Deep Learning for Stock Market Prediction Using Event Embedding and Technical Indicators

Pisut Oncharoen and Peerapon Vateekul
Chulalongkorn University Big Data Analytics and IoT Center (CUBIC)
Department of Computer Engineering, Faculty of Engineering
Chulalongkorn University, Bangkok, Thailand
pisut.o@student.chula.ac.th, peerapon.v@chula.ac.th

Abstract— **Recently, ability to handle tremendous amounts of information using increased computational capabilities has improved prediction of stock market behavior. Complex machine learning algorithms such as deep learning methods can analyze and detect complex data patterns. The recent prediction models use two types of inputs as (i) numerical information such as historical prices and technical indicators, and (ii) textual information including news contents or headlines. However, the use of textual data involves text representation construction. Traditional methods like word embedding may not be suitable for representing the semantics of financial news due to problems of word sparsity in datasets. In this paper, we aim to improve stock market predictions using a deep learning approach with event embedding vectors extracted from news headlines, historical price data, and a set of technical indicators as input. Our prediction model consists of Convolutional Neural Network (CNN) and Long Short-term Memory (LSTM) architectures. We use accuracy and annualized return based on trading simulation as performance metrics, and then perform experiments on three datasets obtained from different news sources namely Reuters, Reddit, and Intrinio. Results show that enhancing text representation vectors and considering both numerical and textual information as input to a deep neural network can improve prediction performance.**

Keywords—Deep Learning; Convolutional Neural Network; Long Short-term Memory; Event Embedding; Stock Market Prediction

I. INTRODUCTION

The goal of most investors is to predict the behavior of the stock market to decide between buying or selling shares of stocks and seeking to maximize profit on investment. However, predicting behavior is a difficult task because stock markets are highly volatile and influenced by many external factors including the global economy, events, politics and investor expectations.

The random walk theory [1] states as a hypothesis for an efficient market that stock prices reflect all the available information and any price movements are random processes. Hence, it is impossible to predict future market movements. However, with recent advances in artificial intelligence and increasing amounts of data, researchers can predict stock market behavior better and more efficiently than a random process. Examples are studies [2-7].

The traditional approach for stock market prediction involves applying knowledge of artificial intelligence with financial techniques, i.e., technical analysis as a study of historical price movements and fundamental analysis as a study of economic business trends and past financial statements to forecast future results. References [2, 3] focus on predicting stock market behavior using machine learning algorithms combined with knowledge of technical analysis, while [4-7] are inspired by fundamental analysis. These studies focus on text mining approaches and machine learning techniques combined with textual information such as news contents, headlines and financial reports to discover relationships and then use them to predict future market behavior.

Modern approaches for stock market prediction are based on deep learning methods. For example, [8, 9] use a deep neural network model and feed event representation vectors extracted from financial news headlines as input to predict S&P 500 index movements. Reference [10] attempts to forecast market movement by applying LSTM networks with historical price data. These studies yield better prediction performance than traditional machine learning algorithms and focus on building a prediction model using either numerical or textual information. However, most investors analyze market behavior based on both types of information to make the best decision. Hence, recent research direction concentrates on applying the deep learning approach with both kinds of information. For example, study [11] uses news headlines and historical prices as input for a prediction model, while [12] uses a set of technical indicators instead of historical price data. These studies give better prediction performance than a single input type but use only traditional methods like word embedding and paragraph vectors to create news headline representation vectors. These may not be suitable for representing the semantics of financial news headlines because of word sparsity in the datasets, which potentially limits the predictive power. Furthermore, the experiments are conducted based on only a single dataset, so we cannot be confident that these models can work well for other datasets.

In this paper, we propose a framework for stock market prediction by feeding textual and numerical information as input to a deep learning model consisting of CNN and LSTM architectures. For textual information, we employ the event embedding technique [9] to extract text representation vectors from news headlines. Long-term and mid-term event vectors are fed to CNN layers to extract feature maps that represent the most important events during these periods. The feature maps are concatenated with short-term event vectors and fed into the next hidden layers. For numerical information, we use historical prices and a set of technical indicators derived from price data. This information is fed to LSTM layers to analyze temporal relationships. Outcomes from numerical and textual parts are then combined and fed

978-1-5386-4804-9/18/$31.00 @2018 IEEE

to the final hidden layers to make a prediction. Experiments are conducted on three datasets from different news sources as Reuters, Reddit and Intrinio. We use accuracy and annualized return based on trading simulations as performance metrics. Results show that prediction performance can be improved by considering both numerical and textual information as input to a deep neural network, thereby enhancing text representation vectors.

