

Received July 2, 2018, accepted September 3, 2018, date of publication September 6, 2018, date of current version October 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2868970

# Deep Learning Approach for Short-Term Stock Trends Prediction Based on Two-Stream Gated Recurrent Unit Network

**DANG LIEN MINH<sup>1</sup>, ABOLGHASEM SADEGHI-NIARAKI<sup>1</sup>, HUYNH DUC HUY<sup>2</sup>, KYUNGBOK MIN<sup>1</sup>, AND HYEONJOON MOON<sup>ID1</sup>**

<sup>1</sup>Department of Computer Science and Engineering, Sejong University, Seoul 143-747(05006), South Korea

<sup>2</sup>Department of Information System, University of Information Technology, Ho Chi Minh City 700000, Vietnam

Corresponding author: Hyeonjoon Moon (hmoon@sejong.ac.kr)

This work was supported by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture, Forestry and Fisheries (IPEF) through the Agri-Bio industry Technology Development Program funded by the Ministry of Agriculture, Food and Rural Affairs (MAFRA) under Grant 316033-4.

**ABSTRACT** Financial news has been proven to be a crucial factor which causes fluctuations in stock prices. However, previous studies heavily relied on analyzing shallow features and ignored the structural relation among words in a sentence. Several sentiment analysis studies have tried to point out the relationship between investors' reaction and news events. However, the sentiment dataset was usually constructed from the lingual dataset which is unrelated to the financial sector and led to poor performance. This paper proposes a novel framework to predict the directions of stock prices by using both financial news and sentiment dictionary. The original contributions of this paper include the proposal of a novel two-stream gated recurrent unit network and Stock2Vec—a sentiment word embedding trained on financial news dataset and Harvard IV-4. Two main experiments are conducted: the first experiment predicts S&P 500 index stock price directions using the historical S&P 500 prices and the articles crawled from Reuters and Bloomberg, and the second experiment forecasts the price trends of VN-index using VietStock news and stock prices from cophieu68. Results show that: 1) two-stream GRU outperforms state-of-the-art models; 2) Stock2Vec is more efficient in dealing with financial datasets; and 3) applying the model, a simulation scenario proves that our model is effective for the stock sector.

**INDEX TERMS** Deep learning, natural language processing, stock trends, sentiment analysis.

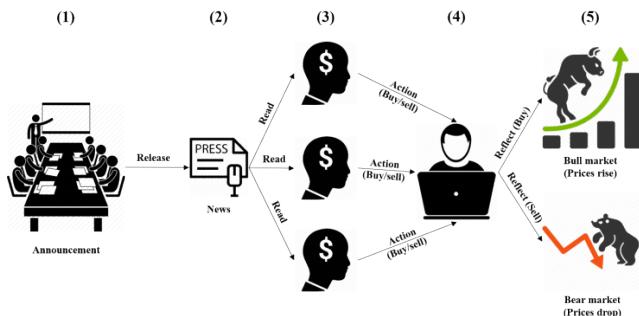
## I. INTRODUCTION

The stock market is one of the most important components forming a country's economy. Through IPO – Initial Public Offering, a company is able to raise a substantial amount of money to expand businesses. It is a great opportunity for investors to buy a brand-new stock and become either a stockholder who gets extra benefit from dividends from the firm's shareholder bonus program or a trader who trades stock in the stock market. If the stock trader predicted the stock price trends correctly, he would gain enormous profits. However, the stock market is volatile [1], daily news events such as developing political situations, the company's performance and other unexpected events affect stock prices immediately in a positive or negative way. As the result, it is impossible to predict the stock prices and their directions (increase, decrease) accurately, instead investor is only to forecast the upcoming short-term

trends. He usually evaluates a company's performance before making the decision to buy stock. The evaluation includes analyzing a company quarterly earnings report and paying attention to the important news to avoid buying overrated or high-risk stocks. However, both the speed of release and the number of daily news outlets have skyrocketed over the last few years which overwhelm investor's ability to thoroughly assess such a huge volume of data. As the result, an automated decision support system is essential as it automatically evaluates and shows the prediction for the upcoming stock trends. For example, if the price of the potential stock was predicted to be "going up" tomorrow, investors could either sell the stocks they held at a higher price or wait for the price drops and buy more. Thus, which algorithm is more effective and how to analyze the financial news to increase profits have drawn much interest from the research community.

Previous studies on this topic were divided into two main approaches: the technical analysis and the fundamental analysis. In the technical analysis, mathematics has been widely used to analyze historical stock price patterns and predict stock prices in the near future. Researchers have applied many algorithms such as multiple kernel learning [3], deep learning [5], [18], [17], stepwise regression analysis [4], etc. Although they achieved good results, it is impossible to predict the stock prices accurately by using only the historical prices because unexpected events can affect the stock prices immediately. On the other hand, the fundamental analysis [2], [5], [6] used natural language processing (NLP) to analyze financial news and financial statements from the company and predict the stock trends in the future (uptrend, downtrend).

In NLP, the bag-of-words technique is commonly applied to extract features from news articles, it measures the occurrence of each word and then converts text information into vector spaces using these occurrences. After that, machine learning algorithms are implemented to learn the connection between word patterns and stock prices movements. Although bag-of-words based approaches have achieved high accuracy in [2] and [5], they ignored one crucial element of the directional predictions, which is the sentiment of the article. As shown in Fig. 1, one important stage is that the published news articles are first interpreted by investors and converted into sentiments (positive, negative); the investors then decide whether to sell/hold/buy stocks based on the sentiment interpretations; finally, market prices aggregate the actions of each investor and reflect them in the final price trends. Therefore, combining the sentiment analysis and natural language processing would become more effective.



**FIGURE 1.** A general plot illustrates how news impacts the stock price movements. (1) Events occur; (2) News are published; (3) Investors read the news; (4) Investors take actions (buy/sell/hold) based on their interpretations; and (5) The actions affect the stock prices. Idea from [17].

Sentiment analysis assesses the document from sentiment aspect by measuring the frequency of words, and each word is analyzed and depicted by a sentiment vector. For instance, “increase” can be described as strong and active by looking up in Harvard IV-4 psychological dictionary; on the other case, the word “believe” indicates commitment, social relation, positive features. In a sentiment dictionary, the dimension of the sentiment feature is fixed. Each news article has a specific sentiment vector value; it is calculated by summarizing each word sentiment vectors. Integrating sentiment

analysis and bag-of-words brings many benefits: (1) Dimension reduction: the bag-of-words approach uses words as features so the larger the dataset the bigger the dimension (tens of thousands of words). On the contrary, the sentiment representation reduces the dimensions to hundreds, e.g. Harvard IV-4 psychological dictionary has 182 dimensions and Loughran–McDonald financial sentiment dictionary has 6 dimensions; (2) Better explanation. It is difficult to explain the mapping generated by bag-of-words. On the other hand, the sentiment values in fixed dimensions give us a more straightforward view of the document.

Finally, machine learning algorithms are implemented to learn the relationship between the extracted features from news and stocks trends. In recent years, deep neural networks (DNNs) – a branch of machine learning has achieved numerous successes in various domains such as speech recognition [15], computer vision [16]. Much of the hype surrounding neural networks is about image-based applications. However, a different approach from the image-based application is Recurrent Neural Networks (RNNs), which have been successfully used in recent years to predict future events in time series as well. RNNs have contributed to the development in a wide variety of fields centered around predicting sequences of events. Because of its efficiency on large-scale datasets [17], researchers have already applied some DNN models on features extracted from news articles and historical stock prices such as [18] and [19].

Based on the above analyses and evaluations, we propose a novel approach to predict daily stock price directions by analyzing financial news articles and historical stock prices using the deep learning approach. The main contributions include four aspects:

- Proposing a two-stream gated recurrent unit (TGRU) for stock price trends prediction.
- Proposing a sentiment Stock2Vec embedding model trained on both the stock news and the sentiment dictionary
- Applying three technical indicators as an additional feature set for stock prices trend prediction and thoroughly analyzing whether the news impact stock prices immediately or after a short period (2 days, one week).

