IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 24, 2019, accepted September 27, 2019, date of publication October 8, 2019, date of current version October 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946223

A Dual-Attention-Based Stock Price Trend Prediction Model With Dual Features

YINGXUAN CHEN[1], **WEIWEI LIN**[1,3], **AND JAMES Z. WANG**[2]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[2]School of Computing, Clemson University, Clemson, SC 29631, USA
[3]Guangdong Luan Industry and Commerce Company Ltd., Guangzhou 510520, China

Corresponding author: Weiwei Lin (linww@scut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772205 and Grant 61872084, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2017B010126002 and Grant 2018KJYZ009, in part by the Guangzhou Science and Technology Projects under Grant 201802010010, Grant 201807010052, Grant 201902010040, and Grant 201907010001, in part by the Nansha Science and Technology Projects under Grant 2017GJ001, in part by the Special Funds for the Development of Industry and Information of Guangdong Province (big data demonstrated applications) in 2017, and in part by the Fundamental Research Funds for the Central Universities, SCUT, under Grant 2019ZD26. The work of J. Z. Wang was supported in part by the NSF under Grant 1759856.

ABSTRACT Modeling and predicting stock prices is an important and challenging task in the field of financial market. Due to the high volatility of stock prices, traditional data mining methods cannot identify the most relevant and critical market data for predicting stock price trend. This paper proposes a stock price trend predictive model (TPM) based on an encoder-decoder framework that predicting the stock price movement and its duration adaptively. This model consists of two phases, first, a dual feature extraction method based on different time spans is proposed to get more information from the market data. While traditional methods only extract features from information at some specific time points, this proposed model applies the PLR method and CNN to extract the long-term temporal features and the short-term spatial features from market data. Then, in the second phase of the proposed TPM, a dual attention mechanism based encoder-decoder framework is used to select and merge relevant dual features and predict the stock price trend. To evaluate our proposed TPM, we collected high-frequency market data for stock indexes CSI300, SSE 50 and CSI 500, and conducted experiments based on these three data sets. The experimental results show that the proposed TPM outperforms the existing state-of-art methods, including SVR, LSTM, CNN, LSTM_CNN and TPM_NC, in terms of prediction accuracy.

INDEX TERMS Stock price, predictive models, neural networks, encoder-decoder, dual feature extraction, dual attention mechanism.

I. INTRODUCTION

The stock price is a highly volatile time series in the financial field. The prices of stocks are affected by many factors, such as interest rates, exchange rates, inflation, monetary policy, investor sentiment, etc. Modeling the relationship between the stock price and these factors and predicting the stock price trend is a challenging task for researchers and investors.

There are many studies on the prediction and analysis of financial time series. In 1970, the Effective Market Hypothesis [1] indicated that the stock price is an immediate reflection of stock market information. Therefore, researchers use traditional statistical methods such as regression methods, exponential average and ARIMA [2]–[4] to predict the stock price based on historical prices. However, since the underlying market information mined from stock price is too little, these statistical methods cannot accurately predict stock price trend. Statistical methods often assume that the time series is generated from a linear process and therefore perform poorly in non-linear stock price prediction. Machine learning and deep learning methods have been relatively successful in financial time series modeling. Compared with statistical methods, machine learning and deep learning methods have better nonlinear mapping ability. A considerable amount of research has been conducted to extract features on specific time points and then use features to model and predict the result. However, they ignore the interaction of data and short-term continuity of data fluctuations. To breach this gap,

The associate editor coordinating the review of this manuscript and approving it for publication was Yongping Pan.

we propose a dual data feature extraction method based on one single time point and multiple time points, combine short-term market features with long-term temporal features to improve the accuracy of prediction. Moreover, the proposed model is based on the encoder-decoder framework [5]–[6], and the attention mechanism [7] is introduced in encoder and decoder stages respectively to solve the problem that the most relevant features could not be concerned in a long time series.

Motivated by above-mentioned problems, this paper proposes a new stock price trend prediction model (TPM) based on dual features and dual attention mechanism. The aim of the TPM is to predict the direction and duration of stock price changes. The main contributions of this paper include:

1) A new dual-feature extraction method based on different time spans is proposed, which can effectively mine the underlying market information and optimize model prediction results. In this paper, the piecewise linear regression method and convolutional neural network are used to extract long-term temporal features and short-term market features of the financial time series in different time spans. Describing the stock market information with dual features can improve the model prediction performance.
2) Using an encoder-decoder framework, a stock price trend prediction model (TPM) based on the dual attention mechanism is proposed. Introducing the attention mechanism in both encoder and decoder stage respectively, the TPM model can adaptively select the most relevant spatial short-term market features and combine them with long-term temporal features for prediction.
3) A performance study on our proposed TPM shows that this proposed method outperforms the state-of-art methods, including SVR, LSTM, CNN, LSTM_CNN and TPM_NC, under various training and testing parameters with different stock index data sets. The performance results show our proposed TPM demonstrates better generalization ability and market forecasting ability.

The rest of this paper is organized as follows. Section 2 introduces the related work on mining the financial time series. The proposed preprocessing method and TPM are discussed in details in Section 3. Section 4 presents experiments which demonstrate the superiority of our TPM. We conclude our work in Section 5.

II. RELATED WORK

In the field of mining the financial time series, most research methods can be divided into two phases: data preprocessing and time series modeling. In data preprocessing phase, some preprocessing procedure such as dimension reduction, feature selection, and feature extraction can be used to transform raw input data into representative features. Then in time series modeling stage, a prediction model is built to learn the temporal dependencies of feature and predict the result.

Data preprocessing is a process that maps the original high-dimensional data into low-dimensional features, filter out the irrelevant data, and obtain the least redundant and most representative features. The data preprocessing results often influence the effect of the prediction. In the field of mining financial time series, feature selection and extraction, time series segmentation are often used for data preprocessing. Feature selection and extraction can be classified as filters and wrappers, depending on different feature evaluation methods [8]. Filter methods choose static statistical characteristics of the data as evaluation criteria, while wrapper methods are refining the results dynamically. Generally, wrapper methods perform better than filter methods but need more expensive computing resources. Therefore, some researchers have suggested a combined filter and wrapper method. Moradi and Gholampour [9] proposed a feature search method which combined feature correlation of local search and particle swarm optimization of global search. Dong *et al.* [10] blended binary genetic algorithm, neighborhood rough set algorithm and ROGA algorithm for feature extraction. Time series segmentation, decomposing historical data into important points or segments, have great help in filtering data noise, reducing dimension and saving computing resources. For example, Chang *et al.* [11] combined genetic algorithm and segmentation methods for identifying trend points and Zhao and Wang [12] used the outliers of stock volume for stock market prediction.