The remainder of this paper is organized as follows. Section II reviews related literature regarding stock market prediction using the deep learning approach. Section III describes the proposed model framework. Section IV explains the experimental setup. Section V presents and discusses the results with conclusions made in Section VI.

II. Related Works

This section focuses on related studies that attempt to predict stock market behavior using the deep learning approach. Each has a different deep neural network structure and model input data. Previous studies can be split into two groups as described below

A. Deep Learning with Textual Information

Traditional approaches for representing textual information use bags-of-words (BoW), term frequency-inverse document frequency (TF-IDF), named entities and noun phrases. However, these simple approaches are not suitable for representing the semantics of news headlines and cannot capture entity-relation information.

In 2014, Ding et al. [8] proposed a framework that utilized Open Information (Open IE) technique to extract structured events from news headlines and news contents and then fed them into a deep neural network to predict stock price movement. Their approach gave improved prediction accuracy over using traditional methods. In 2015, Ding et al. [9] proposed an improved method to represent an event from the news, called the event embedding approach. Using this approach, news describing similar events is encoded into similar event vectors.

In 2017, Huynh et al. [13] introduced a prediction model using Bidirectional Gated Recurrent Unit (BGRU) architecture by feeding word embedding vectors from news headlines as input. The performance of this approach was better than the baseline in [8]. However, it was still lower than [9] on the same dataset because the event embedding approach can generate event vectors that represent improved news headline semantics better than using word vectors.

Here, we employ the event embedding approach to represent textual information due to its effectiveness over other approaches. Details of the event embedding approach will be explained in the next section.

B. Deep Learning with Both Textual and Numerical Information

Due to advances in the artificial intelligence field, as well as the increasing amount of data, many researchers are interested in applying deep learning approaches to analyze both types of information to improve the performance of stock market prediction.

In 2016, Akita et al. [11] proposed a prediction model that used Japanese news headlines and historical prices as input. The news headlines are converted to representation vectors by the paragraph vector approach and concatenated with price vectors. They are then fed into LSTM networks to forecast the following day's stock prices. The performance of the model is evaluated by trading simulation. They found that using both types of information resulted in improved prediction performance, paragraph vector was a better news headline representation than bags-of-words, and LSTM performed better than traditional approaches such as Support Vector Machine (SVM), Multi-layer Perceptron (MLP) and Recurrent Neural Network (RNN).

In 2017, Vargas et al. [12] introduced the deep model for directional movement prediction of Standard & Poor's 500 Index using financial news headlines and a set of technical indicators as input. They focused on architectures such as CNN and RNN, which had previously shown good results in traditional natural language processing (NLP) tasks. Their results showed some improvement compared with baselines from previous studies.

In this paper, we use the same set of technical indicators as proposed by [12, 14]. However, stock prices are highly volatile. As such, using technical indicators for price smoothing might not fully capture the volatility of price movements. We also add opening, highest, lowest and closing prices (OHLC) into numerical input, as our preliminary study indicated that they improved prediction performance.

III. Proposed Framework

Our objective is to improve prediction performance using both numerical and textual information as input to a deep neural network. The process flow of our proposed framework is shown in Fig. 1.

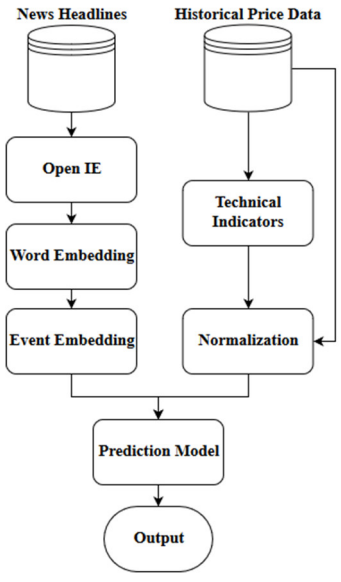

Figure 1. Process flow of the proposed framework

A. Preprocessing

1) Technical Indicator

A new feature is computed using historical price data. In this paper, we use seven technical indicators as proposed by [12, 14]. The formulas of these indicators are summarized in Table 1.

