A Volume-Aware Positional Attention-Based Recurrent Neural Network for Stock Index Prediction

Xinpeng Yu
School of Electronics and Computer Engineering
Peking University
Shenzhen, China
yuxinpeng@pku.edu.cn

Dagang Li*
International Institute of Next Generation Internet
Macau University of Science and Technology
Macau, China
dagang.li@ieee.org

***Abstract*—With the rapid development of deep learning, more researchers have attempted to apply nonlinear learning methods such as recurrent neural networks (RNNs) and attention mechanisms to capture the complex patterns hidden in stock market trends. Most existing approaches to this task employ an attention mechanism that primarily relies on the information extracted from input features but fails to consider the other important factors (e.g., trading volume and position), which can potentially enhance these attention-based approaches. Motivated by the observation, we extend the attention mechanism with features needed for stock performance prediction in this article. Specifically, we propose a volume-aware positional attention-based recurrent neural network (VPA-RNN) for this task. First, we propose a generic method of adding position awareness to the attention mechanism. Next, the trading volume is incorporated into the original attention distribution to form a revised distribution. To evaluate the effectiveness of VPA-RNN, we collected real stock market data for stock indexes S&P 500 and DJIA, and the experimental results show that the proposed VPA-RNN can significantly outperform several existing highly competitive methods.**

Keywords-stock index prediction; recurrent neural network; attention mechanism; volume-aware attention; positional attention

I. INTRODUCTION

Stock performance prediction has received much attention due to its decisive role in stock investment, which aims to predict the future price or trend of stocks in order to achieve the maximum profit from stock investment. Various methods have been proposed to predict stock performance by many economic analysts and stock traders.

Li et al. [1] applied a quantile AR model to analyze the dynamics of stock index returns in China. In addition, the hidden Markov model (HMM) has been used to make nonlinear predictions of stock trends. Zhang et al. [2] presented an approach to predict stock market price trends based on a high-order HMM for the purpose of considering both short and long-term time dependence. However, such traditional solutions have apparent drawbacks, as they lack the capability of modeling the nonstationary and nonlinear nature of stock prices. To address this issue, many methods based on deep learning have been proposed to forecast stock prices in recent years. More researchers have attempted to apply deep learning methods such as multilayer perceptions (MLPs) [3] and recurrent neural networks (RNNs) [4]–[6] to capture the complex patterns hidden in market trends. Although the traditional RNN is capable of processing nonlinear data, it is not sufficient to model long-term dependence on a time series. This motivates the use of gated memory cells; thus, the famous long short-term memory (LSTM) network was proposed to better model long-term dependency on a time series and mitigate the vanishing gradient problem [7]. Accordingly, many studies employ the LSTM neural network in financial prediction [8]–[10].

However, if the time series is very long, LSTM also suffers from the problem of vanishing gradients which results in decreasing performance [11]. To overcome this problem, researchers have proposed the attention mechanism that achieved great success in various fields, including neural machine translation [12], speech recognition [13], and image processing [14]. Therefore, several recent works introduced an attention mechanism to stock-related applications [15]-[22]. Li et al. [16] proposed a multi-input LSTM (MI-LSTM) model, which can extract valuable information from low correlation factors and discard their harmful noise by employing additional input gates controlled by the convincing factors called mainstream. Furthermore, Qin et al. [17] proposed the dual-stage attention-based RNN (DA-RNN), drawing inspiration from the encoder-decoder structure used in machine translation. The DA-RNN model predicts the stock index of the next day using the previous values of stock indexes and constituent stock prices as input. This model consists of an encoder and a decoder. The encoder is composed of LSTM and an input attention mechanism that is used to adaptively extract the relevant features at each time step by referring to the previous encoder hidden state. The output of the encoder serves as the input of the decoder. The decoder is composed of LSTM and a temporal attention mechanism that is used to select the relevant encoder hidden states across all time steps. In this way, the DA-RNN model can not only adaptively select the most relevant input features but also capture the long-term temporal dependencies of a time series appropriately.

