J Inf Process Syst, Vol.15, No.5, pp.1231~1242, October 2019
https://doi.org/10.3745/JIPS.02.0121

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Two-Dimensional Attention-Based LSTM Model for Stock Index Prediction

Yeonguk Yu* and Yoon-Joong Kim*

Abstract

This paper presents a two-dimensional attention-based long short-memory (2D-ALSTM) model for stock index prediction, incorporating input attention and temporal attention mechanisms for weighting of important stocks and important time steps, respectively. The proposed model is designed to overcome the long-term dependency, stock selection, and stock volatility delay problems that negatively affect existing models. The 2D-ALSTM model is validated in a comparative experiment involving the two attention-based models multi-input LSTM (MI-LSTM) and dual-stage attention-based recurrent neural network (DARNN), with real stock data being used for training and evaluation. The model achieves superior performance compared to MI-LSTM and DARNN for stock index prediction on a KOSPI100 dataset.

Keywords

Attention Mechanism, LSTM, Stock Index Prediction, Two-Dimensional Attention

1. Introduction

Predictions of stock prices are a challenge due to the fact that stock prices are non-stationary and non-linear at a certain point in time. In recent years, several studies have applied deep learning approaches to solve the challenge with promising results. The main deep learning models applied to predict stock volatility are convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), and attention architecture, which are commonly utilized in the areas of image processing and neural machine translation.

Cao and Wang [1] exemplified the use of CNN in this area by proposing a stock price forecasting model based on a modified CNN. Further, LSTM has become the state-of-the-art model for sequential data application. Greff et al. [2] investigated and analyzed its various variants and concluded that there were no better variants than the standard LSTM. Saad et al. [3] exploited time delay, recurrent, and probabilistic neural networks for predictability analysis, and concluded that all the networks are feasible. As regards attempts to add financial news or refine data features, Berradi and Lazaar [4] used feature-dimensionality reduction to enhance the accuracy of the RNN model, which resulted in good prediction for stock prices. Minh et al. [5] used both financial news and a sentiment dictionary to predict the directions of stock prices based on the modified gated RNN. Several researchers have also suggested new

Manuscript received July 1, 2019; first revision August 19, 2019; accepted September 4, 2019.
Corresponding Author: Yoon-Joong Kim (yjkim@hanbat.ac.kr)
* Dept. of Computer Engineering, Hanbat National University, Daejeon, Korea (ryk012@naver.com, yjkim@hanbat.ac.kr)

ideas, and modified the structure of LSTM to accommodate those ideas [5,6]. In one such case, Baek and Kim [6] devised an overfitting prevention LSTM module and a prediction LSTM module.

In the above stock price prediction model based on LSTM, the stock price sequence is transformed into the temporal context in a single fixed-length vector, which is fed to the network to produce the predicted stock price of the next day. The performance of the model varies depending on how much information the vector can capture from the source sequence. However, the data on the front-end of the sequence is diminished in the vector representation as the sequence enlarges, which results in decreasing performance—which is commonly referred to as the "long-term dependency problem (LTDP)". To overcome this problem, researchers have proposed an attention mechanism for neural machine translation (NMT) [7], which extends to various fields including stock-related applications [8-10].

Liu and Wang [8] suggested an attention-based method using news and numerical data in stock market prediction. Li et al. [9] identified the stock selection problem (SSP), which is the issue whereby selection of stock prices weakly correlated with the target stock price, can degrade the prediction performance. They then proposed the attention-based multi-input LSTM (MI-LSTM), which can select important stocks from positively and negatively correlated stocks and demonstrated that the MI-LSTM models outperform other stock prediction models using LSTM. Furthermore, Qin et al. [10] proposed the dual-stage attention-based RNN (DARNN), drawing inspiration from the encoder-decoder structure used in machine translation. The DARNN model predicts the stock index of the next day using past stock indexes and prices as input. This model consists of an attention-based encoder and decoder. The attention-based encoder is composed of an LSTM and input attention mechanism that applies attention weights to all stock according to their importance. The encoder output is taken as the attention-based decoder input. The attention-based decoder is composed of an LSTM and temporal attention mechanism that applies attention weights across all time steps for selection of relevant time steps. In other words, Qin et al. [10] proposed the use of an attention-based encoder and decoder to mitigate the problems of stock selection and long-term dependency by allowing the model to focus on informative stocks and times steps. They also demonstrated that their DARNN model outperforms traditional RNN models for stock index prediction.