Modeling and learning the dependencies of financial time series is a challenging issue. In the last decades, machine learning and deep learning methods have become very popular in financial time series modeling. Machine learning methods, such as random forest [13], artificial neural network [14], and support vector machine [15], have good nonlinear mapping ability and easy interpretation. Chen and Hao [16] predicted the stock index by applying feature-weighted SVM and feature-weighted KNN. Thakur and Kumar [17] integrated the random forest and weighted SVM to generate trading signals of decision support systems. Chandar [18] used a discrete wavelet transform to decompose financial time series data and build a fuzzy set based ANN model for predicting the closing price of the stock. Deep learning methods, discover the multi-level abstract data representations of data set by its deep neural network structure and back-propagation algorithms, have achieved great success in image processing, speech recognition and text mining [19], such as Seq2Seq [5], GoogLeNet [20], ResNet [21], BERT [22] and have also been tried in the financial time series field [23]–[25]. For example, Zhang *et al.* [26] proposed a novel event representation model involving RBM and sentence2vec, which extracts and trains the stock price data and news text information for prediction. Minh *et al.* [27] proposed a new two-stream gated recurrent unit network (TGRU) to improve the performance of the prediction model. Pang *et al.* [28] proposed two LSTM models, one based on the embedded layer and the other based on the automatic encoder. The LSTM with embedded layer shows better performance in stock market prediction. As we discussed earlier, most of them applied machine learning method for feature extraction and used a

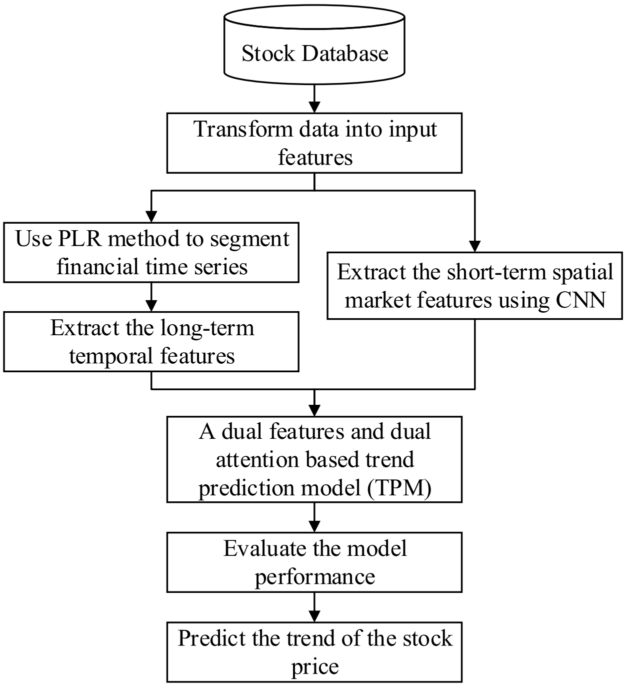

Stock Database
Transform data into input features
Use PLR method to segment financial time series
Extract the short-term spatial market features using CNN
Extract the long-term temporal features
A dual features and dual attention based trend prediction model (TPM)
Evaluate the model performance
Predict the trend of the stock price

FIGURE 1. Detailed processes of the TPM.

single neural network for modeling. However, they ignored that using multiple different intricate structures of deep neural networks, such as CNN, RNN, LSTM, can consider the characteristic of data more comprehensively.

III. TREND PREDICTION MODEL (TPM)

The aforementioned deficiencies are summarized as follows: first, the univariate financial time series contains insufficient information. Second, the traditional feature extraction method is limited in studying market behavior. Third, the information learning from data with a single neural network is incomprehensive.

To address these issues, this paper proposes a new stock trend prediction model (TPM) based on dual features and dual attention mechanism. This model consists of two phases. First, we use the piecewise linear regression method to segment the financial time series and extract the historical long-term temporal features based on the sub-sequences with different time spans. The short-term spatial market features based on each time point are generated through a convolutional neural network. Then, in the second phase of TPM, with the dual features extracted previously, a dual-attention-based trend prediction model is proposed. It is based on the encoder-decoder framework. The encoder stage is in the form of LSTM and the attention mechanism in encoder is applied to extract the most relevant short-term market features adaptively, then encode into a feature vector. The decoder stage, formed by attention-based LSTM, selects and decodes the most relevant fusion features to predict the stock price trend. The detailed process of the TPM is shown in Fig. 1.

Given a financial time series $\mathbf{X} = X_1, X_2, \ldots, X_n$, where $X_n \in \mathbb{R}^d$ represents the input data at time n. We decompose the predict time series into sub-sequence sets, denoted by $\mathbf{L} = (L_1, L_2, \ldots, L_m)$. Each sub-sequence is fitted to a segment and denoted by $L_m = (s_m, d_m)$, s_m is the slope of the segment, and d_m is the duration of the sub-sequence, which is the time length of the segment. The long-term temporal features can be extract through the sliding window ω_l. At time t, the long-term feature can be denoted by

$$Z_t = \left\{(s_m, d_m) \mid \sum\nolimits_{m1}^{m_{\omega_l}} d_m \leq t\right\} \tag{1}$$

The short-term feature can be extracted from the data of the sliding window ω_s and the short-term feature at time t is

$$S_t = \left(X_{t-\omega_s}, X_{t-\omega_s+1}, \ldots, X_t\right) \tag{2}$$

The set of the short-term features is $\mathbf{S} = (S_1, S_2, \ldots, S_k)$ and the length of each element S_k is ω_s.

As we discussed in Section 1, according to the long-term features and the short-term features, our goal is to predict the stock price trend $\widehat{Y_T} = (s_T, d_T)$. More specifically, we aim to learn a nonlinear mapping $F(\mathbf{.})$ that

$$\widehat{Y_T} = F(Z_{T-1}, S_{T-1}) \tag{3}$$

A. PHASE I: DATA PREPROCESSING

1) FEATURE GENERATION

Since the market information provided by univariate financial time series is insufficient, it is hard to model and predict the stock price trend from univariate data. We choose the basic market data such as opening price, closing price, highest price, lowest price, volume and transform them into technical indicators. Technical indicators are the meaningful rules and patterns of the market proposed in [29]–[30]. For example, Moving Average (MA), The Relative Strength Index (RSI) and Moving Average Convergence/Divergence (MACD). In addition, their lagged time series also contained as our features. The specific description of the generated features is shown in Table 1.

Collecting the five-minute interval market data of CSI 300 on March 21, 2017, the closing price and several features mentioned above are plotted in Fig. 2. We can find that in Fig. 2 (a), the closing price change of two sub-periods is very similar, but Fig. 2 (b)-(f), other features of these two sub-periods are completely different, especially the Volume in Fig. 2 (b) and the WMSR%12 in Fig. 2 (e). Therefore, it is unreliable to predict the stock price trend only by the change of the closing price, and we should combine multi-feature to depict the stock market information and improve the accuracy of prediction.

2) DUAL FEATURE EXTRACTION

a: PLR EXTRACTS THE LONG-TERM TEMPORAL FEATURES

Considering the continuity of data changes, we extract the long-term temporal features with multiple time points by piecewise linear regression method (PLR). The PLR method

TABLE 1. Description of generated features.

Field	Feature	Description
Moving Average (MA)	MA5 , MA10 , MA20	The average of n consecutive values forms a moving average, which can smooth short-term fluctuations.
Bias (BIAS)	BIAS5 , BIAS10	The difference between the closing price and the MA, which measures the fluctuation of the closing price and analyzes the market behavior.
The Relative Strength Index (RSI)	RSI6 , RSI12	Compares the magnitude of return and loss recently for observing the overbought and oversold situation.
Stochastics (KD)	K9 , D9	The K and D measure the overbought and oversold situation of the asset and give the trading signal.
Moving Average Convergence/Divergence (MACD)	MACD9	Shows the difference between the fast exponential moving average and the slow exponential moving average of the closing price.
Williams %R (WMS %R)	WMS%R12	Williams %R is a momentum indicator that shows the relationship between the current closing price and the highest and lowest prices in the past n time points.
The Volatility of Closing Price	VOL1	Measure the degree of variation of the closing price over time.
The Volatility of Volume	VOL2	Measure the degree of variation of the volume over time.
Differential Technical Indicators	△MA5, △MA10, △K9, △D9,△MA20,△MACD9, △RSI6, △RSI12,△BIAS5, △BIAS10, △WMS%R12	Describe the changes in previous technical indicators.