The rest of the paper is organized as follows. Section 2 reviews related work in stock price movements prediction, sentiment analysis, and recurrent neural network. In Section 3, we show the proposed model and explain each module in detailed. In Section 4, we explain the implementation detailed and also show experimental results. In Section 5, we give the conclusion and what should be done in the future.

## II. RELATED WORK

### A. SENTIMENT ANALYSIS

#### 1) SENTIMENT ANALYSIS FOR FINANCIAL SECTOR

Sentiment analysis which studies the effect of news articles on stock price movements has been investigated thoroughly in the financial sector. Zhang *et al.* [8] inspected NetEase - one

of the most famous internet content providers in China and tried to prove that Internet news will be reflected in future market price movements. The results showed a remarkably unusual return and excessive trading volume on the date the event occurs which proved their hypothesis. Baruch *et al.* [9] analyzed the effects of positive and negative words news on firms' future performance. They had two conclusions: (1) There is a relationship between the readers' expectation and the writers' intention; (2) readers strongly against the content of the reports which violate their presumption.

## 2) SENTIMENT ANALYSIS FOR COMPUTER SCIENCE SECTOR

In computer science, sentiment analysis refers to word's sentiment level evaluation by checking word sentiment value in the sentiment dictionary. As the results, sentiment dictionaries solely decide the effectiveness of sentiment analysis. A document can be represented as a sentiment vector by stacking up the word sentiment vectors in the document. Integrating sentiment analysis in the model will bring: (1) Reliability: adding sentiment value makes the model more reliable since it takes the emotional aspect into consideration (2) Dimensionality reduction: the sentiment dictionary has a fixed dimension, features extracted from it will effectively reduce the dimension of the original vectors. The construction of the sentiment dictionary can be divided into three categories automatic, semi-automatic and manual as described below:

- Automatic: Initially, a dataset is crawled from the internet, all of the documents from that dataset are categorized into positive or negative by analyzing the positive or negative effect on the market (stock, gold, ...). Finally, the sentiment dictionary is constructed by applying NLP on the dataset.
- The dictionary is initially constructed by manually selecting some seed words. It is then expanded by following a set of rules on a new dataset.
- Manual: The dictionary is collected and analyzed by linguistic experts. It contains fewer words than the one constructed by semi-automatically, but much more accurate.

Dang *et al.* [2] generated a sentiment dictionary by crawling news from the internet. After that, the author classified each news into positive or negative based on the effect on stock prices, then the sentiment dictionary is generated by weighting the sentiment score (TF-IDF method) and selecting 500 negative and 500 positive words which had the highest sentiment score. Experiment results showed that their system achieved high accuracy (up to 73%) in the stock trends prediction. Zhou *et al.* [10] also constructed a Chinese short text sentiment dictionary using Word2Vec algorithm and some initial emotional seed words, the experimental results proved that the dictionary effectively improved the emotional classification of short texts. On the other hand, Zhang *et al.* [11] made three hypotheses: (1) the sentiment dictionary can be expanded by constructing negative word, adverb, network word, and other related dictionaries; (2) the sentiment value

of a document can be acquired by computing the weight and (3) the text document on a specific topic is categorized into positive, negative or neutral. Song *et al.* [12] created a special module namely fine-grained named entity recognizer (NER) which mainly relied on named-entity (NE) dictionary. They constructed NE dictionary by following a semi-automatic approach. The proposed framework automatically generated a pseudo-document for each NE class from Wikipedia. It then computed the similarities between the Wikipedia entries and pseudo-documents by using a vector-space model. Finally, it categorized a new Wikipedia entry into NE classes by referring to the computed similarities. Li *et al.* [13] also proposed a sentiment analysis system containing two main sections, domain new words detection, and word propagation. The first section made it possible to detect user-invented words, proper nouns, converted words and the second section allowed the system to detect multiword expressions in the tourism domain. Experimental results showed that their proposed model significantly outperformed traditional sentiment lexicons. Loughran and McDonald [14] provided a manually made financial sentiment dictionary which contains 6 sentiment dimensions. We summarize the papers reviewed in Table 1.

**TABLE 1. Sentiment dictionary construction summary.**

Paper	Construction type	Dimensions
Jianfeng Zhou et al. [10]	Automatic	2
ShunxiangZhang et al. [11]	Automatic	3
Minh Dang et al. [2]	Automatic	2
Song et al. [12]	Semi-automatic	29
Li, Wei et al. [13]	Semi-automatic	2
T. Loughran et al. [14]	Manual	6

Recently a concept-level approach has become an emerging trend in the sentiment analysis. There are two common feature learning techniques: GloVe and Word2Vec. Both models learn geometrical encodings (vectors) of words from their co-occurrence information. Word2vec [31] learns from the vectors to improve the predictive ability of loss, i.e., the loss of predicting the target words from the context words given the vector representations. On the other hand, GloVe count-based technique [32] learns its vectors by essentially making a dimensionality reduction on the co-occurrence counts matrix. Then it factorizes this matrix to yield a lower-dimensional matrix, where each row contains a vector representation for each word. In general, this is done by minimizing a "reconstruction loss" which tries to find the lower-dimensional representations. Although Pennington *et al.* [32] said that GloVe and Word2Vec were equally effective for natural language processing problems, they originated from datasets irrelevant to the financial field which affects the overall performance when they are used for financial applications.

## 3) APPLICATIONS OF SENTIMENT ANALYSIS

Since sentiment analysis brings enormous benefit to text mining, computer science researchers have tried to apply the sentiment analysis in real applications. Gelbard *et al.* [20]

built a conceptual ontology to evaluate human factors by using digital sources and the sentiment analysis available in any organization's information systems. Peng *et al.* [21] developed a system which can provide invaluable added value to the government, business owners and Internet users by analyzing the sentiment element from countless amounts of new information entries each day through product descriptions, customer reviews. Zhang *et al.* [11] proposed an automated system which can classify Chinese micro-blog topic based on sentiment analysis, it is helpful for the public opinion supervisors to make appropriate decisions. Song *et al.* [12] built a system that analyzed users' opinions about specific entities in text messages. As the result, it facilitated the design of human-centric services by social media. Li *et al.* [13] proposed a system which can be applied to hot scenic spots analysis and precision marketing by applying sentiment analysis on customer's reviews from Qunar.com and Ctrip.com.