The occurrence of a stock-market-related event can affect stock volatility and is reflected in stock prices, and the extent and timing of the changes may vary depending on the stock. We refer to this characteristic of stock-dependent event-driven volatility as the stock volatility delay problem (SVDP). To solve this problem, a stock price prediction model must learn the corresponding temporal change patterns and employ them for prediction. However, DARNN [10] cannot solve the stock volatility delay problem, as the temporal attention mechanism uses the output of the input attention mechanism that applies attention weights for selection of relevant stock. Specifically, the output of the input attention mechanism is modified according to the stock importance; therefore, in the output, the temporal information that reflects the stock volatility pattern is distorted. To prevent this distortion, during calculation of the temporal-attention attention weights, it is necessary to use original stock information for which the stock change pattern has not been distorted by weight-based stock selection.

Therefore, in this paper, we propose the two-dimensional attention-based LSTM (2D-ALSTM), which allows the same source sequence to be used for calculating the alignment scores of both the input attention and temporal attention mechanisms. Thus, this model can overcome the stock selection, long-term dependency, and stock volatility delay problems, which are negatively affected in attention-based models such as MI-LSTM and DARNN.

The remainder of this paper is organized as follows. In Section 2, we introduce traditional LSTM, the attention mechanism, and the problem definition. In Section 3, we present and describe our proposed model. In Section 4, we report the results of a 2D-ALSTM model evaluation experiment, which are compared with results for MI-LSTM and DARNN. Finally, Section 5 gives the conclusion.

2. Preliminaries

In this section, we introduce the LSTM and attention mechanism, which form the foundation of all models considered in this study, i.e., MI-LSTM, DARNN, and 2D-ALSTM. Then, we define the prediction problem we consider for model evaluation.

2.1 Long Short-Term Memory

LSTM is one of the most commonly employed RNNs for time series prediction. The LSTM cell state h_t and hidden state C_t are controlled by three gates: the output, input, and forget gates. For a given input time sequence $X = (x_1, x_2, \cdots, x_T)$, where $x_t \in \mathbb{R}^m$ represents the input at time step t and m is the input size, the LSTM updates states are as follows:

$$h_t = \tanh(C_t) * o_t \tag{1}$$

$$C_t = C_{t-1} * f_t + \tilde{C}_t * i_t \tag{2}$$

$$\tilde{C}_t = tanh\left(W_j\,[h_{t-1};\, x_t\,] + b_j\right) \tag{3}$$

$$f_t = \sigma\left(W_f\,[h_{t-1}; x_t\,] + b_f\right) \tag{4}$$

$$i_t = \sigma(W_i\,[h_{t-1};\, x_t\,] + b_i) \tag{5}$$

$$o_t = \sigma(W_o\,[h_{t-1}; x_t\,] + b_o) \tag{6}$$

Here, h_t and $h_{t-1} \in \mathbb{R}^q$ are the cell states at time steps t and t–1, respectively; q is the number of LSTM hidden units; and $[h_{t-1};\, x_t\,] \in \mathbb{R}^{m+q}$ is a concatenation of h_{t-1} and x_t. Furthermore, W_f, W_i, W_j, and $W_o \in \mathbb{R}^{m\times(m+q)}$ are weight parameters to be learned, while b_f, b_i, b_j, and $b_o \in \mathbb{R}^q$ are bias parameters to be learned. Finally, σ represents the sigmoid function. For convenience, we use a simple function notation f_{lstm} to represent the LSTM calculation described in Eqs. (1)–(6):

$$h_t = f_{lstm}(h_{t-1}, x_t) \tag{7}$$

2.2 Attention Mechanism

The attention mechanism was proposed to solve the long-term dependency problem encountered for the sequence-to-sequence (S2S) model used in machine translation. The S2S model is composed of two LSTMs called the encoder and decoder. The encoder converts input sequences of various lengths into information expressed in a single context vector using the LSTM; the decoder then uses this context vector to produce output sequences. However, when the input sequence length becomes excessive, a long-term dependency problem arises for the S2S model, as the encoder does not properly transmit the front-

end information of the input sequence to the decoder. As noted above, use of an attention mechanism has been proposed to overcome the long-term dependency problem [7]. The attention mechanism causes the decoder to focus on the important encoder LSTM output when calculating its own output. Thus, the decoder input c_t at t is calculated based on the previous decoder cell state s_{t-1} and the encoder cell states H = $(h_1, h_2, \cdots, h_T)$ as follows:

$$c_t = \sum_{i=1}^{T} a_{t,i} h_i \tag{9}$$

$$a_{t,i} = \frac{\exp(score(s_{t-1}, h_i))}{\sum_{j=1}^{T} \exp(score(s_{t-1}, h_j))} \tag{10}$$