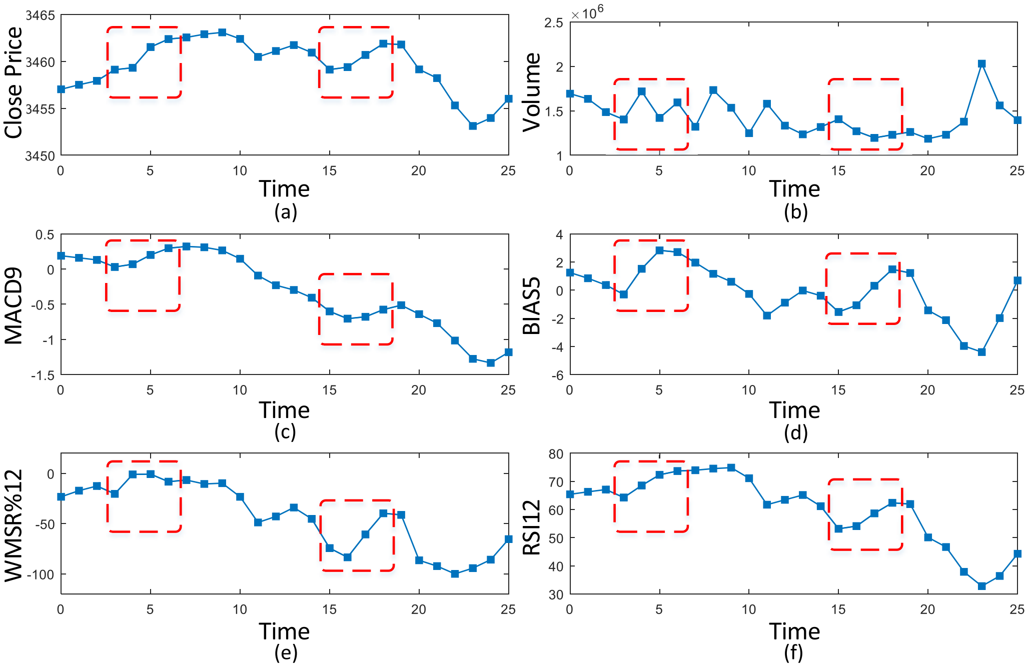


FIGURE 2. Changes in different features of CSI 300, on March 21, 2017.

can smooth the short-term fluctuation noise, reduce the data dimension and improve the computational performance.

There are three traditional PLR methods, bottom-up, top-down and sliding window. The sliding window method is that dividing financial time series into sub-sequences with fixed length. If the window size is not suitable, the sub-sequence will be incorrectly divided, which will influence the effect of the prediction. To avoid this, we choose the bottom-up PLR method which segments the time series more appropriately and has a relatively low fitting error compared with other methods. The detailed algorithm is shown in Algorithm 1.

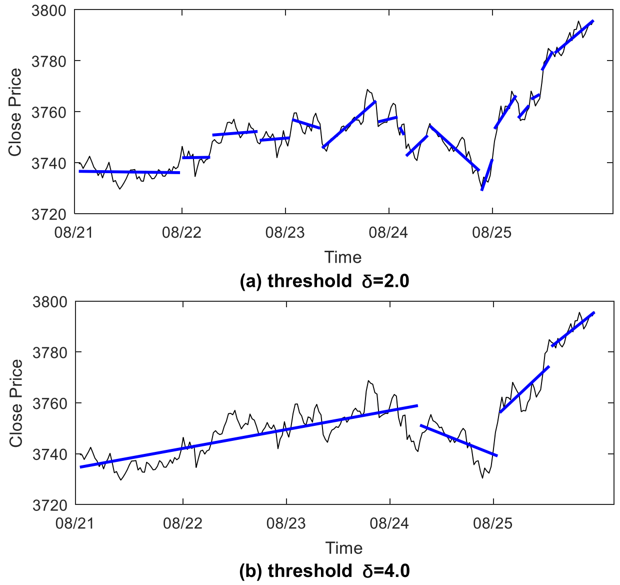
In our task, we assume that the number of data points is n and there are m segments with length d. Each segment with length i needs $\theta(i)$ times to generate so that reach d length segment from 2 length segments needs $\theta(\mathrm{d}^2)$ time. We need to check n/d segments with d length at most, thus, the time complexity is $\mathrm{O}(\mathrm{n} * \mathrm{d})$.

Obviously, the segment result of the time series depends on the maximum error threshold δ. Taking CSI 300 as an example, we use the bottom-up PLR method to segment its historical closing price. In Fig. 3 (a), when the threshold value δ is 2.0, the time series can be divided into

Algorithm 1 Bottom-Up PLR for Financial Time Series Segmentation

```
Input A financial time series X = $x_1, x_2, \ldots, x_n$, max
      error $\delta$
Output A sequence of line segments $L = (L_1, L_2, \ldots, L_m)$

for i = 1: $\lfloor \frac{n}{2} \rfloor$ do
    $L_i$ = create segment($x_{2i-1}, x_{2i}$)
end for
for each $L_i$ in L do
    calculate the merged cost $C_i$ of $L_i$ and $L_{i+1}$ segments
end for
while min(C) < max error $\delta$ do
    i = min(C)
    replace $L_i$ with the merge of $L_i$ and $L_{i+1}$
    delete $L_{i+1}$
    update the merged cost $C_i$ with new $L_i$ and $L_{i+1}$ segments
    update the merged cost $C_{i-1}$ with new $L_{i-1}$ and $L_i$ segments
end while
Output L
```



(a) threshold δ=2.0

(b) threshold δ=4.0

FIGURE 3. The segment results of PLR method with threshold $\delta = 2.0$ (top) and threshold $\delta = 4.0$ (bottom).

16 sub-sequences, while in Fig. 3 (b), when the threshold value δ is 4.0, there are only four sub-sequences can be obtained. With the increase of threshold value, the more data fluctuations are ignored and the fewer sub-sequences are formed. The value of the threshold influence the validity of historical time series features. Each sub-sequence represents the fluctuation of data over a time period. The slope s_m and the duration d_m of each sub-sequence are generated as the long-term temporal features in TPM to predict the stock price trend.

b: CNN EXTRACTS THE SHORT-TERM SPATIAL MARKET FEATURES

Considering the interaction of different data at the same time point, the short-term spatial market feature of each time point is extracted by a convolution neural network. Given financial time series $\mathbf{X} = X_1, X_2, \ldots, X_n$, we construct a Market Matrix to describe the historical stock market which is denoted by $S_{T-1} = (X_1, X_2, \ldots, X_{T-1})$. In the Market Matrix, each row represents one dimension of S_{T-1} and the number of rows is n, while each column represents one time points and the number of columns is *T-1*. Because the CNN preserve the neighborhood relations and spatial locality of the input data [31], CNN can capture the non-linear relationship between the Market Matrix S_{T-1} and stock trend (s_T, d_T), and output the spatial features of the short-term historical time series. $S_{T-1} = (X_1, X_2, \ldots, X_{T-1})$.

