IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 11, 2021, accepted July 14, 2021, date of publication July 26, 2021, date of current version August 5, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3100359*

# Stock Price Manipulation Detection Using Deep Unsupervised Learning: The Case of Thailand

**TEEMA LEANGARUN**[1], **(Graduate Student Member, IEEE),**
**POJ TANGAMCHIT**[1], **(Member, IEEE), AND SUTTIPONG THAJCHAYAPONG**[2], **(Member, IEEE)**
[1]Department of Control Systems and Instrumentation Engineering, King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand
[2]National Electronics and Computer Technology Center, National Science and Technology Development Agency, Khlong Nueng, Pathum Thani 12120, Thailand

Corresponding author: Poj Tangamchit (poj.tan@kmutt.ac.th)

This work was supported in part by the Stock Exchange of Thailand (SET) and in part by the National Science and Technology Development Agency (NSTDA) Thailand under the Graduate Institute of Science and Technology Scholarship (Grant: TG-44-20-59-019D).

**ABSTRACT** Detecting stock price manipulation is a cat-and-mouse game. Manipulators have constantly devised new techniques to avoid detection. The majority of the related work employed supervised learning techniques, which necessitated known manipulation patterns as examples for their models to recognize. To catch unknown and never-before-seen manipulation, we used unsupervised learning to train deep neural networks for detecting stock price manipulation in order to detect unknown and previously unseen manipulation. The models were trained to recognize normal trading behaviors that were expressed in a limit order book. Anomaly trading actions that did not follow to the learned patterns were identified as manipulated. The strength of our method is that it does not require prior knowledge about the characteristics of manipulation. As a result, it is best suited for detecting new or unknown types of manipulation. Two model architectures were evaluated: autoencoder (AE) and generative adversarial networks (GANs). They were put to the test on six prosecuted real manipulation cases from the Stock Exchange of Thailand (SET). With a low false-positive rate, both models could identify five out of six cases. For practical application of the models, a strategy called ''MinManiMax'' was also proposed to optimize the decision boundary.

**INDEX TERMS** Anomaly detection, market abuse, stock markets, stock price manipulation, unsupervised learning.

## I. INTRODUCTION

The term ''manipulation'' was defined by the U.S. Securities and Exchange Commission as ''intentional conduct designed to deceive investors by controlling or artificially influencing an exchange market for security'' [1]. Exchange market regulators have a long way to fight market manipulation in many ways, such as enacting laws and establishing specialized teams to prevent it. Manipulation has an impact on market integrity, which is one of the most important requirements for financial markets to maintain their efficiency [2]. However, the acts are enticing due to the huge benefit that awaits if the operation is successful and uncaught. When one can gain a substantial benefit from market manipulation, the counterparties suffer significantly as well. This is unjust and should be rigorously prevented. In most countries, stock price manipulation is illegal. Governments have established specific organizations to prevent these illegal activities. Despite the fact that there were not many of them, these units have been shown to investigate and prosecute manipulations that cause significant damage. In the case of the Thai stock market, the Securities and Exchange Commission of Thailand (SEC) had a record of 47 prosecuted cases over a 12-year period from 2004-2016 [3]. There were few cases because detecting manipulation and finding clear evidence is difficult and time-consuming. Manipulation detection faces three challenges. First, manipulation can have many different forms and degrees of damage, ranging from minor to severe. The SEC must concentrate on major cases that have a high impact. Second, detecting market manipulation is a cat-and-mouse game [4]. Manipulators will try their best to conceal their intentions by adapting their behaviors over time. Detection techniques that rely on scanning fixed trading patterns become less effective as a result. Third, with the

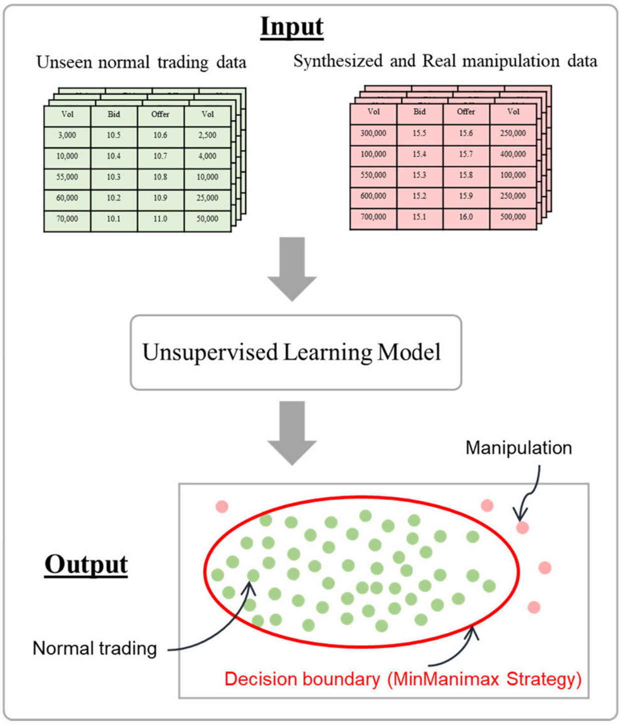
The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

advances in computerized trading and artificial intelligence (A.I.), high-frequency trading now executes more than 55% of trades in the U.S. stock market and more than 40% of trades in Europe [5]. This generates a large number of trading activities, so that illegal trading behaviors can be hidden well inside them. Furthermore, some algorithmic trading employs machine learning, which automatically adapts to the trading styles of the trader. According to Mizuta's [6] research, A.I. can learn illegal trading actions and use them to manipulate a market without awareness of the owners. The owners may be unaware or unconcerned as long as it generates profits for them. For these reasons, market regulators must use smart A.I. to keep them under control.

The use of A.I. and machine learning to detect market manipulation have mostly been studied in a supervised fashion. To train their models, the researchers used manipulative trading patterns. Manipulation patterns can be both real and simulated. The models demonstrated a high level of accuracy in detecting these patterns. However, it is unclear whether market manipulators are still using the same techniques in recent days, or whether they have continuously adapted over time, making their traits undetectable. New manipulation tactics will not be caught by the supervised models, because they can detect only patterns that are similar to the training data. Our research attempted to address this issue by employing an unsupervised learning technique to learn normal trading behaviors in the Thai stock market. Trading activities that deviate significantly from normal behaviors are regarded as suspicious and need further investigation. With recent advancements in A.I. and deep learning, high complexity models are promising to learn and cover most normal trading behaviors, which seemed infeasible in the past due to computation limitations in comparison to the number of various trading patterns. Unseen manipulation techniques can be pinpointed using this method for further examination by a human expert. This technique can also be combined with detecting fixed manipulation patterns to achieve the best results in catching known and unknown manipulation techniques. If an unknown manipulation technique is caught, its trait can be added to the database for future use with the fixed scan method in the future. The following are the main contributions to this study:

- The use of two unsupervised learning methods to detect stock price manipulation in order to fill knowledge gaps for unseen manipulation methods.
- The use of prosecuted real manipulation cases, which are publicized online by the SEC for the evaluation of both unsupervised learning models
- For practical usage, a strategy called "MinManiMax" was proposed to optimize the decision boundary.

Fig. 1 depicts our framework for detecting stock price manipulation. We used real market data from the SET. The event-based data were preprocessed into a time series of limit order books (LOBs) before being put into the models. Two types of models were used: long short-term memory autoencoder (LSTM-AE) and long short-term memory generative adversarial networks (LSTM-GANs). The models will learn normal trading behaviors of good governance stocks and classify outliers as manipulation. Two groups of data, unseen normal trading (green color) and manipulations (both synthesized and real cases) (red color) were expected to be distinguished from one another. To optimize the decision boundary, we introduced the "MinManiMax" strategy (hypothetically shown as the red oval in Fig. 1).


Input
Unseen normal trading data
Synthesized and Real manipulation data
Unsupervised Learning Model
Output
Manipulation
Normal trading
Decision boundary (MinManimax Strategy)

| Vol | Bid | Offer | Vol |
|---|---|---|---|
| 3,000 | 10.5 | 10.6 | 2,500 |
| 10,000 | 10.4 | 10.7 | 4,000 |
| 55,000 | 10.3 | 10.8 | 10,000 |
| 60,000 | 10.2 | 10.9 | 25,000 |
| 70,000 | 10.1 | 11.0 | 50,000 |

| Vol | Bid | Offer | Vol |
|---|---|---|---|
| 300,000 | 15.5 | 15.6 | 250,000 |
| 100,000 | 15.4 | 15.7 | 400,000 |
| 550,000 | 15.3 | 15.8 | 100,000 |
| 600,000 | 15.2 | 15.9 | 250,000 |
| 700,000 | 15.1 | 16.0 | 500,000 |

**FIGURE 1. Framework for detecting stock price manipulation. The stock market data were used in a limit order book time-series format. LSTM-AE and LSTM-GANs were trained on seen normal trading data and tested on unseen normal trading data, real manipulation data, and synthesized manipulation data. To optimize the decision boundary between the two classes, the "MinManiMax" strategy was proposed.**

The remainder of this paper is organized as follows. Section II discusses the study of manipulative trading in stock markets, as well as methods for detecting market manipulation activities. Deep unsupervised learning models for manipulation detection are introduced in Section III. Sections IV and V describe data preparation and two experiments, respectively. Section VI demonstrates how our models can be used to detect stock price manipulation. Finally, section VII summarizes our findings.

## II. RELATED WORK

Market manipulation activities not only contort legitimate market transactions in one marketplace, but also affect the capital allocation, investments, and savings in other markets [7]. Allen and Gale [8] categorized three types of market manipulation: action-based, information-based, and trade-based. Action-based manipulation creates situations that alter the value of companies (e.g., take-over bids [9]).

