IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 June 2022, accepted 12 July 2022, date of publication 19 July 2022, date of current version 26 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192430

RESEARCH ARTICLE

A Combined Model for Multivariate Time Series Forecasting Based on MLP-Feedforward Attention-LSTM

YUNTONG LIU, CHUNNA ZHAO, AND YAQUN HUANG
School of Information Science and Engineering, Yunnan University, Kunming 650504, China

Corresponding author: Chunna Zhao (zhaochunna@ynu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61862062 and Grant 61104035.

ABSTRACT Multivariate time series forecasting has very great practical significance for a long time, and it has been attracting the attention of researchers from a diverse range of fields. However, it is difficult to analyze the relationship and transformation law among multivariate data. Further, it is hard to obtain a relatively accurate prediction. In recent years, long short-term memory (LSTM) has shown high capability in dealing with nonlinearity and long memory of time series data. Although LSTM can also process multivariate data, it is insufficient to pay various degrees of attention to multivariate data. To address this issue, a multivariate time series prediction model based on multilayer perceptron (MLP), feed-forward attention mechanism, and LSTM is proposed in this paper. Firstly, the simulation process utilizes the MLP module to map the multivariate initial sequences into another latent dimensional space, thereby obtaining easily captured mapping features. Then, these features are adaptively assigned attention weights through the feed-forward attention mechanism. Finally, the LSTM module uses these feature sequences with attention weights to make final predictions. The experimental results show that the method that combines the MLP layer with the feed-forward attention layer is effective in extracting multivariate features. Also, the empirical results indicate that our proposed framework (a combined model of MLP-Feedforward attention-LSTM) can achieve better performance than baselines.

INDEX TERMS Multivariate time series, multilayer perceptron, feed-forward attention mechanism, long short-term memory network.

I. INTRODUCTION

Time series data is an irreversible sequence of numbers arranged in chronological order. Modern society contains many types of multivariate time series data, including financial market [1], climate forecasting [2], and renewable energy [3]. One of the most crucial (and arguably the most difficult) tasks of time series analysis is that utilize existing historical data to predict the future. Different from other predictive modeling tasks, multivariate time series have higher time complexity and contain more invalid and disturbing information. Meanwhile, there exists a high degree of complex correlation between the input variables of time series. Therefore, it is crucial to build a model that can capture the complex correlation and highly nonlinear dependency present in real datasets. Multivariate time series forecasting methods assume interdependence between variables, that is, the prediction of a variable can not only observe its historical value, but other related variables also have a non-negligible influence on it. Through many studies of time series, it is found that extracting good features is one of the keys to addressing the problem of time series. However, existing methods cannot extract potential correlations among variables efficaciously.

With the development of applications and the continuous exploration of researchers, the traditional method has become more mature, and it is also one of the most commonly used methods in time series forecasting. For instance, classical statistical methods (e.g., auto-regressive integrated moving average model (ARIMA) [4], hidden Markov models (HMMs) [5], exponential smoothing [6], etc.), have made

The associate editor coordinating the review of this manuscript and approving it for publication was Dost Muhammad Khan.

great progress. However, there are still some unavoidable disadvantages: (1) they can only model the linear relationship, which requires a certain degree of steadiness in datasets; (2) the accuracy of the prediction results is largely determined by the correct choice of the parameter. Due to nonlinearity, highly complex correlation, non-stationarity, randomness, and other properties of real-world datasets, it is difficult for traditional statistical methods to solve the above problems.

In recent years, with the rapid development of deep learning, many machine learning problems have been solved [7]. Unlike traditional methods, deep learning can not only directly adapt data, but also handle the nonlinearity and complex correlation of time series. In time series prediction, one of the most commonly used and effective deep learning models is the recurrent neural network (RNN) [8]. The network has attracted wide attention due to its flexibility in capturing the time characteristics of data. More recently, RNN has also achieved better performance [9]. Nevertheless, in subsequent applications, the RNN cannot process the accumulated information, resulting in an increasing loss. To solve the problem, two variants in RNN, i.e., – gated recurrent unit (GRU) [10] and long short-term memory network (LSTM) [11] were introduced. LSTM controls the balance between information retention and forgetting through three gates (forget gate, input gate, and output gate). So, it successfully achieves the capture of long-term dependence of time series to a greater extent while retaining short-term information. Based on LSTM, researchers have done lots of research on improving the prediction accuracy [12]–[16]. Recently, the research of attention models in the field of computer vision has been attracting the attention of many researchers in other fields. Researchers on the time series have also successfully applied attention mechanisms to the time series analysis. Its main purpose is to enhance the selection of relevant time steps in the past [17]–[21], including the Transformer model [22]. Zheng *et al.* [23] presented a theoretical analysis of LSTM integrated with attention mechanism, and demonstrated that it is capable of generating an adaptive decay rate which dynamically controls the memory decay. If the attention mechanisms are used in other places, they may play a different role. In addition, these deep learning models usually directly take the initial multivariate time series as input data. However, real-world datasets are generally chaotic, different feature vectors conversely interweave as they pass through neural networks. As a result, their features become blurred, and some important features may be filtered by neural networks without being used. If the attention mechanisms are applied to the initial time series, more important feature data may be preserved.