The detailed short-term feature extraction structure of CNN is shown in Fig.4. In our CNN architecture, different size of convolution kernel is chosen such as 1 × 3, 1 × 5 to extract abstract multi-level spatial market features. The convolution neuron for extracting features from input the Market Matrix is given by

$$H_t^c = \emptyset\left(b^c + \sum W^c * X_t\right) \tag{4}$$

where X_t denotes the input Market Matrix, $*$ is the convolution operation, W^c and b^c are the weights and biases of convolution neurons to be trained, $\emptyset(\cdot)$ is a non-linear activation function which is chosen to be the ReLU function [32].

The max pooling layer will be performed after convolution layers and it can reduce the size of feature maps and avoid overfitting. We choose the same size with the convolution kernel 1 × m and the max pooling operation can be described by

$$H_t^{i,p} = \max\left(H_t^{i,c}, H_t^{i+1,c}, \ldots, H_t^{i+m-1,c}\right) \tag{5}$$

After several layers of convolution and max-pooling, we feed the outputs to a projected layer by $W_t = W^p * H_t^p + b^p$, where W^p and b^p are parameters. Finally, the interaction of data can be depicted by the short-term spatial market vector $\mathbf{W}_{Market} = (W_1, W_2, \ldots, W_{T-1})$, where each $W_t \in \mathbb{R}^m$ denotes the spatial market feature at time t.

B. PHASE II: TIME SERIES MODELING BY ENCODER-DECODER FRAMEWORK

The encoder-decoder framework is first proposed in text processing, which is usually in the form of RNN or CNN. In an encoder-decoder framework, the encoder compresses the input information into a fixed-size vector and the decoder processes these vectors into the final result. However, when there is too much input information, the encoder cannot efficiently identify all relative information. As a result, the performance of the encoder-decoder framework will deteriorate. The attention mechanism can optimize the problem by decoding the hidden state of relevant neurons. The encoder-decoder framework is simulating the human information processing process, solving the limitation of

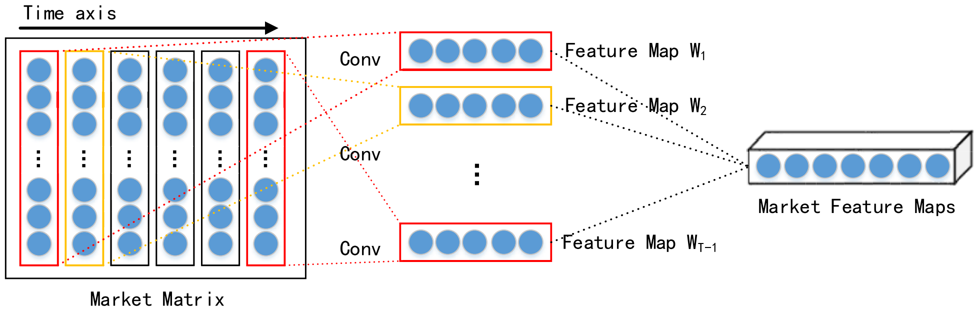


FIGURE 4. The short-term feature extraction structure of CNN.

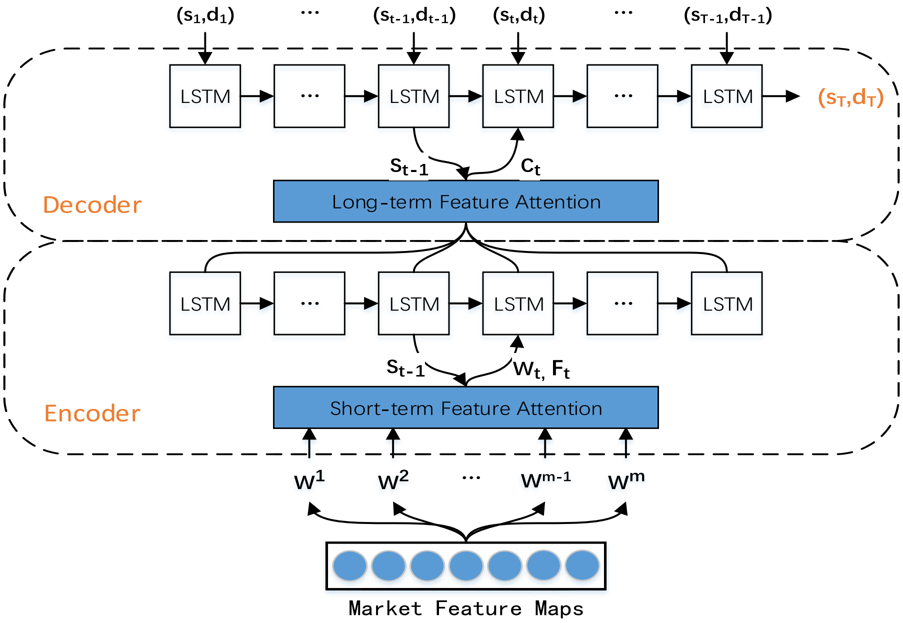


FIGURE 5. The structure of the encoder-decoder framework in TPM.

the same length of the encoding-decoding time series, and refining and compressing the input data to generate better prediction results.

Clearly, there is a problem that the attention-based decoder cannot select relevant input data explicitly, so we introduce the attention in both encoder and decoder stage respectively. The second phase of our proposed TPM is based on the dual attention mechanism which is shown in Fig. 5. The encoder-decoder framework can be divided into two stages. In the first stage we input the short-term spatial market features extracted by CNN into the attention-based LSTM encoder, the relevant short-term features at each time point are selected adaptively and encoded into vectors. In the second stage, the encoded vectors and the long-term temporal features extracted by PLR are input, the LSTM decoder decodes the relevant vectors and features based on the attention mechanism to predict the stock price trend. Through the dual attention mechanism, we can adaptively select the most relevant spatial market features and temporal features to model and predict the trend.

1) ATTENTION-BASED SHORT-TERM FEATURE ENCODER

Given the short-term spatial market features $\boldsymbol{W}_{Market} = (W_1, W_2, \ldots, W_{T-1})$ extracted by the CNN. At each time point t, the encoder learns the mapping relationship between the input feature W_t and the hidden state H_t:

$$H_t = f_{en}\left(W_t, H_{t-1}; \theta_{en}\right) \tag{6}$$

where $H_t \in \mathbb{R}^k$ is the hidden state of the encoder at time t, k is the size of the hidden state, $f_{en}(\centerdot)$ is a nonlinear function, and θ_{en} denotes the parameters of the encoder. We use LSTM [33] as a nonlinear function f_{en} to capture the temporal dependencies and form a short-term feature encoder. A LSTM neuron controls the update and output of the state by a forget gate σ_1, an input gate σ_2 and an output gate σ_3. Their operations are

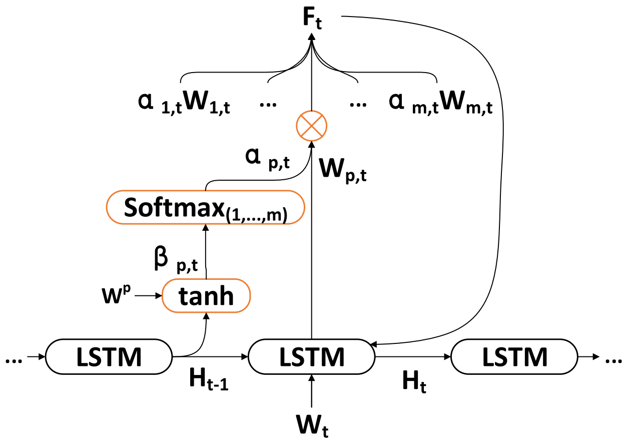


FIGURE 6. The calculation procedure of the attention mechanism.

