Neurocomputing 383 (2020) 122–137

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multistage attention network for multivariate time series prediction

Jun Hu[a,b,*], Wendong Zheng[a]

[a] College of Computer Science and Electronic Engineering, Hunan University, PR China
[b] State Key Lab. for Novel Software Technology, Nanjing University, PR China

ARTICLE INFO

Article history:
Received 26 April 2019
Revised 29 September 2019
Accepted 12 November 2019
Available online 4 December 2019

Communicated by Wei Chiang Hong

Keywords:
Attention mechanism
Multivariate time series prediction
Long short-term memory

ABSTRACT

The deep learning model has been used to predict the variation rule of the target series of multivariate time series data. Based on the attention mechanism, the influence information of multiple non-predictive time series on target series in different time stages is processed as the same weight in the previous studies. However, on real-world datasets, multiple non-predictive time series has different influence (such as different mutation information) on target series in different time stages. Therefore, a new multistage attention network is designed to capture the different influence. The model is mainly composed of the influential attention mechanism and temporal attention mechanism. In the influential attention mechanism, the same and different time stage attention mechanisms are used to capture the influence information of different non-predictive time series on the target series over time. In the temporal attention mechanism, the variation law of data can be captured over time. Besides, the prediction performance of proposed model on two different real-world multivariate time series datasets is comprehensively evaluated. The results show that, the prediction performance of the proposed model beat all baseline models and SOTA models. In a word, the multistage attention network model can effectively learn the information of the influence of different non-predictive time series on the target series in different time stages in the historical data.

1. Introduction

Recently, machine learning algorithms have been frequently applied to multivariate time series prediction tasks [1], such as traffic flow forecasts [2], weather forecasts [3] and financial market forecasts [4]. The classical machine learning model GBRT [5] is suitable for dealing with nonlinear short-term time series prediction. However, high computational overhead is required for the convergence of machine learning algorithms in processing of tens of thousands of records of multivariate time series data. To improve the prediction accuracy of the model, recurrent neural network (i.e., LSTMs/GRUs) was used to learn the time dependence [6–7]. In multivariate time series prediction tasks, different effects of multiple non-predictive time series on the target series cannot be fully captured by standard LSTM. Inspired by the attention learning mechanism of human beings, the two-stage attention network (DA-RNN) [8] was proposed first to solve the multivariate time series prediction problem of the above complex change rules. In the encoder stage, the DA-RNN model uses the input feature attention mechanism to select the most relevant series and learn the correlation information generated by the series. Meanwhile, the DA-RNN model used the temporal attention mechanism in the decoder stage to capture the variation rule of the historical data of the target series. Subsequently, the multi-view attention mechanism [9] was introduced into the encoder-decoder, so that the influence of multiple non-predictive time series on the target series was observed from the perspective of different data sources. Then, inspired by the multi-view attention mechanism, GeoMAN model [10] was proposed. The local attention mechanism was used to learn the time series data changes over time in each sensor, and the global attention mechanism of the model also captured the correlation information between different time series. However, the dynamic influence of multiple non-predictive time series over time on target series is ignored in the above three attention network models. To this end, the dynamic change of influence information is learned in this study by designing specific attention networks according to the change of influence information in different time stages.

Multivariate time series prediction has three major challenges. The complex dynamic relationships are demonstrated within

* Corresponding author at: College of Computer Science and Electronic Engineering, Hunan University, PR China.
E-mail addresses: hujun_111@hnu.edu.cn (J. Hu), wendongz@hnu.edu.cn (W. Zheng).

https://doi.org/10.1016/j.neucom.2019.11.060
0925-2312/

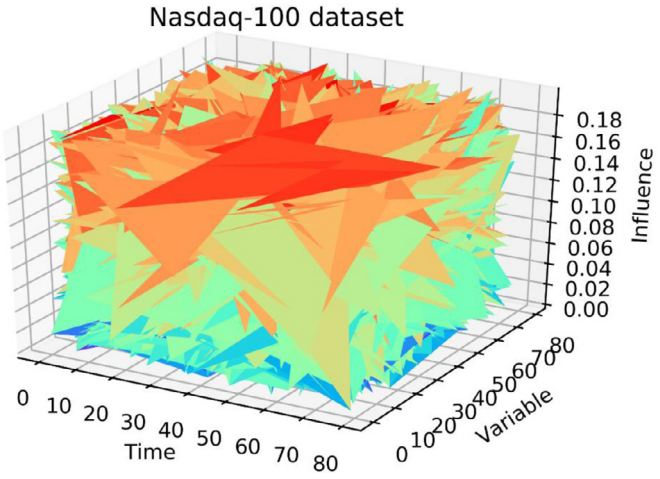


Fig. 1. The complex correlations in the Nasdaq 100 dataset.

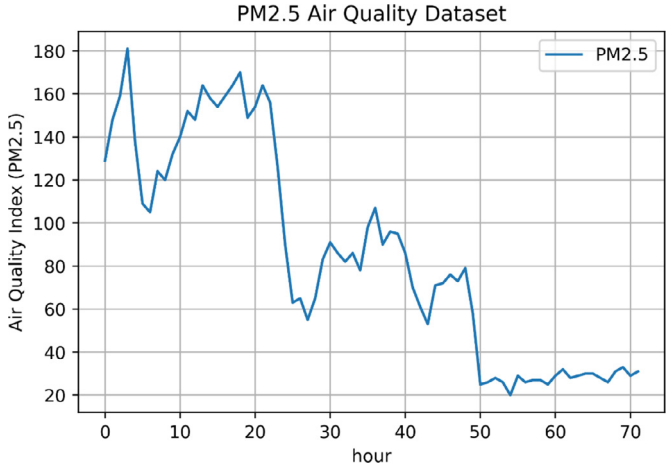


Fig. 2. Target sequence curve of PM2.5 Air Quality dataset.

multivariate time series data through 3D and 2D visualization techniques.

(1) Dynamic temporal correlation of multivariate time series. In a multivariate time series, there is a strong temporal correlation between the data at the time before and after each sequence. During the same time period, there is strong interaction information between non-predicted sequences. Moreover, each of these multiple time variables has different impacts on the target series. The information of mixed influence factors will be lost in the simple pairwise independent analysis of the traditional model, resulting in the insufficient generalization of the model.
(2) The dynamic impact of multiple time variables. The target series in a multivariate time series is affected by time and other non-predictive time series. Therefore, influence of multiple time variables on the target series varies with time. Fig. 1 shows complex temporal correlation relationships in the above two aspects.

As shown in Fig. 1, the three-dimensional graphics are composed of influential three-dimensional data of multivariable time series data obtained by the model pre-training, instead of simple (*x, y, z*) three-coordinate data. In the Numpy, the data shape is represented as [81, 81, 81], which represents 81-time steps, 81 multivariate series, and the influence values of each sequence. From the cross-section of time and influence axis, the color of the influence continuously changes from dark-blue to red at different time stages, indicating the dynamic impact of multivariate data generation over time. Then, from the cross-section of variables and influences, the color of the influence information generated by different variable sequences varies dynamically. It implies that there is different importance of the impact information generated by different variables.

(1) Mutations effect in multivariate time series data on the target series. Based on previous studies, it is known that some mutations in the multivariate time series can affect the trend of the target series (predicted attribute column). In the study of the mutation, we were inspired by the study of multivariate time series data, detection of outlier and novelty points. In practice, the sudden change point can be regarded as the outlier point with significant change over time. Since the above features cannot be fully captured by existing attention networks, a new attention network should be designed to solve the above problems.

More specifically, the phenomenon of a sudden change is visualized in the PM2.5 Air Quality dataset over a certain stage of time. First, the sequence of the target is defined for a time stage: $\boldsymbol{y} = \{y_1, y_2, \cdots, y_T\}, \boldsymbol{y} \in \mathrm{R}^T$. For the PM2.5 Air Quality dataset, the definitional domain of T is defined as [0, 71] for a total of 72 h. As shown in Fig. 2, a significant mutation process can be observed. For example, from 0 - 1 min, PM2.5 index rises rapidly from 130 points to 150 points. Within the 21st minute to the 24th minute, the index value quickly decreases from 157 points to 62 points. Considering the above-mentioned process of sharp rise and rapid decline, a simple formal description of the mutation phenomenon is obtained in the PM2.5 Air Quality dataset. On the one hand, there is a sudden increase $\Delta_{up} = |y_{i+1} - y_i| > 20$ in PM2.5, where Δ_{up} represents the change amount in the rise of the target series in a certain interval. In addition, there is a sudden decrease in the PM2.5 index, namely $\Delta_{down} = |y_{i+1} - y_i| > 20$, the change amount Δ_{down} represents the falling value in a certain interval.

Previous studies (such as the time series de-trending in economics) have formally described the correlation between the complex dynamic multivariate time series data [11]. The law of data variation can be extended for multivariate time series prediction tasks. Firstly, components of the multivariate time series $x_{k,t}$ are considered as Eq. 1:

$$x_{k,t} = u_{k,t} + w_{k,t} \ \forall t, k \tag{1}$$

where the new time series $u_{k,t}$ represents the trend term of the kth column time series over time; $w_{k,t}$ represents the noise term and the cyclic term in the original time series [12]. Before looking for general trend items, the above three challenges that encountered into a formal definition are organized such as problem 1.

Problem 1. Given a triple $A = \{X, S, d\}$, where $X \epsilon \mathrm{R}^{K \times T}$ is the information collected by the K multiple time series in T time steps, $S \epsilon \mathrm{R}^{K \times K}$ denotes a matrix of $K \times K$ dimensional storage multivariate time series information, and d is a vector storing the mapping relationship between X and S. $\hat{X}_{i,t}$ is estimated at the tth moment of the ith non-predicted series, and then the value of $l(X_{i,t}, \hat{X}_{i,t})$ is calculated between the estimated value and the observed value. The value of $l(X_{i,t}, \hat{X}_{i,t})$ is also used to judge whether it is the "mutation index" of the mutation point, where $l(\cdot, \cdot)$ is an evaluation indicator function.

$x_{k,t}$ is used to define the kth non-predicted time series at time t, and $x_{k,t}$ is an instance that can be stored in the data matrix X. $\boldsymbol{x}_k$ is used to represent all the multivariate time series data of the kth column in the window for a period of time $\boldsymbol{x}_k = \{x_{k,1}, ..., x_{k,T}\}$. For the same reason, the trend term of the multivariate time series is presented as $\boldsymbol{u}_k = \{u_{k,1}, ..., u_{k,T}\}$. Eq. (2) can be used to sim-

ply look for trend items and minimize the empirical risk of the following objective functions.

$$\min \sum_{k=1}^{K}\sum_{t=1}^{T}\left(x_{k,t}-u_{k,t}\right)^2+\lambda\sum_{i=1}^{K}\sum_{j=1,j\neq i}^{K}\sum_{t=1}^{T}\left(\nabla_t^2\left(u_{i,t}-C_{ij}u_{j,t}\right)\right)^2 \tag{2}$$

where ∇_t^2 represents the second order difference operator. The weight parameter λ, called as the smoothing parameter, is usually used to adjust the weight value in line with the characteristics of the dataset, so as to guarantee the best performance of the model. Furthermore, C_{ij} represents the co-variance matrix of the i th time series and the jth time series. The association information between different time series is presented by the latter term of Eq. (2) (more details can be found in [13]). Then the deep learning model is used to capture different effects of multivariate time series data on the target series at different time stages.