Information-based manipulation spreads false rumors to misinform others about the perceived value of a company and then makes profits by acting on the deceived price. Trade-based manipulation is the use of trading activities to manipulate the price with the intent of deceiving others and profiting from these deceptive activities. Along with asset value altering and insider trading, trade-based manipulation is one of the most significant threats to market efficiency [10]. Since the flash crash of 2010, one of the most significant turning points in financial markets, innovations and advances in financial technology have significantly contributed to a new financial reality for market regulators [11]. Many trade-based events occurred and brought a large number of studies into market manipulation. Our work is solely focused on detecting trade-based stock manipulative transactions.

The following related work is divided into two sections. The first section provides theoretical studies of market manipulation. These studies contribute to a better understanding of market manipulation and the factors that have a significant impact on it. The second section of the related work is a review of existing research papers on detecting stock price manipulation using supervised and unsupervised learning models.

### A. STUDY OF MANIPULATIVE TRADING IN STOCK MARKETS

Previously, stock manipulation research was conducted in the field of finance. The majority of them studied the characteristics of manipulation, allowing researchers to broaden their knowledge toward its detection. Kyle and Viswanathan [12] proposed that a trading strategy should be classified as illegal activity or price manipulation if the investors who break the rules intend to destroy market liquidity and pricing accuracy. The largest price swing is a key indicator for detecting market manipulation [8], [13]. Manipulative cases tend to raise rather than lower the stock price [14]. Palshikar and Bahulkar [15] also demonstrated that while a stock was manipulated, temporal price and volume patterns repeated themselves. In the context of company sizes, both small and large companies in developed and emerging financial markets were equally likely to be manipulated [14], [16]–[19]. Stock manipulation also involves two common types of investors. Allen and Gorton [20] found that uninformed investors may engage in trade-based manipulation by buying a stock to raise its price and then selling it at a higher price to make a profit. According to Aggarwal and Wu [14], potentially informed parties (e.g., corporate insiders, large shareholders) could be manipulators. Moreover, an examination of the characteristics and patterns of manipulated stocks, as well as their effects on Taiwan stock markets, revealed that many stocks were manipulated by large individual outsiders [16].

Throughout the evolution of financial markets, trading systems have been tremendously transformed from human operations into electronic operations. Market manipulation techniques have evolved from traditional forms (human traders) to modern forms (high-speed computer algorithms) as well [21]. Despite the fact that the ultimate goal of both market manipulation operations is the same, i.e., making profits, their effects are slightly different. In the context of human operations, traditional market manipulations distort natural prices or transactions for the benefit of the manipulators [22]. Cornering, wash trading, pump-and-dump, and other trade-based methods are commonly used to manipulate financial markets. In contrast to traditional market manipulation methods, new market manipulation methods (e.g., pinging, spoofing, and layering) are carried out on high-speed computers with the primary goal for distorting market information. The advancement of high-speed supercomputers that run on smart algorithms made new manipulation methods feasible and profitable [23]. These methods require high-speed submission and cancellation of large limit orders in a matter of seconds.

### B. METHODS FOR IDENTIFYING MARKET MANIPULATION ACTIVITIES

Detecting market manipulation is like a cat-and-mouse game. Manipulators must make their best effort to avoid being caught, while the regulators have to keep up with them. In technological changes, Teall [24] presented challenges for market regulators in terms of resources, detection, and enforcement. The lack of adequate resources has an impact on regulatory technology and expertise, while regulators continue to find a way to tackle new manipulative patterns in the markets. In terms of detection, high-speed transactions make it difficult for regulators to notice manipulative schemes. Transactions in the U.S. financial marketplace are measured in milliseconds [25], [26]. While trading information continues to accelerate, detecting new manipulative patterns will be even more difficult for regulators [27]. In today's financial markets, computer algorithms have been used to process a deluge of trading data [28]. The more digital trading data become available for algorithmic programs in financial markets, the more regulators will be challenged with the effort to manipulate the market. Apart from resources and detection, enforcement is also a challenge for regulators, especially with new methods of market manipulation. Law enforcement historically emphasized that manipulative patterns are caused by human actions, but not the ones created by computerized systems [29]. So, it is difficult to distinguish between some legitimate trading activities and illegal activities by manipulators [30], [31]. For instance, Chartis Research [32] revealed that the greatest surveillance challenge was a large volume of false positives. It is costly to keep the pace of verifying because regulators use human expertise to do so. The most common types of manipulative patterns (e.g., quote stuffing, wash trading, marking the close, layering, and spoofing) are normally listed for rule-based systems that detect suspicious activities based on known patterns.

Exactpro Test [33] stated that the next generation of market surveillance systems tends to apply machine learning algorithms rather than rules-based algorithms to detect different varieties of new manipulative patterns. There are two types of machine learning techniques: supervised learning and unsupervised learning. Supervised learning is effective in detecting known manipulation types and variations. It makes use of manipulative data labeling, which requires prior knowledge of that manipulation type. One of the long-standing challenges for this technique is to handle an insufficient amount of labeled data [34]. Some of the previous work is as follows. Luo *et al.* [35] used voting-based outlier mining and probability-based outlier mining on multiple time series, which outperformed traditional benchmark models. Öğüt *et al.* [36] classified manipulation cases using data mining techniques (artificial neural network and support vector machine) and multivariate statistical techniques (discriminant analysis and logistic regression). They found that the data mining techniques achieved better performance than the multivariate statistical techniques in terms of total classification accuracy, while the multivariate statistical techniques were superior in terms of sensitivity values. Roodposhti *et al.* [37] compared the performance of an artificial neural network (ANN), a multiple discriminant analysis (MDA), and a logit model to verify that those abilities had high accuracy to forecast manipulation. Golmohammadi *et al.* [38] compared the performance of various supervised learning algorithms. In terms of sensitivity and specificity, the results showed that Naïve Bayes outperformed the others. Zhang *et al.* [39] found that a random forest (RF) and a support vector machine (SVM) could make meaningful detections using real-time data in China stock markets. Li *et al.* [40] compared machine learning techniques and found that the most effective methods for detecting market manipulation were a decision tree (DT) and a k-nearest neighbor (k-NN). To detect manipulative behaviors, Zhai *et al.* [41] proposed static and dynamic models. Rather than relying on market indicators, they conducted a comprehensive analysis of the manipulative strategies. The static models (k-NN and one-class SVM) were designed to learn trading behaviors and detect manipulative patterns. The dynamic model, the adaptive hidden Markov technique, was used to explore the contextual relationships of the sequential trading data and detect the patterns of price manipulation.

Even though all the mentioned research papers used supervised learning methods to detect manipulation activities with high success rates, stock manipulators can find a new method very expeditiously [42]. Our work used unsupervised learning, which can detect unknown types of manipulation, even when they have no prior information about how these manipulations are carried out. The techniques are similar to the one used for predicting stock market movement [43], [44], but there have not been many studies that use unsupervised learning to detect stock price manipulation.

The following is a list of previous work in this area. In the Korean stock market, Kim and Sohn [45] proposed peer group analysis (PGA) to detect suspicious patterns in real-time series data. The daily closing price was monitored as a feature. The proposed method could improve the performance of the general peer group analysis by incorporating the weight of peer group members into the summary of their behaviors and also updating the weight based on the newly-observed data. However, close prices were used as features, which contain less information than order books. Golmohammadi and Zaiane [46] proposed a prediction-based contextual anomaly detection (CAD) method on different industrial sectors of the S&P 500. The price of the security was monitored. The study found that the method outperformed both the Naïve predictor and k-NN. Al-Thani *et al.* [47] extended the work in [46] and developed a new preprocessing step for improving the recall of the anomaly detection. The authors used their methods not only in the S&P 500, but also in Qatar Stock Market (QSM). Even though manipulations were effectively detected using their method, many normal cases were pinpointed incorrectly because the precision of the algorithm was low. There are three papers from the same group of researchers who implemented kernel density estimation (KDE) for stock manipulation detection [48]–[50]. They used an open-source LOBSTER database and injected manipulative patterns (for example, sawtooth and spike pattern) into the normal trading data. In their first paper, they used Empirical mode decomposition (EMD) with KDE to extract the level of frequency components without labeling input data. They compared the performance of K-means, principal component analysis (PCA), and Dirichlet process Gaussian mixture model (DPGMM) and concluded that the proposed model outperforms the existing approaches by a maximum of 84%. Their second paper used the Dendritic cell algorithm with KDE to capture the abnormal trends in a dataset without any labeling information. The results showed that the performance evaluation of DCA-KDE based outperformed the k-NN approach. Their third paper employed Kernel PCA to extract important features, which were then passed to the MKDE for detecting manipulation. The result showed that this method outperformed other benchmark approaches, including the adaptive hidden Markov model with anomaly states, Naïve Bayes, Probabilistic Neural Network (PNN), PGA.

The unsupervised learning approach relied on detecting anomalies that did not conform to the normal trading behaviors learned by the models. The limitation of this approach is that there are many varieties of normal trading behaviors by both human and automated trading systems. It could be argued that the models cannot cover all of them, and thus identify them as manipulation. With the need for high-complexity models, deep neural networks emerge as a solution to this problem. From the literature survey, we found two papers (our previous work and work by Rizvi *et al.*) that used deep neural networks for stock price manipulation. Our previous work [51] was a pioneer that used the LSTM-GANs technique for stock price manipulation detection in the SET. The model achieved an accuracy of 68% in detecting only the synthesized pump-and-dump scheme without requiring knowledge about the manipulation patterns. Rizvi *et al.* [52] implemented AE using the information captured by the affinity matrix to detect stock price manipulation. They used the same dataset and injected the same manipulative patterns as in [50]. The input features were extracted using a discrete wavelet transform and were then used to compute affinity and grouped using the proposed clustering techniques. In terms of AUC values, the proposed approach outperformed the benchmark methods, such as k-NN, PCA, OCSVM, and k-means.

This paper fills the gap of the previous work by using prosecuted real manipulation cases in the experiments rather than using synthesized manipulation like the previous work. Because the models, LSTM-GANs and LSTM-AE, were trained with only normal trading behaviors, they have no prior knowledge of these manipulation cases. A strategy to detect real manipulation in practice is also proposed.