To fulfill the above assumption, an improved prediction model based on LSTM is proposed. The main idea of the feed-forward attention mechanism [24] is utilized to implement the function of extracting important information. Real-world datasets are often chaotic, divergent, and non-linear, thus such mechanism cannot directly extract useful information. Inspired by the theory of multilayer perceptron (MLP), that is, through the linear transformation of the feature space, the correlation between features can be more easily captured. An MLP layer is added in front of the feed-forward attention layer so that the data are mapped to a more suitable space to represent their data features. So far, the combined attention layer can achieve a better function in extracting the feature information for prediction. The introduction of this method also solves the defect that traditional LSTM is distracted when dealing with multivariate sequences. In other words, information from different sequences is analyzed equally. Therefore, this model can further improve the predictive performance of LSTM. Experiments on four real datasets verify that the method can obtain better prediction results. Meanwhile, the idea of this model can also be extended to other multivariate recurrent neural network architectures to further improve their performance. To sum up, our main contributions are depicted as follows:

(1) The MLP layer adjusts the feature space of the initial multivariate time series, which makes it easier for multivariate features to be captured in another potential space.
(2) Improve the construction of traditional attention mechanisms and recurrent neural networks. Traditional attention mechanisms are mostly used behind recurrent neural networks to enhance the selection of past relevant time steps. But in our architecture, the feed-forward attention layer is applied in front of the LSTM layer to extract multivariate time series features.
(3) The shortcoming of the feed-forward attention mechanism is avoided. Due to its parallelism, it cannot be used for tasks of chronological importance, which means that if it is used to extract the time steps correlation of recurrent neural networks, the order of time steps will be disturbed. In this paper, it is used to extract multivariate data so that there is no time order limitation between multivariate data at the same time step.
(4) A hybrid MLP-Feedforward Attention-LSTM model is proposed, which improves the ability of the LSTM model to extract multivariate eigenvalues.
(5) Extensive experiments are conducted on four real datasets. The experimental results demonstrate that the proposed framework is more effective than baselines, and ablation experiments validate the necessity of our framework.

II. METHODS

In this section, a hybrid MLP-Feedforward Attention-LSTM model(M-FA-LSTM) is constructed. Our proposed model mainly includes three parts: an MLP model, a feed-forward attention model, and an LSTM Model.

A. MLP LAYER

The most typical MLP includes three layers: input layer, hidden layer, and output layer. The different layers of the MLP are fully connected. Fully connected means that any neuron in the previous layer is connected to all neurons in

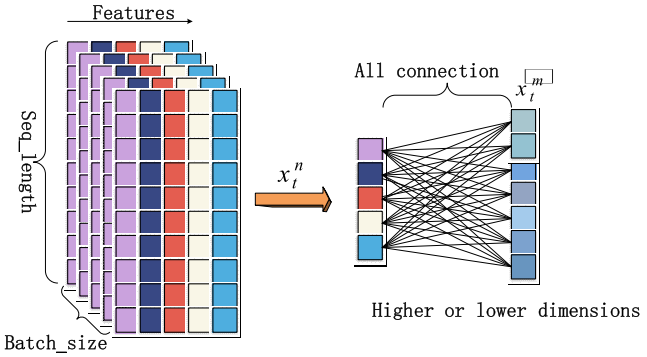


FIGURE 1. Details of MLP layer.

the next layer. In this paper, the output layer is replaced by the following feed-forward attention layer and LSTM layer. The architecture of the MLP layer is shown in Fig. 1. In the MLP, the input layer and hidden layer are used to map the initial sequences into a space that can better express features relationship. The formula is as follows:

$$\widetilde{x_t^m} = w \cdot x_t^n + b \tag{1}$$

where w is the N-dimensional weight vector of the network, b is the N-dimensional bias vector, $w \cdot x_t^n$ is the inner product of w and x_t^n. Besides, the value of the N-dimensional vector of w and b are required to be in the real number domain.

In the MLP layer, only a linear layer is implemented that maps from one space to another, without using activation functions.