Thus, as shown in Eq. (9), c_t is the weighted sum of h_i. Note that Eq. (10) shows calculation of the attention weight $a_{t,i}$ using the softmax function and score function. With the help of the attention mechanism, the dependencies between source and target sequences are not restricted by the in-between distance. Consequently, the attention mechanism was soon extended into various fields, including that of stock price prediction [8-10].

2.3 Notation and Problem Statement

In this paper, we propose a model for time series prediction and evaluate this model using a stock index prediction task. For stock index prediction, we take a stock index sequence and stock price sequence as the model input. We refer to the stock index sequence as $\boldsymbol{y} = (y_1, y_2, \cdots, y_T) \in \mathbb{R}^T$, where T is the time window size. In addition, we refer to the stock price sequence as $\boldsymbol{X} = (\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_T}) \in \mathbb{R}^{n \times T}$, where n is the number of stocks. Hence, $\boldsymbol{x}_t = (x_t^1, x_t^2, \cdots, x_t^n) \in \mathbb{R}^n$ is the price sequence of all stock at time t, and $\boldsymbol{x}^k = (x_1^k, x_2^k, \cdots, x_T^k) \in \mathbb{R}^T$ is the price sequence of the k-th stock.

The goal of the model is to predict the stock index of the next day, for a given $\boldsymbol{y}$ and $\boldsymbol{X}$. We use $\bar{y}_{T+1}$ to represent the predicted real index of the next day given by the model, where y_{T+1}represents the actual value. Thus, our model can be defined as follows:

$$\bar{y}_{T+1} = F(\boldsymbol{y}, \boldsymbol{X}) \tag{11}$$

3. Proposed Model

To mitigate the performance degradation caused by problems such as stock selection, long-term dependency, and stock volatility, we propose a novel 2D-ALSTM model that consists of a CNN layer, an input and temporal attention layer, and an LSTM RNN layer, as shown Fig. 1. We use a CNN to transform the source stock price sequences to the effective feature sequences, which are appropriate for determining the 2D-attention (input and temporal attention mechanisms) attention weights. Finally, the context vector is calculated by the 2D-attention component, and we concatenate the context vector and index for use as the LSTM input. This process is described in more detail in the following subsections.

3.1 CNN of 2D-ALSTM

The CNN produces the feature matrix $\boldsymbol{D}$ for a given stock price sequences $\boldsymbol{X}$, as shown in Fig. 1. The

CNN with filter size 5 is applied to the k stock price sequences through the time axis to produce the feature d_t^k, which forms the stock feature sequence $\boldsymbol{d}^k$ and the temporal feature sequence $\boldsymbol{d}_t$, such that we obtain $\boldsymbol{D}$ as follows for all stock:

$$\boldsymbol{D} = (\boldsymbol{d}^1, \boldsymbol{d}^2, \dots, \boldsymbol{d}^{\mathrm{n}}) \tag{12}$$

$$\boldsymbol{d}^k = (d_1^k, d_2^k, \cdots, d_T^k) \in \mathbb{R}^T \tag{13}$$

$$\boldsymbol{d}_t = (d_t^1, d_t^2, \cdots, d_t^n) \in \mathbb{R}^n \tag{14}$$

$$d_t^k = (W_d\,[x_{t-2}^k;\, x_{t-1}^k;\, x_t^k;\, x_{t+1}^k;\, x_{t+2}^k] + b_d) \tag{15}$$

where $W_d \in \mathbb{R}^{1\times 5}$ is the weight parameter for training and $b_d \in \mathbb{R}^1$ is the bias.