This paper aims to predict multivariate time series with complex correlations with dynamic changes. The influence of these mutation points on the target series is explored to improve the accuracy of the prediction task. Therefore, based on the above formal analysis, a deep learning model is proposed in line with multistage attention mechanism and transformation gated LSTM network in encoder, so as to capture the different effects information of multivariate time series on target series at different time stages. Finally, the change rule of the target series is learned accurately and the value prediction of the target series in the future time is realized. Full details of the proposed model are described in Section 3. The main contributions of this paper are as follows:

(1) A novel deep learning model is constructed with a multistage attention mechanism to accomplish multivariate time series prediction tasks. The proposed model can adaptively capture complex dynamic associations between multiple non-predictive time series and target series by processing the impact information of all non-predictive time series on the target series at different time stages.
(2) The richer impact information, more relevant time series input features and time-dependent information should be captured to enhance the performance of the attention mechanism. Thus, the attention score adjustment module is introduced in the multi-stage attention mechanism inside the encoder-decoder.
(3) There are a huge mutation points in the multivariate time series with complex dynamic correlation. These mutation points may significantly affect the future change of the target series. Based on our previous research, the LSTM network with transformation gating is introduced into the encoder to replace the standard LSTM. Thus, the learning ability of mutation information is enhanced within the neural network.
(4) Empirically, the proposed multi-stage attentional network model defeats all baseline models on two different real-world datasets. Meanwhile, the proposed model can be widely used in multivariate time series prediction in different task.

2. Related works

The time series prediction is an important and challenging issue. The well-known autoregressive moving average (ARMA) model [14] has a good performance in different time series prediction tasks, while nonlinear problems cannot be effectively solved. To this end, multiple nonlinear autoregressive derivation (NARX) models [15] are put forward. The recurrent neural networks (RNNs) [16] in the deep learning have received much attention due to its flexibility in capturing nonlinear relationships. However, long-term dependencies cannot be exactly captured due to the gradient disappearance. Long short-term memory (LSTM) [6] networks and gated recurrent unit (GRU) [7] neural networks have overcome this deficiency. Subsequently, the encoder-decoder combined with LSTM is proposed to improve the performance of the predictive model [17].

To effectively capture the long-term dependencies of encoder-decoder, attention model has been proposed for the first time [18]. Later, researchers developed multi-level attention-based network models to select relevant features and encoder hidden states [19–21]. To efficiently obtain the temporal correlation of driving series, DA-RNN is proposed in this paper to capture the relevant driving series at each time step by using input attention and temporal attention [8]. Multiple attention scores can be generated from multi-view to further obtain complex temporal correlations on a multivariate time series dataset [9]. To predict the time series, this study proposes a multi-level attention mechanism to obtain complex spatiotemporal correlations [10]. However, none of the above models above can accurately capture different effects of different time series at different time stages, which is the key point for the multivariable time series prediction in the real-world tasks. In addition, this study initially explored how to learn mutations in time series [3]. However, the learning module is not included specifically for multivariate time series data mutation information in previous studies. To the best of our knowledge, when the standard LSTM is encountered with a sudden change process, the predicted value will deviate significantly from the true value.

3. Multistage attention model

3.1. Notation

Attribute columns of our dataset are composed of non-predictive series and target time series. Given n time-varying series, i.e., $= (\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^n)^\top = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T) \in \mathrm{R}^{n\times T}$, where T is the size of time window. Contents of a non-predictive series of size T (i in the formula represents the $\mathrm{i}-th$ over time series) are represented by $\mathbf{x}^{\mathrm{i}} = (x_1^i, x_2^i, \cdots, x_T^i)^\top \in \mathrm{R}^T$. Besides, $\mathbf{x}_t = (x_t^1, x_t^2, \cdots, x_t^n)^\top \in \mathrm{R}^n$ is used to define a 1d-tensor (vector) of n time series at time t.

Given the previous values of each non-predictive time series, values of the target series over next k hours are predicted as $\hat{y} = (\hat{y}_{T+1}, \hat{y}_{T+1}, \cdots, \hat{y}_{T+k})^\top \in \mathrm{R}^k$.

3.2. Model framework

Fig. 3 shows the main frame structure of the proposed approach. Specifically, in the encoder, two different attention mechanisms are implemented, i.e., the influence attention mechanisms for the same time stage and the different time stages, respectively. For multi-stage “Influence Att”, the data are first fed to the “Same Att” and the resulting values to the “Diff Att” for processing, and then the information is connected through the “concat” operation. Note that a new module of attention score adjustment is inwardly contained, so as to effectively capture different effects of varying time series on the target series at different time stages. After the above operation, the information obtained by the influence attention mechanism and the hidden state information are passed as input information to the LSTM with transformation gating (i.e., “TG-LSTM”). TG-LSTM can map the multivariate time series input information to the most obvious range of values through internal transformation gating, so as to enhance the capture ability of data changes and even mutation processes (see Section 3.4 for more details). In the decoder, the temporal attention mechanism

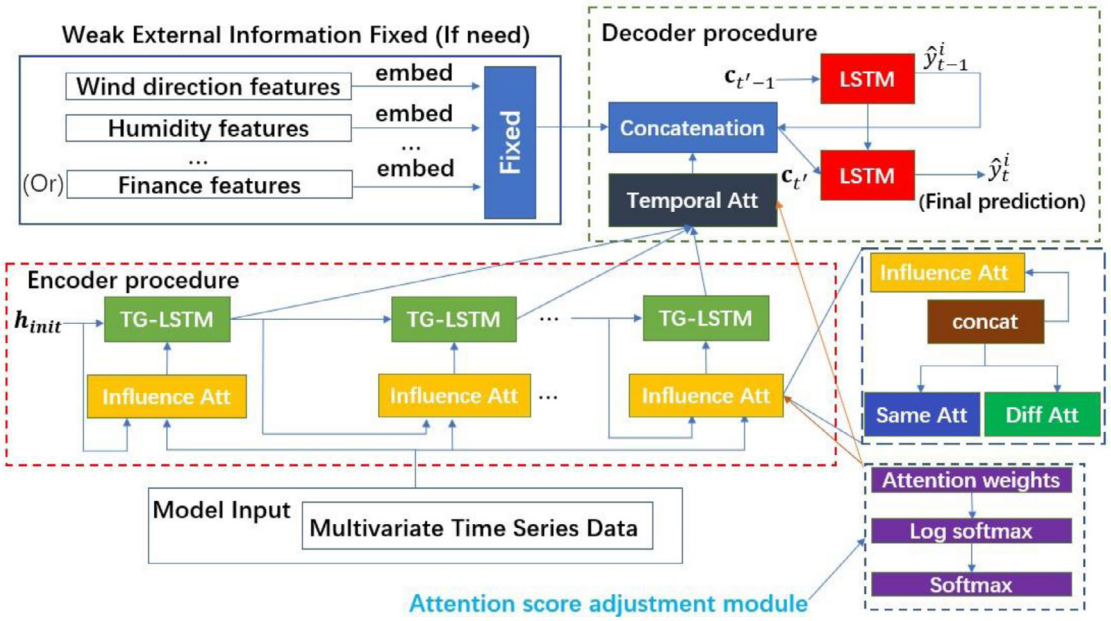


Fig. 3. Framework of the proposed model.
h_{init}: the initial state in encoder.
Att: Attention.
$\boldsymbol{c}_{t'}$: the context vectors at time t'.
TG-LSTM: the LSTM with transformation gating./
Note: The content inside the red dashed box is the main structure of the encoder. The content inside the green dashed box is the main structure of the decoder.

is employed with the attention score adjustment module to adaptively select the relevant previous time steps for the prediction. Considering that there are some weaker external factors in certain multivariate time series prediction tasks, the embedding layer is used in the decoder to add the relevant weak external information (the necessary external factors) in the proposed model. The information are generated by the embedded layer "Fixed", the results by the "Temporal attn" and $\hat{y}^i_{t-1}$ by the concatenation operation, and predicted results $\hat{y}^i_t$ by the standard LSTM.

3.3. Attention score adjustment module

Typically, the attention mechanism uses the traditional Softmax function to calculate the attention weight score. Note that the premise of Softmax function is that only a few related elements should be captured for the attention mechanism as the goal. However, in the context of multivariable time series, such premise does not hold, since multiple time series will provide different impact information to the target series. To address this issue, a new attention score adjustment module is designed to make full use of multivariate time series information. At the same time, this module can bring more rich influence information from the non-predictive time series to the target series for the standard attention mechanism. More formally, we start with a log_Softmax operation on x^i_t, which is mathematically equivalent to taking the logarithm of the Softmax function, in Eq. (3). Log_Softmax function, which can be used to log the standard Softmax, is an effective way to compress the difference in attention weighting scores between different input time series. After the processing, the difference between the attention score of sequences with small weight before processing and those with large attention score before processing is narrowed. In this way, more related time series are selected by the attention mechanism (Because there are dozens of dimensions on multivariate time series dataset, some attention scores in the time series of intermediate value may be lost). The smooth $\tilde{x}^i_t$ with a lot of impact information is obtained, and then the softmax function is used to calculate the final attention score δ^i_t, as follows Eq. 4:

$$\tilde{x}^i_t = \log\left(\frac{\exp\left(x^i_t\right)}{\sum_{j=1}^{n}\exp\left(x^j_t\right)}\right) \tag{3}$$

$$\delta^i_t = \frac{\exp\left(\tilde{x}^i_t\right)}{\sum_{j=1}^{n}\exp\left(\tilde{x}^j_t\right)} \tag{4}$$

3.4. LSTM with transformation gating mechanism

Although more relevant multivariate time series are screened by the multi-stage attention mechanism to generate different impact information at different time stages as input to the encoder, the standard LSTM network in DA-RNN still has the weak capture ability of short-term mutation information. In addition, the activation function (both sigmoid and tanh) inside LSTM has a certain supersaturation region, so that the input information flow cannot change significantly in the value range of the supersaturated region. To enhance the learning ability of LSTM network for the long-term dependencies, this study designs a LSTM where a transformation gating mechanism is added. Relevant research shown that, the long-term dependence learning effect of the LSTM model can be increased by adding Dropout and Zoneout technology to the input gate, forget gate or output gate to lose part of the information flow. The mechanism of the proposed transformation gating is novel. Particularly, it is different from attenuation technology using Zoneout method (In the attenuation technology, the update rule of input and output and forget gate is multiplied by the attenuation coefficient. The information flow into the neural network is lost randomly in proportion to the attenuation coefficient. However, these missing information flow may contain important features). In the proposed transformation gating, the $1-tanh$ function is used to transform the value range of the forget gate output, so that the important characteristics of the input information flow are preserved completely. Besides, the information flow can be compressed into a significant change interval to enhance the capture

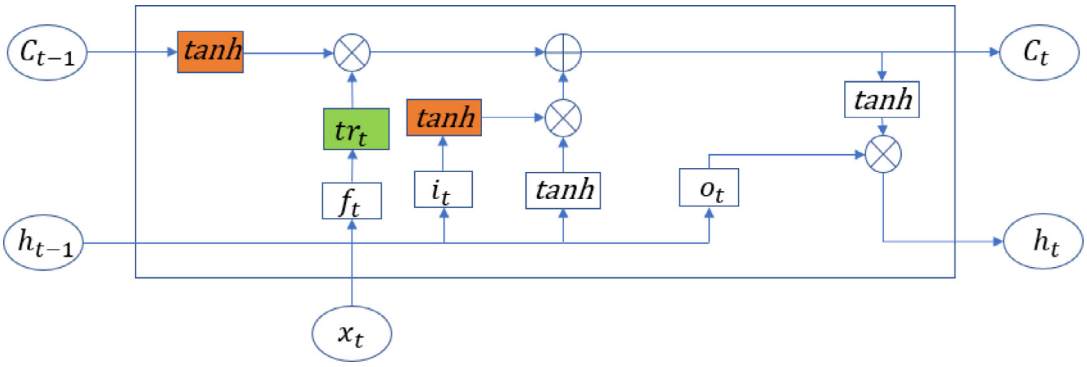


Fig. 4. TG-LSTM.