## III. DEEP UNSUPERVISED LEARNING MODELS FOR MANIPULATION DETECTION

With the rapid growth of deep generative models, unsupervised deep learning has shown its potential in feature representation for anomaly detection. In this paper, two popular generative models, AE and GANs, were used to detect stock price manipulation using LSTM layers to capture time-series behaviors of manipulators and other traders in the LOB data.

### A. LSTM-AE MODEL

An AE is a typical feed-forward neural network containing an encoder and a decoder in which their parameters are trained by back-propagation [53]. To detect market manipulation, an AE was used to extract the characteristics of historical normal trading data and reconstruct the input sequence. Normal trading data, which were used to train the model, will have a lower reconstruction error than that of manipulation data, of which the model has never seen. In this paper, we applied a time-sequential unit in the AE framework using the LSTM network. Fig. 2 depicts the model structure.

There were four layers on both the encoding and decoding sides. A sub-array of size 10 timesteps with 15 features was one of our input samples. The first encoding layer was LSTM with 64 hidden units, a hyperbolic tangent activation function, and a sigmoid recurrent activation function. The following encoding layers were 32, 16, and 8 hidden units for compression. All encoding layers emitted a signal on each cell per timestep, except the encoding layer 4. Only the last timestep cell emitted signals to get a 2D array for the next layer. The next RepeatVector layer replicated the last output of the final encoder layer. This process prepared the input for the first LSTM decoding layer. So, the output from the RepeatVector were then fed into the first decoder layer in every timestep. The decoding layers were stacked in the reverse order of the encoding layers. Their hidden units were 8, 16, 32, and 64 respectively. The final layer was the TimeDistributed layer with the Dense layer to create a vector of length equal to the number of features. The LSTM-AE aimed to make the reconstructed inputs as close to the original inputs as possible. The mean squared error loss function, the stochastic optimizer Adam with default parameters, and Glorot initialization were used to train our model.

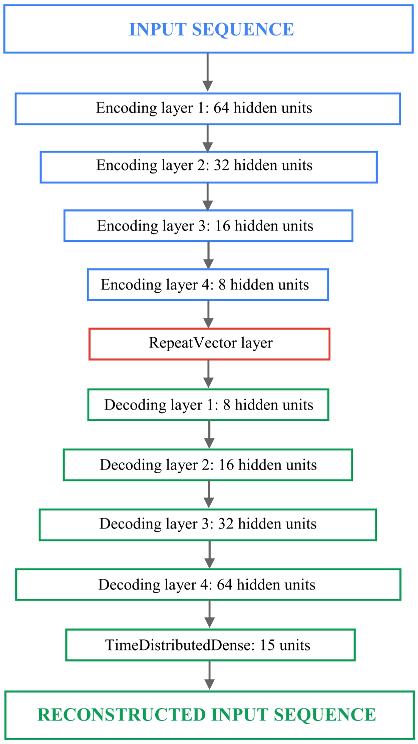

INPUT SEQUENCE
Encoding layer 1: 64 hidden units
Encoding layer 2: 32 hidden units
Encoding layer 3: 16 hidden units
Encoding layer 4: 8 hidden units
RepeatVector layer
Decoding layer 1: 8 hidden units
Decoding layer 2: 16 hidden units
Decoding layer 3: 32 hidden units
Decoding layer 4: 64 hidden units
TimeDistributedDense: 15 units
RECONSTRUCTED INPUT SEQUENCE

**FIGURE 2. The framework of the proposed model: LSTM-AE for stock price manipulation detection.**

### B. LSTM-GANs MODEL

The other proposed unsupervised model is based on GANs, which consists of two networks, a generator G and a discriminator D. The generator G attempts to synthesize samples $G(z)\epsilon R^d$ where $z\epsilon R^m$ is the random variable of some prior distribution and deceive the discriminator D by creating look-alike trading features as much as it can. The goal of discriminator D is to distinguish between genuine normal trading features (a training set) and forgeries (the output of the generator G). So, both are trained in an adversarial process. Fig. 3 depicts an LSTM-GANs diagram. The generator G gets random noise, extracts the features, and generates the fake normal trading features. We implemented the generator G based on LSTM with one-to-many architecture because it attempts to extract a sequence of trading features from single random noise. The discriminator D differentiates normal trading features from the generator G and the real input features in terms of real or fake. The original GANs cost function is defined as the following minimax game.

$$\begin{aligned}\min_{G}\max_{D} V(D,G) &= \mathrm{E}_{x\sim p_{data}(x)}\left[\log D(X)\right]\\ &\quad+\mathrm{E}_{z\sim p_z(z)}\left[\log\left(1-D\left(G\left(Z\right)\right)\right)\right], \qquad (1)\end{aligned}$$

where $p_{data}(x)$ denotes the distribution over normal trading data in the space $X$ and $p_z(z)$ is defined as the distribution over random variables $Z$ in the latent space.

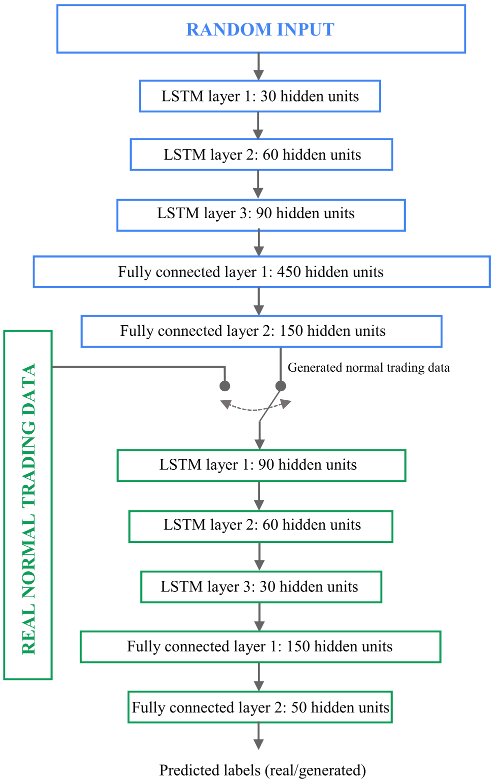

RANDOM INPUT
LSTM layer 1: 30 hidden units
LSTM layer 2: 60 hidden units
LSTM layer 3: 90 hidden units
Fully connected layer 1: 450 hidden units
Fully connected layer 2: 150 hidden units
Generated normal trading data
REAL NORMAL TRADING DATA
LSTM layer 1: 90 hidden units
LSTM layer 2: 60 hidden units
LSTM layer 3: 30 hidden units
Fully connected layer 1: 150 hidden units
Fully connected layer 2: 50 hidden units
Predicted labels (real/generated)

**FIGURE 3.** The framework of the proposed model: LSTM-GANs for stock price manipulation detection.

Because the G model was trained to generate normal trading data, the D model was expected to return a value of 1 when it is fed with the normal trading cases but a value of 0 otherwise. For this study, we used an LSTM network withdepth 3 and hidden units of 30, 60, and 90 for the generator. The outputs of the LSTM layers were then flattened and placed in two fully connected layers, which included 450 and 150 hidden units with ReLU and linear functions respectively. We set our latent space dimension as the number of features. The random values were drawn from a uniform distribution with values ranging from 0 to 1. The LSTM structure of the discriminator is the same depth as the generator with 90, 60, and 30 hidden units. In the discriminator classification task, we used two fully connected layers with 150 and 50 hidden units, respectively. These networks have a total of 709,091 parameters. In the adversarial training stage, our networks were all optimized with Adam with default parameters. We also used Glorotinitialization and tuning to ensure that the training converged to a good solution.

## IV. DATA PREPARATION

This section describes how we prepared normal trading and manipulation data for both models' training and testing.

### A. STOCK SELECTION

We chose stocks with varying price ranges and market capitalizations to be used as normal trading data. Even though there was no report of manipulation on these stocks, it is possible that there were manipulations that were undetected. For the best practice, we only selected stocks from the Good Corporate Governance list by Corporate Governance Report of Thai Listed Companies [54], which is a global standard by the Organization for Economic Co-operation and Development (OECD) Principles of Corporate Governance. This can ensure that there is a low possibility of manipulation in these normal trading stocks. To cover a wide range of normal trading activities, we selected eight large-capitalization companies, four medium-capitalization companies, and four small-capitalization companies from various SET industry groups [55]. Their prices range between 2 and 400 THB [56]. The information is summarized in Fig. 4.

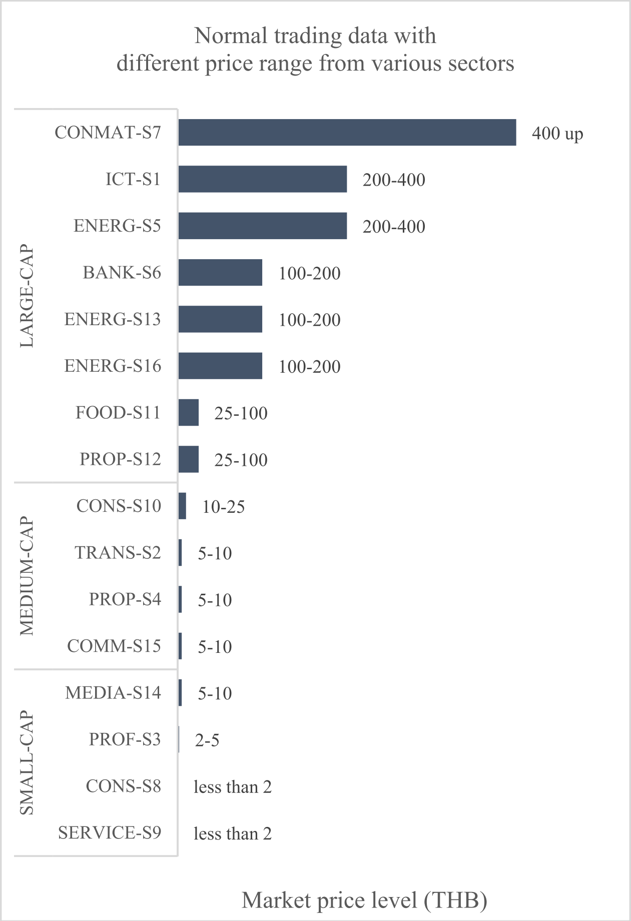


**FIGURE 4.** Normal trading data from various sectors with varying price ranges and market capitalizations.

In most problems involving detecting an anomaly in time series, abnormal cases occur infrequently. From 2004 to 2016, we looked for real manipulation cases prosecuted by the SEC [3]. There were 47 cases in total, with only seven trade-based manipulation cases reported during that period. As shown in Table 1, six of the seven cases were completed and used to conduct our experiments. Real manipulation stocks were named RM-1 to RM-6. The list of Good