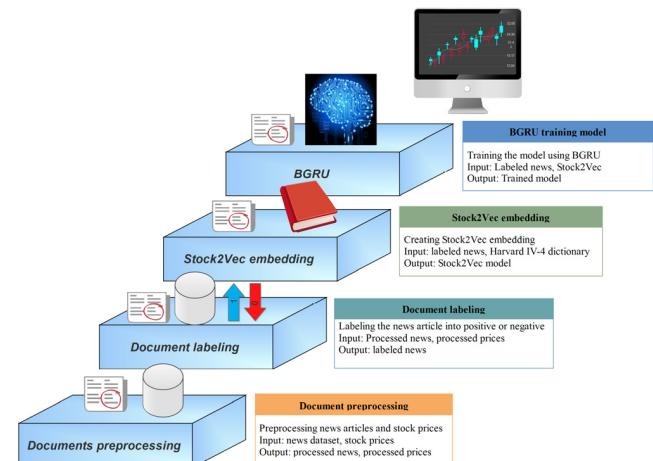
### B. DEEP LEARNING

After extracting the features, machine learning algorithms such as in [33] and [34] are implemented to analyze the relationship between financial news articles and historical stock prices. These works proved that it is possible to effectively forecast the stock prices trend. However, they did not explore the structural relation of the sentence. For example, a news story titled "Apple has finally won \$120 million from Samsung patent battle", if the system only analyzes individual terms "Samsung", "won", "Apple", it cannot differentiate between the winner and the loser thus it is unable to anticipate that the stock price of apple will probably go up in the near future [22]. The previous problem is the reason why Recurrent Neural Network - another class of artificial neural network proved to be more efficient than CNN in dealing with text and speech analysis [24]. A recurrent neural network can be imagined as multiple copies of the same network; each copy takes a sequence of vectors ( $x_1, x_2, \dots, x_n$ ) as input and passes another sequence ( $h_1, h_2, \dots, h_n$ ) that represent a part of the sequence to its successor. Although RNNs have been very effective in text generation and speech recognition tasks [15], Hochreiter [25] revealed a big problem of RNNs; when the gradient is passed back through many time steps, it tends to grow or vanish, they stated that it was very hard to make RNN remember the long-term dependencies. GRU (Gated Recurrent Unit) [23] and LSTM (Long Short-Term Memory) were proposed as the solution for the problem as they have the ability to keep memory/state from previous activations rather than replacing the entire activation like RNN. The GRU unit controls the flow of information like the LSTM unit, but without having to use a memory unit because it just exposes the full hidden content without any control. Moreover, GRU was computationally more efficient than LSTM as pointed out in [23]. GRU successfully solved the drawback of RNN. however, it can be further improved by increasing the amount of input information available to the network. Inspired by the model proposed by Schuster *et al.* [28], which connected two opposite directions hidden layers in the orig-

inal RNN network to the same output to be able to get information from not only past states but also future states. we apply this structure to form a two-stream GRU which will be effective for the text analysis.

### III. METHODOLOGY

As depicted in Fig. 2, the proposed method includes four sections. 1) In the document preprocessing step, the extracted articles are preprocessed to remove useless information such as stop words, punctuation. 2) The next step includes two tasks: labeling news articles based on stock prices. 3) In Stock2Vec embedding section, we create a new sentiment word embedding model based on financial dataset and sentiment dictionary. 4) In the final step, TGRU network is implemented on the financial news dataset and then multiple scenarios are conducted.



**FIGURE 2. Proposed system for stock trends prediction.**

### A. DOCUMENT PREPROCESSING

#### 1) NEWS ARTICLES

Because the news articles were crawled from the internet and in HTML format, all unnecessary tags were removed. After that, the articles were exported in plain text. Next, each sentence was tokenized into a sequence of tokens; each token represented a single word. Finally, the following steps were implemented:

- Stop word removal: all stop words such as "and", "or", "by", etc., were removed based on the English stop words dictionary proposed by Porter and Martin [26].
- All punctuation and numbers were also eliminated because they are unnecessary and occur frequently which affect the role of other important words.
- A Porter Stemmer [26] was also used to reduce word inflectional form to a common base form.

#### 2) DAILY STOCK PRICES

Raw daily S&P 500 Index stock prices downloaded from Yahoo Finance was in CSV format, each row contained six

pieces of information including date (trading date), open (opening price), close (closing price), high (highest price), low (lowest price), adj. close (adjusted closing), and volume. Two steps were conducted:

- Removing unnecessary information: “adj. closes” is unnecessary so it was removed to save system resources.
- Importing data into the database: for efficient query speed and data manipulation, we imported all historical stock prices into MySQL; the database contains a total of 2,519 rows.

### B. DOCUMENT LABELING

The goal of labeling is to classify each article into either positive, which drives the price up, or negative, which leads to a downward trend of the price. In this study, we choose two classes approach (positive or negative) to reflect the direction of the stock prices [2].

There are two techniques in labeling the news articles using historical stock prices: an open-to-close return (daytime return) and a close-to-close return (overnight return). The open-to-close price return approach was used to determine the label of the document. The reason why we chose open-to-close return instead of close-to-close return originated from the practical viewpoint pointed out in [29], the records of the open-to-close are more similar to the total of the same stock, suggesting that the open-to-close return contributes more to the total return.

We investigated the model at different time periods (i.e., 1 day, 2 days, 5 days, 7 days, and 10 days) to examine the impact of time period on stock prices. The Open-to-Close price return  $R_{dt}$  of day d and time period t ( $t \in [1,2,5,7,10]$ ) is calculated as follows:

$$R_{dt} = O_{d+t} - C_d$$

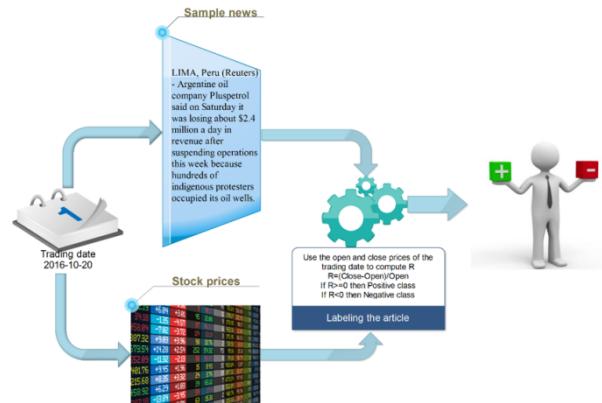
$$\text{Article label} = \begin{cases} \text{Positive} & \text{if } R_i \geq 0 \\ \text{Negative} & \text{if } R_i < 0 \end{cases}$$

where  $O_{d+t}$  is the opening price of the day after day d a period of t ( $t \in [1,2,5,7,10]$ ). For example, if day d is November 13th, 2016 and time period t is 2 then  $O_{d+t}$  is the opening price on November 15th, 2016.  $C_d$  is the closing price of the stock on day d. If  $R_i$  is greater or equal to 0 then the article on that day is classified as positive (stock price will likely go up) whereas if  $R_i$  is less than 0 then the news article is labeled as negative (stock price will likely go down).

Fig. 3 shows an overall process of labeling a news article; each article has a published date which is used to query the opening and closing stock prices from the database, then the Open-to-Close return is computed using opening and closing prices. The return value is then used for labeling the news article into the positive or negative news.

### C. WORD EMBEDDING

After the labeling step, a labeled dataset is generated, it is then used to train the proposed Stock2Vec model. The model



**FIGURE 3. Document labeling using historical stock prices. Idea from [17].**

was implemented using Python programming language; the total training time was approximately 40 minutes, each word is presented by a vector with a maximum length of 300, and the total number of words in the Stock2Vec was limited to 5,000. There are three main functions:

- build\_vocab takes Harvard IV-4 dictionary and stock news dataset as the input and the output of the function is word\_frequency array which contains a unique word\_id and its frequency in the entire dataset. The sentiment value has a range spread from -2 to 2, where -2 is really negative, -1 is negative, 1 is positive, and 2 is really positive so at the end of this step, the word which has high sentiment value will have high word\_frequency value.
- build\_cooccur accepts a word\_frequency array, stock news dataset, and required parameters: the context window size and minimum count were initially set to 10. The context window applies to the number of words used to determine the context of each word whereas the minimum count was used to drop rare word co-occurrence pairs. The distance from m\_word to context\_word is computed by counting the number of words between m\_word and context\_word. This function builds a sparse matrix which yields co-occurrence tuples in the form (word\_id, context\_word\_id, x<sub>ij</sub>), where x<sub>ij</sub> is the co-occurrence value.
- train\_stock2vec initializes the parameters of the model and manages the training process. It uses co-occurrences data to re-compute weight vectors, biases, and accompanying gradient histories every iteration.

```

1: Initialize vector_size=100, iterations=1, learning_rate = 3*10e-5
2: While iterations < 25 do
3:   W = averaging vocab and vector_size
4:   biases = compute based on vocab and vector_size
5:   cost = update_cost(vocab, data)
6:   Re-update gradient based on cost
7: End While
8: Return the optimal W

```

### D. TECHNICAL INDICATORS

In addition to the features extracted from the news dataset, we also use another feature set containing three indicators that

are commonly used in technical analysis. Technical indicators are mathematical calculations based on the price, volume, or open interest of a security or contract. By analyzing historical data, technical analysts use indicators to predict future price movements [27]. We add them to examine whether the system performance is improved. Three technical indicators are a Stochastic oscillator, William, and Relative Strength Index.