In the abovementioned attention-based stock price prediction model, the temporal attention mechanism primarily relies on the information extracted from input features but fails to consider the other important factors (e.g., trading volume and position), which can potentially enhance these attention-based approaches. Motivated by the observation, we extend the temporal attention

DOI reference number: 10.18293/SEKE2021-191

mechanism with features needed for stock performance prediction in this article. On the one hand, the position of the time step is a key factor in the stock performance prediction task. It is natural that the time steps closer to the predicted time step are more important. However, the abovementioned attention methods take no account of the effects of positions of different time steps, i.e., identical or very similar time steps are scored equally regardless of their positions in the sequence. Therefore, we introduce the positional attention mechanism to the task of stock performance prediction, which has achieved success in various fields, including natural language processing (NLP) [23,24] and speech recognition (SR) [25]. On the other hand, trading volume is also an important feature that provides valuable information, as past trading volume predicts both the magnitude and the persistence of future price momentum [26], i.e., the time step with higher trading volume is generally more important, and the attention mechanism should pay more attention to such time steps. Thus, inspired by several task-oriented attention mechanisms [27,28], we take advantage of this feature of stock performance prediction and propose volume-aware attention to incorporate the trading volume into the original attention distribution to achieve attention recalibration.

Combining these two gives a volume-aware positional attention-based recurrent neural network (VPA-RNN) with markedly better stock index prediction performance. To justify the effectiveness of the VPA-RNN, we compare it with the state-of-the-art approach using the S&P 500 dataset and the DJIA dataset. Our proposed VPA-RNN achieves the RMSE that is 6.80% and 47.83% lower than that of the best previous model DA-RNN [17], respectively.

II. Related theory and technology

In this section, we introduce the LSTM and the attention mechanism, which are the foundations of both the proposed model and the comparative models in this article.

A. Long short-term memory neural networks (LSTM)

Due to its memory blocks, the LSTM network [7] has a strong capability of capturing the long-term memory of sequential data with high prediction capability on chaotic time series. Hence, many related works adopt LSTM to learn long-term temporal dependencies from stock data time series [16,17]. For a similar reason, we also use LSTM in this paper. LSTM is a variant of RNNs that uses a gating mechanism to control the flow of information into or out of memory. For convenience, in this study, we use the function LSTM$(\cdot,\cdot,\cdot)$ as shorthand for the LSTM model in (1):

$$(h_t, c_t) = \text{LSTM}(x_t, h_{t-1}, c_{t-1}, W, b), \tag{1}$$

where *W* and *b* include all of the weight matrices and bias vectors, which are determined in the training process.

B. Attention mechanism

Based on recurrent neural networks, sequence-to-sequence models (S2S) have become popular due to their success in machine translation [29]-[31], which is composed of encoder and decoder. The encoder is used to convert the input sentences into a fixed-length vector and then used by the decoder to produce output sequences. However, encoder-decoder networks encounter the long-term dependency problem that their performance will deteriorate rapidly as the length of the input sequence increases. To resolve this issue, the attention mechanism is employed to select parts of hidden states across all the time steps by allocating adequate attention to key information [12]. The general attention mechanism is often implemented by scoring each encoder hidden states h_j in $\mathrm{H} = (h_1, h_2, \cdots, h_n)$ separately based on the previous decoder hidden state s_{i-1} and normalizing the scores $e_{i,j}$ by a softmax function to generate the attention weight $\alpha_{i,j}$:

$$e_{i,j} = \text{Score}\big(s_{i-1}, h_j\big), \tag{2}$$

$$\alpha_{i,j} = \frac{\exp\big(e_{i,j}\big)}{\sum_{i=1}^{n} \exp\big(e_{i,j}\big)}. \tag{3}$$

Then, the decoder input c_i at i is the weighted sum of h_j and calculated as follows:

$$c_i = \sum_{i=1}^{n} \alpha_{i,j} h_j. \tag{4}$$

Under the attention mechanism, the dependencies between the source and target sequences are not restricted by the intermediate distance. Consequently, it is helpful for over-coming the long-term dependency problem, and it was soon extended into various fields, including stock-related applications [15]-[22].

III. Volume-Aware Positional Attention-Based RNN

In this section, we first introduce the notation used in this article and the problem we aim to study. Then, the motivation and details of the proposed VPA-RNN model for stock index prediction are presented.