B. FEED-FORWARD ATTENTION LAYER

Attention Mechanisms are a complex weighted sum. A small amount of important information is selectively filtered and focused from a large amount of information by assigning weights. They are also considered a soft addressing process by many researchers [25]. The weight coefficient indicates the importance of information. The larger the weight coefficient is, the more it focuses on its corresponding value.

A typical attention mechanism used in recurrent neural networks is proposed by [26]. It weights the sum of the correlation between the state of the previous time step s_{t-1} and the current sequence of hidden states h_j to obtain the attention weights of the hidden state sequence h. Finally, a context vector c_t with attention weights is calculated through this information. This attention mechanism changes the dependency among the time-step hidden states of recurrent neural networks. The state of the current time step not only depends on the state of the previous time step, but also can extract the required information from the state of other interval time steps, thus making it easier to model long-term dependencies. The above process is calculated by using the following formula:

$$e_{tj} = a\left(s_{t-1}, h_j\right) \tag{2}$$

$$\alpha_{tj} = \frac{\exp\left(e_{tj}\right)}{\sum_{k=1}^{T} \exp\left(e_{tk}\right)} \tag{3}$$

$$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j \tag{4}$$

where T is the number of time steps in each sequence, e_{tj} is a scalar importance value of the given pre-state s_{t-1} and the hidden state h_j, which is calculated by the learnable function a. And α_{tj} are the weight coefficients of the corresponding hidden stateh_j at each time step t. And c_t is a context vector that combines the weighted sum of the state sequence h in time T.

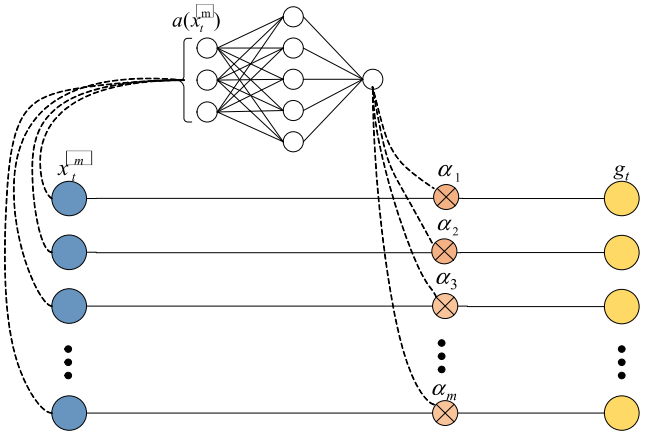


FIGURE 2. Details of the feed-forward attention layer.

A consequence of using attention mechanisms is the ability to integrate information over time. In [24], a simplified version of the above-mentioned attention mechanism is proposed. It mainly simplifies $\left(s_{t-1}, h_j\right)$ to a single vector h_t, so that the learnable function a can only be obtained by h_t, namely, the context vector c is obtained by calculating the adaptive attention weights of the state sequence h_t. The calculation process of this attention mechanism can be expressed as:

$$e_t = a\left(h_t\right) \tag{5}$$

$$\alpha_t = \frac{\exp\left(e_t\right)}{\sum_{k=1}^{T} \exp\left(e_k\right)} \tag{6}$$

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{7}$$

However, the feed-forward attention mechanism cannot be used in experiments where temporal order matters [24]. Because of the parallelism of the mechanism, they are commutable, referencing other sequences anywhere in the input when calculating the output. To avoid this limitation, the attention mechanism is applied to extract multivariate important information from the initial sequence in our method and each dimension sequence is interchangeable. Therefore, the parallelism of this mechanism has no negative impact on our model. The calculation process is reorganized as follows:

$$e_m = a\left(\widetilde{x_t^m}\right) \tag{8}$$

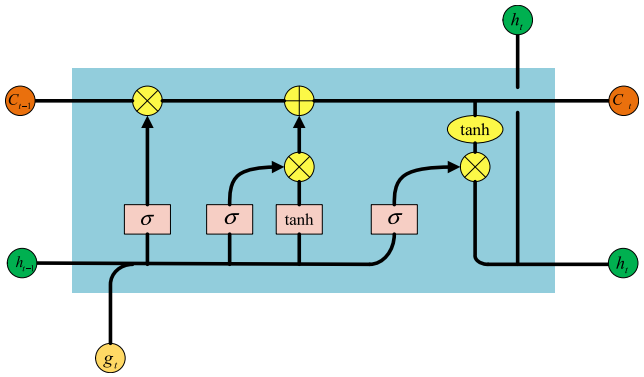


FIGURE 3. Details of the LSTM layer.

$$\alpha_m = \frac{\exp(e_m)}{\sum_{k=1}^{M} \exp(e_k)} \tag{9}$$

where M is the number of dimensions in each time step.