- Stochastic oscillator (%K): a momentum indicator which compares the closing price of a stock to its price range over a period. It can be used to foreshadow reversals when the indicator reveals bullish or bearish divergences.

$$\%K = 100 \times \left( \frac{C - L_p}{H_p - L_p} \right)$$

where C is the closing price of the day under consideration,  $L_p$  and  $H_p$  are lowest and highest prices in the last  $p$  days, respectively.

- William (%R): a momentum indicator which supports investors in detecting overbought and oversold conditions. It is based on a comparison between the current close and the highest high for a user-defined look back period p.

$$\%R = 100 \times \left( \frac{H_p - C}{H_p - L_p} \right)$$

where C is closing price of the day under consideration, and  $L_p$  and  $H_p$  are the lowest and highest prices in the last  $p$  days after the day under consideration, respectively.

- Relative Strength Index (RSI): a momentum oscillator which evaluates speed and trend of price movements. RSI fluctuates between 0 and 100. In practice, investors usually sell when RSI value  $\geq 80$  and buy when it is  $\leq 20$ .

$$RSI = 100 - \left( \frac{100}{100 + RS} \right)$$

where RS is the average gain of positive periods during a specified time frame / average loss of negative periods during the specified time frame.

#### E. TGRU NEURON NETWORK ARCHITECTURE

Each recurrent unit in the GRU network [30] captures the dependencies of different time scales adaptively. The activation n-dimensional  $h_t$  of the network at time t is a linear interpolation between element-wise multiplication  $\odot$  of the previous activation  $h_{t-1}$  and update gate  $z_t$  and element-wise multiplication  $\odot$  of the candidate activation  $h_t$  and  $(1-z_t)$ .

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h_t$$

where update gate  $z_t$  decides the level the unit updates its activation or content. The update gate is computed by a linear sum between the newly computed state and existing state with the bias parameter  $b_z$ .  $x_t$  is the m-dimensional input vector at

time t,  $\sigma$  is the logistic nonlinearity, and  $W_z$  ( $n \times m$  matrix),  $U_z$  ( $n \times n$  matrix),  $b_z$  ( $n \times 1$  vector) are fixed sized parameters (two weights and bias) which are shared across an entire network.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

GRU exposes to the whole state each iteration. The candidate activation  $h_t$  is calculated similarly to the traditional recurrent unit.

$$h_t = \tanh(Wx_t + U(r_t \odot h_{t-1}) + b_r)$$

in which  $r_t$  is a set of reset gates. When the status is off ( $r_t$  near 0), the reset gate forces the unit to act as it is processing the first symbol in the sequence of input, allowing it to remove previously calculated state  $h_{t-1}$ .

The reset gate  $r_t$  is computed similarly to the update gate but with different values:  $W_r$  ( $n \times m$  matrix),  $U_r$  ( $n \times n$  matrix),  $b_r$  ( $n \times 1$  vector).

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

By using gating mechanism, GRU can keep memory significantly longer than RNN. However, through observation, we figured out that when GRU analyzes a word it only considers the forward lingual context, so it is impossible for GRU to learn the backward context. As the result, the learning process is partly completed because, in any language model, the meaning of a word in a sentence is affected by not only the forward context but also on the backward context. TGRU was proposed to solve the above issue; it lets the model learn the lingual context of a word from both sides. TGRU is inspired by the bidirectional recurrent neural networks (BRNNs) in [28]. It divides each training sequence into two separate recurrent nets forwards and backward, both of them are combined to be the output layer. The formulas for update gate, reset gate, activation, and candidate activation of the forward and backward GRU are shown below:

Forward pass:

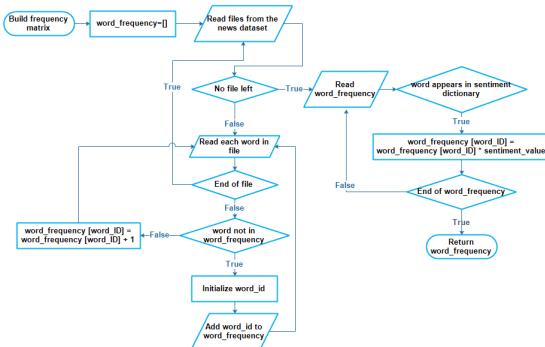
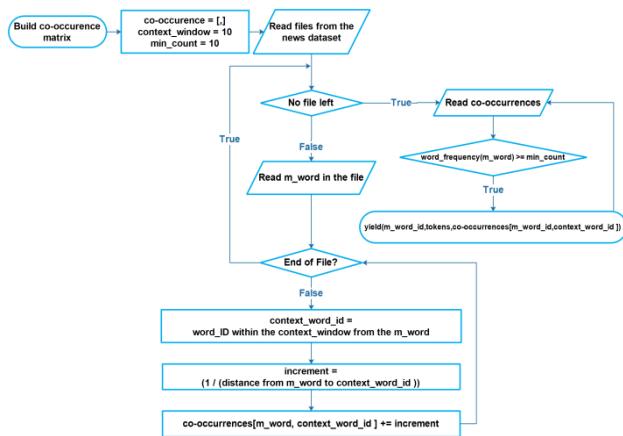
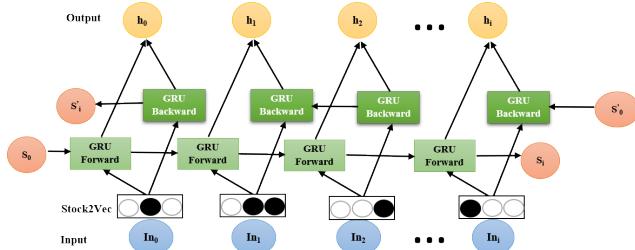
$$\begin{aligned} \vec{z}_t &= \sigma(\vec{W}_z x_t + \vec{U}_z \vec{h}_{t-1} + \vec{b}_z) \\ \vec{r}_t &= \sigma(\vec{W}_r x_t + \vec{U}_r \vec{h}_{t-1} + \vec{b}_r) \\ \vec{h}_t &= \tanh(\vec{W}_x x_t + \vec{U}(\vec{r}_t \odot \vec{h}_{t-1}) + \vec{b}_r) \\ \vec{h}_t &= \vec{z}_t \odot \vec{h}_{t-1} + (1 - \vec{z}_t) \odot \vec{h}_t \end{aligned}$$

The backward pass is the step we added to our model to explore more valuable information.

Backward pass:

$$\begin{aligned} \overleftarrow{z}_t &= \sigma(\overleftarrow{W}_z x_t + \overleftarrow{U}_z \overleftarrow{h}_{t-1} + \overleftarrow{b}_z) \\ \overleftarrow{r}_t &= \sigma(\overleftarrow{W}_r x_t + \overleftarrow{U}_r \overleftarrow{h}_{t-1} + \overleftarrow{b}_r) \\ \overleftarrow{h}_t &= \tanh(\overleftarrow{W}_x x_t + \overleftarrow{U}(\overleftarrow{r}_t \odot \overleftarrow{h}_{t-1}) + \overleftarrow{b}_r) \\ \overleftarrow{h}_t &= \overleftarrow{z}_t \odot \overleftarrow{h}_{t-1} + (1 - \overleftarrow{z}_t) \odot \overleftarrow{h}_t \end{aligned}$$

For the activation of a word at time t:  $h_t = [\vec{h}_t, \overleftarrow{h}_t]$  for a random sequence  $(x_1, x_2, \dots, x_n)$  containing n words, every word at time t is represented as a dimensional vector.

**FIGURE 4.** Flow diagram for the build\_vocab function.**FIGURE 5.** Flow diagram for the build\_cooccur function.**FIGURE 6.** Structure of the TGRU network.

The forward GRU computes  $\vec{h}_t$ , which represents the left to the right context of the sentence whereas the backward GRU step took the right to left context  $\overleftarrow{h}_t$  into consideration. Then the forward and backward context representations were concatenated into a single context. Fig. 6 shows the detailed structure of TGRU.