ability of changing information. The TG-LSTM process is defined as follows:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \tag{5}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{6}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \tag{7}$$

$$g_t = \tanh\left(W_g x_t + U_g h_{t-1} + b_g\right) \tag{8}$$

$$tr_t = 1 - tanh(f_t) \tag{9}$$

$$c_t = tr_t \odot tanh(c_{t-1}) + \tanh(i_t) \odot g_t \tag{10}$$

$$h_t = o_t \odot tanh(c_t) \tag{11}$$

From Eqs. 5 and 6 to Eq. (7), i_t, f_t, and o_t represent the values of the input gate, the forget gate, and the output gate processed at time t by the activation function σ (i.e., hard sigmoid is used to speed up the training process). W_i, W_f, and W_o represents the input weight matrix of the corresponding gate, respectively; U_i, U_f, and U_o represents the recurrent weight matrix, and b_i, b_f; and b_o represent the bias. x_t and h_{t-1} represents the input feature information at time t and the hidden state information at the previous t-1 time, respectively. g_t in Eq. (8) is an output value obtained by performing the element-wise dot multiplication by x_t and h_{t-1}, respectively, and then it is activated with the tanh function. c_t in Eq. (10) represents the output value of the memory cell state at time t; and $\odot$ indicates that the matrix is multiplied element-wise. Finally, the hidden state information at time t is obtained by Eq. (11). The internal structure of the TG-LSTM is illustrated in Fig. 4. The green rectangle indicates the introduced transformation gating, and the two orange-yellow rectangles are the introduced activation functions.

Note that the transformation gating can change the value range of the forget gate outputs, so that the dependencies between the short-term information of the data are enhanced, while the long-term dependencies can be preserved in Eq. (9). Fig. 5 shows the specific process of the range change of functions. The output rule of the forget gate is known as follows, when the sigmoid function is activated, the values will be completely discarded if it is close to 0; values will be passed completely if it is close to 1. Thus, the value range of the output of the forget gate is [0, 1], and the output of the forget gate is mapped to the interval of [0.25, 1.0] by the proposed transformation gate. After transformation gate, the value close to 1 will be converted to the value smaller to 0.25; value close to 0 will be converted to the value close to 1. Note that the values near the middle will be compressed to 0.5. Compressing the value range of the data to the most obvious interval is more conducive to capturing the dependencies between the data. To compress the output information of c_{t-1} and i_t to the significant interval, the tanh function is also used in Eq. (10). TG-LSTM is used in the encoder as a nonlinear transformation function in Section 3.5.

3.5. Encoder with multistage influence attention

First, the influence attention mechanism of the same time stage is introduced. In practice, the target series is often affected differently by other time series, leading to extremely complex impact information. To this end, given the i-th input feature vector $\mathbf{x}^i = (x_1^i, x_2^i, \cdots, x_T^i) \in \mathrm{R}^T$, we use this attention mechanism to adaptively capture the different influence between the target series and each time series features with Eq. 12:

$$e_t^i = \mathbf{v}_s^\top \tanh\left(\mathbf{W}_s[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_s \mathbf{x}^i + \mathbf{b}_s\right) \tag{12}$$

where $[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}]$ is a concatenation operation of the previous hidden state $\mathbf{h}_{t-1}$ and the cell state $\mathbf{s}_{t-1}$. $\mathbf{v}_s$, $\mathbf{b}_s \in \mathrm{R}^T$, $\mathbf{W}_s \in \mathrm{R}^{T\times\ 2m}$ and $\mathbf{U}_s \in \mathrm{R}^{T\times T}$ are parameters that need to be learned by the model. The attention score adjustment module is used to obtain new attention weights. The final attention score α_t^i represents the importance of the influence contributed by each time series. The output vector for the influence attention mechanism of the same time stage at time t is calculated as Eq. 13:

$$\tilde{\mathbf{x}}_t^{same} = \left(\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \cdots, \alpha_t^n x_t^n\right)^\top \tag{13}$$

Actually, the influence degree can change dynamically with time in the real world. Hence, a new influence attention mechanism is designed to capture the dynamic influence between different time series. Given our predictive series as target series and another time series l, we compute the attention weight (i.e., the influence weights) as follows Eq. 14:

$$d_t^l = v_{dif}^\top \tanh\left(\mathbf{W}_{dif}[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}]\right) + \mathbf{U}_{dif} y^l + \mathbf{W'}_{dif} \mathbf{x}^l \mathbf{u}_{dif} + \mathbf{b}_{dif} \tag{14}$$

$\mathbf{W'}_{dif} \in \mathrm{R}^{T\times n}$ are parameters that need to be learned by the model. Considering the target series and the features of the same time stages of other time series, this attention mechanism can adaptively select relevant time series for prediction. Furthermore, $[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}]$ is obtained by concatenating the previous hidden state $\mathbf{h}_{t-1}$ with the previous cell state $\mathbf{s}_{t-1}$ in the encoder; and the extension of the historical time stage information in the time step is fully considered. We also use the attention score adjustment module in this attention mechanism. When attention scores are calculated by the attention score adjustment module (i.e., the influence weights, $\beta_t^l, l \in \{1, \cdots, n\}$), the output vector of the different stage the influence attention is calculated as follows Eq. 15:

$$\tilde{\mathbf{x}}_t^{dif} = \left(\beta_t^1 y_t^1, \beta_t^2 y_t^2, \cdots, \beta_t^n y_t^n\right)^\top \tag{15}$$

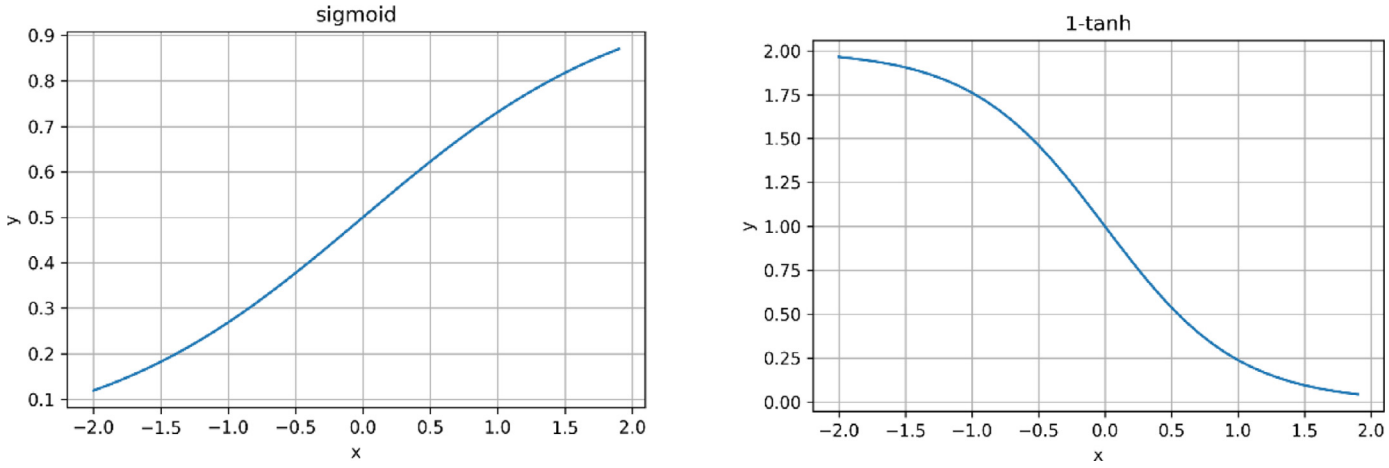


Fig. 5. Sigmoid activation function and 1-tanh transformation function.

We simply merge output vectors, which are obtained from the same time stage influence attention mechanism and the different stages influence attention mechanism as follows Eq. 16:

$$\tilde{x}_t = \left[\tilde{\mathbf{x}}_t^{same}; \tilde{\mathbf{x}}_t^{dif}\right] \tag{16}$$

where $\tilde{x}_t \in \mathrm{R}^{n+n}$ is the output vector obtained after merging. The combined $\tilde{x}_t$ is fed to the encoder as a new input, using both the previous encoder hidden state $\mathbf{h}_{t-1}$ and $\tilde{x}_t$ into $\mathbf{h}_t = f_{en}(\mathbf{h}_{t-1}, \tilde{x}_t)$ to get the current hidden state $\mathbf{h}_t$. In addition, the f_{en} is a TG-LSTM network.

3.6. Decoder with temporal attention

Previous studies have shown that, the performance of an encoder-decoder decreases rapidly as the length of the encoder increases. Here, a time attention mechanism with an attention score adjustment is used to mitigate this performance degradation, which adaptively selects the relevant hidden state from the encoder and obtains more hidden state information. Thus, the model can capture the dynamic temporal correlation of the target series. More specifically, we compute the attention vector at each output time t' over each hidden state of the encoder, as follows Eq. 17:

$$u_{t'}^k = \mathbf{v}_d^\top \tanh\left(\mathbf{W}_d'\left[\mathbf{d}_{t'-1}; \mathbf{s}_{t'-1}'\right] + \mathbf{U}_d \mathbf{h}_k + \mathbf{b}_d\right) \tag{17}$$

$$\mathbf{c}_{t'} = \sum_{k=1}^{T} \gamma_{t'}^k \mathbf{h}_k \tag{18}$$

where $\mathbf{W}_d' \in \mathrm{R}^{m\times 2p}$, $\mathbf{U}_d \in \mathrm{R}^{m\times m}$ and $\mathbf{v}_d, \mathbf{b}_d \in \mathrm{R}^m$ are learned by the model. The attention score adjustment module is used to calculate the temporal attention scores $\gamma_{t'}^k$. The temporal attention mechanism calculates the context vector $\mathbf{c}_{t'}$ as a weighted sum of all encoder hidden states in Eq. (18). The context vector $\mathbf{c}_{t'}$ is combined with the previous output of decoder $\hat{y}_{t'-1}$ and the weak external information $w\mathbf{x}_{t'}$ (if need) to update the decoder hidden state by $\mathrm{d}_{t'} = f_{de}(\mathrm{d}_{t'-1}, [\hat{y}_{t'-1}; \mathbf{c}_{t'}; w\mathbf{x}_{t'}])$, where f_{de} is a LSTM network in decoder. Then a new hidden state is generated through the concatenation operation of the context vector $\mathbf{c}_{t'}$ and the hidden state $\mathrm{d}_{t'}$, the final prediction results are obtained as Eq. 19:

$$\hat{y}_{t'} = \mathbf{v}_y^\top(\mathbf{W}_y[\mathrm{d}_{t'}; \mathbf{c}_{t'}] + \mathbf{b}_w) + b_y \tag{19}$$

where the parameters matrix $\mathbf{W}_y \in \mathrm{R}^{p\times(p+m)}$ and the vector $\mathbf{b}_w \in \mathrm{R}^p$. The $[\mathrm{d}_{t'}; \mathbf{c}_{t'}] \in \mathrm{R}^{p+m}$ is a concatenation of the decoder hidden state and the context vector. Finally, this linear function is used to generate the final prediction result.