A. Notation and Problem Statement

The goal of this work is to predict the closing price of the next day. Given the previous values of the target as $Y = (y_1, y_2, \ldots, y_T)^\top \in \mathbb{R}^T$ where T represents the size of the time window and y_t is the target at time t. Similarly, the time series of all features would be denoted as $X = (X_1, X_2, \ldots, X_T)^\top \in \mathbb{R}^{T\times N}$ where N specifies the number of features. Hence, $X_t = (x_t^1, x_t^2, \cdots, x_t^n) \in \mathbb{R}^N$ is a vector of all the N features at time t and $X^n = (x_1^n, x_2^n, \cdots, x_T^n) \in \mathbb{R}^T$ is the time series of the *n*-th feature in time window T. Thus the VPA-RNN model aims to learn a nonlinear mapping function $F(\cdot)$ as follows:

$$\hat{y}_{T+1} = F(y_1, y_2, \ldots, y_T, X_1, X_2, \ldots, X_T). \tag{5}$$

The features used in this paper include *open*, *close*, *high*, *low*, *adj_close*, *and volume* in the granularity of the trading day. *Adj_close* is an abbreviation of the adjusted closing price, which amends a stock's closing price to accurately reflect that stock's value after adjustments for splits and dividend distributions. Deemed as the true price of stocks, it is often used when examining historical returns or performing a detailed analysis of historical returns. Therefore, this study uses *adj_close* of the next day as the target Y.

Among all the features, only the feature *volume* does not belong to the type of stock price, which refers to the number of

transactions in a trading day. Specifically, we represent the historical series of *volume* as $V = (v_1, v_2, \dots, v_T)^\top \in \mathbb{R}^T$, and it is used to achieve attention recalibration in the next subsection.

B. Proposed Model

The overall structure of our proposed VPA-RNN model is shown in Fig. 1. Inspired by existing work, we employ a dual-stage attention-based encoder-decoder neural network. In the encoder, we introduce an input attention mechanism proposed by [17], which is used to select the relevant features adaptively. In the decoder, our proposed volume-aware positional attention is used to automatically select relevant encoder hidden states across all time steps. With the help of the proposed attention mechanism, the decoder can take account of the effects of volumes and positions of different time steps in order to assign weight more appropriately.

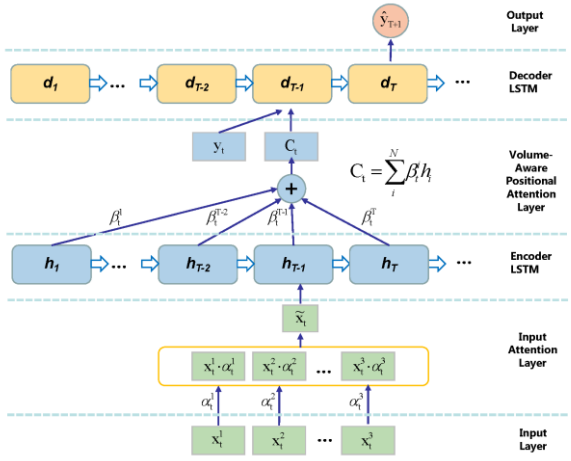

Figure 1. Graphical illustration of the volume-aware positional attention-based recurrent neural network.

1) Encoder with input attention

In this paper, the encoder is essentially an LSTM used to encode the input sequences into a hidden feature representation. As described above, given the time series of all features $X = (X_1, X_2, \dots, X_T)^\top \in \mathbb{R}^{T \times N}$ where N specifies the number of features, the encoder learns a mapping from X_t to h_t:

$$h_t = \mathrm{LSTM}_1(h_{t-1}, X_t), \tag{6}$$

where $h_t \in \mathbb{R}^m$ denotes the hidden state of the encoder at time t, m is the size of the hidden state. In order to select relevant features in the early stages, an input attention mechanism is employed, the time series of the n-th feature in time window T are denoted as $X^n = (x_1^n, x_2^n, \cdots, x_T^n) \in \mathbb{R}^T$, the input attention mechanism is implemented by a multilayer perceptron referring to the previous hidden state h_{t-1} and the cell state c_{t-1} in the encoder LSTM unit:

$$e_t^n = v_e^\top \tanh(W_e[h_{t-1}; c_{t-1}] + U_e X^n), \tag{7}$$

where $v_e \in \mathbb{R}^T$, $W_e \in \mathbb{R}^{T \times 2m}$, and $U_e \in \mathbb{R}^{T \times T}$ are parameters to learn, and the bias terms are omitted for succinctness. Then, a softmax function is applied to the alignment score e_t^n to ensure all the attention weights sum to 1:

$$\alpha_t^n = \frac{\exp(e_t^n)}{\sum_{i=1}^N \exp(e_t^i)}, \tag{8}$$

where α_t^n is the attention weight measuring the importance of the n-th input feature. Finally, we can adaptively select the features as follows:

$$\tilde{X}_t = (\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \cdots, \alpha_t^n x_t^n)^\top. \tag{9}$$

Thus, (6) can be updated as:

$$h_t = \mathrm{LSTM}_1(h_{t-1}, \tilde{X}_t), \tag{10}$$

where X_t is replaced by $\tilde{X}_t$ that considers the weights of different features. Therefore, the encoder can adaptively select certain features rather than pay attention to all the features equally.

2) Decoder with volume-aware positional attention

In the decoder, we use another LSTM to decode the encoded input information. In order to adaptively select relevant encoder hidden states across all time steps, we employ a temporal attention mechanism. Specifically, the attention weight l_t^i of each encoder hidden state is calculated based upon the previous decoder hidden state $d_{t-1} \in \mathbb{R}^p$ and cell state $c'_{t-1} \in \mathbb{R}^p$:

$$l_t^i = v_d^\top \tanh(W_d[d_{t-1}; c'_{t-1}] + U_d h_i), 1 \le i \le T, \tag{11}$$

where $[d_{t-1}; c'_{t-1}] \in \mathbb{R}^{2p}$ is a concatenation of the hidden state and cell state of the previous LSTM unit. $v_d \in \mathbb{R}^m$, $W_d \in \mathbb{R}^{m \times 2p}$, and $U_d \in \mathbb{R}^{m \times m}$ are parameters to be learned. We omit the bias terms here for clarity. The attention weight β_t^i that represents the importance of the i-th encoder hidden state h_i is calculated by the formula:

$$\beta_t^i = \frac{\exp(l_t^i)}{\sum_{j=1}^T \exp(l_t^j)}. \tag{12}$$

However, such temporal attention mechanism suffers from two problems: (1) identical or very similar time steps are scored equally regardless of their positions in the sequence. But the position of each time step is a key factor in the task of stock performance prediction. (2) It does not explicitly model the effect of volume of each time step in the input sequences, which is another important feature that provides valuable information. Therefore, we propose a new volume-aware positional attention mechanism to tackle these challenges, as shown in Fig. 2, which can evaluate the relative contribution of each time step not only on the information of encoder hidden states but also on the global position and volume of each time step.

First, inspired by the position encoding vectors used in [23], we define a position vector p_i for each time step in the time window T as follows:

$$p_i = \left(\frac{i}{T}, \frac{i}{T}, \cdots, \frac{i}{T}\right) \in \mathbb{R}^m, 1 \le i \le T, \tag{13}$$

where m is the dimension of position encoding vectors, that is the same as the size of the decoder hidden state in order to facilitate calculation, then we add the position encoding vector to the calculation formula of l_t^i, and (11) is updated as follows:

$$l_t^i = v_d^\top \tanh(W_d[d_{t-1}; c'_{t-1}] + U_d h_i + E_d p_i), 1 \le i \le T, \tag{14}$$