In general, the complexity of an algorithm is computed using  $O(W)$  and the number of estimated parameters of the algorithm is  $W$ . Two common pieces of information used to compute  $W$  is the dimension of the input vector **m-dimensional** and hidden layer dimension **n-dimensional**. Table 2 shows the estimated parameters of GRU, LSTM, and TGRU.

**TABLE 2.** The number of parameters in GRU, LSTM and BGRU.

Model	Number of parameters
GRU	$3 \times (n^2 + nm + n)$
LSTM	$4 \times (n^2 + nm + n)$
Proposed BGRU	$6 \times (n^2 + nm + n)$

The complexity of TGRU doubles the complexity of GRU because the number of parameters are double. As a result, TGRU needs more time and resources for computation than GRU or LSTM. However, TGRU is able to explore valuable information which hugely improves the accuracy of the stock trends prediction.

#### IV. EXPERIMENTAL RESULTS

In this section, we conduct several experiments to show how well TGRU performs compared to state-of-the-art methods. There are four main parts, the first part describes the dataset, the second part explains carefully about evaluation protocols used in our study, the third part contains three experiments on examining the financial indicators, the impact of a different time period and the effectiveness of TGRU. Finally, TGRU is being compared with recent studies and a trading simulation is conducted.

#### A. DATASET

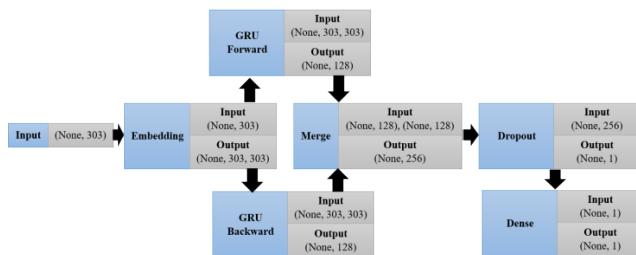
This study contains two different datasets: (1) Daily financial news from Reuters and Bloomberg from between October 2006 and November 2013, totaling 106,521 and 447,145 news articles from each source respectively. (2) Daily S&P500 stock prices in the same period are downloaded from Yahoo Finance. We then split the dataset (1) into three parts which were similar to those in [18] and [19] for comparison; the training part contains news articles between 2006-10-01 and 2012-12-31 (439,304 articles), the validation part involves news from 2013-01-01 to 2013-06-15 (2,305 articles) and the testing part includes news from 2013-06-16 to 2013-12-31 (98,468 articles).

#### B. EVALUATION PROTOCOLS

The performance of the system is evaluated by calculating the confusion matrix. Table 3 describes the components of the confusion matrix which includes TP, TN indicating the right classification for the corresponding class while FP, and FN declaring the false classification for the corresponding class. After obtaining the value in the confusion matrix, precision, recall, and accuracy are used to evaluate system performance. Accuracy is the number of correctly classified samples on the entire dataset. In general, the higher the accuracy, the better the classifier. However, accuracy cannot give us a comprehensive evaluation of the system. As a result, precision and recall are two widely used measurements beside accuracy. High precision means most of the samples which were classified into positive were correct. On the other hand, recall or sensitivity

**TABLE 3.** Confusion matrix for classification evaluation.

		Predicted class	
		Positive	Negative
Actual class	Positive	TP	FN
	Negative	FP	TN

**FIGURE 7.** The configuration of TGRU network.

is the proportion of positive trend news that was predicted by our classifier as positive trend news.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

### C. INITIAL EXPERIMENTS

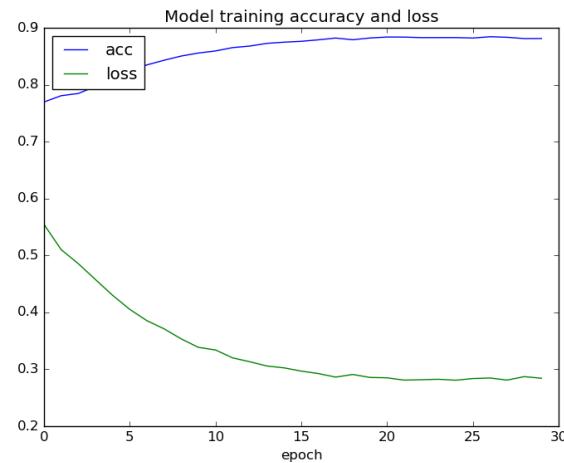
The implementation is conducted on the NVIDIA DIGITS toolbox with Keras API version 1.2.2 using Python version 2.7.3. The model was trained through 30 epochs, and the average training time for each epoch was 64 minutes. A Linux machine with Ubuntu 14.04 was used for the entire process; the machine specification was: Intel®Core i7-5930K processor with four NVIDIA Titan X 12GB GPUs, four 3072 Cuda cores, and 64GB of DDR4 RAM.

Fig. 7 describes each layer in the TGRU in detail. In the input layer, because the article length was different, we limited the length to 300 words, truncating long news articles and padding shorter news articles with zero values. Before combining three financial indicators into the features extracted from the news article, they were normalized to a range between 0 and 1, then the normalized features were combined with the previous features to form the input vectors; each vector now has a dimension of 303. This layer is followed by the Embedding layer which accepts a maximum of 303 input sequences for each news article. The total number of words in the vocabulary was set to the 10,000 most frequent words. The next layer is the TGRU layer with 128 memory units for each forward and backward pass. For each iteration, we applied a dropout for the GRU layer; the input Dropout-W was set to 30 percent while hidden state Dropout-U was set

to 30 percent. After merging the forward GRU and backward GRU, another dropout layer was added to drop 50% of the input to cope with the overfitting problem. The last layer is the Dense layer which has a single neuron, and it used sigmoid activation function to decide 0 or 1 for the two classes prediction (positive or negative). Moreover, through each epoch, the mean squared error loss value is calculated to minimize this value; this model uses Adam optimizer with batch size  $n = 64$ , learning rate  $\eta = 0.001$ , and learning rate decay  $t = 0.0001$ . The time-based learning rate schedule was used to anneal the learning rate over time. Therefore, the learning rate for epoch  $k$ -th was determined by:

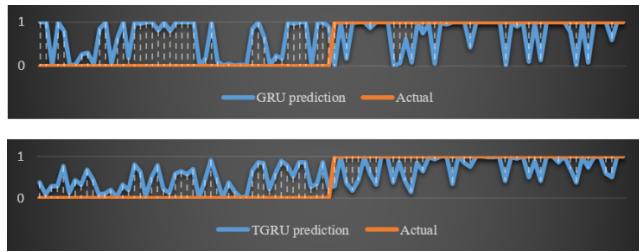
$$\eta_{t+1} = \frac{\eta_t}{(1 + k \times t)}$$

The detailed accuracy and loss model is given in the Fig. 8. The model is trained in 30. As shown in the figure, the accuracy of training and validation set increase significantly over 85% and corresponding loss of training and validation decrease dramatically below 35% after 10 epochs. Those numbers go stability with very little fluctuation before stopping at over 88% for accuracy and 30% for the loss.

**FIGURE 8.** Impact of stock news on the price of stock at different time periods.

### 1) TGRU AND GRU COMPARISON

We evaluate TGRU effectiveness over GRU by selecting 100 random news samples from the original dataset, which contain 50 positive news and 50 negative news. As shown in Fig. 9, the orange line represents the actual class by calculating the Open-to-Close approach (0 stands for negative or 1 stands for positive), while the blue line shows the trends predicted by GRU and TGRU. Because the Dense layer uses the sigmoid function, it returns a real value output between 0 and 1. TGRU achieved an accuracy of 67% while GRU obtained the accuracy of 58%. It is noticeable that the blue line in the TGRU model is nearer to the orange line compared to GRU which indicates TGRU prediction is better than GRU.