3.7. Training procedure

Although the Adam optimization algorithm [22] is widely applied in different fields, recent research indicates that there is a convergence problem under certain circumstances. In response to such problems, our previous research proposed an adaptive first-order stochastic optimization algorithm called AdaHMG [23] to solve the convergence problem of Adam, and there is the significant accuracy improvement in simple time series prediction tasks. It is worth noting that the AdaHMG algorithm considers the squared gradient information before and after time in the calculation of the second-order moment estimation. This algorithm is used to capture the correlation information within the data, especially on time-critical tasks with strong temporal correlation (More detailed time complexity analysis and algorithm convergence analysis can be referred to our previous research.). Therefore, in this paper we further extend the application range of the AdaHMG algorithm, try to use it on multivariable time series data with more complex correlation. In the next Section 4.3, the mean absolute error (MAE) performance is illustrated on the validation set for the AdaHMG optimization algorithm.

The proposed method is smooth and differentiable, so all parameters can be learned with mean squared error (MSE) as a loss function which is widely used in time series prediction. The loss function is obtained as Eq. 20:

$$\mathcal{L}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_t^i - \hat{y}_t^i\right)^2 \tag{20}$$

where n represents the number of training samples; y_t^i represents the true value of the target at time t; and $\hat{y}_t^i$ represents the predicted value at time t.

4. Experiments

4.1. Settings

Table 1 clearly describes two different datasets used in our experiment, namely Nasdaq100 and PM2.5 Air Quality. In the Nasdaq100 Stock dataset [8], the stock prices of 81 main corporations are selected as the multivariable time series. The NDX of the Nasdaq100 is used as the target series. The dataset of PM2.5 Air Quality [24] is obtained from UCI, containing the related air data collected near the US Embassy in China during early 2010 to late 2014. We employ PM2.5 values as the target series in this dataset.

To facilitate comparison with previous studies, the Nasdaq100 data are partitioned into training, validation and test data by a

Table 1
Detail of the datasets.

Dataset	Nasdaq100	PM2.5 Air Quality
Target series	NDX	PM2.5
Attributes	82	9
Time Spans	7/26/2016–12/22/2016	1/1/2010–12/31/2014
Time Intervals	1 min	1 h
Instances	40,560	43,801

Table 2
Detail of the hyperparameters.

Hyperparameters	Nasdaq100	PM2.5 Air Quality
Time steps size T	$T\epsilon${6, **10**, 12, 16, 24}	$T\epsilon${8, **10**, 12, 16, 24}
Hidden states size m	$m\epsilon${16, 32, **48**, 64, 128}	$m\epsilon${12, 24, **32**, 48, 64, 128}

ratio of 13:1:1 in this experiment. The PM2.5 Air Quality dataset is partitioned into training, validation and test data by a ratio of 3:1:1. The two metrics, namely mean absolute error (MAE) and root mean squared error (RMSE) are considered in time series prediction [25–27]. During the training model stage, the batch size is 128, the learning rate is 0.001 and the LSTM unit is 2. The bold numbers in Table 2 are the best hyperparameter settings.

4.2. Baseline methods

To evaluate the performance of our proposed approach, the performances of six baseline models and four variants of our proposed model are compared. (1) ARIMA-GARCH: A hybrid model of two classical statistical models, autoregressive integrated moving average (ARIMA) and generalized autoregressive conditional heteroskedasticity (GARCH) [28]. (2) Gradient boosting regression tree (GBRT) is a very popular machine learning method used in engineering to solve regression problems [29]. The hyper-parameters of GBRT are set as follows: the number of weak regressors is 1000, the maximum depth of the tree is 2 and the initial learning rate is 0.003. (3) LSTM+Zoneout: Adding Zoneout technology [30] based on two simple LSTM networks on the stack can greatly improve the long-term dependency weak learning ability of single-layer LSTM. This structure brings satisfactory accuracy over other traditional models on multivariate time series prediction tasks. The number of neurons per layer of LSTM is 128, the random loss ratio of Zoneout in context state c is 0.5, and the random loss ratio of hidden state h is 0.05. (4) The Attention-RNN model is applied to machine translation tasks. The initial version of Attention Network adds an attention mechanism to the decoder, where the output of each time step of the decoder is used to generate a relative probability distribution of words in a translation dictionary. To meet the output requirements of the multivariate time series prediction task, we used the squared loss function in the training phase and changed the output to the actual predicted value of the future T time steps. (5) The structure of the Input-Attn-RNN model contains only encoders with input feature attention mechanisms and standard decoders. The model's encoder-decoder has a same hidden state size of 128. (6) A dual-staged attention model (DA-RNN) which owns the state-of-the-art performance in time series prediction. Section 4.1 and Table 2 have given all the basic hyperparameters of our proposed model and its variants. (1) The MsA-no-adjustment-module model is a variant that retains only the multi-stage attention network. Specifically, it has a multi-stage input multivariate attention mechanism inside the encoder and a temporal attention mechanism inside the decoder. (2) "MsA": This is a model that adds an attention score adjustment mechanism to each attention mechanism based on the MsA-no-adjustment-module variant model. (3) Starting from the MsA+AdaHMG model, the optimization algorithm of the subsequent variant model training phase uses an adaptive first-order stochastic optimization algorithm for time series data proposed by our previous research. This algorithm has a faster convergence rate than that of Adam, so as to make the more accurate predictions. The additional parameters k_1 and k_2 of the AdaHMG algorithm are set at 0.5 and 0.3, respectively. (4) MsA+AdaHMG+TG-LSTM is the best performing model we have reported. Based on the previous variant, the standard LSTM inside the encoder is replaced with the TG-LSTM designed in our previous study.

The hyperparameter-tuned models all show the best performance. The CPU of the experimental device is i7-7850H, and the GPU is NVIDIA's GTX 1080. Our proposed method and the baseline methods are implemented with TensorFlow1.4.0-GPU [31] and keras-2.1.5 [32].

4.3. Method comparison and analysis

In this section, performances of the proposed method and the nine baseline methods are compared on the two datasets, as shown in Table 3. We report the average of the different evaluation metrics for all models on the test set.

The experimental results show that, DA-RNN has the best performance among the six baseline models in Table 3. On both datasets, the different evaluation metrics for the ARIMA-GARCH model are the weakest. The result indicates that, the law of dynamic changes cannot be obtained, and the ARIMA-GARCH model is more suitable for short-term stable time series data. The GBRT model is second to the deep learning method with the attention mechanism in the PM2.5 Air Quality dataset. However, this method does not perform well on the Nasdaq100 datasets with more non-predictive time variables. Due to the limitation of the depth of the tree and the number of weak regressors, the learning ability of the different influence information generated by its many time variables is insufficient. Although the long-term dependency learning ability of the LSTM network with Zoneout technology has been improved, the complex association of multivariate is ignored since multivariate data are fed directly to this model. The Attention-RNN model is originally applied to machine translation tasks. Since attention scores of several input non-predictive time series are only calculated over time, and the influence of several non-predictive time series on target series is equally processed, its performance is significantly weaker than that of most machine learning models and Input-Attn-RNN models. Input-Attn-RNN model is designed for the different input time series of multivariate time series data. Attention mechanism is implanted into the encoder to filter the most relevant input feature time series. Thus, the Input-Attn-RNN model can better grasp the different influence of different non-predictive time series on target series. Since the dynamics of the multivariate on the time axis are neglected, the performance of this model is weaker than that of the DA-RNN with the temporal attention mechanism. In Nasdaq100 dataset, the MsA+AdaHMG+TG-LSTM method shows 33.01% and 37.55% improvements beyond the DA-RNN on MAE and RMSE respectively. In PM2.5 air quality task, the MsA+AdaHMG+TG-LSTM method shows 10.16% and 12.81% improvements beyond the DA-RNN on MAE and RMSE. The reason can be explained as follows. the DA-RNN model simply treats all the non-target time series with equal effect on the target series and feeds them directly into the encoder to select the relevant series by input attention mechanism, whereas our model can effectively capture the different effects of multiple time series on the target series at the same time stage. In addition, the performance improvement of our proposed model comes from the capture of the strong temporal correlation between the current time and historical moments by the used of AdaHMG Algorithm and the function transformation of the input stream by TG-LSTM in the neural network to map the information to the "appropriate" interval. Fi-

Table 3
Performance comparison of different methods.

Model	Nasdaq 100 Index			PM2.5 Air Quality		
	RMSE	MAE	MAPE ($\times 10^{-2}$%)	RMSE	MAE	MAPE ($\times 10^{-2}$%)
ARIMA-GARCH	0.9798	0.7346	1.51	0.8323	0.5678	1.134
GBRT	0.6370	0.4751	0.97	0.3156	0.1674	0.332
LSTM+Zoneout	0.5074	0.4565	0.93	0.3310	0.1638	0.317
Attention-RNN	0.9543	0.7011	1.43	0.7504	0.4349	0.865
Input-Attn-RNN	0.3903	0.2602	0.53	0.3118	0.1623	0.302
DA-RNN	0.3017	0.2251	0.43	0.3031	0.1592	0.298
MsA-no-adjustment-module	0.2533	0.1738	0.34	0.2957	0.1560	0.285
MsA	0.1973	0.1636	0.31	0.2794	0.1454	0.276
MsA+AdaHMG	0.1899	0.1525	0.29	0.2781	0.1445	0.269
MsA+AdaHMG+TG-LSTM	**0.1884**	**0.1508**	**0.27**	**0.2723**	**0.1388**	**0.258**

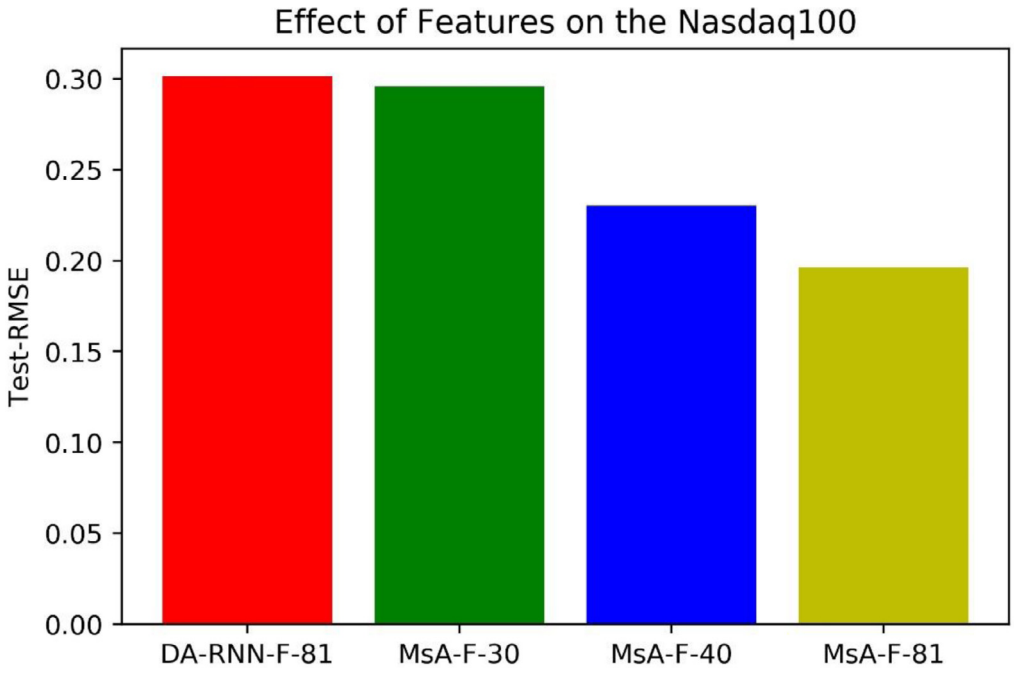


Fig. 6. Evaluation on the number of features.

nally, the MAPE metrics obtained by our experiment is basically consistent with the results reported by the DA-RNN model in the dataset of NASDAQ-100 in the previous study. Specifically, the test-MAPE value by MsA+AdaHMG+TG-LSTM in the NASDAQ-100 and PM2.5 Air Quality datasets improved by 37.20% and 13.42%,% respectively. Compared with the DA-RNN model, the test-MAPE metrics also fully demonstrated that the proposed model had significant advantages over the prediction performance of the DA-RNN model.