TABLE I. SUMMARY OF INDICATORS

Feature	Formula	Feature	Formula
Stochastic %K	$\frac{C_t - LL_n}{HH_n - LL_n}$	William's %R	$\frac{H_n - C_t}{H_n - L_n} x100$
Stochastic %D	$\frac{\sum_{i=0}^{n-1} \%K_{t-i}}{n}$	A/D Oscillator	$\frac{H_t - C_{t-1}}{H_t - L_t}$
Momentum	$C_t - C_{t-n}$	Disparity 5	$\frac{C_t}{MA_5} x100$
Rate of Change	$\frac{C_t}{C_{t-n}} x100$		

where C_t is the closing price at day t, H_t is the highest price at day t, L_t is the lowest price at day t, MA_n is moving average of the past n days, and HH_n and LL_n are the highest high and the lowest low in the past n days, respectively.

2) Normalization

It is necessary to standardize the input vectors that represent technical indicators and prices because each indicator has a different range of value. We apply the z-score to normalize the input vectors using the following formula:

$$z = \frac{x-\mu}{\sigma} \quad (4)$$

where μ is the mean of the input x and σ is the standard deviation of the input x.

3) Event Representation

We apply the idea in [8] that attempts to extract event representation from daily news headlines using Open Information Extraction framework (Open IE). This framework transforms a news headline into a tuple of three values denoted as (Actor, Action, Object). For example, a news headline such as "Microsoft agrees to buy Nokia's mobile phone business for $7.2 billion." is transformed to (Actor = Microsoft, Action = buy, Object = Nokia's mobile phone business). Instead of implementing our own Open IE framework, we use the free Open IE software developed by Stanford [15]. The tuples extracted by Open IE are converted to word vectors using the pre-trained word vectors namely GloVe [16] with 100 dimensions. The output word vectors are used as input for event embedding construction.

4) Event Embedding

Event embedding is introduced in [9]. This process creates event vectors by feeding Actor (O_1), Action (P) and Object (O_2) vectors into the Neural Tensor Network as shown in Fig. 2.

The vector R_1 in Fig. 2 represents a relation between Actor (O_1) and Action (P). This vector can be computed by (1) where f is tanh function, k is a dimension of the input vector, W is $k\ x\ 2k$ weights matrix, and b is biases vector.

$$R_1 = f(O_1^T \cdot T_1^{[1:k]} \cdot P + W \begin{bmatrix} O_1 \\ P \end{bmatrix} + b) \quad (1)$$

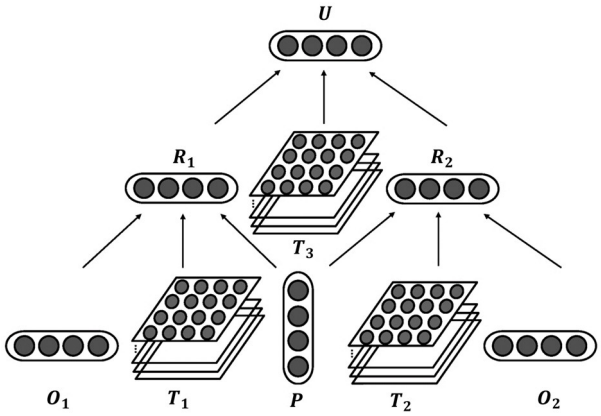

Figure 2. Structure of Neural Tensor Network [9]

The vector R_2 that represents a relation between Object (O_2) and Action (P) can be computed by replacing O_1 in (1) with O_2. Finally, the vector U or event vector that represents the relationship between vector R_1 and R_2 is computed as follows:

$$U = f(R_1^T \cdot T_3^{[1:k]} \cdot R_2 + W \begin{bmatrix} R_1 \\ R_3 \end{bmatrix} + b) \quad (2)$$

The Neural Tensor Network is trained by the proposed loss function from [9], as seen below:

$$Loss(E, E^r) = max\big(0, 1 - f(E) - f(E^r)\big) + \lambda \|\phi\|_2^2 \quad (3)$$

where the event tuple (E) is denoted as (O_1, P, O_2), the corrupted event (E^r) generated by randomly replacing the actor is denoted as (O_1^R, P, O_2), λ is L_2 regularization parameter and $\phi = (\mathrm{T_1}, \mathrm{T_2}, \mathrm{T_3}, \mathrm{W}, \mathrm{b})$ is the set of parameters. The training algorithm is described in Fig. 3.