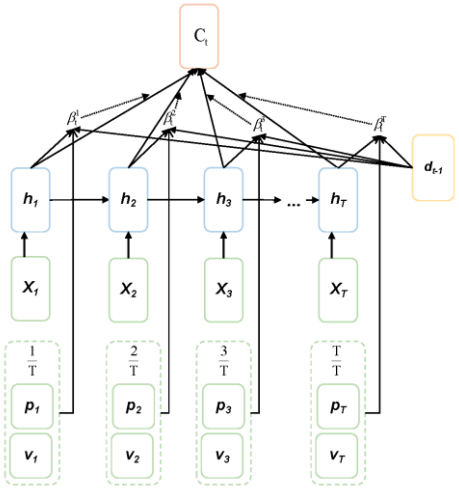

Figure 2. Our proposed volume-aware positional attention mechanism.

where $E_d \in \mathbb{R}^{m\times m}$ is a learnable parameter. In this way, the position encoding vector p_i can provide the important spatial information of each time step.

Second, inspired by the study of trading volume in [26], we argue that the volume of a time step to some extent reflects the importance of this time step. Hence, during the learning process of the attention mechanism, the effect of volume should be considered explicitly, we change (14) by multiplying l_t^i by the volume of this time step $v_t \in \mathbb{R}$, and the updated formula is as follows:

$$l_t^i = v_d^\top \tanh(W_d[d_{t-1}; c'_{t-1}] + U_d h_i + E_d p_i)v_t, \quad 1 \le i \le T \tag{15}$$

then, when the trained model is used to test on the test sequences, v_t will be replaced by the volume of the corresponding time step in the test sequences. In this way, the model can learn the attention weights under the condition that the effect of volume is considered explicitly, and the original learned attention distribution without being multiplied by the volume v_t can exclude the effect of the trading volume. Then, it can be multiplied by the volumes of the corresponding time steps in the test sequences to achieve attention recalibration when testing.

The context vector C_t is a weighted sum of all the encoder hidden states h_i, and it is distinct at each time step in the decoder.

$$C_t = \sum_{i=1}^{T} \beta_t^i h_i. \tag{16}$$

Then, the given target series $Y = (y_1, y_2, \dots, y_T)^\top \in \mathbb{R}^T$ can be combined with the context vectors:

$$\tilde{y}_t = \tilde{w}^\top [y_t; C_t] + \tilde{b}, \tag{17}$$

where $\tilde{w} \in \mathbb{R}^{m+1}$ and $\tilde{b} \in \mathbb{R}$ are used to map the concatenation $[y_t; C_t] \in \mathbb{R}^{m+1}$ to the size of the decoder input. The newly calculated $\tilde{y}_t$ is used to update the decoder hidden state:

$$d_t = \mathrm{LSTM}_2(d_{t-1}, \tilde{y}_t). \tag{18}$$

Finally, the final prediction result can be obtained by a linear function:

$$\hat{y}_{T+1} = v_y^\top \left(W_y[d_T; C_T] + b_w\right) + b_v, \tag{19}$$

where $[d_T; C_T] \in \mathbb{R}^{p+m}$ is a concatenation of the last decoder hidden state and context vector, and p is the size of the decoder hidden states. $W_y \in \mathbb{R}^{p\times(p+m)}$ and $b_w \in \mathbb{R}^p$ are the parameters used to map the concatenation to the size of the decoder hidden states. The linear function with weights $v_y \in \mathbb{R}^p$ and bias $b_v \in \mathbb{R}$ produces the final prediction result.

C. Training procedure

The minibatch stochastic gradient descent (MGD) and Adam optimizer are used for the model training with mean squared error (MSE) as the loss function:

$$\text{loss} = \frac{1}{N}\sum_{i=1}^{N} \left(\hat{y}_{T+1}^i - y_{T+1}^i\right)^2. \tag{20}$$

IV. Experiments

In this section, we first describe the two datasets used for empirical studies. Then, we present the experimental setup, including parameter settings and evaluation metrics. Finally, we conduct extensive experiments to evaluate the performance of the proposed VPA-RNN by comparing it to different baselines.