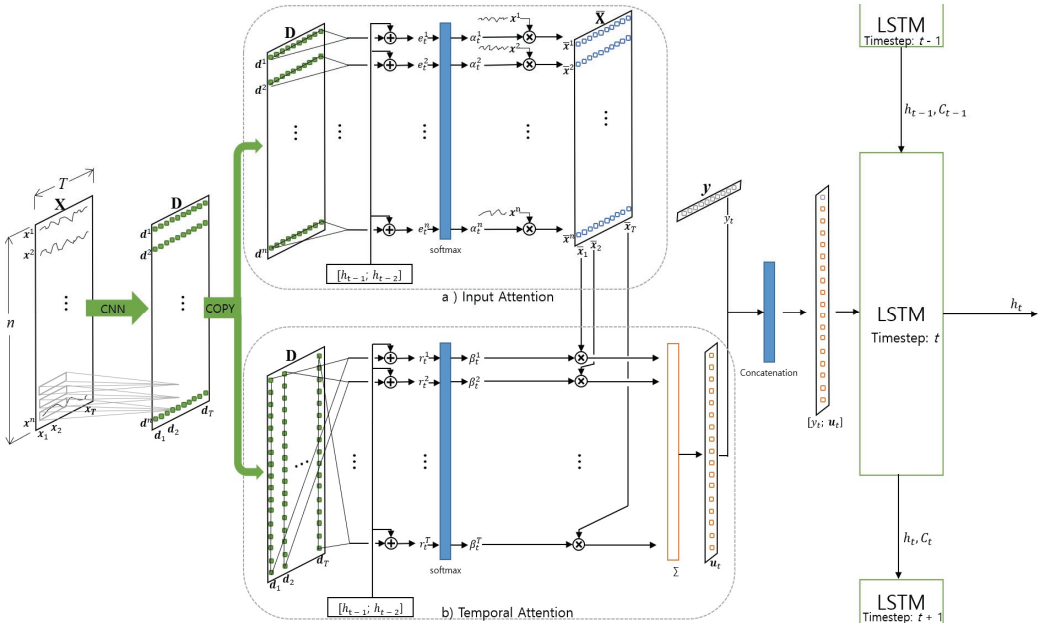

Fig. 1. Conceptual diagram of the 2D-ALSTM model composed of a CNN layer, an input and temporal attention layer, and an LSTM RNN layer.

3.2 Input Attention Mechanism of 2D-ALSTM

The input attention mechanism of the 2D-ALSTM is shown in the section labeled a) of Fig. 1. We use the stock feature sequence $\boldsymbol{d}^k$ and the previous LSTM cell states h_{t-1} and h_{t-2} as the alignment score function input. The score function applies linear regression to $\boldsymbol{d}^k$ and the concatenated state $[h_{t-1};\, h_{t-2}]$ and uses query vector v_e to calculate the alignment score e_t^k. Following calculation of the e_t^k values for all stock, the softmax function is applied to convert e_t^k to attention weight a_t^k. We obtain $\overline{\boldsymbol{X_t}}$, which is the input-attention output for all stock, by multiplying a_t^k by $\boldsymbol{x}^k$ as follows:

$$e_t^k = v_e^{\mathrm{T}} \tanh(W_e[h_{t-1}; h_{t-2}] + W_{indic}\boldsymbol{d}^k) \tag{16}$$

$$a_t^k = \frac{\exp(e_t^k)}{\sum_{j=1}^{n} \exp(e_t^j)}, 1 \le k \le n \tag{17}$$

$$\overline{\boldsymbol{X}}_t = (a_t^1 \boldsymbol{x}^1, a_t^2 \boldsymbol{x}^2, \cdots, a_t^n \boldsymbol{x}^n) \tag{18}$$

where $v_e \in \mathbb{R}^{\mathrm{T}}, W_e \in \mathbb{R}^{\mathrm{T}\times 2\mathrm{m}}$, and $W_{indic} \in \mathbb{R}^{\mathrm{T}\times \mathrm{T}}$ are the weight parameters to be trained. We omit the bias for succinctness.

The alignment score e_t^k indicates the important score of the k-th stock feature sequence $\boldsymbol{d}^k$ for the LSTM state h_t. As the normalized score a_t^k is a probability variable of the importance probability distribution over the stocks, the input-attention output $\overline{\boldsymbol{x}}^k$ represents the updated stock price of the k-th stock $\boldsymbol{x}^k$ weighted by the importance score a_t^k, and thus constitutes data for solution of the stock selection problem.