Meanwhile, we find that the level of performance improvements is not similar across the two datasets, because the different number of the time series can impact the target series in the two datasets, i.e., 81 time series in the Nasdaq100 dataset and eight time series in the PM2.5 dataset. Thus, the performance of the DA-RNN and "MsA" method are further tested when encountered with a different number of time series. The DA-RNN-F-81 and MsA-F-81 use the full nasdaq100 dataset. The first 30 or 40 columns of the dataset and the target series are used for the MsA-F-30 and the MsA-F-40, respectively. Fig. 6 shows that our method has better predictive performance when there are more time series. The predictive performance of "MsA" method decreases to the performance level of DA-RNN when only 30-time series features are available. It suggests that multi-stage attention mechanism can fully capture the effect on the target series with the increase of the multivariable time series.

4.4. Variant comparison

In order to further investigate the performance of the DA-RNN and our proposed model variants, RMSE and MAE of the test set of two real-world tasks are plotted during the testing phase, as shown in Fig. 7(a), (b). In the two subgraphs of Fig. 7, the red histogram represents our first variant model, which has only a multi-stage attention mechanism and a temporal attention mechanism. On the Nasdaq 100 dataset, the test RMSE and test MAE for this variant increases by 16.0% and 22.7% beyond the DA-RNN, respectively. On another dataset, PM2.5 Air Quality, the test RMSE and test MAE for this variant increases by 2.4% and 2.0% beyond the DA-RNN, respectively. The difference in the magnitude of the increase implies that this variant model has different performance on multivariate time series data with different number and complexity of variables. However, such experimental results demonstrate that the variant can capture different effects of different non-predictive variables on the target series at different time stages through a multi-stage attention mechanism. This is an ability that was not available in previous DA-RNN model. Then,

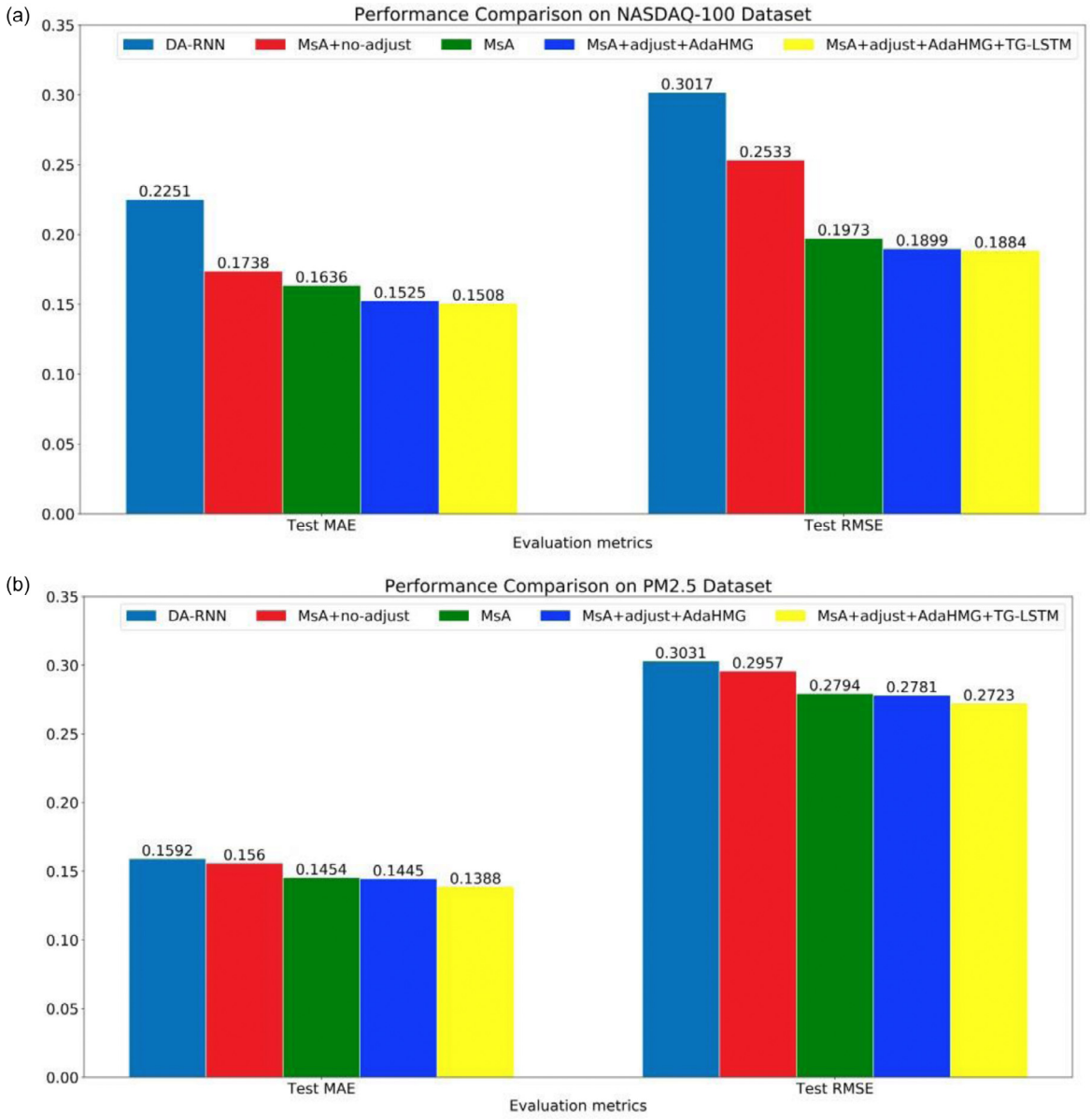


Fig. 7. (a) RMSE and MAE of the test set of Nasdaq 100 dataset. (b) RMSE and MAE of the test set of PM2.5 Air Quality dataset.

because there is a series of non-predictive time series in the multivariate time series data, different effects are generated. In order to obtain more abundant influence information, we introduce the attention score adjustment module. In Nasdaq100 dataset, the multi-stage attention with attention score adjustment module method shows 27.3% and 34.6% improvements beyond the DA-RNN on test MAE and test RMSE respectively. In the PM2.5 air quality task, this variant method shows 7.8% and 8.6% improvements beyond the DA-RNN on test RMSE and test MAE. Such experimental results reveal that the attention score adjustment module can enhance the representation of the change rule of the target series by obtaining richer influence information. The dark blue rectangle in Fig. 7 indicates that the AdaHMG optimization algorithm is used during model training. The used optimization algorithm is a mini-batch of stochastic gradient descent method, but the internal update of the square gradient term in algorithms such as Adam may ignore the strong temporal correlation between the current time and the historical time, especially for multivariate time series prediction tasks with complex temporal correlations. As shown in Fig. 7, evaluation indicators on two datasets have been further improved when the AdaHMG algorithm is used. To precisely evaluate the variant model, the new optimization algorithm is used and the curves for validation MAE on two datasets are presented in Fig. 8(a) and (b).

As shown in Fig. 8, the variants represented by the green lines using the AdaHMG algorithm are faster in the first five epochs than in the model using the Adam optimization algorithm. When other models are oscillated, the curves of the variants continue to decline with a weaker oscillate. In particular, the Multi-stage-Att+adjust+AdaHMG+TG-LSTM model validates the MAE curve on the PM2.5 Air Quality dataset, and most epochs have lower validation MAE values than all other baseline and variant models. Fig. 8 suggests that our model (Multi-stage-Att+adjust+AdaHMG+TG-LSTM) can achieve faster training procedure and better convergence.

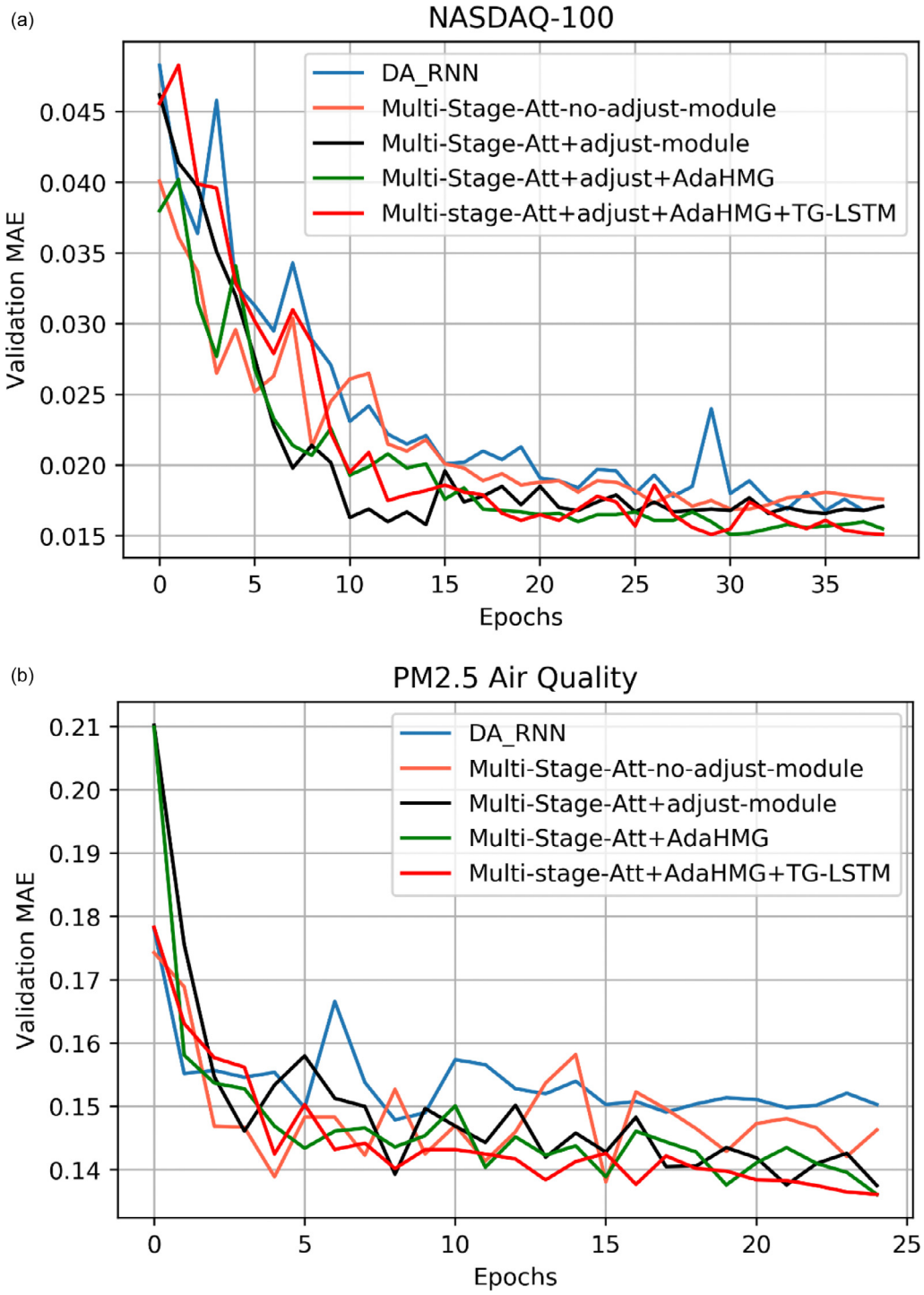


Fig. 8. (a) The validation MAE curve on the NASDAQ 100 dataset. (b) The validation MAE curve on the PM2.5 Air Quality dataset.