After getting the attention weights α_m of the initial sequences, the input sequence g_t with attention weights is obtained:

$$g_t = \beta_t \cdot \widetilde{x_t^m} \tag{10}$$

$$\beta_t = (\alpha_1, \alpha_2, \ldots, \alpha_m)^M \tag{11}$$

After the calculation of the above steps, the conversion of the initial sequences into sequences with attention weights has been completed, which is illustrated in Fig. 2. The sequences filter out key information and discard redundant information. The next work is to use these sequences as the input of LSTM, which can improve the prediction accuracy of the model.

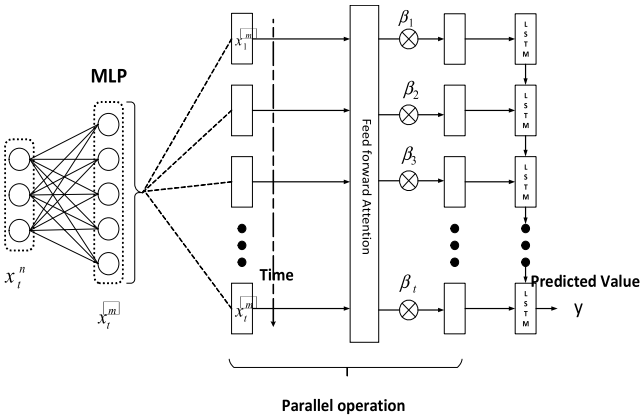


FIGURE 4. Overview of the proposed model, M-FA-LSTM.

C. LSTM LAYER

Long short-term memory (LSTM) [11] is a particular RNN, mainly to solve the problem of the gradient in the training process of long sequences, shown in Fig. 3. In short, compared with ordinary RNN, LSTM can perform better in longer sequences. The specific calculation process is as follows:

$$f_t = \sigma\left(W_f \times [h_{t-1}, g_t] + b_f\right) \tag{12}$$

$$i_t = \sigma\left(W_i \times [h_{t-1}, g_t] + b_i\right) \tag{13}$$

$$\tilde{C} = \tanh(W_C \times [h_{t-1}, g_t] + b_C \tag{14}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{15}$$

$$o_t = \sigma\left(W_o \times [h_{t-1}, g_t] + b_o\right) \tag{16}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{17}$$

where g_t is the sequence with attention weights calculated by the MLP layer and the feed-forward attention layer. h_{t-1} is the state of the hidden layer at the previous moment, σ is a logistic sigmoid function, and *tanh* is a hyperbolic tangent activation function. The connection vector $[h_{t-1}, g_t]$ is multiplied by the weight parameterW_f, W_i, and W_C respectively. Then, the bias value b_f, b_i, and b_C are added to the results respectively. They are converted to values between 0 and 1 through an activation function σ. This is called the gated state. There are three states, which are the forget gate, input gate, and output gate. The cell state C_t and the hidden state h_{t-1} in each LSTM are controlled by the above three gates. They decide to remember the important information at the time t.

D. MLP-FEEDFORWARD ATTENTION-LSTM

Traditional RNN-based time series forecasting frameworks usually feed the raw time series directly into the networks. Such models process all input sequence features indiscriminately, arguably a distracting behavior. The proposed M-FA-LSTM model can address this problem, which adaptively captures important feature sequences by assigning attention weights. Fig. 4 shows the flowchart method of the proposed model. The prediction process of multivariate time series based on M-FA-LSTM is as follows:

Step 1: Use the MLP layer to map the initial multivariate time series x_t^n to another dimensional space. The dimensional space can obtain more obvious feature information. The grid search method is applied to find the mapping hyperparameter. Then the time series $\tilde{x_t^m}$ is obtained after the change of dimension space. This dimensional space may be a low-dimensional space or a high-dimensional space according to the characteristics of the corresponding multivariate data. It is worth noting that in this process, activation functions are not used, and only linear mapping is performed.

Step 2: The feed-forward attention layer is used to obtain a multivariate sequence g_t with attention weights. Using the learnable function a, the multivariate sequence $\tilde{x_t^m}$ is learned to generate the corresponding attention weight coefficients α_m. Then each sequence is multiplied by the corresponding α_m to get the multivariate sequence g_t with attention weight coefficients.

Step 3: Enter the sequence g_t into LSTM. The LSTM layer is used to capture the long-term correlation and then predict the final predicted value.

TABLE 1. Statistics of all datasets, where Dimension is the initial Dimension of the dataset, Sensor is the hyperparameter of the MLP layer, Train represents the number of training sets, and Test indicates the number of test sets.