A. Datasets and Setup

We collected real-world historical data of two stock indexes from the Yahoo! Finance website: the S&P 500 Index (S&P 500) and the Dow Jones Industrial Average Index (DJIA), traded from Jan 3rd, 2000, to Dec 30th, 2020, at a daily frequency, for a total of 21 years, to test the effectiveness of the proposed VPA-RNN model. For a fixed time window of size T+1 and a stride of 1, each sample incorporated input sequences of T time steps and a target index value for model training and evaluation. All data from both datasets were normalized to between 0 and 1 using min-max scaling. Then, we divided each dataset into 7:1.5:1.5 ratios in the time dimension as the training set, validation set, and test set. The basic process of evaluation was to use the training set to train the model and obtain a classifier every epoch. Then, the best classifier was selected based on the validation set and was finally evaluated on the test set.

B. Parameter Settings and Evaluation Metrics

In the experiments of the previous study [17] for DA-RNN, the length of time window 10 yielded the best results. Hence, in this study, all the compared models use the same length of time window as 10 for comparison. In addition, other parameter settings for the baseline models are selected based on optimal experimental results. Other parameter settings of our VPA-RNN model are shown in Table I.

To measure the effectiveness of various methods for stock index prediction, we consider three different evaluation metrics: root mean square error (RMSE), mean absolute error (MAE), and coefficient of determination (R^2). Specifically, the smaller the RMSE and MAE, the closer the predicted value to the true value; the closer the coefficient R^2 to 1, the better the fit of the model.

TABLE I. PARAMETER SETTINGS

Parameter	Parameter Description	Value
lr	Learning rate	0.001
epoch	Number of epochs	1000
batch_size	Batch size	128
encoder_lstm_unit	Neuron number in encoder LSTM	64
decoder_lstm_unit	Neuron number in decoder LSTM	64
activation	Activation function	Tanh

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |\hat{y}_i - y_i|, \tag{21}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}, \tag{22}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}, \tag{23}$$

where y_i is the true value, $\bar{y}$ is the mean value of all true values, and $\hat{y}_i$ is the predicted value.

C. Results

To evaluate the effectiveness of the proposed VPA-RNN, we conduct experiments to compare our results with those of the compared models, including a standard long short-term memory neural network (LSTM), the encoder-decoder network (Encoder-Decoder) proposed in [32], we change it to perform stock index prediction as Qin et al. did in [17], and the dual-stage attention-based recurrent neural network (DA-RNN) proposed in [17]. Furthermore, we compare our proposed VPA-RNN model against the setting that only employs its positional attention mechanism without the volume-aware attention mechanism (PA-RNN) and the setting that only employs its volume-aware attention mechanism without the positional attention mechanism (VA-RNN). All models take the same input for a fair comparison. For all the compared methods, we train them ten times and report their average performance. The comparison results of all the models over the two datasets are shown in Table II.

As illustrated in Table II, the DA-RNN model outperforms Encoder-Decoder, which has no attention layer, indicating the effectiveness of the dual-stage attention mechanism since it is capable to adaptively extract relevant features and select relevant hidden states across all time steps. In addition, our proposed PA-RNN, VA-RNN, and VPA-RNN all show better performance than Encoder-Decoder and DA-RNN. This suggests that taking the position of each time step into account and extending the attention mechanism to be volume-aware can both provide more reliable attention weights to make more accurate predictions. With the integration of the positional attention as well as the volume-aware attention, our proposed VPA-RNN achieves the best MAE, RMSE, and R^2, that increase of 11.76%, 6.80%, and 0.41% and 56.81%, 47.83%, and 15.82% for the S&P 500 and DJIA datasets, respectively, compared to the DA-RNN model, indicating the effectiveness of our overall model structure.

TABLE II. STOCK INDEX PREDICTION RESULTS OVER THE S&P 500 DATASET AND DJIA DATASET (BEST PERFORMANCE DISPLAYED IN BOLDFACE)

Models	S&P 500 Dataset			DJIA Dataset		
	MAE ($\times 10^{-2}$%)	***RMSE*** ($\times 10^{-2}$%)	R^2 ($\times 10^{-1}$%)	***MAE*** ($\times 10^{-2}$%)	***RMSE*** ($\times 10^{-2}$%)	R^2 ($\times 10^{-1}$%)
LSTM	0.96	1.41	9.77	1.81	2.19	9.18
Encoder-Decoder	1.28	1.75	9.65	2.92	3.34	8.09
DA-RNN	1.02	1.47	9.75	2.57	3.22	8.22
PA-RNN	0.98	1.42	9.77	1.98	2.75	8.70
VA-RNN	0.94	1.38	9.78	1.49	2.05	9.28
VPA-RNN	**0.90**	**1.37**	**9.79**	**1.11**	**1.68**	**9.52**