Algorithm 1: Event Embedding Training Process

Input: $\mathcal{E} = (E_1, E_2, \cdots, E_n)$ a set of event tuples; the model $EELM$

Output: updated model $EELM'$

random replace the event argument and got the corrupted event tuple
$\mathcal{E}^r \leftarrow (E_1^r, E_2^r, \cdots, E_n^r)$
while $\mathcal{E} \neq [\,]$ **do**
  $loss \leftarrow max(0, 1 - f(E_i) + f(E_i^r) + \lambda \|\mathbf{\Phi}\|_2^2$
  if $loss > 0$ **then**
    $Update(\mathbf{\Phi})$
  else
    $\mathcal{E} \leftarrow \mathcal{E}/\{E_i\}$
return $EELM$

Figure 3. Event embedding training process [9]

Using this approach, news headlines describing similar events are encoded into similar event vectors even if they do not share common words. Examples of event embedding are illustrated in Fig. 4.

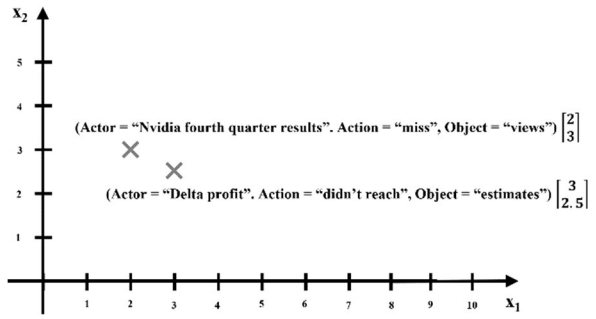

Figure 4. Examples of event vectors

B. Proposed Model

The proposed model in this work was modified from Ding's model [9] by adding numerical data as input to improve the prediction performance. The input used for this model is divided into two parts.

The first part is the average event embedding vectors that are extracted from daily news headlines. The event vector input is separated into three parts: (i) long-term events represented by the past 30-days event vectors, (ii) mid-term events as past 7-days event vectors, and (iii) short-term events as 1-day event vectors. Long-term and mid-term event vectors are fed into 1D CNN and max-pooling layers to extract feature maps that represent the most important events during these periods. The feature maps generated from the past 7 and 30 days event vectors are then concatenated with short-term event vectors to feed the concatenated event vectors into hidden layers. Output from the hidden layers is connected to two perceptrons that use softmax as the activation function. Output of the softmax function is a prediction based on event vectors.