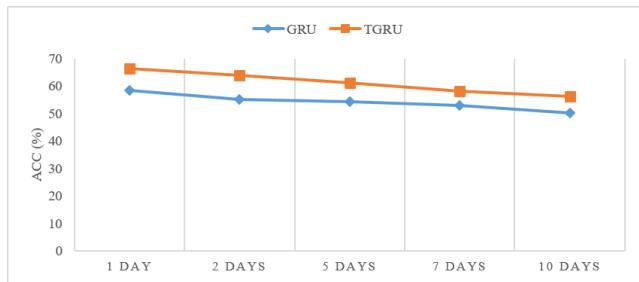


**FIGURE 9.** The configuration of TGRU network.

## 2) EVALUATING THE BEST TIME PERIOD

In this section, we evaluate the effect of news articles on stock prices on different time periods, by changing the  $i$  value ( $i \in [1, 2, 5, 7, 10]$ ) of the Open-to-Close price return which indicates the time intervals (1 day, 2 days, 5 days, 7 days, and 10 days). For example,  $i = 1$  means the news articles will affect the stock prices within 24 hours after the news was published, so the open and close prices at that day are queried from the database and Open-to-Close price equation will be calculated to decide whether to set a positive or negative label for the news; the remaining time intervals follow the same rule.

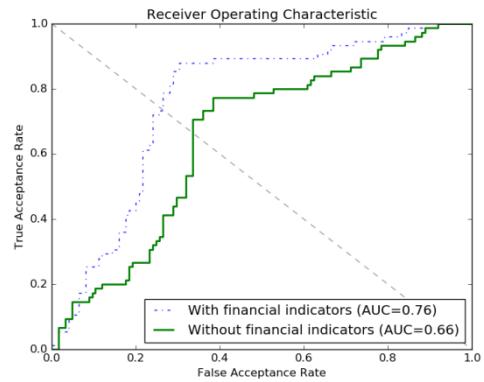
Fig. 10 shows the performance of the system at different time intervals. The results prove that the model is suitable for daily stock movement rather than the long-term movement with the highest accuracy of 66.32 percent for TGRU. The system performance decreases gradually as the time gets longer which proves the hypothesis given in [2] that the news article immediately influences the investors trading actions (buy, sell) in a short period ( $<=24$  hours); as the period increases, the news has a lesser impact on the investor's decisions. Take the "Brexit" event in 2016 as an example; it impacted the entire UK stock market immediately in negative ways after "Brexit" was announced by the government.



**FIGURE 10.** Impact of stock news on the price of stock at different time periods.

## 3) FINANCIAL INDICATORS EVALUATION

In this section, an experiment was conducted to verify the effectiveness of the model when the financial indicators feature was added. Fig. 11 shows the Receiver Operating Characteristic (ROC) for stock movement prediction in two cases: using the financial indicators feature set and without the



**FIGURE 11.** Receiver Operating Characteristic curves with and without indicators.

financial indicators feature. The area under the curve (AUC) was 0.66 for the model which did not use financial indicators features, but when the financial indicators feature was used, we achieved a significant improvement of 0.1 in the area under the ROC curve to 0.76. Through the ROC, it is clear that applying the financial indicators noticeably improved system performance because these indicators are used to verify the labeling procedure.

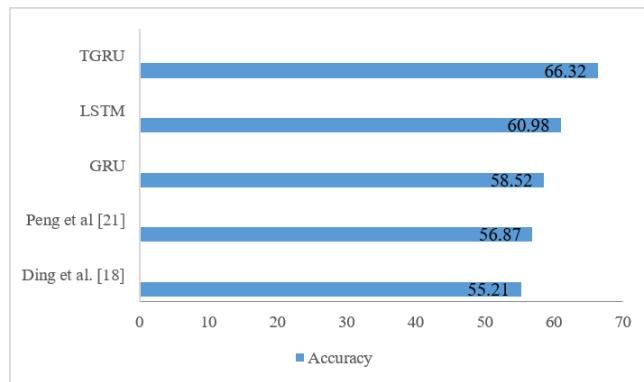
## D. COMPARING TGRU WITH RECENT STUDIES

### 1) STOCK TRENDS PREDICTION ON S&P 500

This section evaluates the system performance compared with recent studies [18], [21]. Ding *et al.* [18] proposed using Open Information Extraction for event-based stock price trend prediction to extract structured events from large-scale public news articles; the structured event was denoted as  $(O_1, P, O_2)$  where  $O_1$  is the first object (ticker name, company name, etc.),  $O_2$  is the second object (ticker name, company name, etc.), and  $P$  indicates the relationship between them. They applied the feedforward neural network and achieved an accuracy of 55.21%. Using the same dataset, Peng *et al.* [21] employed the word embedding method and used DNNs to predict the future stock movements of the S&P 500 index based on the extracted features, with an accuracy that was slightly improved to 56.87%.

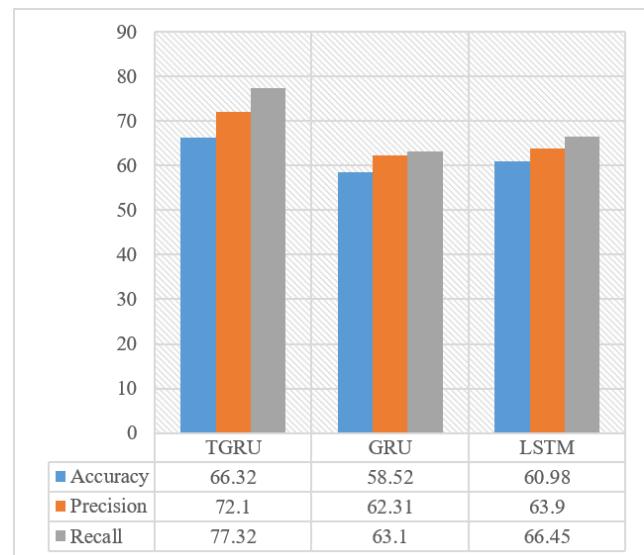
GRU and TGRU were implemented on the same dataset and same period as these two studies. LSTM (long short-term memory), which is an extension of RNN that has been widely used in recent years [24], was also added to this experiment to allow for further comparisons. Both GRU and LSTM are RNN extensions so which algorithm is better? We deploy GRU and LSTM on our dataset with the parameters similar to those in [21].

The results in Fig. 12 proved that our TGRU outperforms other methods (better than Ding at 55.21% and Peng at 56.87%) at an accuracy of 66.32%. The improvement is explainable because not only did we construct the word embedding, add the sentiment analysis and three technical indicators but we also combined the forward and backward

**FIGURE 12.** Our proposed model compared with two previous models.

context to learn more valuable information from the news article. It is also noticeable that LSTM and GRU performed better than the results from two previous studies at 60.98% and 58.52%, respectively.

We also evaluate the system performance on other metrics. Fig. 13 shows the accuracy, precision, and recall metrics used to judge the system performance using LSTM, GRU, and TGRU. Overall, TGRU has the highest accuracy compared to the others. Also, TGRU has the highest f-measure at 77.3% and precision at 72.1%. The high precision indicates the number of positive prediction samples which are true compared to the number of actual positive samples.

**FIGURE 13.** TGRU, LSTM, and GRU performance on different measurements.

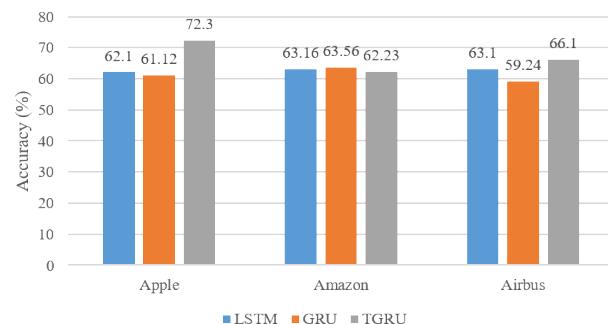
The main purpose of the model is to predict the price trend in the S&P 500 index - the stock index for 500 large companies which have stock tickers on the NYSE or NASDAQ. Besides, we want to verify the model performance on individual company stock indexes because this approach is more useful for investors who invest in several companies.