**TABLE 1.** Real manipulation cases (Information as revealed to public).

| Cases | Descriptions | Manipulation periods |
|---|---|---|
| 1) RM-1<br>• Market cap.<br>○ Small<br>• Industrial sector<br>○ SERVICE<br>• Price range<br>○ 25-100 THB | The manipulator traded RM-1 shares continuously in such a way that the price of such shares was manipulated, sustained, and influenced in order to attract others to buy and sell the shares. During the manipulation period, the share price and the average daily trading volume increased by approximately 130% and 1070%, respectively. | 28 Apr – 30 May, 2014<br>(22 trading days) |
| 2) RM-2<br>• Market cap.<br>○ Small<br>• Industrial sector<br>○ STEEL<br>• Price range<br>○ 0-2 THB | Manipulator's action caused the price and trading volume of RM-2 shares during the trading period to be inconsistent with the normal market conditions, misleading others into believing that RM-2 shares were in a high demand at the time and temping them to trade such shares accordingly. | 7 Jan, 2013<br>(1 trading day) |
| 3) RM-3<br>• Market cap.<br>○ Small<br>• Industrial sector<br>○ STEEL<br>• Price range<br>○ 0-2 THB | During the manipulation period, the highest level of the day was an increase of 11% compared to the opening price or a 14% increase compared to the previous date's opening price. The overall trading volume increased by 87%. | 16 Sep, 2013<br>(1 trading day) |
| 4) RM-4<br>• Market cap.<br>○ Small<br>• Industrial sector<br>○ CONMAT<br>• Price range<br>○ 25-100 THB | Manipulators pushed the share price higher and matched the trading orders within the group in concealment to mislead others into believing that the fair price of RM-4 had changed. They also traded RM-4 shares on an ongoing basis, causing the share price to deviate from normal market conditions. | 3 Oct, 2012<br>(1 trading day) |
| 5) RM-5<br>• Market cap.<br>○ Small<br>• Industrial sector<br>○ CONMAT<br>• Price range<br>○ 5-10 THB | Manipulators used multiple trading accounts to trade RM-5 shares in concealment and on a continuous basis to mislead others into believing that RM-5 shares were traded in large volumes. | 9 Jan – 28 Feb, 2013<br>(37 trading days) |
| 6) RM-6<br>• Market cap.<br>○ Small<br>• Industrial sector<br>○ PETRO<br>• Price range<br>○ 2-5 THB | Manipulators used multiple trading accounts to trade RM-6 shares in concealment and on a continuous basis to mislead others into believing that RM-6 shares were traded in large volumes. | 10-27 Sep, 2013<br>(14 trading days) |

Corporate Governance did not include any real manipulative stocks. The first column consists of market capitalization, industrial sector, and price range of each real manipulative case. The second column contains the information of about each manipulation case. The manipulation periods are revealed in the last column.

Despite knowing which stock was manipulated and on which day the manipulations occurred, the regulator did not reveal details about manipulative patterns (e.g., which patterns of trade-based manipulations, the exact time of starting and ending points in manipulative stocks, and when non-bona fide orders are placed or canceled) due to market security concerns. We assumed that all trading activities in those intraday trading periods were labeled as manipulated, based on best practices in stock price manipulation detection, even though manipulation activities may not occur all of the time.

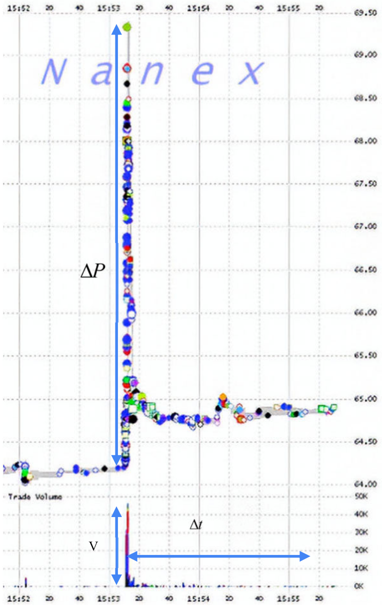


**FIGURE 5.** Manipulation of WAB on the NYSE on December 14, 2011 [79] (Photo Courtesy of Nanex). In four seconds, the short-term pump-and-dump pattern occurred. In a single second, the price increased by 8% before returning to slightly above the initial level three seconds later.

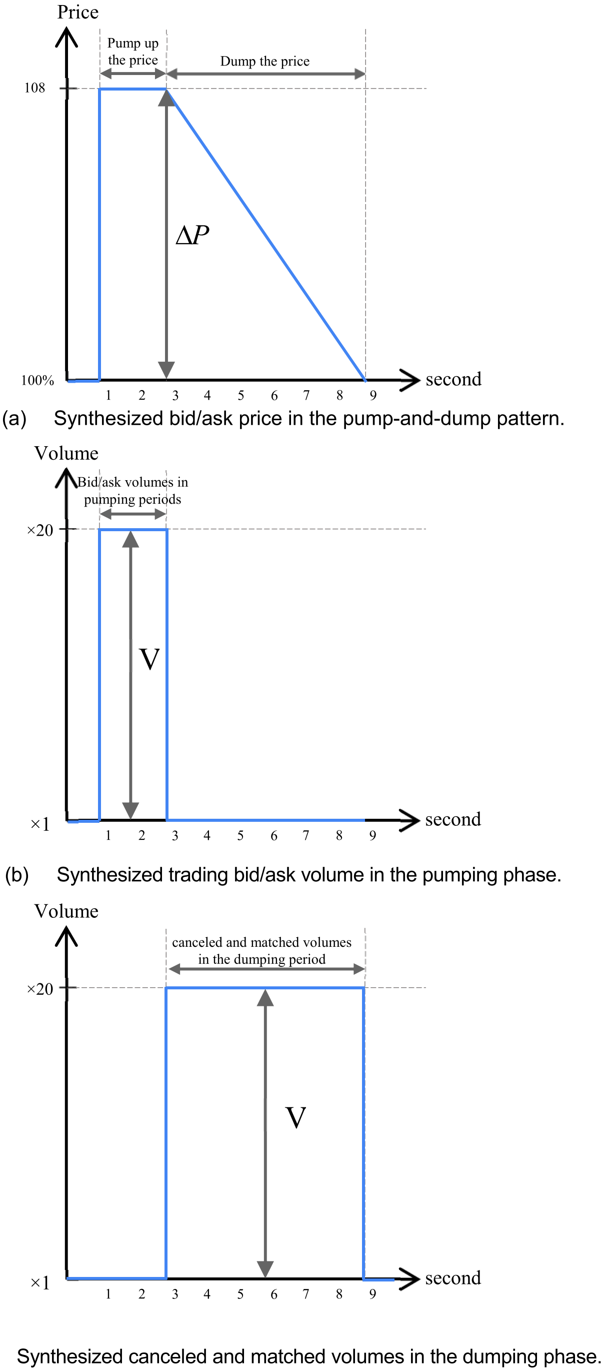
This may cause our label to become noisy, making it difficult for our models to achieve high accuracy.

Furthermore, due to the lack of real manipulation cases as testing data, we synthesized a manipulative pattern and injected it into any normal trading data. This synthesized method is an acceptable method in this research field [57], especially when labeled data are limited and difficult to obtain. The same manipulation pattern was injected into normal trading data from various companies. Pump-and-dump was chosen as a template in our study based on a real case from Nanex, a company that provides streaming market data services and real-time analysis. It is the most prevalent type of stock market manipulation [16]. The stock (WAB) was manipulated on the New York Stock Exchange (NYSE) on December 14, 2011, as shown in Fig. 5 [58].

A pump-and-dump pattern occurs when a manipulator enters large non bona-fide buy orders to raise the price. This act seeks to artificially inflate stock demand levels. During this time, genuine orders are entered into the trading system by innocent investors using trend following strategies. The manipulator cancels his orders shortly after the first one is placed and then executes a trade on the opposite side of the innocent investors. As a result, the manipulator can profit from this strategy by purchasing at a lower price than a regular buy order. At its peak in the real manipulation case (Fig. 5), the stock price increased by 8% ($\Delta P$) in one second. The trading volume (V) was also increased by 20-30 times from its initial level. Three seconds later, the stock price and the trading volume both dropped and returned to slightly above the initial level. Based on the real case from Nanex, we synthesized a pump-and-dump pattern with the amount of price and volume changes. Because of the different timeframes, the length of the artificial manipulation period $\Delta t$ differed from that of the real case. The real event had a total length of four seconds, while our timespan had eight seconds. The price change percentage ($\Delta P$) was set proportionally. During the manipulation period, the trading volume V was increased 20 times the average volume. The manipulative pattern was randomly placed within 10 timesteps of input features. For example, the pump-and-dump period could begin at the second timestep and end at the last timestep. Fig. 6 depicts the details of eight-second pump-and-dump pattern injection. The first two seconds were spent pumping and the remaining time was spent dumping. Fig. 6(a) displays synthesized bid/ask price in the pump-and-dump pattern. Initially, a start-to-peak bid/ask price was raised at a constant rate to its peak of 8% during the pumping period. The first orders were then canceled, and the price returned to the initial level proportionally six seconds later. The trading volume in five-depth was increased 20 times the average volume in the first two seconds of the pumping period in Fig. 6(b). The details of canceled and matched volumes during the dumping period are shown in Fig. 6(c). When the first orders were withdrawn, they were immediately executed. The trading volumes have returned to the initial level.