For visual comparison, we show the prediction results of Encoder-Decoder, DA-RNN, and VPA-RNN over the DJIA dataset in Fig. 3. We can see that our proposed VPA-RNN generally fits the ground truth much better than Encoder-Decoder and DA-RNN, which shows the proposed volume-aware positional attention mechanism is indeed effective in the problem of stock index prediction.

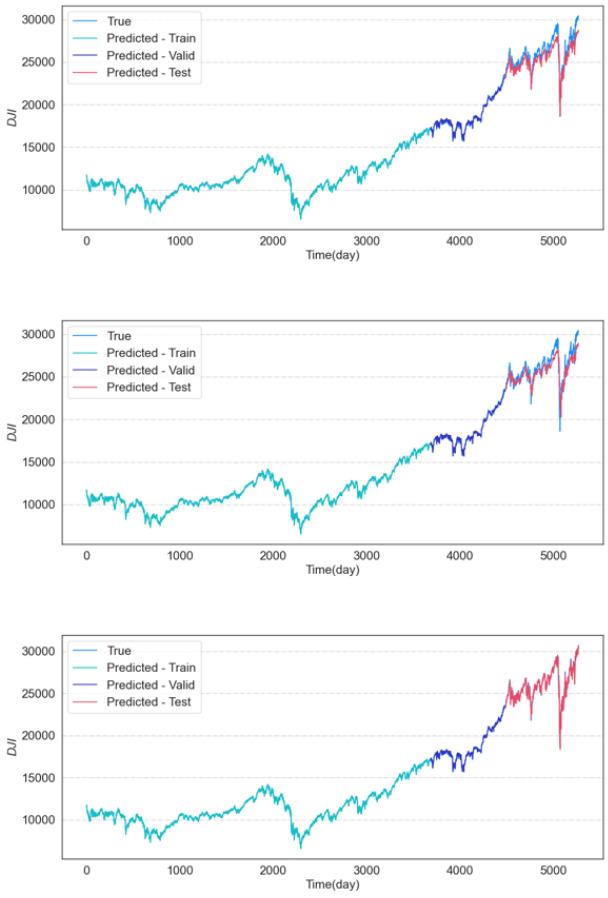

Figure 3. DJIA Index vs. Time. Encoder-Decoder (top) and DA-RNN (middle) are compared with VPA-RNN (bottom).

V. CONCLUSION

In this paper, we note two important factors (e.g., trading volume and position), which can potentially enhance the

attention mechanism for stock index prediction. Motivated by the observation, this study proposes a novel volume-aware positional attention recurrent neural network (VPA-RNN). Specifically, we add a position vector for each time step in the input sequences into the calculation formula of attention score to take the important spatial information into account. Then, we incorporate the trading volume into the original attention distribution to achieve attention recalibration. Based upon these two improvements, the VPA-RNN can take advantage of the features of stock index prediction and thus provide more reliable attention weights to make more accurate predictions. Extensive experiments on the S&P 500 dataset and the DJIA dataset demonstrated the superior performance of the proposed VPA-RNN relative to the original LSTM, Encoder-Decoder, and DA-RNN, indicating the VPA-RNN model has broad application prospects and is highly competitive. In summary, this work provides new insight into attention-based stock index prediction research and can help to develop better predicting models.

In the future, we will investigate whether feeding more technical indicators and basic information or adding predictions based on stock-related news can result in more accurate predictions. Furthermore, it is also promising to apply the proposed model to more granular trading data, such as hourly or per-minute transaction data.

ACKNOWLEDGMENT

This work was supported by National Key R&D Program of China (2019YFB1804400).

REFERENCES

[1] L. Li, S. Leng, J. Yang, and M. Yu, "Stock market autoregressive dynamics: a multinational comparative study with quantile regression," Mathematical Problems in Engineering, vol. 2016, 2016.