Dataset	Dimension	Sensor	Train	Test
Stock 000513	6	32	1984	351
Stock 600396	6	5	2140	378
Weather forecast	7	6	133847	23620
Solar-energy	10	8	6406	1130

III. EXPERIMENTS

In this section, 7 methods (including baseline models and hybrid models) were selected as competitors, and verified the performance of our proposed model from four real datasets.

A. COMPARE METHODS AND THEIR CONFIGURATIONS

Five kinds of methods are selected to test on 4 datasets and compared with our model results.

(1) MLP: In MLP, the settings of the input layer and output layer are consistent with those of other methods. The hidden layer is set as a layer, and the hidden neuron is set as 32. And the RELU is used as the activation function to train the model.

(2) RNN: Recurrent neural network is a kind of network with short-term memory ability, which can efficiently process sequence data. The model selects only one hidden layer, and it is set to 32 neurons.

(3) GRU: Gated recurrent unit is an effective variant of LSTM. It simplifies the forget gate and input gate of LSTM into an update gate, and simultaneously carries out two steps of forgetting and remembering. It can also solve the long dependency problem in RNN. The parameter settings here are consistent with the above RNN model.

(4) LSTM: Long short-term memory. LSTM filters the information that needs to be retained through gating rules. The parameter settings here are consistent with the above RNN model.

(5) LSTM-FA: a hybrid model. The model is composed of the LSTM layer and the feed-forward attention layer. The feed-forward attention layer is used to extract the correlation between time steps of the LSTM layer. The parameter settings here are consistent with the above LSTM model. This experiment is mainly to prove the limitation proposed in [24]. That is, the feed-forward attention mechanism will likely cause the failure of chronologically important tasks because of its parallelism.

B. EXPERIMENTAL EXAMPLES

In order to prove the effectiveness of our model on different types of time series datasets, empirical research on four datasets in different fields is conducted. Table 1 describes the relevant information about the datasets used in the experiments:

Stock datasets: two different types of stocks are used for this kind of dataset from 2011-01-04 to 2021-11-01, and stock codes are 000513 and 600396 respectively. Stock data is scraped on the NetEase Finance website using a web crawler. The input features include closing price, high price, low price, opening price, transaction volume, and transaction amount.

Weather forecast dataset[1]: This dataset contains the period from 2011-01-01 to 2016-12-31, and is the Jena climate dataset recorded by the Max Planck Institute for Biogeochemistry. The dataset consists of 14 features such as temperature, pressure, and humidity, which are recorded every 10 minutes. Because no significant change is expected for 60 minutes, one point per hour is resampled. In addition, among these features, some redundant features are not strongly correlated. Such as ''Vapor pressure'' and ''Vapor pressure deficit'' contain repetitive information, so these features are actively screened before starting the experiment. Then seven features with strong correlation are left (Pressure, Temperature, Saturation vapor pressure, Vapor pressure deficit, Specific humidity, Airtight, and Wind speed).

Solar-energy dataset [27]: This dataset contains the period from 2016-02-01 to 2017-11-31, and hourly resolution is used, from 6 am to 5 pm. The dataset consists of ten features: Hour, Cloud coverage, Visibility, Temperature, Dew point, Relative humidity, Wind speed, Station pressure, Altimeter, Solar-energy.

C. PARAMETER SETTINGS AND SENSITIVITY

The test environment is as follows: CPU is Intel(R) Core (TM) i7-6500U CPU @ 2.50GHz 2.59 GHz, and 12.0 GB (11.9 GB usable) RAM. OS is Windows 10. Based on Python 3.7, Keras is used as the deep learning framework for all the model development and its performance evaluations.

In the three-layer architecture of the model, the hyperparameter of the MLP layer is mainly tuned. Because if the MLP layer maps the initial sequence into a space that better expresses the feature information, the prediction performance can also be improved accordingly. Therefore, grid search [28] is used in our work to find the relative optimal hyperparameter. Hyperparameters are chosen based on the number of input features for different tasks and datasets. The grid of hidden neurons is {5,8,10,16,32,64} for both the stock datasets, {5,6,7,8,10,16,32} for the Weather forecast dataset, and {6,8,10,12,16,32} for the Solar-energy dataset. Our purpose is to prove the validity of the model, so there is no intensive search for the best hyperparameter. As for the feed-forward attention layer, it mainly adjusts the attention weights through self-learning, so there is no hyperparameter to be adjusted. For the LSTM layer,

[1]https://storage.googleapis.com/tensorflow/tk-keras-datasets/jena_climate_2009_2016.csv.zip

TABLE 2. Comparison of prediction measures of different methods. The best results are highlighted in bold.