### B. DATA PREPROCESSING AND NORMALIZATION

This research obtained restricted stock trading data in a proprietary format from SET. The raw data were converted from the proprietary format to a five-depth LOB reconstruction, which is a standard format in most markets. A limit order book (LOB), also known as the market depth, is a list of all valid limit buy and sell orders that have yet to be fulfilled. A limit order is only be executed when a buyer and a seller mutually agree to exchange stocks at their pre-specified price. We assumed that all traders' actions and their intentions are reflected in the LOB dynamic. A trader's decision (i.e., insert buy or sell order, cancel orders) is also influenced by the state of the LOB [59], [60]. Manipulators seek to exploit this by duping other traders into misinterpreting the market by manipulating the state of the LOB. We used ten direct features from the LOB: bid-ask prices and volumes. The number of canceled and matching orders, which was considered a good indicator for stock manipulation [35], is another feature that involves stock manipulation activities. Montgomery [61] discovered that large market orders can be presumed that are possible to be manipulated and correlated with canceling trading orders. Furthermore, according to Chan and Ma's work [62], the canceled order is one of the activities that involved manipulation activities. These are the primary reasons why we included both of them as input features with LOB. However, the best bid-ask price was only chosen as one of the input features after the missing data were filled in. As a result, the five depths of bid-ask price had the same space between values based on tick sizes prescribed by the SET for securities trading.



(a) Synthesized bid/ask price in the pump-and-dump pattern.

(b) Synthesized trading bid/ask volume in the pumping phase.

(c) Synthesized canceled and matched volumes in the dumping phase.

**FIGURE 6. Synthesized pump-and-dump pattern.**

Our models were trained using regular trading data from the selected companies with varying price ranges and market capitalization. We had to train one model that could receive input data from any of these stocks because there were so few manipulation cases to test. These input features should be adjusted to be on the same scale in order to generalize the models to any stock. The price rate of change (ROC) was calculated to rescale a set of bid-ask prices at the current and previous times. As described in (2), ROC is one of the key

**TABLE 2.** List of normalized input features explanation.

| No. | Features | Symbols |
|---|---|---|
| 1 | Best bid price | $\Delta P_b$ |
| 2 | Best ask price | $\Delta P_a$ |
| 3 | Best bid volume | $Z_{b1}$ |
| 4 | Second best bid volume | $Z_{b2}$ |
| 5 | Third best bid volume | $Z_{b3}$ |
| 6 | Forth best bid volume | $Z_{b4}$ |
| 7 | Fifth best bid volume | $Z_{b5}$ |
| 8 | Best ask volume | $Z_{a1}$ |
| 9 | Second best ask volume | $Z_{a2}$ |
| 10 | Third best ask volume | $Z_{a3}$ |
| 11 | Forth best ask volume | $Z_{a4}$ |
| 12 | Fifth best ask volume | $Z_{a5}$ |
| 13 | Matched volume | $Z_m$ |
| 14 | Canceled volume (bid side) | $Z_{cb}$ |
| 15 | Canceled volume (ask side) | $Z_{ca}$ |

indicators generally used to confirm price movements [63]. (2). Because the tick size in the Thai stock market is quite large, we included only the best bid and the best ask prices. Prices are always distributed evenly across the five depths. Therefore, there is no reason to include all of them as features. We used Z-score normalization of the logarithmic volume to normalize volumes in the LOB. First, we smoothed the data over 10 timesteps using a simple moving average (SMA). The mean and the standard deviation used to calculate the Z-score were averaged over the course of its intraday trading.

$$\Delta P = \frac{P_t - P_{t-n}}{P_{t-n}} \times 100 \tag{2}$$

$$z_i = \frac{x_{\log_{10}(SMA_{Vol})_i} - \mu_{log_{10}(SMA_{Vol})}}{\sigma} \tag{3}$$

In total, 15 normalized input features were used: the ROC of the best bid-ask prices, the Z-score of log-SMA-volume in five market depths, the Z-score of log-SMA matched volume, and the Z-score of log-SMA canceled volumes. Table 2 describes the structure of input features.

## V. EXPERIMENTS

We conducted two experiments (Exp. 1 and Exp. 2) and evaluated the proposed stock price manipulation detection techniques on (1) normal trading data, (2) real manipulation data, and (3) synthesized manipulation data. Our two experiments were created to answer the following two research questions (RQ) listed below.

- *RQ1:* How effective are the LSTM-AE and LSTM-GANs in detecting stock price manipulation when they were trained with normal trading data and without prior knowledge about manipulations?
- *RQ2:* How well do both models perform in detecting real manipulation cases?

**TABLE 3.** List of datasets based on the experiments.

| Datasets | Sub-groups of datasets | Training | Exp.1 | Exp.2 |
|---|---|---|---|---|
| I | (1) seen normal trading data | x | | |
| I | (2) unseen normal trading data 1 | | x | x |
| II | (3) unseen normal trading data 2 | | x | x |
| III | (4) Synthesized manipulation data 1 | | x | |
| III | (5) Synthesized manipulation data 2 | | x | |
| III | (6) Real manipulation data | | | x |

### A. DATASETS

We divided our data after normalization into three datasets: (I) normal trading cases – companies in group 1, (II) normal trading cases – companies in group 2, and (III) manipulation cases. The datasets can be divided into six categories, as shown in Table 3.

#### 1) NORMAL TRADING CASES-Yyyyy–Xxxxx-COMPANIES IN GROUP 1

We chose eight companies from the Good Corporate Governance list that displayed normal trading behaviors. The SEC has no reported any price manipulation activities to any of them. Group 1 companies were divided into two groups: *(1) seen normal trading data* and *(2) unseen normal trading data 1*. Sixty percent of the available companies in group 1 were assigned to training. The remaining 10% and 30% of unseen normal trading data 1 were referred to as validation and test sets, respectively.

#### 2) NORMAL TRADING CASES-Yyyyy–Xxxxx-COMPANIES IN GROUP 2

*(3) Unseen normal trading data 2*, an additional eight companies were chosen solely for testing purposes. We also chose the entire dataset from the Good Corporate Governance list.

#### 3) MANIPULATION CASES

Manipulation cases were only used for testing purposes. They were divided into two major groups. In experiment 1, the

**TABLE 4.** Confusion matrix.

| | | Actual values | |
|---|---|---|---|
| | | *Manipulation* | *Normal trading* |
| Predicted values | *Manipulation* | True positives (TP) | False positives (FP) |
| | *Normal trading* | False negatives (FN) | True negatives (TN) |

synthetic manipulation data were used, while in experiment 2, the real manipulation data were used. The synthesized manipulation data contain two subgroups taken from unseen normal groups 1 and 2. The sizes of the unseen normal trading data and the synthesized manipulation data were the same. The real manipulation data were taken from prosecuted stocks during the advertised time period and were all labeled as manipulated. The real manipulation cases account for 19% of all normal trading cases.

## B. RESULTS AND PERFORMANCE EVALUATIONS

### 1) EXPERIMENT 1: PERFORMANCE COMPARISON IN DETECTING SYNTHESIZED MANIPULATION

The main goal of this experiment is to see how effective LSTM-AE and LSTM-GANs are at detecting stock price manipulation. Both models were trained to learn regular trading activities and to recognized anomalies as manipulation. Because they did not use prior knowledge about manipulations in their training, the models were unaffected by variation in manipulation patterns. The performance measure was calculated according to the confusion matrix, using the expression given in Table 4, to summarize the performance of our classification algorithms.

We used the following standard metrics for classification tasks to assess the effectiveness of true class detection. The probability of real synthesized manipulation being correctly predicted as manipulation is measured by Recall (REC). Specificity (SPC) is a metric that measures the proportion of real normal trading that was correctly predicted as normal trading. Precision (PRE) is the proportion of the synthesized manipulation data given by the classifiers that is used to assess the models' differentiation capability. Accuracy (ACC) reflects the total proportion of correctly classified unseen normal trading data and synthesized manipulation data for overall performance. F-Beta Score $F_\beta$ enables the combination of recall and precision into a single value that captures both properties. The specificity value is the most important of these metrics for our problem. Unsupervised learning is used in stock manipulation detection to identify unknown manipulative patterns. We anticipated a very low false-positive rate (high specificity), as otherwise the model would generate too many alerts and become unusable. The model should detect only highly suspicious data. As a result, some of the small manipulated patterns may evade detection, which is acceptable for unsupervised models. Users can use other supervised models in conjunction with our unsupervised models to effectively catch all known patterns.

$$REC = \frac{TP}{TP + FN} \tag{4}$$

$$SPC = \frac{TN}{FP + TN} \tag{5}$$

$$PRE = \frac{TP}{TP + FP} \tag{6}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$F_\beta = \left(1 + \beta^2\right) \times \left(\frac{PRE \times REC}{\beta^2 PRE + REC}\right) \tag{8}$$

To predict a class label in a classification task, a decision threshold must be set. The final layer of the LSTM-AE model does not output the probability of manipulation, but it does output the reconstruction of the input. The reconstruction error can be calculated. If it is set to a high value, the input data is regarded as abnormal or manipulated. We must determine an appropriate threshold value. Because our problem requires high specificity, the LSTM-AE decision threshold was determined by the maximum value of the reconstruction errors from the validation set to achieve the highest specificity value. In the case of LSTM-GANs, the discriminator D can provide the probability of one of two classes representing the real or fake case. A real case is one that involves normal trading; otherwise, it is considered an anomaly. The threshold for the discriminator's output was set to 0.5 by default to differentiate predicted probabilities between 0 and 1.

Table 5 displays the results of Experiment 1. Although the models had never seen the manipulative trading data, the LSTM-AE and LSTM-GANs achieved more than 99% specificity and precision. On the synthesized manipulation data, however, the recall, which describes how well LSTM-GANs captures all of the synthesized manipulation cases, was less than 15%, whereas LSTM-AE had a higher recall rate. Both models have an accuracy of more than 56% in all conditions. Both models have an F-Beta Score of more than 33%. With a beta score of 0.5, we are more concerned with precision, which attempts to reduce false positives. In summary, the results clearly demonstrated that LSTM-AE and LSTM-GANs can detect both groups of previously unseen normal data.