4.5. Case study

Since the Nasdaq 100 dataset has more non-predictive time series and more complex dynamic change information than the PM2.5 air quality dataset, we use the 81 series from the Nasdaq 100 dataset as an example to visualize attention weights for all encoder time step t in Fig. 9.

First, in the Nasdaq 100 dataset, multiple non-predictive time series has different effects on the target sequence at the same time. This phenomenon can be significantly observed by printing the encoder attention weight score, and our model can learn such information very well. Specifically, when the encoder step is zero, the sequence No. 0 produces a maximum influence of 0.014 (i.e., The color is red), the influence of the No. 27 sequence is 0.011 (i.e., The color is bright green), and the No. 69 sequence exhibits a minimum influence of 0.009 (i.e., The color is purple). The above example implies that, an encoder with attention score adjustment mechanism and a LSTM with transformation gating successfully

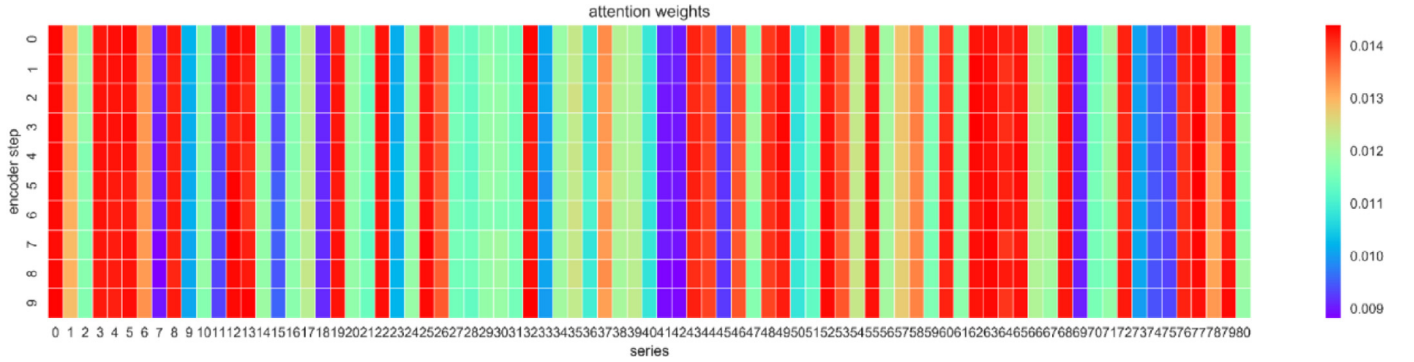


Fig. 9. Attention weights heatmap.

capture different non-predictive time series that has different effects on the target series at the same time.

Second, multiple non-predicted time series produces different attention scores over different time stages. Our method can effectively capture this information through a multistage attentional mechanism observed by the attention weights heatmap. For example, in the encoder step from 2 to 6, the influence of the sequence No. 35 varies in the range of [0.011, 0.0125]. Sequence No. 78 dynamically changes in the interval of [0, 4] during the period when the encoder step is [0.0127, 0.0135]. These examples demonstrate that the multi-stage attention mechanism can learn the different effects of different non-predicted time series on target series over different time stages.

Last but not least, for the mutation phenomenon, our attention weights heatmap can only show a sudden change point in a short time window due to space limitations. When the abrupt phenomenon occurs, the color corresponding to the influence value generated by a non-predicted sequence from i to $i+1$ undergoes a significant change. At the time stage of encoder step 7 through 8, the color corresponding to the influence of the 36th sequence changes from blue to light blue. This example illustrates that there is a sudden increase in impact, indicating a sudden rise in stock time data in column 36. In addition, in the time stage of the encoder step 1 to 2, the color corresponding to the influence of the 78th sequence changes from soil yellow to light yellow. This example illustrates a sudden change in the impact, which may imply a sudden drop in stock time series data in column 78. The above examples demonstrate the attention mechanism of our encoder stage and the good learning ability of TG-LSTM for mutations.

These results indicate that our model not only captures the different influences from different time series, but also obtains the information of the dynamic influences in different encoder time steps. Finally, the encoder's internal TG-LSTM network and multistage attention mechanism capture the abrupt changes contained in multivariate time series data at different levels.

4.6. Statistical test

To accurately assess the proposed approach, significant tests are performed statistically in two different metrics of two datasets. Two-tailed T-test is used, and when the significance degree $\alpha = 0.05$, the value of the two-tailed T-test ($p < 0.01$) is greater than the critical value of the table lookup. If $\hat{\varepsilon}_i \le \varepsilon_0$ cannot be rejected, it indicates that proposed methods are significantly different and the differences are caused by errors.

We give examples of two-tailed T-tests with different evaluation metrics on two datasets. Specifically, on the Nasdaq 100 dataset, the average test RMSE value of method "MsA+AdaHMG+TG-LSTM" is calculated as $\mu = \frac{1}{m}\sum_{i=1}^{m}\hat{\varepsilon}_i = 0.1884$ (where $\hat{\varepsilon}_i$ represents the ith test RMSE value). Then, the variance is $\sigma^2 = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{\varepsilon}_i - \mu)^2 = 3.25 \times 10^{-7}$ ($m = 5$). Finally, the critical value is obtained as $\tau_t = \sqrt{m}|\mu - \varepsilon_0|/\sigma = 3.138$ (where ε_0 is the assumed maximum RMSE value, and $\varepsilon_0 = 0.1892$). The critical value τ_t is larger than that given 2.776 by the two-tailed T-test table. This results show that the performance of "MsA+AdaHMG+TG-LSTM" model is less than the assumed test RMSE (0.1892) with a confidence degree ($1 - \alpha = 0.95$).

Two-tailed T-test is used, and when the significance degree $\alpha = 0.01$, the value of the two-tailed T-test ($p < 0.01$) is greater than the critical value of the table lookup. On the PM2.5 Air Quality dataset, the average test MAE value of method "MsA+AdaHMG+TG-LSTM" is calculated as $\mu = \frac{1}{m}\sum_{i=1}^{m}\hat{\varepsilon}_i = 0.1388$ (where $\hat{\varepsilon}_i$ represents the ith test MAE value). Then, the variance is $\sigma^2 = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{\varepsilon}_i - \mu)^2 = 3.54 \times 10^{-8}$ (The value of m is 5). Finally, the critical value is obtained as $\tau_t = \sqrt{m}|\mu - \varepsilon_0|/\sigma = 7.132$ (the assumed maximum MAE value ε_0 is 0.1394). The critical value τ_t is larger than that given 4.604 by the two-tailed T-test table. This results show that the performance of "MsA+AdaHMG+TG-LSTM" model is less than the assumed test MAE (0.1394) with a confidence degree ($1 - \alpha = 0.99$).

As shown in the above statistical tests, evaluation indicators obtained by our proposed model on the two data sets are significantly better than that of DA-RNN model.

5. Discussion

5.1. Preliminary theoretical analysis of the effect of AdaHMG algorithm

As a first-order adaptive stochastic optimization algorithm, AdaHMG algorithm is proposed in our previous research to avoid the convergence of Adam algorithm to a local minimum under certain conditions. In the Adam algorithm, the convergence speed of the algorithm is evaluated by giving the time complexity of the upper bound of Regret. Specifically, the theoretical computational complexity of Adam algorithm is O $(\sqrt{T} + \sqrt{1+\log T} \times \sum_{i=1}^{d} g_{1:T,i_2})$, and the theoretical computational complexity of our AdaHMG algorithm is O $(\sqrt{T} + \sqrt{1+\log T} \times \sum_{i=1}^{d} g_{1:T,i_2^{\frac{1}{2}}})$, which is obviously lower than Adam algorithm. Because the second item $\sum_{i=1}^{d} g_{1:T,i_2^{\frac{1}{2}}}$ in the time complexity of AdaHMG algorithm is only half the power of that in the time complexity of Adam algorithm (i.e., $\sum_{i=1}^{d} g_{1:T,i_2^{\frac{1}{2}}} < \sum_{i=1}^{d} g_{1:T,i_2} \ll \sqrt{dT}$). Therefore, AdaHMG algorithm has faster convergence speed and lower computational over-

head in the process of model training. As is known to all, when the convergence rate of the optimization algorithm is not as fast as the sublinear convergence rate, the smaller Regret upper bound the faster the convergence of the model, so that model parameters that minimize the empirical risk can be obtained. This is the fundamental reason that AdaHMG algorithm can improve the prediction accuracy of the proposed model.

The update operation of AdaHMG algorithm's internal hybrid history and current second-order moment estimation information (i.e., $\hat{v}_t = k_1\hat{v}_{t-1}^2 + k_2 v_t$) can directly learn the variation law of gradient information over time. This operation in the back-propagation phase is helpful to indirectly capture the temporal correlation information between the moments before and after continuous temporal data. In the convergence proof of the iterative algorithm, the fusion equation involving the gradient information before and after moments is obtained as follows Eq. 21:

$$\sum_{t=1}^{T}\alpha_t\left\|\hat{v}_t^{-\frac{1}{4}}m_t\right\|^2 \le \sum_{t=1}^{T-1}\alpha_t\left\|\hat{v}_t^{-\frac{1}{4}}m_t\right\|^2 + \frac{\alpha}{4\sqrt{\frac{4k_1k_2}{1-\beta_2}}(1-\beta_1)\sqrt{T(1-\beta_2)^2}} \times \sum_{i=1}^{d}\sum_{j=1}^{T-1}\frac{\beta_1^{T-1-j}g_{j,i}^2}{\left(\beta_2^{T-1-j}g_{j,i}^2\right)^{\frac{3}{4}}} + \frac{\alpha}{4\sqrt{\frac{4k_1k_2}{1-\beta_2}}(1-\beta_1)\sqrt{T(1-\beta_2)^2}} \times \sum_{i=1}^{d}\frac{g_{T,i}^2}{\left(\sum_{j=1}^{T-1}\beta_2^{T-1-j}g_{j,i}^2\right)^{\frac{3}{4}}} \tag{21}$$

where the second and third term respectively shows the fusion process of history of $T-1$ moment to T moment gradient information (i.e., $g_{j,i}^2$). This fusion operation suggests that dynamic temporal variation of multivariate time series data can be learned by the proposed method (i.e., the moment before and after the temporal correlation information is learned from the aspect of the optimization algorithm). The detailed proof can be found in our previous research [23].