as follows:

$$f_t = \sigma_1(W_f[H_{t-1}; W_t] + b_f) \tag{7}$$
$$i_t = \sigma_2(W_i[H_{t-1}; W_t] + b_i) \tag{8}$$
$$\tilde{C}_t = \tanh(W_c[H_{t-1}; W_t] + b_c) \tag{9}$$
$$o_t = \sigma_3\left(W_o\left[H_{t-1}; W_t\right] + b_o\right) \tag{10}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{11}$$
$$H_t = o_t * \tanh(C_t) \tag{12}$$

where $\sigma_1, \sigma_2, \sigma_3$ are three sigmoid functions, $*$ is the element-wise operator, H_{t-1} is the hidden state of the previous time point *t-1*, W_t is the input at time point t, $W_f, W_i, W_c, W_o \in \mathbb{R}^{k\times(k+m)}$ and $b_f, b_i, b_c, b_o \in \mathbb{R}^k$ are parameters. LSTM is capable of modeling the dynamic temporal behavior of time series effectively and avoiding gradient vanishing or exploding issues in RNN [34].

As shown in Fig. 6, we introduce the attention mechanism [7] in encoder stage and divide the input feature $\boldsymbol{W}_{Market}$ into $\left(W^1, W^2, \ldots, W^m\right)$ according to the feature dimension m, where $W^p = (W_{p,1}, W_{p,2}, \ldots, W_{p,T-1})$ represents the p-th dimension feature at each time point. Given the hidden state H_{t-1} and the cell state C_{t-1} calculated at time *t-1*, the relevant dimensions of the input features are identified and used to update the input features of the next time t.

$$\beta_{p,t} = v_a \tanh\left(W_a\left[H_{t-1}; C_{t-1}\right] + U_a W^p\right) \tag{13}$$
$$\alpha_{p,t} = \frac{\exp\left(\beta_{p,t}\right)}{\sum_{p=1}^{m} \exp(\beta_{p,t}))} \tag{14}$$
$$F_t = \left(\alpha_{1,t}W_{1,t}, \alpha_{2,t}W_{2,t}, \ldots, \alpha_{m,t}W_{m,t}\right) \tag{15}$$

where $v_a \in \mathbb{R}^{T-1}$, $W_a \in \mathbb{R}^{(T-1)\times 2k}$ and $U_a \in \mathbb{R}^{(T-1)\times(T-1)}$ are parameters, the softmax function is chosen for calculating the importance $\alpha_{m,t}$ of each dimension feature, update all dimensions of W_t to F_t and input them to the encoder, then the hidden state of the time point t is:

$$H_t = f_{en}\left(F_t, H_{t-1}; \theta_{en}\right) \tag{16}$$

Through the above steps, at each time point t, we can select the relevant dimensions of spatial market features, update the input feature and the hidden state of the encoder successively, and generate the most relevant short-term feature encode vector.

2) ATTENTION-BASED LONG-TERM FEATURE DECODER

The decoder is in the form of the LSTM neurons to predict the stock price trend. Given the long-term temporal feature $Z_{T-1} = (L_1, L_2, \ldots, L_{T-1})$ extracted by PLR method, where *T-1* is the sequence length, and $L_t = (s_t, d_t)$ denotes the long-term temporal features at time point t. At each time point t, the decoder LSTM learns the mapping relationship between the encode vector W_t, the long-term feature L_t and the hidden state $H_t^{'}$.

$$H_t^{'} = f_{de}\left(L_t, W_t, H_{t-1}^{'}; \theta_{de}\right) \tag{17}$$

where $H'_t \in \mathbb{R}^g$ is the hidden state of the decoder at time t, g is the size of the hidden state, $f_{de}(\boldsymbol{.})$ is a nonlinear function, and θ_{de} denotes the parameters of the decoder. Similarly, we use LSTM as a nonlinear function f_{de} to capture the temporal dependencies and form a long-term feature decoder. The calculate procedure is similar to the encoder stage.

We also introduce the attention mechanism in decoder stage to get the related encoder hidden states of all time points. Given the hidden state $H'_{t-1} \in \mathbb{R}^g$ and the cell state $C'_{t-1} \in \mathbb{R}^g$ of the decoder, the hidden state H_i of the encoder, the importance of the hidden state $\gamma_{i,t}$ in the i-th encoder at time t can be obtained by

$$\eta_{i.t} = v_b \tanh\left(W_b\left[H_{t-1}^{'}; C_{t-1}^{'}\right] + U_b H_i\right) \tag{18}$$
$$\gamma_{i,t} = \frac{\exp\left(\eta_{i.t}\right)}{\sum_{j=1}^{T-1} \exp\left(\eta_{i.j}\right)} \tag{19}$$

where $1 \leq i \leq T-1$, $v_b \in \mathbb{R}^k$, $W_b \in \mathbb{R}^{k\times 2g}$ and $U_b \in \mathbb{R}^{k\times k}$ are parameters. Then, the context vector that we feed to the decoder is given through all hidden states of the encoder $(H_1, H_2, \ldots, H_k)$ by

$$C_t^{'} = \sum_{i=1}^{T-1} \gamma_{i,t} H_i \tag{20}$$

After obtaining the context vector $C_t^{'}$, we can combine C'_t with the long-term temporal feature L_t to generate the mixed feature y_t, that is

$$y_t = w_c^T\left[L_t; C_t^{'}\right] + b_c \tag{21}$$

where $w_c^T \in \mathbb{R}^{k+2}$ and $b_c \in \mathbb{R}^2$ are parameters to be learned. Finally, instead the feature L_t with the mixed feature y_t, we can get the hidden state of the decoder H'_t by

$$H_t^{'} = f_{de}\left(y_t, H_{t-1}^{'}; \theta_{de}\right) \tag{22}$$

Through the aforementioned formula, at each time point t, the most relevant encoder hidden state of all time points and the long-term temporal features will be chosen to generate the mixed feature vectors.

Finally, we learn the nonlinear mapping function $F(\centerdot)$ between the stock price trend and the dual features. The prediction of stock price trend at time point t $\widehat{Y_T} = (s_T, d_T)$ is given by

$$\begin{aligned}\widehat{Y_T} &= F\left(Z_{T-1}, S_{T-1}\right) \\ &= v_d^T(W_d\left[H_{T-1}^{'}; C_{T-1}^{'}\right] + b_d) + b_d^{'}\end{aligned} \tag{23}$$

where H'_{T-1} and C'_{T-1} represent the hidden state and content vector of the decoder at time point *T-1*, $W_d \in \mathbb{R}^{g\times(g+k)}$ and $b_d \in \mathbb{R}^g$ are parameters. And at last, we use a linear function to get the stock price trend prediction at time point T, where $v_d^T \in \mathbb{R}^g$ and $b_d^{'} \in \mathbb{R}$ are weights and bias of the last linear function.