Precision is more important than recall in the context of detecting stock price manipulation because the type I error is more problematic. According to Baader and Helmut [64], the goal of their study was to reduce the number of false positives in fraud detection, which is a case of the purchase-to-pay business process. Analyzing false positives takes time and consequently incurs unnecessary costs. Similarly, Anurag [4] revealed that the vast majority of manipulation detection in options trading are false positives. Analysts are being

**TABLE 5.** Results of experiment 1.

| Measures | LSTM-AE | LSTM-GANs |
|---|---|---|
| Recall | 0.3863 | 0.1519 |
| Specificity | 0.9999 | 0.9999 |
| Precision | 0.9999 | 0.9999 |
| Accuracy | 0.6839 | 0.5640 |
| F-Beta Score | 0.7589 | 0.3289 |

**TABLE 6.** The frequency of warning occurrences to the whole dataset.

| | Proposed models | |
|---|---|---|
| | *LSTM-AE (millionth)* | *LSTM-GANs (millionth)* |
| Unseen normal trading data | 3 | 2 |
| Real manipulation data | 5,769 | 197,513 |

inundated with false alerts due to a high volume of false alerts.

### 2) EXPERIMENT 2: PERFORMANCE COMPARISON IN DETECTING REAL MANIPULATION

In the second experiment, our models were put to the test against six real manipulation cases from the Stock Exchange of Thailand. The same training was used to train both models as in experiment 1. Both groups of the unseen normal data from 16 good governance companies were mixed with these real manipulation cases so that our models could classify them into two classes: normal and manipulated order book data. As an indicator, the ratios of the number of manipulation predictions from the models to the total number of samples on each stock were used. This indicator will be referred to as the frequency of warning occurrences, $F(s)$, later in this paper. Table 6 compares the frequency of warning occurrences for the entire dataset. The frequency of warning occurrences was significantly lower in the unseen normal trading data than in the real manipulation data. Despite the fact that the frequency of warnings on the real manipulation data was low, the discrepancy was sufficient to recognize that there was something wrong with these data and that further investigation was required. Following that, we will look into the results of each stock individually. The results from the models with various stocks as input are shown below. Real manipulation stocks range from RM-1 to RM-6. The list of unseen normal trading stocks ranges from UN-1 to UN-16.

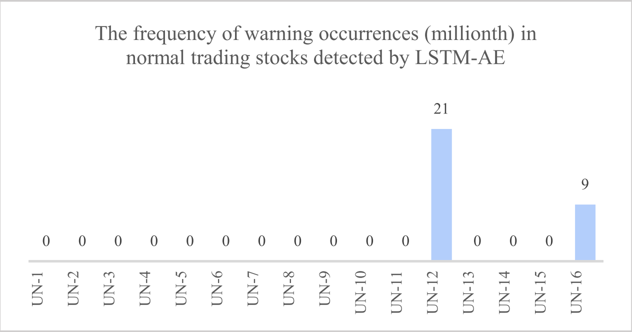


**FIGURE 7.** The frequency of warning occurrences in normal trading stocks detected by LSTM-AE.

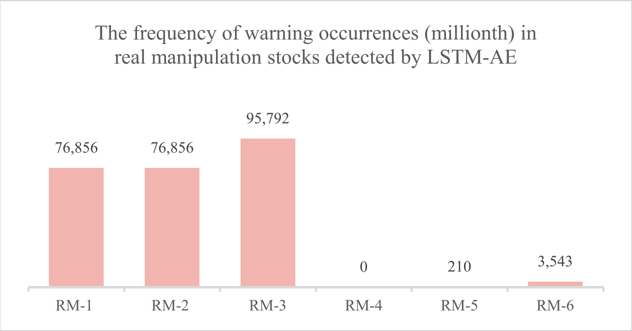


**FIGURE 8.** The frequency of warning occurrences in real manipulation stocks detected by LSTM-AE.

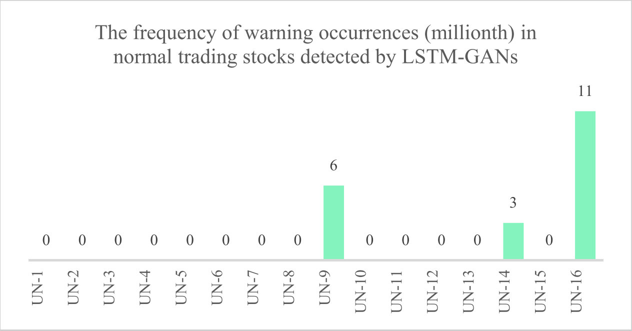


**FIGURE 9.** The frequency of warning occurrences in normal trading stocks detected by LSTM-GANs.

The frequency of warning occurrences detected by the LSTM-AE and LSTM-GANs in both data groups is illustrated in Fig. 7 to 10. For example, on normal stocks, the frequency of warning occurrences detected by LSTM-AE was mostly zero, indicating that the model did not detect an anomaly. Some false positives occurred on UN-4 and UN-8, which had the frequency of warning occurrences of 21 and 9 respectively. Except for RM-4, where the models see no anomalies, the frequency of warnings was noticeably higher for manipulated stocks than the normal stock group.

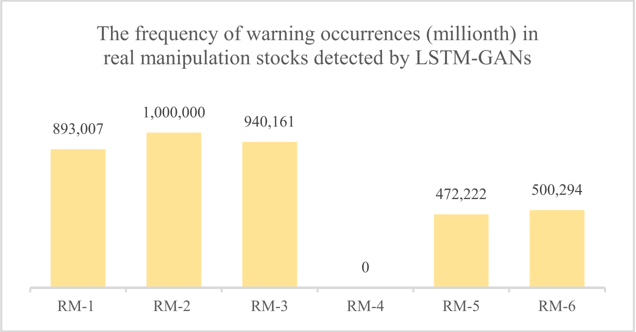


**FIGURE 10. The frequency of warning occurrences in real manipulation stocks detected by LSTM-GANs.**

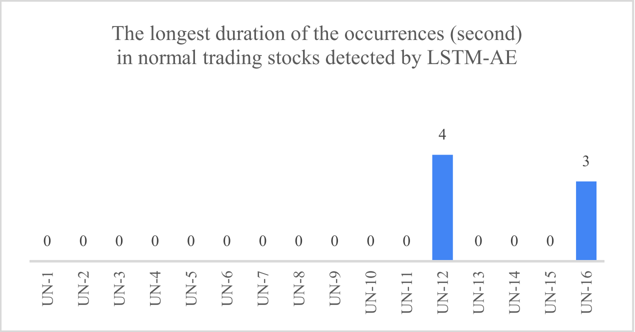


**FIGURE 11. The longest warning duration in normal trading stocks detected by LSTM-AE.**

For manipulated stocks, the outputs from the models were expected to give a mix of the two classes because real manipulations usually occur in on and off patterns (see Arnoldi [65]). We used a statistical analysis to confirm that the outputs of the models from the two groups of stocks differed significantly. Because there were only few stocks in our experiments, the Welch's t-test was selected to analyze the results. The null hypothesis $H_0$ was defined as no difference in the indicators between normal and manipulated stocks. The alternative hypothesis $H_1$ proposed that the frequency of warning occurrences differed between the two groups. The p-values of the two-tailed t-test for the LSTM-AE and LSTM-GANs models were 0.0664 and 0.0026, respectively. We can reject $H_0$ and accept $H_1$. This means that the classification of manipulated stocks, both models achieved a confidence level of 90%, and the LSTM-GANs could achieve a confidence level of up to 99%.

Even though the models performed well, the knowledge that stock manipulation occurs in on and off patterns can be expended upon. We propose using the second indicator, which is the length of time that the model continuously produces anomalies as output. We chose the longest length in each stock to represent it. $T_{max}(s)$, the second indicator, will be referred to as "the longest warning duration." The longest warning duration for various stock groups is depicted in Fig. 11 to 14. The minimum longest warning duration in real manipulation stocks is significantly longer than the maximum longest warning duration in normal trading stocks.

## VI. MINIMUM MANIPULATION-MAXIMUM NORMAL STRATEGY

The public information about manipulations from the SEC of Thailand revealed the periods of manipulative actions in a daily timeframe. They did not specify the exact time of the day when the manipulators carried out their actions. Due to the market's security concerns, these details were not made public. When we used these data in our experiments, we labeled all trading activities during the publicized period as manipulative, even if manipulation occurred only at a specific point on that day. Arnoldi [65] discovered that manipulators typically used algorithmic trading to deceive others in an intermittent pattern. He also investigated three manipulation cases and concluded that they were repeated patterns that occurred several times throughout the trading day. In another stock manipulation study, an algorithmic trader layered the stock of W.W. Grainger (NYSE: "GWW") on NASDAQ and the Boston Stock Exchange [66]. His actions indicated that the layering strategy was used repeatedly on that trading day. These studies revealed that manipulation was typically executed in on and off patterns. Similarly to the results of experiment 2, our models did not recognize manipulation data as manipulating all of the time during the time period announced in the news. For all of these reasons, we cannot