5.2. Preliminary theoretical analysis of the TG-LSTM network

The TG-LSTM network inside the encoder is the core neural network component learning the complex time dependence of the input time series. Through the numerical analysis of the partial derivative function in the back-propagation stage, temporal correlation information or even mutation information can be captured by the network. First, analyzing the function of the partial derivative corresponding to the transformation gating mechanism is the key step. Specifically, the chain rule is used to get the partial derivative calculation procedure involving transformation gating $\frac{\partial L_t}{\partial f_t} = \frac{\partial L_t}{\partial tr_t}\cdot\frac{\partial tr_t}{\partial f_t} = \delta tr_t \odot (\tanh^2(f_t) - 1)$, where L_t represents the Loss at the moment t; f_t and tr_t represent the values of forget gate and transformation gate, respectively. As shown in Fig. 10, the function image of the partial derivative function $\tanh^2(f_t) - 1$ within the domain of the time series data after regularization is illustrated. When the domain of the partial derivative function ranges from [0, 2], the function image is based on the Y-axis symmetry within the significantly changed range of $[-0.08, -1.0]$. Meanwhile, the partial derivative function is convex and has smooth function property, which is beneficial for learning the temporal correlation information between gradient error flow information in the backpropagation stage. The detailed analysis of backpropagation can be found in our previous similar research [3].

5.3. Theoretical analysis of attention score readjustment mechanism

Since the selected time series datasets contain multiple dimensions of non-predicted time series (especially the NASDAQ-100 dataset with dozens of dimensions of non-predictive sequences), how to capture more important temporal correlation information among the numerous non-predictive time series is of great importance. Log_Softmax is introduced after the traditional attention-mechanism score calculation function Softmax. It can compress the numerical difference between different non-predictive series' attention-weight scores, so that more non-predictive time series with attention-weight scores in the median attachment can be screened and learned by the attention-mechanism. Eq. (22) shows the compression process ($\sigma(\cdot)$ is Softmax function):

$$\log\sigma(x_i) = \log\frac{exp(x_i)}{\sum_j exp(x_j)} = x_i - log\left(\sum_j exp(x_j)\right) \tag{22}$$

As we know, the log function based on 2 has the property of the compression function range, and the log function further smooths the distribution of the output values of the Softmax function. Owing to the rational use of the above attention score remodulation mechanism, more non-predictive time series can be fully captured in the process of the time series data with a large number of non-predictive time series.

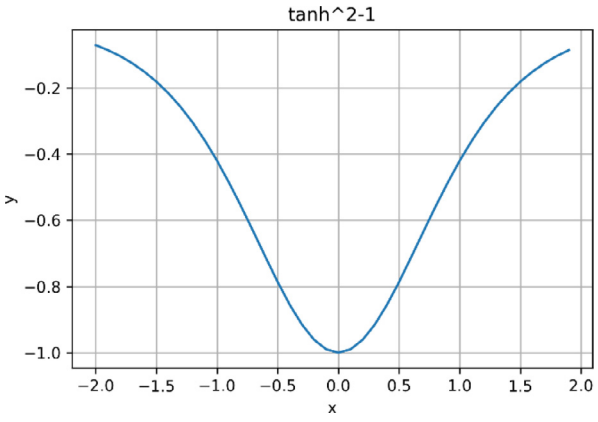


Fig. 10. $\tanh^2(f_t) - 1$ function curve.

5.4. Preliminary theoretical analysis of the multistage attention mechanisms

Finally, TG-LSTM neural network and multistage attention mechanisms are combined to analyze positive effects of multistage attention mechanisms on prediction performance. A recent study on attention mechanisms supports a similar view from the field of natural language sequences [33]. In this paper, the weight score operation within the attention mechanism plays a key role in the following two aspects: 1. Selecting the most relevant multiple non-predictive time series; 2. Selecting the time stage when the non-predictive time series has a great impact on the target series. Attention weight score is the parameter acquired by the integral calculation component (i.e., $e_t^i = \mathbf{v}_s^\top \tanh(\mathbf{W}_s[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_s\mathbf{x}^i + \mathbf{b}_s)$, where the hidden state $\mathbf{h}_{t-1}$ and the cell state $\mathbf{s}_{t-1}$ are processed by TG-LSTM network. The above two state information directly determines the lower bound of the learning effect of attention mechanism to the information) TG-LSTM neural network during iterative training. This indicates that if the complex temporal variation rules of multiple non-predictive time series cannot be effectively captured by the recurrent neural network within the encoder, then the most relevant temporal information cannot be selected by the attention mechanism. Up to now, there is no intuitive theory to explain why certain non-predictive time series can correspond to a specific attention weight score. Thus, it is

easier to understand the effect of attention score on prediction performance from the model. After all, if attention-weighted score calculation components are removed in different positions of the model, the performance is reduced to different degrees.

6. Conclusion

In this paper, we propose a novel multistage attention networks model, which consists of an encoder with multistage influence attention mechanism and a decoder with temporal attention mechanism. It is worth noting that the circulating neural network inside the encoder is a LSTM with transformation gating, which is specifically designed to capture the abrupt phenomenon. Each attention mechanism contains an attention score adjustment module, so as to adaptively select more relevant input time series that produces impact information for the target series, and capture the dynamic long-term temporal dependent information. In addition, the AdaHMG optimization algorithm, which is accuracy improvement in multivariate time series prediction tasks. We evaluate our model on two multivariable time series datasets. The experimental results show that our model has better performance over all other baseline models. In future work, we will further explore whether the proposed model can be extended to more complex spatial-temporal correlation multivariate time series prediction by supplementing the learning ability of spatial dependence.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is partially supported by State Key Lab. for Novel Software Technology, Nanjing University, P.R. China under Grant KFKT2019B09.

Appendix

Fig. A.1 and A.2–Fig. A.3 and A.4 help readers better understand how the model's hyper-parameters affect the predicted performance. In Fig. A.1, we find that the proposed model is insensitive to the number of hidden states encoded and decoded internally, and 48 state units are required to achieve the best prediction effect (less than 64 hidden state units required by DA-RNN model). When $T = 10$, the proposed model and DA-RNN both have the best performance. The proposed model has a more stable performance of multi-step prediction than that of DA-RNN model. Fig. A.3 shows that the prediction performance of DA-RNN model declines or rises significantly in four-time steps.

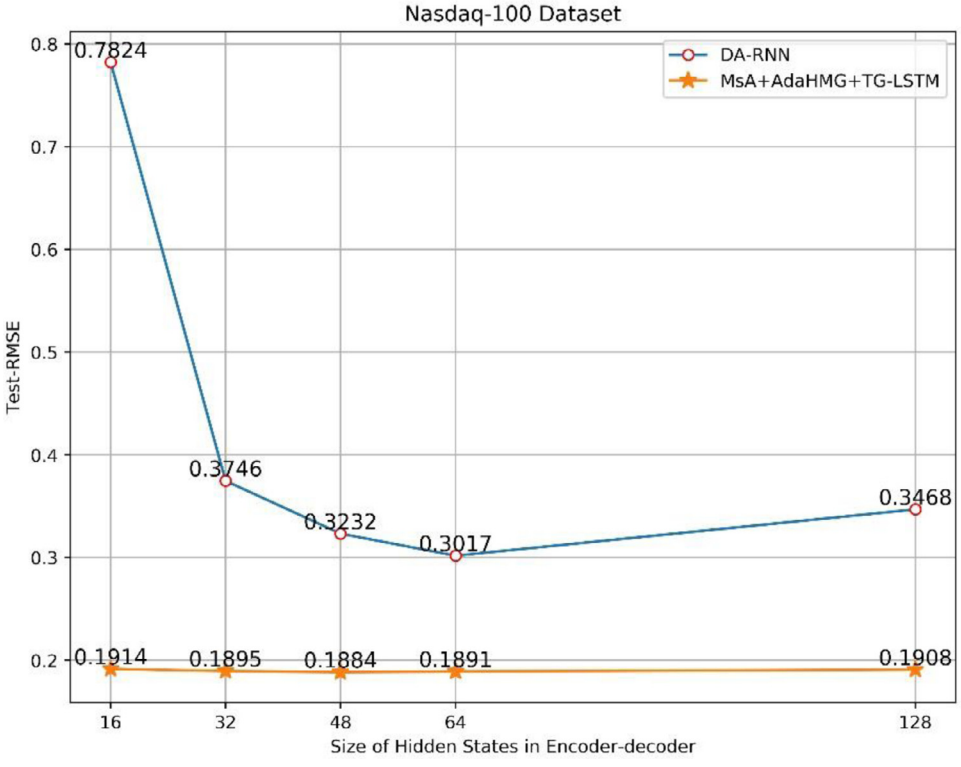


Fig. A.1. The comparison of the number of the encoder-decoder hidden states between the DA-RNN model and our model on the NASDAQ-100 dataset.

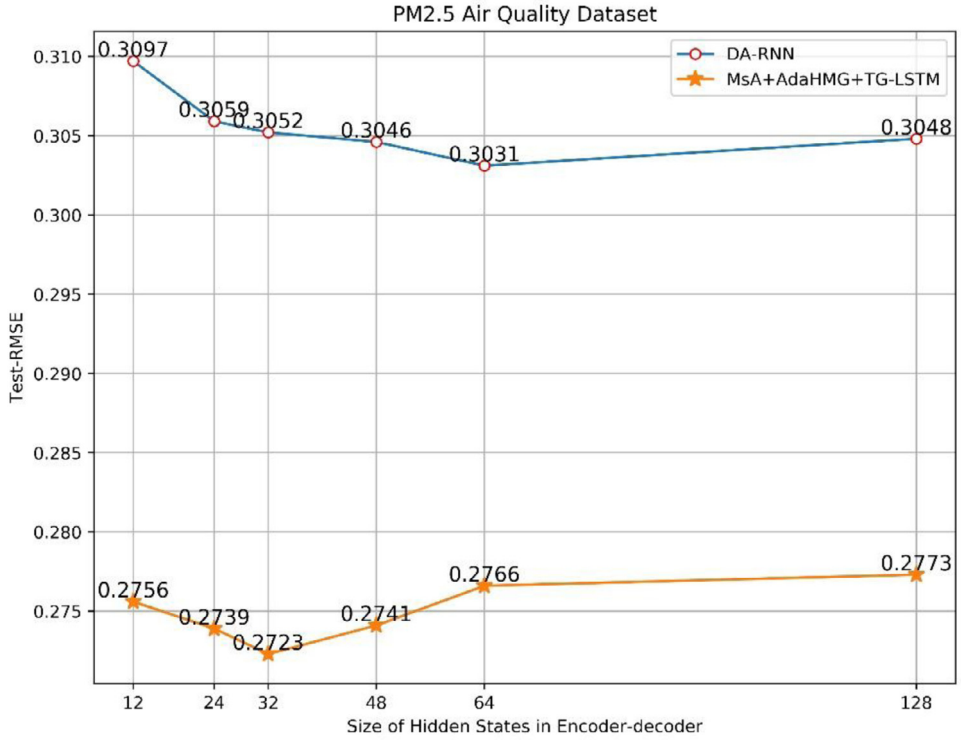


Fig. A.2. The comparison of the number of encoder-decoder hidden states between the DA-RNN model and our model on the PM2.5 Air Quality dataset.