We used a stochastic gradient descent method and a momentum optimizer to train the proposed model with the batch size of 64 and the learning rate of 0.001. The squared error function with regular terms is our object function, and the parameters of the model will be learned through the back propagation. The loss function is given by

$$\text{Loss (W, b)} = \frac{1}{N}\sum_{i=1}^{N}(\widehat{Y}_T^i - Y_T^i)^2 + \lambda \left\|W\right\|_2 \tag{24}$$

where W and b represent the weights and biases to be learned, N is the number of training samples, λ is the hyper-parameter of L2 regularization, and Y_T^i denotes the slope and duration of time T. By feeding on the extracted dual features, the slope and duration of the stock price trend can be obtained.

IV. EXPERIMENTS

In this section, three different stock indexes from China's A-share market are chosen for experiments, including CSI 300, SSE 50 and CSI 500. We collected the stock market data including opening price, closing price, highest price, lowest price, trading volume and turnover with a 5-minute interval. The data covers the period from August 31, 2005, to August 31, 2018. In our experiments, we used CSI 300 data of 1,132 days including 150,336 data points, SSE 50 data of 3,133 days including 150,384 data points, and CSI 500 data of 3,042 days including 1,460,016 data points. We split these three data sets into training, validation, and testing set with ratio 8:1:1. With these stock index data sets, we conducted extensive experiments to evaluate the predictive performance of our proposed TPM and other models.

A. COMPARISON MODELS AND EVALUATION METRIC

The Support Vector Regression, LSTM, CNN and LSTM_CNN models are implemented for comparison. These models are described briefly as follows:

1) Support Vector Regression (SVR): the concatenating of the short-term features is feed to the SVR, and the parameters of radial basis functions (RBF) are set to c = 1, $\gamma = 0.1$ and d = 3. The prediction of the stock price trend will be generated with the RBF-based SVR.
2) LSTM: we implemented a recurrent neural network based on LSTM neurons and the hidden size is set to 64. We feed the long-term temporal features into LSTM to model and predict the stock price trend.
3) CNN: The short-term time series are used as input data, and the stock price trends are trained and predicted by a two-layer convolutional neural network with 3 × 3 convolution kernels.
4) LSTM_CNN [35]: it is a hybrid structure of CNN and LSTM. We feed the financial time series to this network and the CNN and LSTM extract and hybrid the features to learn and generate the prediction.
5) TPM_NC: We implemented the TPM_NC model by removing the CNN neurons from our TPM model. The encoder and decoder of TPM_NC model consist of LSTM neurons. The TPM_NC model encodes the short-term time series directly and decodes them with the long-term temporal features to predict the stock price trend.

We set the CNN of the TPM to consist of a 1×3 convolution kernel, and the number of LSTM neurons in the encoder and decoder is 64. The maximum training epochs of these models are set to 100. Meanwhile, we adopt the early stop method, the dropout layer with 0.5 dropout ratio, and the L2 regularization with $\lambda = 0.0001$ to prevent over-fitting.

In order to evaluate the performance of our TPM and other models in the trend prediction of financial time series, the root mean square error (RMSE) is used as evaluation metric. Specifically, assuming N is the number of samples, $\hat{Y}_t$ and Y_t denote the predicted value and the true value respectively at time t, we can calculate the RMSE by

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{Y}_t - Y_t\right)^2} \tag{25}$$

The lower the RMSE, the closer the predicted value is to the true value and the better performance of the model.

Based on these parameters settings, the market data of the three China Stock Indexes: CSI 300, SSE 50 and CSI 500 are trained respectively and their stock price trends are predicted separately.

B. EXPERIMENTS RESULTS

1) PREDICTION RESULTS

After analyzing the historical stock data of China's stock index CSI 300, SSE 50 and CSI 500, we set the maximum error thresholds of PLR method in long-term temporal features extraction as $\delta_{CSI\ 300} = 2.5$, $\delta_{SSE\ 50} = 1.6$, $\delta_{CSI\ 500} = 0.025$, and their time step lengths $T - 1$ all are 96. We conducted the experiments with the settings in Section A. and the experimental results are shown in Table 2.

It can be observed that in all these three data sets, our proposed TPM out-performs other models in predicting the

TABLE 2. The experimental results of different methods in three stock indexes.

Methods	CSI 300		SSE 50		CSI 500	
	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$
SVR	16.197	15.099	8.048	15.319	0.059	64.043
LSTM	9.425	9.676	6.268	8.010	0.062	26.609
CNN	9.857	8.974	6.277	9.161	0.062	30.669
LSTM_CNN	10.094	8.714	6.585	8.112	0.062	27.636
TPM_NC	8.620	8.595	5.801	7.970	0.060	27.210
TPM	**7.800**	**7.909**	**5.633**	**7.823**	**0.059**	**25.309**

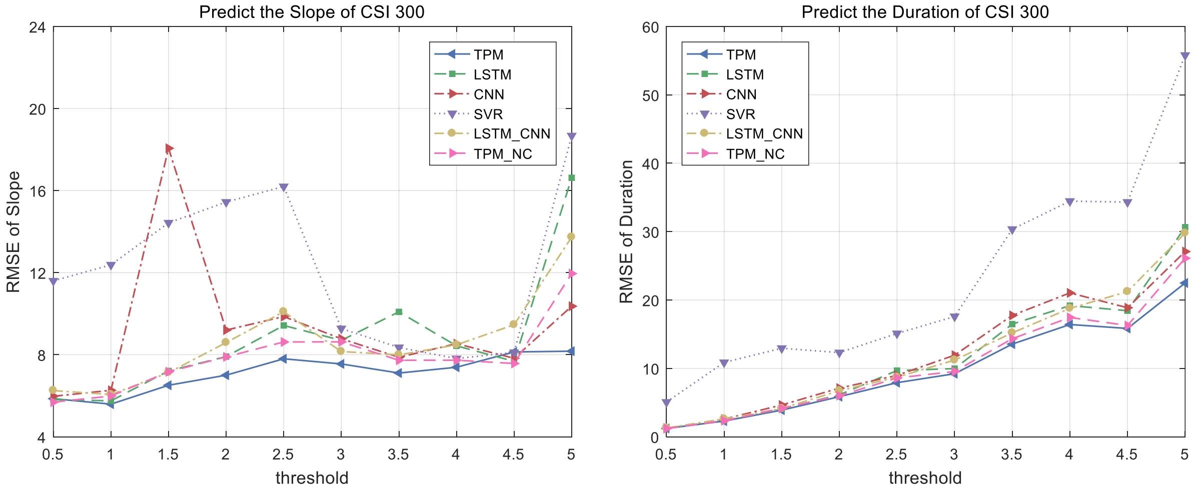


FIGURE 7. The slope prediction results with different thresholds of CSI 300 (left) and the duration prediction results with different thresholds of CSI 300 (right).