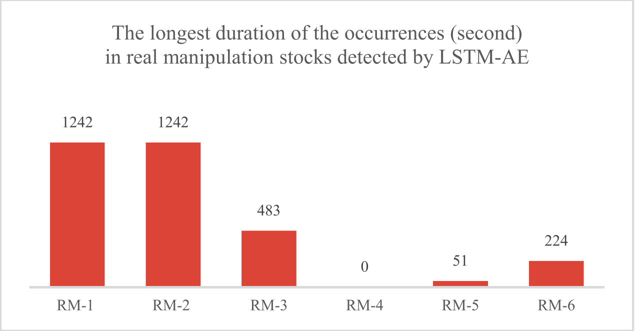


**FIGURE 12. The longest warning duration in real manipulation stocks detected by LSTM-AE.**

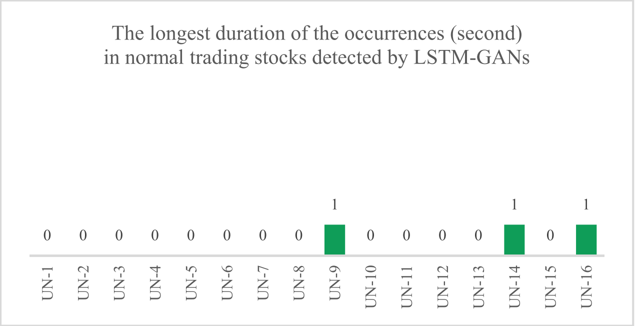


**FIGURE 13. The longest warning duration in normal trading stocks detected by LSTM-GANs.**

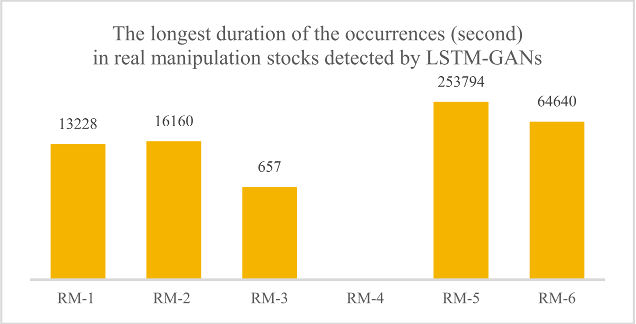


**FIGURE 14.** The longest warning duration in real manipulation stocks detected by LSTM-GANs.

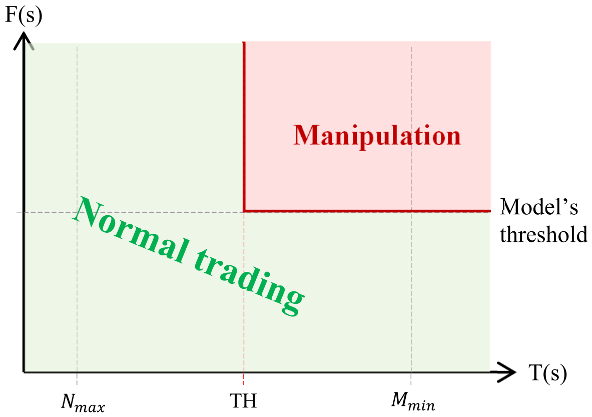


**FIGURE 15.** The MinManiMax strategy.

rely solely on the label, because the manipulation data contained both regular and manipulative patterns. We must devise a strategy for making the best use of the models. The previous section reported the frequency of warning occurrences $F(s)$ and the longest warning duration $T(s)$ on each stock were reported in the previous section. We proposed using them as a strategy for distinguishing between normal and manipulated stocks. The minimum value of $T(s)$ in the manipulated stocks and the maximum value of $T(s)$ in the normal stocks differed significantly. $M_{min}$ is defined as the minimum value among the longest duration in the real manipulation stocks. $N_{max}$ is defined as the maximum value among normal stocks with the longest duration. To optimize the decision boundary, the threshold $TH$ was set in the middle of both groups, as shown in Fig. 15. Two thresholds define the red area (manipulation). The threshold for $F(s)$ depended on each model.

$$M_{\min} = \min_{S \in RM} T_{\max}(S) \tag{9}$$

$$N_{\max} = \max_{S \in UN} T_{\max}(S) \tag{10}$$

$$TH = \frac{M_{\min} + N_{\max}}{2} \tag{11}$$

The threshold value for the LSTM-AE is 27.5 seconds, calculated from 51 and 4 seconds in RM-5 and UN-12, respectively. Except for RM-4, the model could detect all manipulation cases with this value. The threshold value for the LSTM-GANs is 329 seconds, calculated from 657 and 1 seconds in RM-3 and UN-9-14-16, respectively. With this value, the model would be able to detect all manipulation activities (except those in RM-S4) while avoiding false positives from unseen normal stocks. This strategy is named as Minimum Manipulation-Maximum Normal, abbreviated as "*MinManiMax*". It can be used in unsupervised learning for practical manipulation detection.

**TABLE 7.** MinManiMax parameters.

| | $N_{max}$ (sec) | $M_{min}$ (sec) | $TH$ (sec) |
|---|---|---|---|
| *LSTM-AE* | 4 | 51 | 27.5 |
| *LSTM-GANs* | 1 | 657 | 329 |

## VII. CONCLUSION

For detecting stock price manipulation, we proposed an unsupervised deep learning approach. The performance of two deep unsupervised models: LSTM-AE and LSTM-GANs was compared in this study. Both models were trained to recognize normal trading behaviors and treat other behaviors as manipulated. Our dataset was divided into three categories: normal trading data, real manipulation data, and synthesized manipulation data. Normal trading data were used for training and testing of the models, while real and synthetic manipulation data were only used for performance evaluation. For the synthesized manipulation data, the results of the study indicated that both models were effective in detecting stock price manipulation with low false positives, despite the fact that they had no prior knowledge about stock manipulations. Both models can identify manipulations in five out of six cases in the real manipulation data. For practical manipulation detection, we proposed a strategy called *MinManiMax*, which optimizes the decision boundary between the two classes. Our method's strength is its ability to detect new types of manipulation by using only normal trading data as training samples, which is widely available. However, if new incoming trading data show significant deviations from the norm, it is a good practice to retrain the models and re-adjust their threshold values to find the best decision boundaries.

## REFERENCES

[1] *Market Manipulation*. US Securities and Exchange Commission. Accessed: Jan. 15, 2021. [Online]. Available: https://www.investor.gov/introduction-investing/investing-basics/glossary/market-manipulation

[2] M. J. Aitken, A. Aspris, S. Foley, and F. H. de B. Harris, "Market fairness: The poor country cousin of market efficiency," *J. Bus. Ethics*, vol. 147, no. 1, pp. 5–23, Jan. 2018.

[3] (2019). *SEC News*. The Securities and Exchange Commission. Thailand. [Online]. Available: https://www.sec.or.th/EN/ Pages/NEWS_LISTVIEW.aspx

[4] M. Anurag. (May 4, 2020). *Winning the Compliance Battle on Multiple Fronts: Tackling Market Abuse and False Positives*. NICE. [Online]. Available: https://www.niceactimize.com/compliance /blog-winning-the-compliance-battle-on-multiple-fronts-tackling-market-abuse-and-false-positives.html

[5] V. Burkhard. (Feb. 14, 2019). *Fast, Faster. High-frequency Traders*. Credit Suisse. [Online]. Available: https://www.credit-suisse.com/ch/en/articles/private-banking/schnell-schneller-hochfrequenzhaendler-201902.html

[6] T. Mizuta, "Can an AI perform market manipulation at its own discretion?—A genetic algorithm learns in an artificial market simulation," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Canberra, ACT, Australia, Dec. 2020, pp. 407–412.

[7] M. B. Fox, L. R. Glosten, and G. V. Rauterberg, "The new stock market: Sense and nonsense," *Duke LJ*, vol. 65, no. 2, p. 191, 2015.

[8] F. Allen and D. Gale, "Stock-price manipulation," *Rev. Financial Stud.*, vol. 5, no. 3, pp. 503–529, Jul. 1992.

[9] M. Bagnoli and B. L. Lipman, "Stock price manipulation through takeover bids," *RAND J. Econ.*, vol. 9, no. 9, pp. 124–147, Apr. 1996.

[10] M. J. Aitken, F. H. Harris, and S. Ji, "Trade-based manipulation and market efficiency: A cross-market comparison," in *Proc. 22nd Australas. Finance Banking Conf.*, 2009, p. 18.

[11] T. C. W. Lin, "The new market manipulation," *Emory LJ*, vol. 66, no. 6, pp. 1253–1314, 2016.

[12] A. S. Kyle and S. Viswanathan, "How to define illegal price manipulation," *Amer. Econ. Rev.*, vol. 98, no. 2, pp. 79–274, 2008.

[13] R. A. Jarrow, "Market manipulation, bubbles, corners, and short squeezes," *J. Financial Quant. Anal.*, vol. 27, no. 3, pp. 311–336, 1992.

[14] R. K. Aggarwal and G. Wu, "Stock market manipulation-theory and evidence," in *Proc. AFA San Diego Meetings*, 2003, pp. 1–49.

[15] G. K. Palshikar and A. Bahulkar, "Fuzzy temporal patterns for analysing stock market databases," in *Proc. Int. Conf. Adv. Data Manage. (COMAD)*, 2000, pp. 135–142.

[16] Y. C. Huang and Y. J. Cheng, "Stock manipulation and its effects: Pump and dump versus stabilization," *Rev. Quant. Finance Accounting*, vol. 44, no. 4, pp. 791–815, May 2015.

[17] A. Khwaja and A. Mian, "Unchecked intermediaries: Price manipulation in an emerging stock market," *J. Financial Econ.*, vol. 78, no. 1, pp. 203–241, Oct. 2005.

[18] A. Kaul, V. Mehrotra, and R. Morck, "Demand curves for stocks do slope down: New evidence from an index weights adjustment," *J. Finance*, vol. 55, no. 2, pp. 893–912, 2000.

[19] S. Q. Shah, I. Ismail, and A. R. B. Shahrin, "Stock market manipulation: A comparative analysis of east Asian emerging and developed financial markets," *Manage. Sci. Lett.*, vol. 9, no. 1, pp. 183–192, 2019.

[20] F. Allen and G. Gorton, "Stock price manipulation, market microstructure and asymmetric information," *Eur. Econ. Rev.*, vol. 36, nos. 2–3, pp. 624–630, Apr. 1992.

[21] T. J. Putniņš, "Market manipulation: A survey," *J. Econ. Surveys*, vol. 26, no. 5, pp. 952–967, Dec. 2012.