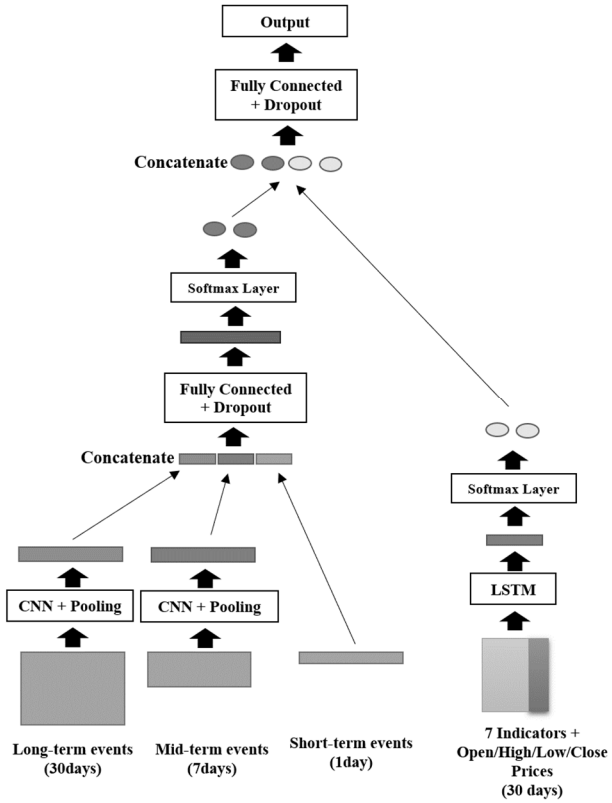

Figure 5. Proposed model

The second part is the numerical input as a vector that represents the past 30-days of seven indicators and OHLC prices. This vector is fed into the LSTM network, which is widely used for time series input. The output from LSTM is fed into hidden layers that are connected with two perceptrons that use a softmax activation function. The output of the softmax layer can imply a prediction based on technical indicators and price data.

Finally, we concatenate all output from the softmax layers and feed them into the final hidden layers to predict the following day's stock market movement. The output of the model is a binary classification, where class +1 represents that the stock price will increase, and class -1 represents that the stock price will decrease. The structure of the proposed model is shown in Fig. 5.

IV. Experimental Setup

A. Datasets

Data input used in this experiment consists of historical prices and news headlines. We extract the historical price data from Yahoo! Finance website. The following stock indexes are used to represent the stock market:

- Standard & Poor's 500 Index (S&P500) which is the U.S. stock market index based on market capitalizations of 500 large companies.
- Dow Jones Industrial Average (DJIA) which represents a price-weighted average of 30 significant stocks traded on the New York Stock Exchange and the Nasdaq.

For new headlines, we use information from three sources as follows:

- Financial news headlines from Reuters during 20 October 2006 to 19 November 2013. We use data published by [8] and select only headlines that contain company names or abbreviations.
- News headlines from Reddit WorldNews Channel (/r/world news) from 8 June 2008 to 1 July 2016. Reddit users vote to rank the headlines and only the top 25 are considered for a single date. This data is available on the Kaggle website [17].
- News headlines of 30 companies that are components of Dow Jones Industrial Average from 1 August 2016 to 12 December 2017. We obtain this dataset from a financial data provider, namely Intrinio.

We combine historical prices from Yahoo! Finance and news headlines from different sources and then split the three datasets, as summarized in Table 2.

TABLE II. Summary of Information on Datasets

Information	**Dataset #1**	**Dataset #2**	**Dataset #3**
Historical Prices	S&P 500	DJIA	DJIA
News Sources	Reuters	Reddit	Intrinio
# of Headlines	27,158	49,725	138,516
Splitting Periods			
- *Training*	20/10/06 – 18/06/12	8/08/08 – 31/12/12	1/08/16 – 14/07/17
- *Validation*	19/06/12 – 21/02/13	2/01/13 – 31/12/14	15/07/17 – 30/09/17
- *Test*	22/02/13 – 19/11/13	2/01/15 – 1/07/16	1/10/17 – 12/12/17

B. Baseline Models

In this study, the following baseline models are implemented for performance comparison:

(i) Event Only: the proposed model from [9] that uses only event embedding vectors as input.

(ii) News and Indicators: we modify the proposed model from [12], which takes both news headlines and technical indicators as input. Instead of using seven technical indicators as numerical input, we also add the historical OHLC price data into the model.

(iii) News Only: the modified model from (ii) by removing technical indicator part.

(iv) Indicator Only: the modified model from (ii) by removing news headlines part.

C. Performance Evaluation

In this paper, we use the following metrics for performance evaluation:

- Accuracy

The accuracy of the model is computed as:

$$Accuracy = \frac{number\ of\ correct\ predictions}{number\ of\ total\ predictions} \quad (5)$$

- Trading Simulation

We simulate daily stock trading following the strategy proposed by Lavrenko et al. [18] which is given by:

$$Action(t) = \begin{cases} buy \rightarrow sell,\ when\ predicted\ price = +1 \\ sell \rightarrow buy,\ when\ predicted\ price = -1 \end{cases} \quad (6)$$

where $buy \rightarrow sell$ denotes a transaction that purchases stocks at the opening price and the stock is sold at the closing price. We apply the same strategy for short selling. $sell \rightarrow buy$ denotes a transaction that sells stocks at the opening price and the stock is bought at the closing price. The transaction cost is assumed to be 0.1% of daily traded volume.