	Stock 000513			Stock 600396			Weather forecast			Solar energy		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
MLP	0.7187	0.8534	2．01%	0.2453	0.3044	9.52%	3.0789	3.3500	205.94%	593.1131	664.5358	253.36%
RNN	0.5877	0.6764	1.65%	0.1324	0.1579	5.11%	2.1204	2.2062	136.70%	479.4548	580.3198	2.0614%
GRU	0.5863	0.8148	1.64%	0.1285	0.1632	4.94%	1.5814	1.8516	109.04%	372.2161	444.3662	128.97%
LSTM	0.6112	0.8538	1.71%	0.1235	0.1631	4.72%	1.6285	2.1038	117.11%	341.2768	429.8723	102.76%
LSTM_FA	0.6666	0.8074	1.86%	0.1199	0.1708	4.55%	1.6173	1.8747	94.75%	386.4271	451.4381	131.09%
M-FA-LSTM	**0.4109**	**0.6293**	**1.15%**	**0.0942**	**0.1265**	**3.60%**	**1.2718**	**1.4579**	**89.44%**	**289.6937**	**403.7671**	**73.12%**

all our models use a single-layer LSTM with 32 hidden units.

Since both the stock datasets and the Solar-energy dataset are small, a batch size of 16 is used in the experiments. The weather forecast dataset is relatively large, so the batch size is set to 256. We set different time steps for different datasets, specifically: the step size is set to 5 for the two stock datasets, 120 for the Weather forecast dataset, and 12 for the Solar-energy dataset. To keep our model from overfitting, a layer of dropout [29] is added after the LSTM layer and its dropout rate is set to 0.2. Moreover, the Adam algorithm [30] is used to optimize our deep learning model, and its optimizer sets the initial learning rate to 0.001. Finally, to improve the efficiency of the model with a maximum number of 100 epochs, an automatic pause is preset. For five times running, if the loss value of the test set is not improved, then the model stops training. MAE is set as the loss function. What's more, it is possible to improve the predictive performance of the model by adjusting the number of layers or parameters of the LSTM. However, keeping the hyperparameters of the model basically consistent can better highlight the main message of the paper.

In addition, for the division of datasets, the first 85% of the data are used as the training set and the last 15% are used as the test set. To eliminate the dimensional influence between the features, the normalization method is used to preprocess the data.

D. METRICS

To compare the prediction performance of different methods, three general evaluation metrics are adopted: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \widetilde{y}_i| \tag{18}$$

$$RMSE = \left(\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widetilde{y}_i)^2\right)^{\frac{1}{2}} \tag{19}$$

$$MAPE = \sum_{i=1}^{n}\left|\frac{Y_i - \widetilde{y}_i}{Y_i}\right| \times \frac{100}{n} \tag{20}$$

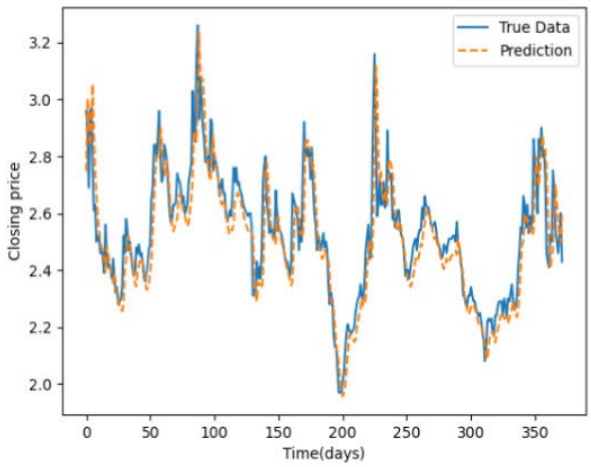
where Y_i is the predicted value and $\widetilde{y}_i$ is the true value. The lower these three values are, the better the effect of the corresponding model is.

