Akita, A. Yoshihara, T. Matsubara and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in Computer and Information Science (ICIS), 2016.

[19] Y. Deng, F. Bao, Y. Kong, Z. Ren and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," IEEE transactions on neural networks and learning systems, vol. 28, no. 3, pp. 653-664, 2017.

[20] 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.

[21] A. Abdallah, M. A. Maarof and A. Zainal, "Fraud detection system: A survey," Journal of Network and Computer Applications, vol. 68, pp. 90-113, 2016.

[23] M. Ahmed, A. N. Mahmood and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, pp. 19-31, 2016.

[24] S. Qian, L. Xiaoping and L. Xia, "Road Detection from Remote Sensing Images by Generative Adversarial Networks," IEEE Access, 2017.

[25] A. Madani, M. Moradi, A. Karargyris and T. S. Mahmood, "Semi-supervised learning with generative adversarial networks for chest X-ray classification with ability of data domain adaptation," in IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, 2018.

[26] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele and H. Lee, "Generative adversarial text to image synthesis," arXiv preprint arXiv:1605.05396, 2016.

[27] L. Yu, W. Zhang, J. Wang and Y. Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," pp. 2852-2858, 2017.

[28] T. Schlegl, P. Seeböck, W. M. Waldstein, U. Schmidt-Erfurth and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in International Conference on Information Processing in Medical Imaging, 2017.

[29] R. B. Kiran, M. D. Thomas and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," arXiv preprint arXiv:1801.03149, 2018.

[30] H. Zenati, C. S. Foo, B. Lecouat, G. Manek and V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," arXiv preprint arXiv:1802.06222, 2018.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," in Advances in neural information processing systems, 2014.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[33] S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," in 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Cambridge, 2018.

[34] Z. Zhao, R. Rao, S. Tu and J. Shi, "Time-weighted LSTM Model with Redefined Labeling for Stock Trend Prediction," in International Conference on Tools with Artificial Intelligence, Boston, 2017.

[35] S. Kaushik, A. Choudhury, N. Dasgupta, S. Natarajan, L. A. Pickett and V. Dutt, "Using LSTMs for Predicting Patient's Expenditure on Medications," in International Conference on Machine learning and Data Science, Noida, 2017.

[36] A. Karpathy , "The Unreasonable Effectiveness of Recurrent Neural Networks," Andrej Karpathy blog, 21 May 2015. [Online]. Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/. [Accessed 10 Jan 2017].

[37] L. Cao, Y. Ou and Y. S. Philip, "Coupled behavior analysis with applications," IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 8, pp. 1378-1392, 2012.

[38] SET, "Constituents," 16 December 2013. [Online]. Available: https://www.set.or.th/en/market/constituents.html. [Accessed 5 April 2018].

[39] "Equities Trading," The Stock Exchange of Thailand, [Online]. Available: https://www.set.or.th/en/products/trading/equity/tradingsystem_p5.html. [Accessed 7 February 2018].

[40] M. J. Aitken, F. H. Harris and S. Ji, "Trade-based manipulation and market efficiency: A cross-market comparison," in 22nd Australasian Finance and Banking Conference, 2009.

[41] R. F. Roodposhti, F. M. Shams and H. Kordlouie, "Forecasting stock price manipulation in capital market," World Academy of Science, Engineering and Technology, vol. 80, pp. 151-161, 2011.