Daily profits or losses are accumulated until the end of the simulation period and the final result is presented as an annualized return, which is computed by:

$$Annualized\ Return = \left(\frac{Final\ Balance}{Initial\ Balance}\right)^{\frac{365}{\#SimulationDay}} - 1 \quad (7)$$

D. Hyperparameters and Training

We apply a modern approach, namely Batch Normalization [19] to our deep neural networks. This approach can help to accelerate deep network training by reducing internal covariate shift. The Dropout technique [20] is also used to prevent overfitting. Our deep neural network is trained using Adaptive Moment Estimation (Adam) algorithm [21] with the initial learning rate of 0.001. The grid search technique is used for hyperparameters tuning. The optimal hyperparameters used in each neural network layer based on this experiment are summarized in Table 3.

TABLE III. SUMMARY OF HYPERPARAMETERS

Layer	Hyperparameter
Convolutional 1	(128 filters, kernel size = 3)
Convolutional 2	(128 filters, kernel size = 3)
Fully Connected 1	256
Dropout 1	0.2
LSTM	128
Fully Connected 2	256
Dropout 2	0.2

V. EXPERIMENT RESULTS

We perform the study using the news headlines and historical price data described in Section IV. Data are split into three parts comprising training data, validation data, and test data. We train the models using training data and then select the final models based on maximum accuracy of the validation data. Finally, the performance of each model is evaluated using the test data.

A. The Effectiveness of Event Embedding

We measure the effectiveness of event embedding by comparing the accuracy among model #1 to #3 in Table 4. These models use only one type of input as OHLC prices and technical indicators, news headlines, and event embedding, respectively.

As shown in Table 4, using event embedding as text representation gives better results in dataset #1 and dataset #3. This observation is consistent with [9]. For dataset #2, the result is worse because of the nature of data as general news form, which it is difficult to extract significant financial events. Moreover, the embedding vectors derived from regular news headlines may not be suitable for stock prediction.

TABLE IV. ACCURACY ON TEST DATA

Model	Dataset #1	Dataset #2	Dataset #3	
#1: Indicator Only	56.59%	**52.38%**	63.01%	
#2: News Headline Only	58.14%	50.53%	63.01%	
#3: Event Only	58.53%	48.94%	64.38%	
#4: News Headline and Indicator	59.69%	51.85%	65.75%	
#5: Event and Indicator (Proposed)	**62.02%**	50.26%	**69.86%**	
Change in Accuracy (model #5 minus other models)				**Average**
#1: Indicator Only	5.43%	-2.12%	6.85%	3.39%
#2: News Headline Only	3.88%	-0.27%	6.85%	3.49%
#3: Event Only	3.49%	1.32%	5.48%	3.43%
#4: News Headline and Indicator	2.33%	-1.59%	4.11%	1.62%

B. The Effectiveness of Adding Numerical Information

The impact of using news headlines with technical indicators is measured by comparing model #4 against model #2. Our experiment shows better prediction accuracy for all datasets. This observation is also consistent with [12], who performed experiments based on news headlines and technical indicators.

Comparing our proposed model with model #3, it is apparent using both event embedding and numerical information improves prediction performance, especially for dataset #3, where accuracy increased significantly from 64.38% to 69.86%.

This experiment shows that the accuracy of stock market prediction can be improved by considering both numerical and textual information as input into a deep neural network.

C. Annualized Return Based on Trading Simulation

As shown in Table 5, results are positive for dataset #1, while others are negative. However, results between datasets should not be compared since they are based on different market conditions. Thus, we compare the results only for models that use the same dataset. Even though we get a negative return, our experiment shows that adding numerical information improves the annualized return of the models. This observation is evaluated by comparing model #5 against model #3 and model #4 against model #2.