E. RESULTS AND DISCUSSION

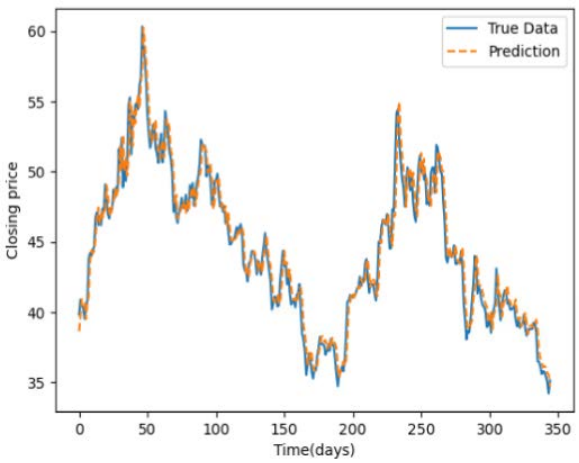
1) MAIN RESULTS

Table 2 summarizes the experimental results of six methods on four datasets. The experiments are conducted on three different types of time series prediction tasks, in which four datasets have different dimensions and sequence lengths. Correspondingly, for two stock datasets, the experiments use the data of the past five days (5-time points) to predict the closing price of the sixth day (1-time point). For the Weather forecast dataset, the experiments use the data of the past 120 hours (120-time points) to predict the temperature 12 hours later (1-time point). For the Solar-energy dataset, the experiments use data collected 12 hours (12-time points) to predict the amount of Solar-energy an hour in future collection periods (1-time point). As shown in Table 2, The best values of the metrics are highlighted in bold. Our model achieves the highest prediction performance on all tasks.

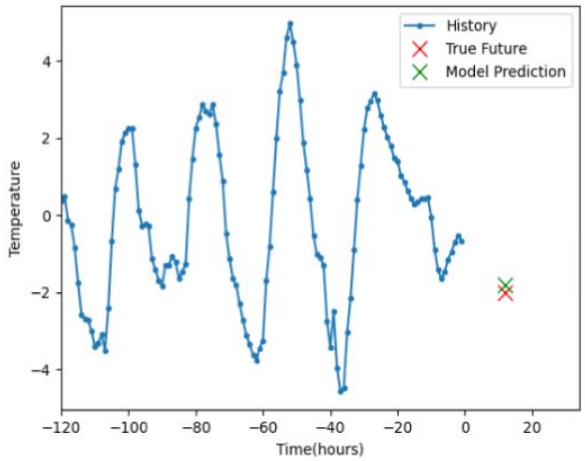
To better display the prediction results, the experimental results of our model on four tasks are visualized, as shown in Fig. 5. In Fig. 5 (a), (b), and (d), the solid blue line represents the true value and the dashed orange line represents the forecast value. In subgraph (c), one forecast result from the test dataset is shown. The red cross represents the true value, and the green cross represents the predicted value. The preceding



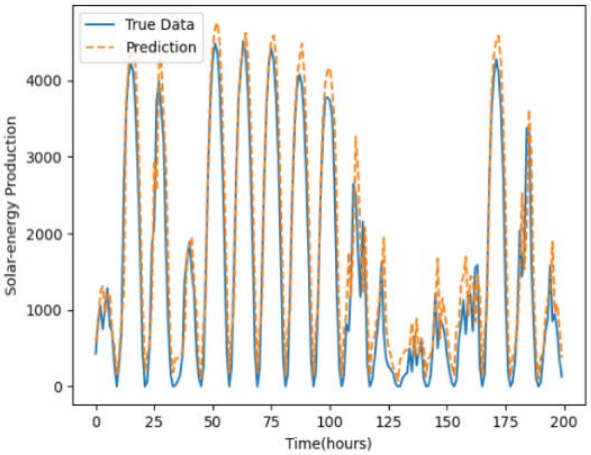
(a) Stock 000513



(b) Stock 600396



(c) Weather forecast



(d) Solar-energy

FIGURE 5. Prediction results of the M-FA-LSTM method on the datasets Stock 000513, Stock 600396, Weather forecast, and Solar-energy.

blue curve represents the historical temperature curve. From Fig. 5, it can be concluded that our model is adaptable to different types of time series forecasting tasks with good generality. Incidentally, compared to more complex and unstable datasets such as the stock datasets, the relatively stable Solar-energy dataset and the Weather forecast dataset can fit better.

Figs. 6 and 7 show examples of loss plots for the stock 000513 dataset and the Weather forecast dataset under the six methods. From these two sets of graphs, it can be seen that the performance of our proposed model is the best. And the curve of the loss graph decreases steadily. This proves that our model captures the important features needed for prediction well. Therefore, the loss graphs of the proposed model perform better than other methods during the training process.

TABLE 3. Test values of hyperparameters and corresponding experimental results for the Stock 000513 dataset.

		Stock 000513		
	Metrics	MAE	RMSE	MAPE
	LSTM	0.6112	0.8538	1.71%
MLP-layer neurons	5	0.4927	0.6933	1.38%
	8	0.4651	0.7129	1.30%
	10	0.4606	0.6581	1.29%
	16	0.4082	0.5930	1.15%
	32	0.4109	0.6293	1.15%
	64	0.4141	0.6093	1.16%

2) DISCUSS THE MAPPING SPACE OF THE FOUR DATASETS AT THE MLP LAYER

Both the stock datasets have the same six initial feature dimensions. For the Stock 000513, it can be seen from

TABLE 4. Test values of hyperparameters and corresponding experimental results for the Stock 600396 dataset.