[22] J. W. Moore and F. M. Wiseman, "Market manipulation and the exchange Act," *Univ. Chicago Law Rev.*, vol. 2, no. 1, p. 46, 1934.

[23] S. Patterson, *Dark Pools: High-Speed Traders, AI Bandits, and the Threat to the Global Financial System*, 1st ed. New York, NY, USA: Crown Business, 2012.

[24] J. L. Teall, *Financial Trading and Investing*. New York, NY, USA: Academic, 2018.

[25] F. Fabozzi, S. M. Focardi, and C. Jonas, "High-frequency trading: Methodologies and market impact," *Rev. Futures Markets*, vol. 9, pp. 7–38, 2011.

[26] N. Johnson, G. Zhao, E. Hunsader, H. Qi, N. Johnson, J. Meng, and B. Tivnan, "Abrupt rise of new machine ecology beyond human response time," *Sci. Rep.*, vol. 3, no. 1, p. 2627, Dec. 2013.

[27] S. K. Mehra, "Antitrust and the robo-seller: Competition in the time of algorithms," *Minnesota Law Rev. 204*, 2016. [Online]. Available: https://scholarship.law.umn.edu/mlr/204

[28] *Findings Regarding the Market Events*, Rep. Staffs CFTC SEC Joint Advisory Committee Emerg. Regulatory Issues, S. CFTC and U. SEC, Washington, DC, USA, May 2010, p. 104.

[29] J. W. Markham, *Law Enforcement and the History of Financial Market Manipulation*. Armonk, NY, USA: ME Sharpe, 2013.

[30] A. Verstein, "Benchmark manipulation," *BCL Rev.*, vol. 56, no. 1, p. 215, 2015.

[31] T. E. Levens, "Too fast, too frequent: High-frequency trading and securities class actions," *Univ. Chicago Law Rev.*, vol. 82, no. 3, p. 1511, 2015.

[32] Chartis Research Ltd. (Oct. 2017). *The Future of Trader Surveillance*. [Online]. Available: https://assets.ey.com/content/dam/ey-sites/ey-com/en_gl/topics/emeia-financial-services/ey-trader-surveillance-report.pdf

[33] *Exactpro Test and Automation Approaches: A Case Study in Market Surveillance*. Exactpro. Accessed: Jun. 5, 2020. [Online]. Available: https://exactpro.com/case-study/market-surveillance

[34] J. Shi, S. Zhou, X. Liu, Q. Zhang, M. Lu, and T. Wang, "Stacked deep polynomial network based representation learning for tumor classification with small ultrasound image dataset," *Neurocomputing*, vol. 194, pp. 87–94, Jun. 2016.

[35] C. Luo, Y. Zhao, L. Cao, Y. Ou, and L. Liu, "Outlier mining on multiple time series data in stock market," in *Proc. Pacific Rim Int. Conf. Artif. Intell.*, 2008, pp. 1010–1015.

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

[37] F. R. Roodposhti, M. F. Shams, and H. Kordlouie, "Forecasting stock price manipulation in capital market," *Int. J. Econ. Manage. Eng.*, vol. 5, no. 8, pp. 957–967, 2011.

[38] K. Golmohammadi, O. R. Zaiane, and D. Diaz, "Detecting stock market manipulation using supervised learning algorithms," in *Proc. Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2014, pp. 435–441.

[39] J. Zhang, S. Wang, S. Xu, and M. Yu, "Stock price manipulation detection based on machine learning technology: Evidence in China," in *Proc. Int. Conf. Geo-Inform. Resour. Manage. Sustain. Ecosyst.*, 2016, pp. 150–158.

[40] A. Li, J. Wu, and Z. Liu, "Market manipulation detection based on classification methods," *Procedia Comput. Sci.*, vol. 122, pp. 788–795, Jan. 2017.

[41] J. Zhai, Y. Cao, and X. Ding, "Data analytic approach for manipulation detection in stock market," *Rev. Quant. Finance Accounting*, vol. 50, no. 3, pp. 897–932, Apr. 2018.

[42] A. Zaborovskaya, V. Zaborovskiy, and K. Pletnev, "Possibilities of preventing manipulative transactions on the stock market in the conditions of new industrialization," in *Proc. 2nd Int. Sci. Conf. Industrialization, Global, Nat., Regional Dimension (SICNI)*, 2019, pp. 154–160.

[43] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, "Stock market prediction based on generative adversarial network," *Procedia Comput. Sci.*, vol. 147, pp. 400–406, Jan. 2019.

[44] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180944.

[45] Y. Kim and S. Y. Sohn, "Stock fraud detection using peer group analysis," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 8986–8992, Aug. 2012.

[46] K. Golmohammadi and O. R. Zaiane, "Time series contextual anomaly detection for detecting market manipulation in stock market," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2015, pp. 1–10.

[47] H. Al-Thani, H. Hassen, S. Al-Maadced, N. Fetais, and A. Jaoua, "Unsupervised technique for anomaly detection in Qatar stock market," in *Proc. Int. Conf. Comput. Appl. (ICCA)*, Aug. 2018, pp. 116–122.

[48] B. Abbas, A. Belatreche, and A. Bouridane, "Stock price manipulation detection using empirical mode decomposition based Kernel density estimation clustering method," in *Proc. SAI Intell. Syst. Conf.*, 2018, pp. 851–866.

[49] B. Rizvi, A. Belatreche, and A. Bouridane, "Immune inspired dendritic cell algorithm for stock price manipulation detection," in *Proc. SAI Intell. Syst. Conf.*, 2019, pp. 352–361.

[50] 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.

[51] T. Leangarun, P. Tangamchit, and S. Thajchayapong, "Stock price manipulation detection using generative adversarial networks," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 2104–2111.

[52] B. Rizvi, A. Belatreche, A. Bouridane, and K. Mistry, "Stock price manipulation detection based on autoencoder learning of stock trades affinity," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[54] *Corporate Governance Report of Thai Listed Companies 2014*. Accessed: Jun. 20, 2019. [Online]. Available: http://www.thai-iod.com/imgUpload/CGR%202014%20Report(3).pdf

[55] *The Stock Exchange of Thailand*. Accessed: Jan. 10, 2020. [Online]. Available: https://www.set.or.th/en/products /index/setindex_p2.html

[56] *Equities Trading*. The Stock Exchange of Thailand. [Online]. Available: https://www.set.or.th/en/products/trading/equity/ tradingsystem_p5.html

[57] L. Cao, Y. Ou, and S. Y. Philip, "Coupled behavior analysis with applications," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 8, pp. 1378–1392, Jun. 2011.

[58] D. Ian, "Market abuse and surveillance," Foresight, Government Office Sci., London, U.K., Econ. Impact Assessment EIA17, 2012.

[59] F. Abergel, C. Huré, and H. Pham, "Algorithmic trading in a microstructural limit order book model," *Quant. Finance*, vol. 20, no. 8, pp. 1–21, 2020.

[60] Q. Q. Yang, W. K. Ching, J. Gu, and T. K. Siu, "Trading strategy with stochastic volatility in a limit order book market," *Decis. Econ. Finance*, vol. 43, no. 1, pp. 1–25, 2020.

[61] J. D. Montgomery. (May 3, 2016). *Spoofing, Market Manipulation, and the Limit-Order Book*. Ankura Consulting Group. Accessed: Apr. 5, 2019. [Online]. Available: https://ankura.com/insights/spoofing-market-manipulation-and-the-limit-order-book/

[62] C.-H. Chan and A. K. C. Ma, "Order-based manipulation: Evidence from Hong Kong stock market," *J. Financial Crime*, vol. 21, no. 1, pp. 111–118, Dec. 2014.

[63] D. Moldovan, M. Mircea, and S. Nitchi, "A stock trading algorithm model proposal, based on technical indicators signals," *Inf. Economica*, vol. 15, no. 1, pp. 183–188, 2011.

[64] G. Baader and H. Krcmar, "Reducing false positives in fraud detection: Combining the red flag approach with process mining," *Int. J. Accounting Inf. Syst.*, vol. 31, pp. 1–16, Dec. 2018.

[65] J. Arnoldi, "Computer algorithms, market manipulation and the institutionalization of high frequency trading," *Theory, Culture Soc.*, vol. 33, no. 1, pp. 29–52, Jan. 2016.

[66] (Sep. 25, 2012). *United States of America Before the Securities and Exchange Commission*. [Online]. Available: https://www.sec.gov/litigation/admin/2012/34-67924.pdf

**TEEMA LEANGARUN** (Graduate Student Member, IEEE) received the B.Eng. degree in control systems and instrumentation engineering and the M.Eng. degree in electrical engineering from the King Mongkut's University of Technology Thonburi, Thailand, in 2014 and 2016, respectively, where she is currently pursuing the Ph.D. degree in computer and electrical engineering. Her research interest includes using artificial intelligence for anomaly detection of irregular behaviors in stock markets.

**POJ TANGAMCHIT** (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, USA, in 2003. He is currently an Associate Professor with the Department of Control Systems and Instrumentation Engineering, King Mongkut's University of Technology Thonburi, Bangkok, Thailand. His current research interest includes modern AI techniques for financial intelligence. He also serves as a Technical Committee Member for the IEEE Computational Finance and Economics Society.

**SUTTIPONG THAJCHAYAPONG** (Member, IEEE) received the B.S. and M.S. degrees in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, and the Ph.D. degree in electrical and electronic engineering from Imperial College London, London, U.K. He is currently a Senior Researcher with the National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSDTA), Pathum Thani, Thailand. His research interests include intelligent transportation systems, data analytics, anomaly detection, signal processing, and machine learning. He served as the Project Manager of the Thai People Map and Analytics Platform (TPMAP), Thailand's Data-Driven Target Poverty Alleviation Project, National Economic and Social Development Council. He served for the Senate of Thailand, as a Sub-Commissioner on the National Strategy and Country Reform Analysis and Monitoring; the Thai Government, as an Assistant Secretary of the National Big Data Steering Committee; and the Geo-Informatics and Space Technology Development Agency, as a member of the Actionable Intelligence Policy Working Group.