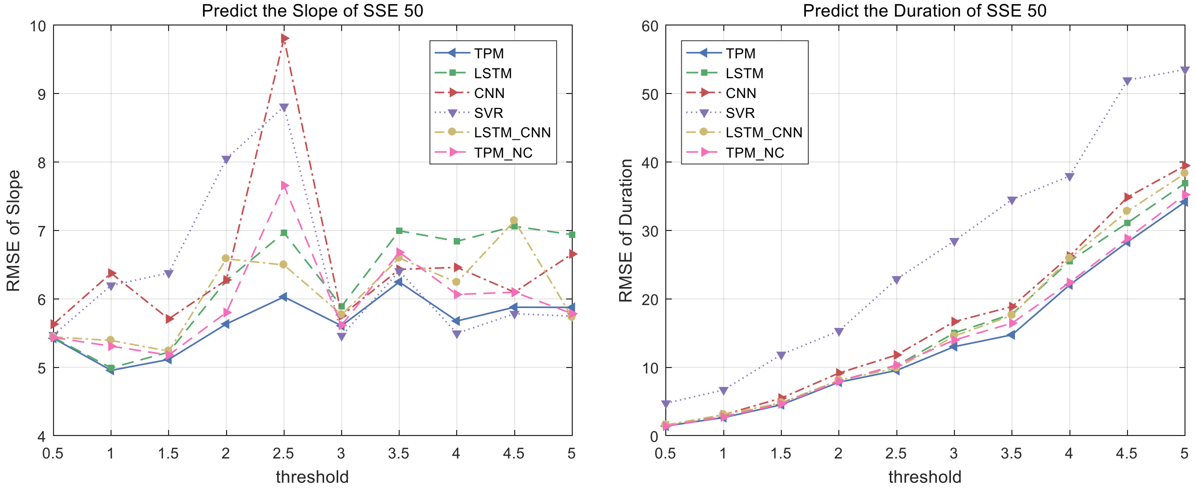


FIGURE 8. The slope prediction results with different thresholds of SSE 50 (left) and the duration prediction results with different thresholds of SSE 50 (right).

TABLE 3. The slope RMSE and duration RMSE of different methods with the time step length varies in CSI 300.

T-1	SVR		LSTM		CNN		LSTM_CNN		TPM_NC		TPM	
	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$
48	12.231	13.174	9.265	8.177	9.674	9.303	8.024	9.283	8.621	8.058	**8.330**	**7.785**
96	16.197	15.099	9.425	9.676	9.857	8.974	10.094	8.714	8.620	8.595	**7.800**	**7.909**
144	15.074	15.064	8.951	8.388	9.339	9.684	8.103	9.198	7.779	8.009	**8.308**	**7.929**
192	14.851	14.850	8.783	7.916	13.034	8.776	8.677	8.246	9.028	7.780	**7.921**	**7.491**
240	15.291	14.783	9.568	8.223	9.456	9.508	8.972	8.521	8.681	7.661	**8.493**	**7.816**

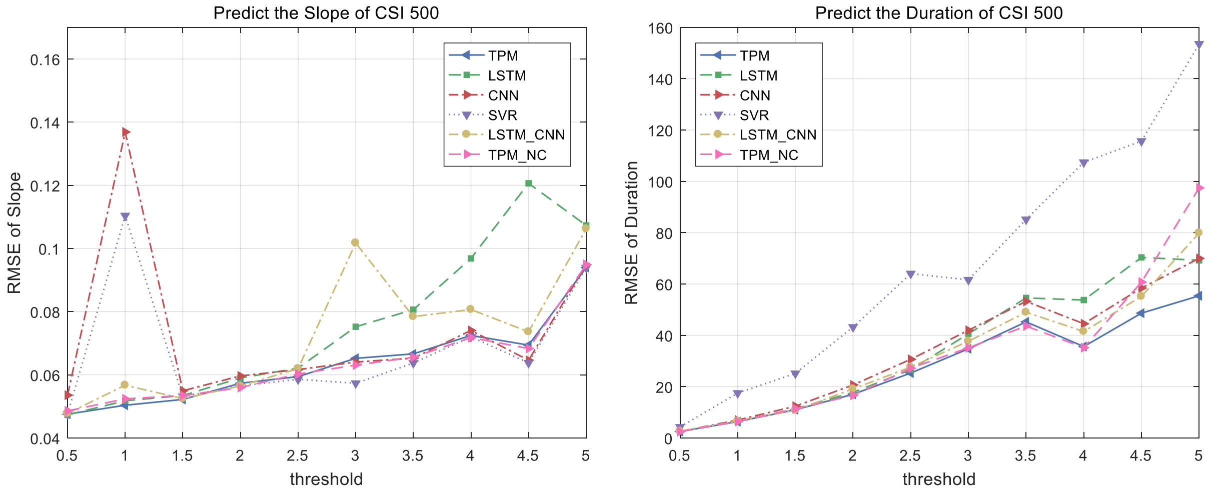


FIGURE 9. The slope prediction results with different thresholds of CSI 500 (left) and the duration prediction results with different thresholds of CSI 500 (right).

TABLE 4. The slope RMSE and duration RMSE of different methods with the time step length varies in SSE 50.

T-1	SVR		LSTM		CNN		LSTM_CNN		TPM_NC		TPM	
	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$
48	7.694	18.912	5.632	8.087	6.182	8.837	5.748	8.035	5.242	**7.653**	**5.240**	7.789
96	8.048	15.319	6.268	8.010	6.277	9.161	6.585	8.112	5.801	7.970	**5.633**	**7.823**
144	7.147	15.268	5.707	8.928	5.977	10.036	5.132	8.733	5.841	8.752	**5.190**	**8.516**
192	7.303	14.404	6.234	7.332	6.428	8.364	6.638	7.821	5.520	7.002	**5.508**	**6.951**
240	7.289	14.840	5.478	6.640	5.761	7.951	5.661	6.683	5.261	6.539	**5.003**	**6.391**

slope and duration of the trend. SVR shows the worst performance, while LSTM performs better, indicating that the model capturing the non-linear temporal relationship well will improve the performance. Analogously, CNN also performs better than SVR because of the spatial modeling ability. The LSTM_CNN achieves better performance than CNN and LSTM, validating the effectiveness of temporal and spatial modeling ability. Furthermore, the TPM outperforms the TPM_NC by a considerable margin since the CNN encoder enhances the predictive performance. The TPM combines the dual features with different time spans and optimizes feature selection and fusion operations through the dual-attention-based encoder-decoder network. In terms of the prediction of CSI 300, our TPM shows 9.51% and 7.98% improvements beyond the best baseline model on slope and duration prediction. It demonstrates that the dual features and dual attention mechanism can be successfully applied to the trend prediction problem and improve the prediction accuracy.

2) PREDICTION RESULTS WITH DIFFERENT ERROR THRESHOLDS

In the extraction process of long-term temporal features, the extracted features change with different maximum error thresholds δ. For the same data set, with the increase of the threshold value, the more data fluctuations are ignored and the fewer long-term features are extracted. Therefore, we set different thresholds δ according to the characteristics of the data set, and observe the influence of the threshold on the prediction accuracy. We set the time point lengths in three data all are 96 and the maximum error threshold varies by $\delta_{CSI300} \in [0.5, 5]$, $\delta_{SSE50} \in [0.4, 4]$, $\delta_{CSI500} \in [0.005, 0.05]$. The experimental results are shown in Fig.7-Fig.9.

It can be observed that as the threshold increases, the prediction performance of all models decreases. However, compared with other models, the proposed TPM has relatively low prediction error and is robust to the data variation.