3.3 Temporal Attention Mechanism of 2D-ALSTM

We use the temporal attention to obtain the context vector $\boldsymbol{u_t}$, as shown in the section label 'b' in Fig. 1. We use the temporal feature sequence $\boldsymbol{d}_i$, h_{t-1} and h_{t-2} for linear regression. We perform a query with query vector v_r to obtain the alignment score r_t^i, which indicates the important score of input $\boldsymbol{d}_i$ at time i for the LSTM state h_t. Then, the softmax function is applied to obtain the attention weight β_t^i, which represents the probability variable of the importance probability distribution over the time. Finally, we obtain the context vector $\boldsymbol{u_t}$ for the LSTM state h_t by multiplying β_t^i by $\overline{\boldsymbol{x}}_i$ of the input-attention output at time i. This procedure can be summarized as follows:

$$r_t^i = v_r^{\mathrm{T}} \tanh(W_e[h_{t-1}; h_{t-2}] + W_{time}\boldsymbol{d}_i) \tag{19}$$

$$\beta_t^i = \frac{\exp(r_t^i)}{\sum_{j=1}^{T} \exp(r_t^j)} \tag{20}$$

$$\boldsymbol{u_t} = \sum_{j=1}^{T} \beta_t^j \overline{\boldsymbol{x}}_j \tag{21}$$

where $v_r \in \mathbb{R}^n, W_r \in \mathbb{R}^{\mathrm{n}\times 2\mathrm{m}}$, and $W_{time} \in \mathbb{R}^{\mathrm{n}\times \mathrm{n}}$ are the weight parameters for training. We omit the bias for succinctness.

The role of the alignment score r_t^i is to allow the model to compute the importance score of i-th temporal feature sequence $\boldsymbol{d}_i$ at time i for the LSTM state h_t from the target perspective. This approach differs from that of the DARNN model, which uses the input-attention output $\overline{\boldsymbol{x}}_i$ instead of $\boldsymbol{d}_i$. As $\overline{\boldsymbol{X_t}}$ corresponds to the weighted $\boldsymbol{X}$ for selection of important stock and is distorted along the time axis, it is unsuitable as input for learning the stock change pattern of the original sequence $\boldsymbol{X}$. Therefore, we use $\boldsymbol{d}_i$, which is the temporal feature sequence produced from $\boldsymbol{X}$, to produce r_t^i in our proposed method.

As we consider the time and stock axes to obtain $\boldsymbol{u_t}$, use of $\boldsymbol{u_t}$ overcomes the long-term dependency problem, because stock prices are weighted according to the importance of each data of sequence regardless of distance. In addition, $\mathbf{u_t}$ overcomes the stock volatility delay problem by selecting important times without using the temporally distorted input-attention output.

3.4 LSTM of 2D-ALSTM

After $\boldsymbol{u_t}$ is produced by the 2D-attention component, it is concatenated with the stock index at time t,

i.e., y_t, and used as the LSTM input to produce the 2D-ALSTM output h_t. Therefore, for a given $\boldsymbol{X}$ and $\boldsymbol{y}$, 2D-ALSTM uses the CNN, input attention mechanism, temporal attention mechanism, and LSTM to produce h_t. We can summarize the 2D-ALSTM procedure as follows:

$$\boldsymbol{d}^k, \boldsymbol{d}_i = CNN(\boldsymbol{X}) \tag{22}$$

$$\bar{\boldsymbol{X}}_t = input\text{-}attention(\boldsymbol{h}_{t-2}, \boldsymbol{h}_{t-1}, \boldsymbol{d}^k, \boldsymbol{X}) \tag{23}$$

$$\boldsymbol{u}_t = temporal\text{-}attention(\boldsymbol{h}_{t-2}, \boldsymbol{h}_{t-1}, \bar{\boldsymbol{X}}_t, \boldsymbol{d}_i) \tag{24}$$

$$\boldsymbol{h}_t = f_{lstm}(h_{t-1}, [\boldsymbol{u}_t; y_t]) \tag{25}$$

$$\boldsymbol{H} = (\boldsymbol{h_1}, \boldsymbol{h_2}, \dots, \boldsymbol{h_T}) \tag{26}$$

We define a function $2D\text{-}ALSTM$ to represent the 2D_ALSTM described by Eqs. (22)–(26). As the parameter input $\boldsymbol{X} \in \mathbb{R}^{n\times T}$ and the output $\mathbf{H} \in \mathbb{R}^{p\times T}$ have the same length, i.e., T, the following multi-layer structure is possible:

$$\boldsymbol{H}^0 = 2D\text{-}ALSTM(\mathbf{X}, \mathbf{y}) \tag{27}$$

$$\boldsymbol{H^z} = 2D\text{-}ALSTM(\boldsymbol{H^{z-1}}, \mathbf{y}),\ 1 \le z \le L \tag{28}$$

where L is the number of layers.