The evaluation was carried out on the three biggest companies in different sectors. Apple represents the Information Technology sector; Amazon stands for the E-commerce sector, and Airbus represents the manufacturing sector (classified by The Global Industry Classification Standard). From the original dataset, news relating to these three companies are selected (news contains the company name, stock ticker, etc.). Table 4 shows the amount of news for each company.

**TABLE 4.** Amount of news for each company.

Company name	Training	Evaluation	Testing
Apple	1,802	450	1,124
Amazon	1,188	296	741
Airbus	1,664	416	1,039

The results in Fig. 14 prove that overall our TGRU achieves better performance than GRU and LSTM. The results from GRU and LSTM are quite similar. Another interesting fact is that the accuracy of Apple-related news using TGRU is over 72.3%, which proves the effectiveness of our system.

**FIGURE 14.** LSTM, GRU and TGRU performance on individual stock trend prediction.

## 2) STOCK TRENDS PREDICTION ON VNINDEX

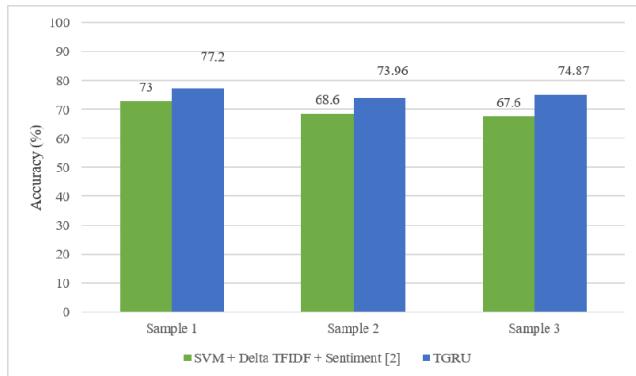
Besides the S&P 500 index, we also perform an additional experiment on Vietnamese stock dataset collected by Dang et al. [2] to compare TGRU to Support vector machine algorithm. The news dataset contains a total of 2,471 news crawled from Vietstock.vn financial news website between 01/2014 and 05/2015. We divided it similar to those in [2], the reason it was separated into three samples is to examine investor's behavior on different time periods. The detailed information is described in Table 5.

Dang et al. [2] used TF-IDF combined with sentiment dictionary as term weighting technique, they also used OCFS (orthogonal centroid feature selection) as term reduction. Finally, in the classification step, they chose Linear SVM using a linear kernel with  $C = 0.5$ . They also applied LibLinear, which is a famous library for linear kernel support vector machines (SVM). We implemented TGRU on this dataset with the same parameters as described above.

**TABLE 5.** VNindex news dataset description.

Company name	Training	Evaluation	Testing
Sample 1 (01/2015-04/2015)	1,092	463	1,555
Sample 2 (09/2014-04/2015)	1,499	640	2,139
Sample 3 (05/2014-04/2015)	1,715	756	2,471

By observing the results shown in Fig. 15, our TGRU yields higher prediction accuracy compared to the approach using SVM + TF – IDF. Overall, TGRU achieves the average accuracy of over 70% and it yields a higher accuracy compared to SVM, it is a noticeable accuracy of 77.2% which is 4.2% higher than [2]. In sample 3, the accuracy for TGRU is 7% better than SVM. As the result, TGRU is proved to outperform SVM when the dataset gets bigger.

**FIGURE 15.** Comparison between our method and SVM on the dataset proposed by Dang et al. [2].

### 3) MARKET SIMULATIONS

In the final experiment, an automated trading system is conducted to evaluate the profits under real trading conditions. The stock investment was carried out on the blue-chip stock Google (GOOGL). The initial investment is assumed to be 10,000 USD, and the transaction fee (buy/sell) is charged at 0.25% of the trading amount. The dataset was collected between October 9th, 2017 and November 9th, 2017, while the number of news articles for each day was limited to one. As a result, a total of 24 sets of news were gathered. Figure 16 shows the price movements and the predicted trend of GOOGL during the period. Also, each day is limited to only one transaction (buy/sell) to avoid too many transaction charges resulting from too many trading actions. It operated under the following rules [6].

- If there is no news released, do nothing.
- If the stock was bought and the prediction was negative for the day under consideration, the stock will be sold when the market opens.

- If the stock was bought while the prediction was positive, do nothing.
- If the stock was sold while the prediction was positive, do nothing.
- If the stock was sold and the prediction was negative, buy with all the money when the market opens.

Fig. 16 shows the statistics for the actual trend (orange line) and predicted trend (blue line) over a one-month period. Table 6 describes the detailed transactions of the system. Our system made eight trades in total within a month based on the prediction model. From the initial 10,000 USD investment, at the end of the month we held 10,531 USD, so the profit was over 500 USD. The profit proves that our system is reliable and can be used to support investors in making trading decisions to some extent.

**FIGURE 16.** Predicted trend and actual price trend of GOOGL between October 2nd and November 2nd, 2017.**TABLE 6.** GOOGL trading activities on one-month period.

Date (mm/dd)	Transaction type	Shares	Money left (USD)	Opening price (USD)
10/04	Buy	10	406	957
10/06	Sell	0	10,049	966.7
10/11	Buy	10	336	973.72
10/12	Sell	0	10,186	987.45
10/19	Buy	10	351	986
10/30	Sell	0	10,466	1,014
10/31	Buy	10	339	1,015.22
10/02	Sell	0	<b>10,531</b>	1,021.76

### V. CONCLUSIONS

The topic of stock trend prediction using deep learning interests many researchers and investors because the improved prediction accuracy will probably bring enormous profit. In this paper, we introduced a stock movement prediction framework to support investors in trading stock on the market.

The proposed Two-stream Gated Recurrent Unit (TGRU) overall accuracy was 66.32% which outperforms the performance of the previous model including GRU, LSTM, our model has two states of learning including backward and forward so it is able to learn more useful information, especially for text processing problem. Then a sentiment Stock2Vec embedding is created by using financial dataset and Harvard IV-4 sentiment dictionary, through the experiments, the Stock2Vec embedding proved to be more effective than the original embedding method such as Glove and Word2Vec because it takes the sentiment value of the word into consideration. Moreover, the financial indicators, which are one

of the most important factors in the financial analysis, are also added. The model shows its robustness in both the S&P 500 index and individual stock trend prediction. In addition, a simulation system was conducted to compute the actual profits investors earn when they use our system; it proved to be robust against market volatility and the ability to adjust itself to the risk from the real market. This framework can also be integrated into an automated system to support investors in trading specific stocks.

There are several ways to extend this work in the future. In our study, only the daily period was considered, so it would be better if the model can be applied to intraday trading which means the system could predict the trend minutes or hours after the news is published. In addition, only three financial indicators are used in this study, so exploring other indicators such as P/E ratio and earnings growth is necessary. Another weakness of the system is in the TGRU network because the complexity of the network double that of GRU so the framework requires long training time and huge computational resources.