3) PREDICTION RESULTS WITH DIFFERENT TIME STEP LENGTHS

The time step length of the input features is adjustable and the obtained prediction results are different with different time step lengths. As the time step length increases, more and more data can be fed to the model. Therefore, we conduct experiments with different time step lengths. The length of time points $T-1$ in three data set is chosen from the range $\{48, 96, 144, 192, 240\}$ which is $\{1, 2, 3, 4, 5\}$ days. Three maximum error thresholds each are set to $\delta_{CSI\ 300} = 2.5$, $\delta_{SSE\ 50} = 1.6$, $\delta_{CSI\ 500} = 0.025$. Table 3-5 shows the effect of the length on the model performance in the three stock indexes, respectively. With the prediction results of CSI 300 in Table 3, we find that when the length becomes longer, the RMSE of TPM decreases slightly and the improvement is little. Because the TPM focuses on the most relevant data which contains extensive market information. These data have a strong influence on the prediction results and increasing the amount of data will improve prediction performance

TABLE 5. The slope RMSE and duration RMSE of different methods with the time step length varies in CSI 500.

T-1	SVR		LSTM		CNN		LSTM_CNN		TPM_NC		TPM	
	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$	$RMSE_s$	$RMSE_d$
48	0.085	55.798	0.080	33.712	0.078	36.505	0.078	35.172	0.076	33.668	**0.075**	**32.389**
96	0.059	64.043	0.062	26.609	0.062	30.669	0.062	27.636	0.060	27.210	**0.059**	**25.309**
144	0.062	63.680	0.072	26.534	0.065	25.692	0.066	24.502	0.067	23.444	**0.064**	**22.537**
192	0.071	59.662	0.075	32.456	0.075	33.381	0.076	33.583	0.074	30.428	**0.072**	**28.799**
240	0.082	63.899	0.084	25.848	0.083	27.594	0.091	26.900	0.082	22.995	**0.082**	**22.312**

marginally. Compared to other models, the TPM model has a lower RMSE and has similar performance in predicting the SSE 50 and CSI 500 indexes, as shown in Table 4 and Table 5. It demonstrates that in stock price trend prediction, the TPM is successful with its high accuracy and robustness.

V. CONCLUSION

Traditional methods cannot extract relevant features for mining the financial time series. To address this issue, we propose a dual phase trend prediction model (TPM) based on dual features and dual attention mechanism for financial stock markets. First, in the data preprocessing phase, we use the PLR method and CNN to extract the dual features which represent the long-term trend of historical data and the short-term underlying market information. Second, in the time series modeling phase, we propose a new framework with short-term feature encoder and long-term feature decoder. We introduced the attention mechanism both in encoder and decoder so that the most relevant dimensions of features of all time points will be selected and merged adaptively. Finally, the TPM can accurately predict the slope and duration of the trend. Our experimental results show that our proposed TPM reduces the RMSE by 13.74% and 17.63% on average in comparison to other models, including SVR, LSTM, CNN, CNN_LSTM, and TPM_NC. In addition, experiments conducted with different thresholds and time step lengths demonstrate that the proposed TPM is not only better in prediction performance but also robust to time and data variation.

REFERENCES

[1] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, pp. 383–417, May 1970.

[2] R. Gencay, "Non-linear prediction of security returns with moving average rules," *J. Forecasting*, vol. 15, no. 3, pp. 165–174, Apr. 1996.

[3] S. M. Idrees, M. A. Alam, and P. Agarwal, "A prediction approach for stock market volatility based on time series data," *IEEE Access*, vol. 7, pp. 17287–17298, 2019.

[4] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 479–489, Jan. 2010.

[5] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Oct. 2014, pp. 1724–1734.

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

[8] R. C. Cavalcante, R. C. Brasileiro, V. L. F. Souza, J. P. Nobrega, and A. L. I. Oliveira, "Computational intelligence and financial markets: A survey and future directions," *Expert. Syst. Appl.*, vol. 55, pp. 194–211, Aug. 2016.

[9] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft. Comput.*, vol. 43, pp. 117–130, Jun. 2016.

[10] H. Dong, T. Li, R. Ding, and J. Sun, "A novel hybrid genetic algorithm with granular information for feature selection and optimization," *Appl. Soft. Comput.*, vol. 65, pp. 33–46, Apr. 2018.

[11] P. C. Chang, C. Y. Fan, and C. H. Liu, "Integrating a piecewise linear representation method and a neural network model for stock trading points prediction," *IEEE Trans. Syst., Man, C (Appl. Rev.)*, vol. 39, no. 1, pp. 80–92, Jan. 2009.

[12] L. Zhao and L. Wang, "Price trend prediction of stock market using outlier data mining algorithm," in *Proc. IEEE 5th Int. Conf. Big Data Cloud Comput.*, Aug. 2015, pp. 93–98.

[13] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 1, pp. 259–268, 2015.

[14] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7046–7056, Nov. 2015.

[15] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An adaptive SVR for high-frequency stock price forecasting," *IEEE Access*, vol. 6, pp. 11397–11404, 2018.

[16] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," *Expert Syst. Appl.*, vol. 80, pp. 340–355, Sep. 2017.

[17] M. Thakur and D. Kumar, "A hybrid financial trading support system using multi-category classifiers and random forest," *Appl. Soft Comput.*, vol. 67, pp. 337–349, Jun. 2018.

[18] S. K. Chandar, "Fusion model of wavelet transform and adaptive neuro fuzzy inference system for stock market prediction," in *Journal of Ambient Intelligence and Humanized Computing*. Berlin, Germany: Springer, 2019. doi: 10.1007/s12652-019-01224-2.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*, [Online]. Available: https://arxiv.org/abs/1810.04805

[23] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 2327–2333.

[24] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2627–2633.

[25] D. M. Nelson, A. C. Pereira, and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1419–1426.

[26] X. Zhang, X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, "Stock market prediction via multi-source multiple instance learning," *IEEE Access*, vol. 6, pp. 50720–50728, 2018.

[27] 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.

[28] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, "An innovative neural network approach for stock market prediction," in *The Journal of Supercomputing*. Springer, 2018. doi: 10.1007/s1122.

[29] J. J. Murphy, *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications*. New York, NY, USA: Prentice-Hall, 1999.

[30] C.-H. Su and C.-H. Cheng, "A hybrid fuzzy time series model based on ANFIS and integrated nonlinear feature selection method for forecasting stock," *Neurocomputing*, vol. 205, pp. 264–273, Sep. 2016.

[31] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59.

[32] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 8609–8613.

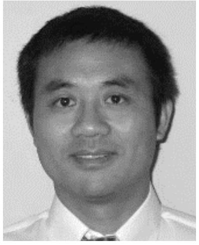
[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[35] T. Lin, T. Guo, and K. Aberer, "Hybrid neural networks for learning the trend in time series," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2273–2279.

YINGXUAN CHEN received the B.Eng. degree in computer science and technology from North China Electric Power University, in 2017. She is currently pursuing the M.S. degree with the School of Computer Science and Engineering, South China University of Technology. Her research interests include deep learning and machine learning.

WEIWEI LIN received the B.S. and M.S. degrees from Nanchang University, in 2001 and 2004, respectively, and the Ph.D. degree in computer application from the South China University of Technology, in 2007, where he is currently a Professor with the School of Computer Science and Engineering. He has published more than 80 articles in refereed journals and conference proceedings. His research interests include distributed systems, cloud computing, big data computing, and AI application technologies. He is a Senior Member of CCF.



JAMES Z. WANG received the B.S. and M.S degrees in computer science from the University of Science and Technology of China and the Ph.D. degree in computer science from the University of Central Florida. He is currently a Professor with the School of Computing, Clemson University, SC, USA. His research interests include storage networks, database systems, distributed systems, cloud computing, and multimedia technologies. He is a Senior Member of ACM.