In the experiments performed in this study, we used three 2D-ALSTM layers, with the last cell state $\boldsymbol{h}_T^3$ of the last layer $\boldsymbol{H^3}$ being taken as the linear regression layer input for the final prediction $\bar{y}_{T+1}$:

$$\bar{y}_{T+1} = W_y \boldsymbol{h}_T^3 + b_y \tag{29}$$

where $W_y \in \mathbb{R}^{1\times p}$ and $b_y \in \mathbb{R}^1$ are weight parameters to be learned.

3.5 Loss Function

In our proposed method, we use the root mean-squared error (RMSE) as the loss function for training. The loss function for the mini batch $\mathrm{M} = (\boldsymbol{X}^i, \boldsymbol{y}^i), 1 \le i \le N$ can be defined as follows:

$$\text{loss} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\bar{y}_{T+1}^i - y_{T+1}^i)^2} \tag{30}$$

We use the Adam optimizer for the model weight parameter training.

4. Experiments

In this section, we describe the experiments performed for comparison of the 2D-ALSTM model performance with those of existing models. First, we describe the dataset used for model training and evaluation. Second, we describe the parameter settings of the 2D-ALSTM model and the other baseline models. Third, we report and discuss the experiment results.

All experiments were conducted using the TensorFlow deep learning framework, and an NVIDIA GeForce GTX 1070 GPU was used to speed up the model training.

4.1 Datasets

We used two datasets, KOSPI100 and NASDAQ100, as described in Table 1.

Table 1. Datasets for experiment

Dataset	Indicator	Training set	Validation set	Test set
KOPSI100	85	1791	0	199
NASDAQ100	81	28378	6081	6091

The first dataset was the KOSPI100 dataset obtained from the KOSPI stock market. The KOSPI100 dataset consists of the KOSPI100 index and the 100 stock prices used for index calculation. The stock closing prices were used as the model stock price sequence and the index closing prices were used as the model index sequence. The dataset frequency was day-by-day. The index and stock closing prices spanned the period from June 10, 2010, to July 16, 2018, corresponding to a total of 2000 open market days. Only stock for which all data were available for the full 2000 days were selected; thus, a total of 85 stock were selected from the original 100. Samples were created using the obtained 2000-day stock and index data. Each sample incorporated input sequences and a target index value for model training and evaluation. For a fixed time window of size $T = 10$ and a stride of 1, we generated 1990 samples from the KOSPI100 dataset. We used 1791 samples for training and the remaining 199 samples for evaluation (Table 1).

The second dataset was the NASDAQ100 dataset obtained from the NASDAQ stock market. The NASDAQ100 dataset consists of the NASDAQ100 index and the 100 stock prices used for index calculation. Note that this identical dataset was also employed in the study in which DARNN was proposed [10]. As for the KOSPI100 dataset, the stock and index closing prices were used as the stock price sequence and index sequence of the model, respectively. The dataset frequency was minute-by-minute. The index and stock closing prices spanned the period from July 26, 2016, to December 22, 2016, corresponding to a total of 105 days. The data for each day contained 390 index and 390 stock prices, excluding November 25 and December 22. Samples were created using the obtained 50560-minute-long stock and index datasets. Each sample incorporated input sequences and a target index value for model training and evaluation. For a fixed $T = 10$ and a stride of 1, we generated 50,550 samples from this NASDAQ100 dataset. We set the training:validation:evaluation ratio to 7:1.5:1.5. Therefore, we used 28378, 6081, and 6091 samples for training, validation, and evaluation, respectively. All data from both datasets were normalized to between 0 and 1 using min-max scaling.

4.2 Parameter Settings

Our 2D-ALSTM can be implemented as a multi-layer structure; thus, the value of L must be decided. In this study, we employed three 2D-ALSTM layers because this structure yielded the best result for the validation samples obtained from the NASDAQ100 dataset. We set the hidden unit size to 24 for all three layers and T to 10, as for DARNN and MI-LSTM. In the experiments of previous studies [9,10] for DARNN and MI-LSTM, T values of 10 yielded the best results. In addition, the parameter settings for these baseline models were selected according to the best experiment results. When the MI-LSTM model

was used for stock price prediction, the Pearson correlation coefficient was used to select positively and negatively correlated stocks. Therefore, in our experiments, we employed two models: MI-LSTM_PN, which used the positively and negatively correlated stocks, and MI-LSTM, which did not use those stocks.