[2] M. Zhang, X. Jiang, Z. Fang, Y. Zeng, and K. Xu, "High-order hidden markov model for trend prediction in financial time series," Physica A: Statistical Mechanics and its Applications, vol. 517, pp. 1–12,2019.

[3] Y. Song and J. Lee, "Importance of event binary features in stockprice prediction," Applied Sciences, vol. 10, no. 5, p. 1597, 2020.

[4] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in2016 IEEE/ACIS 15th International Conference on Computer andInformation Science (ICIS). IEEE, 2016, pp. 1-6.

[5] Q. Gao, "Stock market forecasting using recurrent neural network," Ph.D. dissertation, University of Missouri–Columbia,2016.

[6] A. M. Rather, A. Agarwal, and V. Sastry, "Recurrent neural networkand a hybrid model for prediction of stock returns," Expert Systemswith Applications, vol. 42, no. 6, pp. 3234–3241, 2015.

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

[8] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018.

[9] Y. Gu, T. Shibukawa, Y. Kondo, S. Nagao, and S. Kamijo, "Prediction of stock performance using deep neural networks," Applied Sciences, vol. 10, no. 22, p. 8142, 2020.

[10] L. Zhang, C. Aggarwal, and G.-J. Qi, "Stock price prediction via discovering multi-frequency trading patterns," in Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 2141–2149.

[11] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157–166, 1994.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[13] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in 2013 IEEE international conference on acoustics, speech and signal processing. Ieee, 2013, pp. 6645–6649.

[14] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3128– 3137.

[15] G. Liu and X. Wang, "A numerical-based attention method for stock market prediction with dual information," Ieee Access, vol. 7, pp. 7357–7367, 2018.

[16] H. Li, Y. Shen, and Y. Zhu, "Stock price prediction using attentionbased multi-input lstm," in Asian Conference on Machine Learning. PMLR, 2018, pp. 454–469.

[17] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," arXiv preprint arXiv:1704.02971, 2017.

[18] Y. Yu and Y.-J. Kim, "Two-dimensional attention-based lstm model for stock index prediction," Journal of Information Processing Systems, vol. 15, no. 5, pp. 1231–1242, 2019.

[19] Y. Chen, W. Lin, and J. Z. Wang, "A dual-attention-based stock price trend prediction model with dual features," IEEE Access, vol. 7, pp. 148 047–148 058, 2019.

[20] Q. Chen, W. Zhang, and Y. Lou, "Forecasting stock prices using a hybrid deep learning model integrating attention mechanism, multilayer perceptron, and bidirectional long-short term memory neural network," IEEE Access, vol. 8, pp. 117 365–117 376, 2020.

[21] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long- short term memory neural network based on attention mechanism," PloS one, vol. 15, no. 1, p. e0227222, 2020.

[22] H. Li, Y. Shen, and Y. Zhu, "Stock price prediction using attention-based multi-input lstm," in Asian Conference on Machine Learning. PMLR, 2018, pp. 454–469.

[23] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 35–45.

[24] B. Yang, Z. Tu, D. F. Wong, F. Meng, L. S. Chao, and T. Zhang, "Modeling localness for self-attention networks," arXiv preprint arXiv: 1810.10182, 2018.

[25] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," arXiv preprint arXiv: 1506.07503, 2015.

[26] C. M. Lee and B. Swaminathan, "Price momentum and trading volume," the Journal of Finance, vol. 55, no. 5, pp. 2017–2069, 2000.

[27] J. Yu, J. Jiang, and R. Xia, "Entity-sensitive attention and fusion network for entity-level multimodal sentiment classification," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 429–439, 2019.

[28] V. Piratla, S. Sarawagi, and S. Chakrabarti, "Topic sensitive attention on generic corpora corrects sense bias in pretrained embeddings," arXiv preprint arXiv: 1906.02688, 2019.

[29] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1700–1709.

[30] K. Cho, B. Van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," arXiv preprint arXiv: 1409.1259, 2014.

[31] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," arXiv preprint arXiv: 1409.3215, 2014.

[32] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, ¨ H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," arXiv preprint arXiv: 1406.1078, 2014.