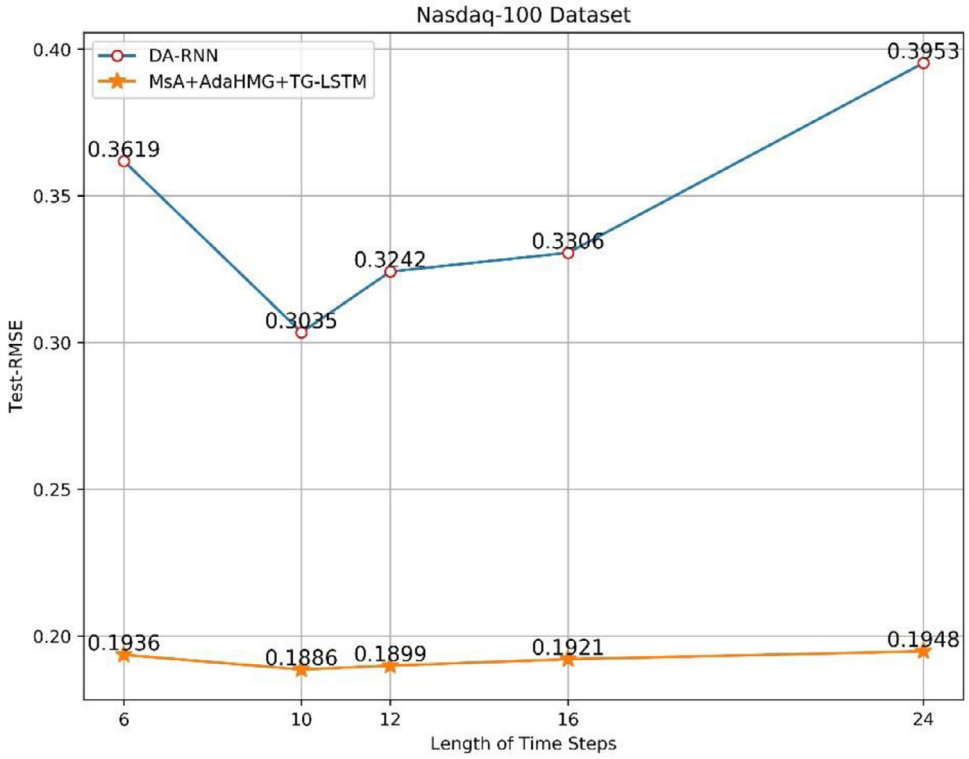


Fig. A.3. The comparison of the length of time steps between the DA-RNN model and our model on the NASDAQ-100 dataset.

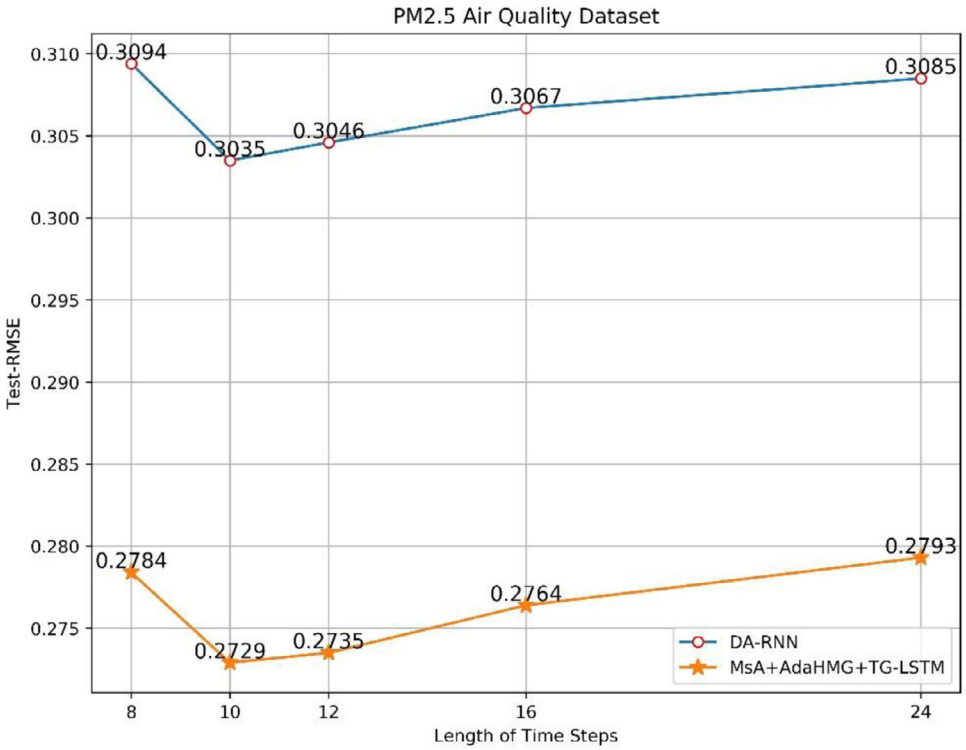


Fig. A.4. The comparison of the length of time steps between the DA-RNN model and our model on the PM2.5 Air Quality dataset.

References

[1] R. Rawassizadeh, E. Momeni, C. Dobbins, J. Gharibshah, M. Pazzani, Scalable daily human behavioral pattern mining from multivariate temporal data, IEEE Trans. Knowl. Data Eng. 28 (11) (2016) 3098–3112.

[2] Y. Haimin, P. Zhisong, T. Qing, Q. Junyang, Online learning for vector autoregressive moving-average time series prediction, Neurocomputing 315 (2018) 9–17, doi:10.1016/j.neucom.2018.04.011.

[3] J. Hu, W. Zheng, Transformation-gated LSTM: efficient capture of short-term mutation dependencies for multivariate time series prediction tasks, in: Proceedings of the International Joint Conference on Neural Networks, 2019, pp. 1–8, doi:10.1109/IJCNN.2019.8852073.

[4] A.A. Ricardo de, N. Nadia, L.I.O. Adriano, S.R. de, L. Meira, A deep increasing–decreasing-linear neural network for financial time series prediction, Neurocomputing (2019), doi:10.1016/j.neucom.2019.03.017.

[5] X. Li, R. Bai, Freight vehicle travel time prediction using gradient boosting regression tree, in: Proceedings of the 15th IEEE International Conference on Machine Learning and Applications, ICMLA 2016, Anaheim, CA, USA, 2016, pp. 1010–1015. December 18-20, 2016.

[6] T. Lin, B.G Horne, P. Tino, C. Lee Giles, Learning long-term dependencies in narx recurrent neural networks, IEEE Trans. Neural Netw. 7 (6) (1996) 1329–1338.

[7] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv:1412.3555, 2014.

[8] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, G. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia, 2017, pp. 2627–2633. Aug. 2017, Morgan.

[9] Y. Yuan, G. Xuny, F. May, Y. Wangy, N. Duz, K. Jia, L. Suy, A. Zhang, MuVAN: a multi-view attention network for multivariate temporal data, in: Proceedings of the ICDM, IEEE, Singapore, 2018.

[10] Y. Liang, S. Ke, Junbo Zhang, Xiuwen Yi, Yu Zheng, GeoMAN: multi-level attention networks for geo-sensory time series prediction, in: Proceedings of the IJCAI, Morgan, Stockholm, Sweden, Jul. 2018, pp. 3428–3434.

[11] M. Gupta, J. Gao, C.C. Aggarwal, J. Han, Outlier detection for temporal data: a survey, IEEE Trans. Knowl. Data Eng. 26 (9) (2014) 2250–2267.

[12] W. Enders, Applied econometric time series, by walter, Technometrics 46 (2) (2004) 264.

[13] Y. Zhou, H. Zou, et al., Non-parametric outliers detection in multiple time series a case study: power grid data analysis, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 4605–4612.

[14] P. Whittle, Ph.D. thesis, 1951.

[15] L. Yan, A. Elgamal, G.W. Cottrell, Substructure vibration narx neural network approach for statistical damage inference, J. Eng. Mech. ASCE-AMER 139 (2013) 737–747.

[16] J.L. Elman, Distributed representations, simple recurrent networks, and grammatical structure, in: Machine Learning, 7, Springer, 1991, pp. 195–225.

[17] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, in: Proceedings of the NIPS, MIT Press, Montreal, Quebec, Canada, Dec.2014, pp. 3104–3112.

[18] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, CoRR (2014) abs/1409.0473.

[19] L. Wang, Z. Cao, G. de Melo, Z. Liu, "Relation classification via multi-level attention cnns, in: Proceedings of the ACL, ACL, Berlin, Germany, 2016.

[20] D. Yu, J. Fu, T. Mei, Y. Rui, "Multi-level attention networks for visual question answering, in: Proceedings of the CVPR, IEEE, Honolulu, HI, USA, Jul. 2017, pp. 4187–4195.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, N.G. Aidan, K. Lukasz, P. Illia, Attention is all you need, in: Proceedings of the Neural Information Processing Systems (NIPS), MIT Press, Long Beach, CA, USA, Dec. 2017, pp. 6000–6010.

[22] J. Ba, D. Kingma, Adam: a method for stochastic optimization, in: Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, May. 2015.

[23] J. Hu, W. Zheng, An adaptive optimization algorithm based on hybrid powerand multidimensional update strategy, IEEE Access 7 (1) (2019) 19355–19369.

[24] X. Liang, et al., Assessing beijing's pm2.5pollution: severity, weather impact, apec and winter heating, in: Proceedings of the Royal Society A Mathematical Physical & Engineering Sciences, 471, Feb. 2015.

[25] J.P. Nobrega, A.L.I. Oliveira, A sequential learning method with Kalman filter and extreme learning machine for regression and time series forecasting, Neurocomputing 337 (2019) 235–250 Apr 14, doi:10.1016/j.neucom.2019.01.070.

[26] Y. Rizk, M. Awad, On extreme learning machines in sequential and time series prediction: a non-iterative and approximate training algorithm for recurrent neural networks, Neurocomputing 325 (2019) 1–19, doi:10.1016/j.neucom.2018.09.012.

[27] A. Sagheer, M. Kotb, Time series forecasting of petroleum production using deep LSTM recurrent networks, Neurocomputing 323 (2019) 203–213, doi:10.1016/j.neucom.2018.09.082.

[28] C. Ding, J. Duan, Y. Zhang, X. Wu, G. Yu, "Using an arima-garch modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility, IEEE Trans. Intell. Transp. Syst. 19 (4) (2018) 1054–1064.

[29] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 1189–1232.

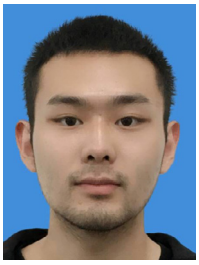
[30] D. Krueger, et al., Zoneout: regularizing RNNs by randomly preserving hidden activations, in: Proceedings of the ICLR, 2017.

[31] M. Abadi et al., "TensorFlow: large-Scale machine learning on heterogeneous systems," https://www.tensorflow.org/, 2015.

[32] F. Chollet et al., "Keras," https://github.com/fchollet/keras, 2015.

[33] S. Wiegreffe and Y. Pinter, “Attention is not not explanation”, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2019), in press, Nov. 3-7, Hong Kong, China.

Jun Hu was born in 1971 and received M.*Sc.* in Computer Application from Kunming University of Science and Technology, Kunming, China, and Ph.D. in Computer Science and Technology from Zhejiang University, Hangzhou, China. In 2010, he was an academic visitor at University of Southampton working on multi-agent system. Currently, he is an associate professor of Hunan University, Changsha, China. He is a senior member of China Computer Federation (CCF). His research interests include multi-agent system, deep learning and software engineering.



Wendong Zheng was born in Tai Yuan, Shan Xi, China in 1994. He received the B.S. degrees in software engineering from the North University of China, in 2017. Currently, he is a master student in computer science at Hunan University. His main research interests are time series prediction, deep learning and first-order stochastic optimization algorithm. He has published 2 peer reviewed journal and conference papers. He is a student member of International Neural Networks Society (INNS) and IEEE. In addition, he is a reviewer of IEEE transactions on neural networks and learning systems and IEEE access. Also, he is a PC member of IEEE ICIST-2019 and IEEE HPCC-2019.