The numbers of weight parameters for the 2D-ALSTM, DARNN, MI-LSTM_PN, and MI-LSTM models are listed in Table 2. Although our 2D-ALSTM model contained three layers, it had fewer weight parameters than the other models because it had a small hidden unit size of 24 (the hidden unit size was 64 for all other models).

Table 2. Numbers of trainable weight parameters (without bias parameters) of considered models

	MI-LSTM	MI-LSTM_PN	DARNN	2D-ALSTM
Trainable weight parameters	74112	90496	77288	35924

4.3 Results

We conducted five experiments. In each experiment, the models were trained and then evaluated using the loss function presented in section 3.5. The experiment results for all models are presented in Table 3. Note that the loss values for the KOSPI100 and NASDAQ100 datasets are equal to the corresponding values listed in Table 3 multiplied by 10^{-2} and 10^{-3}, respectively.

All models were trained and evaluated five times in the same environment. For training with the KOSPI100 dataset, we used a mini batch size of 1791, equivalent to the number of training samples, and we set the learning rate to 0.01. The training was repeated until no further decreases in loss were observed for the test samples, with repetition over 300 epochs. For training with the NASDAQ100 dataset, we used a mini batch size of 1024 and set the learning rate to 0.01. The training was repeated until there were no further decreases in loss for the test samples, with repetition over 30 epochs.

As detailed in Table 3 and Fig. 2, the 2D-ALSTM model exhibited the best performance in 9 of 10 experiments. Comparing the average performance of each model for each dataset, 2D-ALSTM exhibited prediction RMSE, MAE (mean absolute error), and MAPE (mean absolute percentage error) increases of 3.83%, 6.64%, and 6.64% and 1.06%, 1.50%, and 1.44% for the KOSPI100 and NASDAQ100 datasets, respectively, compared to the second-best-performing model, DARNN. The MI-LSTM and MI-LSTM_PN models exhibited lower performance than the DARNN and 2D-ALSTM models. This seems to have been because MI-LSTM was created for stock price prediction and generates a time step weight for the LSTM output sequence using a relatively simple attention mechanism compared to the other models.

Table 3. Average performance comparison of models MI-LSTM, MI-LSTM_PN, DARNN, and 2D-ALSTM

Models	KOSPI100 dataset			NASDAQ100 dataset		
	MAE	MAPE	RMSE	MAE	MAPE	RMSE
MI-LSTM	1.709	2.059	2.225	6.030	10.321	8.551
MI-LSTM_PN	2.179	2.612	2.794	5.033	8.418	7.196
DARNN	1.686	2.023	2.169	3.244	5.709	5.062
2D-ALSTM	**1.581**	**1.897**	**2.089**	**3.196**	**5.628**	**5.009**

The best score indicated by boldface.

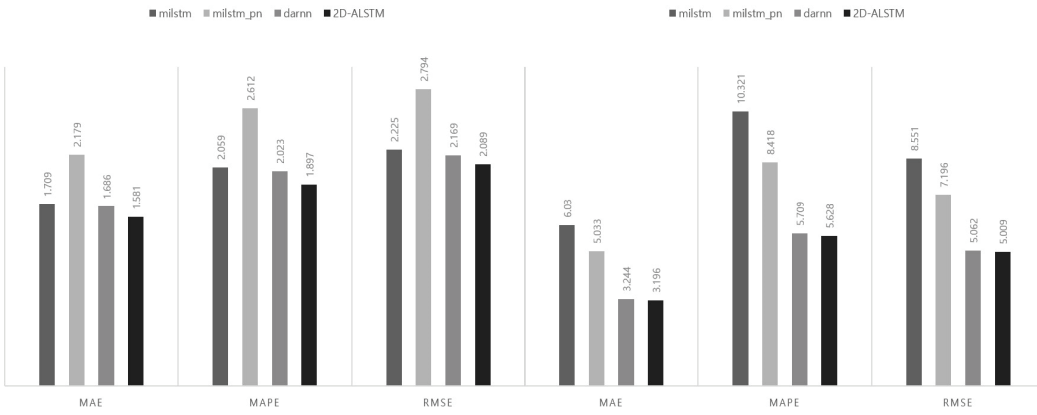

Fig. 2. Performance comparison of models MI-LSTM, MI-LSTM_PN, DARNN, and 2D-ALSTM: the experimental results with KOSPI 100 dataset (a) and NASDAQ100 dataset (b).