## REFERENCES

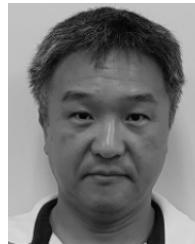
- [1] A. Bastianin and M. Manera, "How does stock market volatility react to oil price shocks?" *Macroeconomic Dyn.*, vol. 22, no. 3, pp. 666–682, 2018.
- [2] M. Dang and D. Duong, "Improvement methods for stock market prediction using financial news articles," in *Proc. IEEE 3rd Nat. Found. Sci. Technol. Develop. Conf. Inf. Comput. Sci. (NICS)*, Sep. 2016, pp. 125–129.
- [3] I. Hassan, "Exploiting noisy data normalization for stock market prediction," *J. Eng. Appl. Sci.*, vol. 12, no. 1, pp. 69–77, 2017.
- [4] S. Jeon, B. Hong, and V. Chang, "Pattern graph tracking-based stock price prediction using big data," *Future Gener. Comput. Syst.*, vol. 80, pp. 171–187, Mar. 2018.
- [5] E. Chong, C. Han, and F. C. Park, "Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies," *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.
- [6] B. Weng, M. A. Ahmed, and F. M. Megahed, "Stock market one-day ahead movement prediction using disparate data sources," *Expert Syst. Appl.*, vol. 79, pp. 153–163, Aug. 2017.
- [7] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. IEEE Int. Conf. Adv. Comput., Commun. Inform. (ICACCI)*, Sep. 2017, pp. 1643–1647.
- [8] Y. Zhang, W. Song, D. Shen, and W. Zhang, "Market reaction to Internet news: Information diffusion and price pressure," *Economic Model.*, vol. 56, pp. 43–49, Aug. 2016.
- [9] S. Baruch, M. Panayides, and K. Venkataraman, "Informed trading and price discovery before corporate events," *J. Financial Econ.*, vol. 125, no. 3, pp. 561–588, 2017.
- [10] J. Zhou, B. Chen, Y. Lin, "An approach to constructing sentiment collocation dictionary for Chinese short text based on Word2Vec," in *Proc. Int. Symp. Emerg. Technol. Educ.* Cham, Switzerland: Springer, 2017, pp. 548–556.
- [11] S. Zhang, Z. Wei, Y. Wang, and T. Liao, "Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary," *Future Gener. Comput. Syst.*, vol. 81, pp. 395–403, Apr. 2018.
- [12] Y. Song, S. Jeong, and H. Kim, "Semi-automatic construction of a named entity dictionary for entity-based sentiment analysis in social media," *Multimedia Tools Appl.*, vol. 76, no. 9, pp. 11319–11329, 2017.
- [13] W. Li, K. Guo, Y. Shi, L. Zhu, and Y. Zheng, "DWWP: Domain-specific new words detection and word propagation system for sentiment analysis in the tourism domain," *Knowl.-Based Syst.*, vol. 146, pp. 203–214, Apr. 2018.
- [14] T. Loughran and B. McDonald, "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks," *J. Finance*, vol. 66, no. 1, pp. 35–65, 2011.
- [15] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Appl. Intell.*, vol. 42, no. 4, pp. 722–737, 2015.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [17] X. Li, X. Huang, X. Deng, and S. Zhu, "Enhancing quantitative intra-day stock return prediction by integrating both market news and stock prices information," *Neurocomputing*, vol. 142, pp. 228–238, Oct. 2014.
- [18] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Using structured events to predict stock price movement: An empirical investigation," in *Proc. EMNLP*, 2014, pp. 1–11.
- [19] Y. Peng and H. Jiang, "Leverage financial news to predict stock price movements using word embeddings and deep neural networks," in *Proc. NAACL-HLT*. San Diego, CA, USA: Association for Computational Linguistics, Jun. 2016, pp. 374–379.
- [20] R. Gelbard, R. Ramon-Gonen, A. Carmeli, R. M. Bittmann, and R. Talyansky, "Sentiment analysis in organizational work: Towards an ontology of people analytics," *Expert Syst.*, p. e12289, May 2018. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/exsy.12289>
- [21] H. Peng, E. Cambria, and A. Hussain, "A review of sentiment analysis research in Chinese language," *Cogn. Comput.*, vol. 9, no. 4, pp. 423–435, 2017.
- [22] I. Medovikov, "When does the stock market listen to economic news? New evidence from copulas and news wires," *J. Banking Finance*, vol. 65, pp. 27–40, Apr. 2016.
- [23] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation." [Online]. Available: <https://arxiv.org/abs/1406.1078>
- [24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [25] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.
- [26] M. F. Porter. (2001). *Snowball: A Language for Stemming Algorithms*. [Online]. Available: <http://snowball.tartarus.org/texts/introduction.html>
- [27] A. Jabbarzadeh, S. Shavvalpour, H. Khanjarpahani, and D. Dourvash, "A multiple-criteria approach for forecasting stock price direction: Non-linear probability models with application in S&P 500 index," *Int. J. Appl. Eng. Res.*, vol. 11, no. 6, pp. 3870–3878, 2016.
- [28] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [29] F. Wang, S.-J. Shieh, S. Havlin, and H. E. Stanley, "Statistical analysis of the overnight and daytime return," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 79, no. 5, p. 056109, 2009.
- [30] R. Dey and F. M. Salemt, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Boston, MA, USA, Aug. 2017, pp. 1597–1600.
- [31] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: <https://arxiv.org/abs/1301.3781>
- [32] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1–12.
- [33] T. Le, M. Y. Lee, J. R. Park, and S. W. Baik, "Oversampling techniques for bankruptcy prediction: Novel features from a transaction dataset," *Symmetry*, vol. 10, no. 4, p. 79, 2018.
- [34] T. Le, L. H. Son, M. T. Vo, M. Y. Lee, and S. W. Baik, "A cluster-based boosting algorithm for bankruptcy prediction in a highly imbalanced dataset," *Symmetry*, vol. 10, no. 7, p. 250, 2018.



**DANG LIEN MINH** received the B.S. degree in information systems from the University of Information Technology, VNU HCMC, Vietnam, in 2016. He is currently pursuing the Ph.D. degree in computer science with Sejong University, Seoul, South Korea. In 2017, he joined the Computer Vision Pattern Recognition Laboratory. His current research interests include computer vision, natural language processing, and artificial intelligence.



**ABOLGHASEM SADEGHI-NIARAKI** received the B.Sc. and M.Sc. degrees from the Department of Geospatial Information System (GIS) Engineering, Geodesy and Geomatics Engineering Faculty, K. N. Toosi University of Technology, Tehran, Iran, and the Ph.D. degree in GIS engineering and IT (geo-informatics) from the Department of Geoinformatic Engineering, Inha University, South Korea. He is currently an Assistant Professor at the Department of Computer Science and Engineering, Sejong University. He taught various GIS and IT & ubiquitous fields to undergraduate and graduate students. He received the Distinguished Researcher Award from the Iranian President. Moreover, he was a GIS Project Manager with the IT Office, Road Maintenance Transportation Organization, Road Ministry, Tehran. In 2003, he was selected as a National Distinguished Researcher of the Year in Iran. His current research interests include geo-informatic engineering and GIS ubiquitous.



**KYUNGBOK MIN** received the B.S. degree in electronics and computer engineering from Korea University. He is currently pursuing the Ph.D. degree in computer science with Sejong University, Seoul, South Korea. He has been with the Computer Vision Pattern Recognition Laboratory since 2017. His current research interests include computer vision, natural language processing, and artificial intelligence.



**HYEONJOON MOON** received the B.S. degree in electronics and computer engineering from Korea University in 1990, and the M.S. and Ph.D. degrees in electrical and computer engineering from the State University of New York at Buffalo in 1992 and 1999, respectively. From 1996 to 1999, he was a Senior Researcher with the Electro-Optics/Infrared Image Processing Branch, U.S. Army Research Laboratory, Adelphi, MD, USA. He developed a face recognition system evaluation methodology based on the Face Recognition Technology Program. From 1999 to 2003, he was a Principal Research Scientist with Viisage Technology, Littleton, MA, USA. Since 2004, he has been with the Department of Computer Science and Engineering, Sejong University, where he is currently a Professor and the Chairman. He has extensive background on still image and real-time video based computer vision and pattern recognition. His main interests include research and development in real-time facial recognition system for access control, surveillance, and big database applications. His current research interests include image processing, biometrics, artificial intelligence, and machine learning.



**HUYNH DUC HUY** received the M.S. degree in computer science from the University of Information System in 2017. He has been an Instructor with the E-Commerce Department, UIT, since 2014. He has strong background on machine learning, natural language processing, and big data. His research interests include strategic and technical issues in corporate information systems.