TABLE V. ANNUALIZED RETURN BASED ON TRADING SIMULATION

Model	Dataset #1	Dataset #2	Dataset #3
#1: Indicator Only	4.04%	-23.78%	-12.07%
#2: News Headline Only	1.33%	-24.79%	-13.56%
#3: Event Only	2.37%	-34.45%	-10.20%
#4: News Headline and Indicator	10.02%	**-23.12%**	-11.05%
#5: Event and Indicator (Proposed)	**21.44%**	-33.76%	**-9.75%**

The negative annualized return might come from the trading strategy in this work, which makes buy or sell transactions every day. This strategy can lead to high transaction costs and negative returns when the capital gain from price movements is less than the trading cost. Moreover, the final models are selected based on maximum prediction accuracy on validation data, which could also lead to negative profit if the model makes accurate predictions during small price movement periods and makes wrong predictions when prices change significantly.

VI. CONCLUSION

In this paper, we aim to improve the performance of stock market prediction using event vectors and numerical information as input to a deep neural network. Our proposed model consists of CNN and LSTM architectures and uses event embedding vectors extracted from news headlines, historical prices and technical indicators as input. We conduct experiments on three datasets and employ accuracy and trading simulation as performance metrics. On average, the proposed model gives better prediction accuracy than all baseline models. Furthermore, annualized return based on trading simulation gives better results for considering both types of input yield.

For future studies, the deep neural network structure should be revised for suitability in analyzing both types of information. Introducing Recurrent Neural Networks and attention mechanisms into textual input may further improve prediction accuracy.

REFERENCES

[1] B. G. Malkiel, A random walk down Wall Street: the time-tested strategy for successful investing. WW Norton & Company, 2007.

[2] H. Mizuno, M. Kosaka, H. Yajima, and N. Komoda, "Application of neural network to technical analysis of stock market prediction," Studies in Informatic and Control, vol. 7, no. 3, pp. 111-120, 1998.

[3] W. Leigh, R. Purvis, and J. M. Ragusa, "Forecasting the NYSE composite index with technical analysis, pattern recognizer, neural network, and genetic algorithm: A case study in romantic decision support," Decision Support Systems, vol. 32, no. 4, pp. 361-377, 2002.

[4] H. Gunduz and Z. Cataltepe, "Borsa Istanbul (BIST) daily prediction using financial news and balanced feature selection," Expert Systems with Applications, vol. 42, no. 22, pp. 9001-9011, 2015.

[5] B. Wang, H. Huang, and X. Wang, "A novel text mining approach to financial time series forecasting," Neurocomput., vol. 83, pp. 136-145, 2012.

[6] R. P. Schumaker and H. Chen, "Textual analysis of stock market prediction using breaking financial news: The AZFin text system," ACM Transactions on Information Systems, vol. 27, no. 2, 2009, Art. no. a12.

[7] G. Gidofalvi, "Using news articles to predict stock price movements," Department of Computer Science and Engineering, University of California, San Diego, 2001.

[8] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Using structured events to predict stock price movement: An empirical investigation," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1415-1425.

[9] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," presented at the IJCAI International Joint Conference on Artificial Intelligence, 2015.

[10] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in Neural Networks (IJCNN), 2017 International Joint Conference on, 2017, pp. 1419-1426: IEEE.

[11] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in Proceedings of the 15th International Conference on Computer and Information Science (ICIS), 2016, pp. 1-6.

[12] M. R. Vargas, B. S. L. P. De Lima, and A. G. Evsukoff, "Deep learning for stock market prediction from financial news articles," in Proceedings of the 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2017, pp. 60-65.

[13] H. D. Huynh, L. M. Dang, and D. Duong, "A New Model for Stock Price Movements Prediction Using Deep Neural Network," in Proceedings of the Eighth International Symposium on Information and Communication Technology, 2017, pp. 57-62: ACM.

[14] Y. Zhai, A. Hsu, and S. K. Halgamuge, "Combining news and technical indicators in daily stock price trends prediction," in International symposium on neural networks, 2007, pp. 1087-1096.

[15] Stanford Open Information Extraction. Available: https://nlp.stanford.edu/software/openie.html

[16] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 2014, pp. 1532-1543.

[17] Daily News for Stock Market Prediction. Available: https://www.kaggle.com/aaron7sun/stocknews/

[18] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan, "Mining of concurrent text and time series," in KDD-2000 Workshop on Text Mining, 2000, vol. 2000, pp. 37-44.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.