In the experiments with the KOSPI100 dataset, the proposed 2D-ALSTM model exhibited approximately 6.64% higher prediction MAE than the DARNN model, as noted above, although the number of weight parameters was only approximately 50% those of the other models. This result shows that the input attention mechanism of the 2D-ALSTM ameliorates the LSTM stock selection problem, and that the temporal attention of the 2D-ALSTM reduces the LSTM long-term dependency and stock volatility delay problems. These problems are not overcome by the MI-LSTM and DARNN models. In the experiments with the NASDAQ100 dataset, however, the 2D-ALSTM model performance was not significantly higher than that of DARNN. We believe this is because the stock volatility delay problem does not seem to arise for the NASDAQ100 dataset, which incorporates minute-by-minute data.

5. Conclusions

This paper presented a 2D-ALSTM model for stock index prediction, incorporating input attention and temporal attention mechanisms for the weighting of important stocks and important time steps, respectively. The proposed model is designed to overcome the long-term dependency, stock selection, and stock volatility delay problems that affect existing models. This model can not only be used for stock index prediction, but also has the potential to show great performance in other time series prediction tasks.

Recently, the attention-based model has achieved positive results in stock index prediction, as the attention mechanism has been suggested in the neural machine translation field. One particular attention-based model is the DARNN, which consists of input attention and temporal attention. The input attention allows model to focus on important stock prices rather than treating all the input stocks prices equally. The subsequent temporal attention allows the model to selectively focus on important times of the stock price pattern, but the temporal attention has the drawback of performance degradation due to the use of distorted temporal information caused by input attention. The proposed 2D-ALSTM model is designed to resolve the problem by using the original data, which are not distorted, to calculate the temporal-attention weight.

The 2D-ALSTM model is validated in a comparative experiment with previous models such as DARNN and MI-LSTM, with real stock data being used for training and evaluation, and achieves superior performance relative to other attention-based models for stock index prediction when applied to a KOSPI100 dataset.

Acknowledgement

This work was supported by the research fund of Hanbat National University in 2018.

References

[1] J. Cao and J, Wang, "Stock price forecasting model based on modified convolution neural network and financial time series analysis," *International Journal Communication and Systems*, vol. 32, no. 12, article no. e3987, 2019.

[2] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, 2016.

[3] E. W. Saad, D. V. Prokhorov and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1456-1470, 1998.

[4] Z. Berradi and M. Lazaar, "Integration of principal component analysis and recurrent neural network to forecast the stock price of Casablanca stock exchange," *Procedia Computer Science*, vol. 148, pp. 55-61, 2019.

[5] D. L. Minh, A. Sadeghi-Niaraki, H. D. Huy, K. Min, and H. Moon, "Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network," *IEEE Access*, vol. 6, pp. 55392-55404, 2018.

[6] Y. Baek and H. Y. Kim, "ModAugNet: a new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module," *Expert Systems with Applications*, vol. 113, pp. 457-480, 2018.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014 [Online]. Available: https://arxiv.org/abs/1409.0473.

[8] G. Liu and X. Wang, "A numerical-based attention method for stock market prediction with dual information," *IEEE Access*, vol. 7, pp. 7357-7367, 2019.

[9] H. Li, Y. Shen, and Y. Zhu. "Stock price prediction using attention-based multi-Input LSTM," in *Proceedings of the 10th Asian Conference on Machine Learning*, Beijing, China, 2018, pp. 454-469.

[10] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia, 2017, pp. 2627-2633.

Yeonguk Yu https://orcid.org/0000-0003-2147-4718

He is currently an undergraduate student in the Department of Computer Engineering of Hanbat National University. His research interests include artificial intelligence and deep learning.

Yoon-Joong Kim https://orcid.org/0000-0002-5451-5558

He received a B.S. degree in electronics engineering in 1981, an M.S. degree in computer engineering in 1983, and a Ph.D. degree in computer engineering in 1999 from Choongnam National University, Daejeon, Korea. He is a member of recognized scientific institutions such as the IEEE Computational Intelligence Society, the Korean Institute of Information Scientists and Engineers, Korea Information Processing Society, the Korean Institute of Communications and Information Sciences. Since 1984, he has been a full professor in the Department of Computer Engineering of Hanbat National University. He has conducted more than 20 government related projects in his field of expertise. He is an author of more than 30 articles in journals and conference proceedings. His research interests include artificial intelligence, deep learning, and speech recognition.