		Stock 600396		
	Metrics	MAE	RMSE	MAPE
	LSTM	0.1235	0.1631	4.72%
MLP-layer neurons	5	0.1359	0.1717	5.23%
	8	0.1239	0.1469	4.77%
	10	0.0995	0.1363	3.79%
	16	0.1163	0.1458	4.46%
	32	0.0942	0.1265	3.60%
	64	0.1059	0.1290	4.07%

TABLE 5. Test values of hyperparameters and corresponding experimental results for the Weather forecast dataset.

		Weather forecast		
	Metrics	MAE	RMSE	MAPE
	LSTM	1.6285	2.1038	117.11%
MLP-layer neurons	5	1.2999	1.6120	92.29%
	6	1.2718	1.4579	89.44%
	7	1.4409	1.6723	96.95%
	8	1.858	2.111	125.74%
	10	2.3762	2.7362	149.16%
	16	1.8515	2.2056	121.22%
	32	1.9253	2.2646	124.21%

TABLE 6. Test values of hyperparameters and corresponding experimental results for the Solar energy dataset.

		Solar energy		
	Metrics	MAE	RMSE	MAPE
	LSTM	341.2768	429.8723	102.76%
MLP-layer neurons	6	289.6937	403.7671	73.12%
	8	290.1015	380.0861	75.12%
	10	336.0065	440.3732	108.88%
	12	381.2413	445.8683	133.31%
	16	360.4628	453.0015	128.54%
	32	378.3226	469.2581	125.53%

Table 3 that our model achieves better results than LSTM on all six tested hyperparameters. In the dimensional space of 16, 32, and 64, the model can achieve better prediction results. Furthermore, in the process of many experiments, it is found that the results are more stable under the dimensional space of 10 and 16. We believe these are the latent dimension spaces that best represent this dataset, so the feed-forward attention layer can extract better feature information from them. For the Stock 600396 dataset, Table 4 shows that under the dimensional space of 10, 16, 32, and 64, the model can achieve better prediction results than LSTM. Similarly, in the process of several experiments, it is found that the prediction performance of the model is more stable under the dimensional space of 32 and 64. The reason why these two datasets are stable in different dimensional spaces is mainly due to the chaos and complexity of stock data. Different stock datasets have different feature representations.

For the Weather prediction dataset, it can be seen from Table 5 that in low dimensional spaces of 5, 6, and 7, better results can be obtained compared with LSTM. The main reason why this dataset does not require large changes to the mapping space is that it has relatively stable periodicity and regularity. For the solar dataset, it can be seen from Table 6 that the model can achieve better prediction results than LSTM in low dimensional spaces of 6, 8, and 10. This dataset is similar to the Weather forecast dataset, with regular and stable characteristics, so their features are more obvious and do not need to be mapped to a high-dimensional space. The experimental results show that our model achieves good performance on different types of datasets.

3) DISCUSS THE CLASSIC USAGE OF THE FEEDFORWARD ATTENTION MECHANISM IN LSTM

In this subsection, the feed-forward attention mechanism is used to extract relevant time steps of the LSTM layer (LSTM-FA). This is a classic usage of recurrent neural networks integrated with attention mechanism to enhance the selection of relevant time steps. Table 7 shows that the addition of the feed-forward attention layer can improve the prediction performance of some datasets, while its performance is inferior to baseline LSTM in other datasets. The main reason is that the parallel calculation of the feed-forward attention mechanism causes the model to lose important time information, which makes the experimental results random.

4) ABLATION STUDIES

In order to explore the validity of each module in M-FA-LSTM, ablation studies are carefully conducted:

FA-LSTM: Remove the MLP module, leaving only the feed-forward attention module.

MLP-LSTM: The feed-forward attention module is deleted, leaving only the MLP feature mapping module.

Generally speaking, the feed-forward attention layer, which is responsible for capturing feature relationships, plays a key role in the predictive performance of the model. Therefore, we conduct validation experiments on the FA-LSTM model with MLP layers removed. However, Table 3 shows that the feed-forward attention layer does not capture useful data without feature mapping of the chaotic initial data in advance. Conversely, the feed-forward attention layer captures the wrong information, causing the FA-LSTM to perform worse than LSTM. This is enough to prove the necessity of our design, and the function of the MLP mapping layer is essential. Similarly, in MLP-LSTM with the feedforward attention layer removed, its performance is not well improved compared to LSTM. This proves that neither the MLP layer nor the feedforward attention layer is the

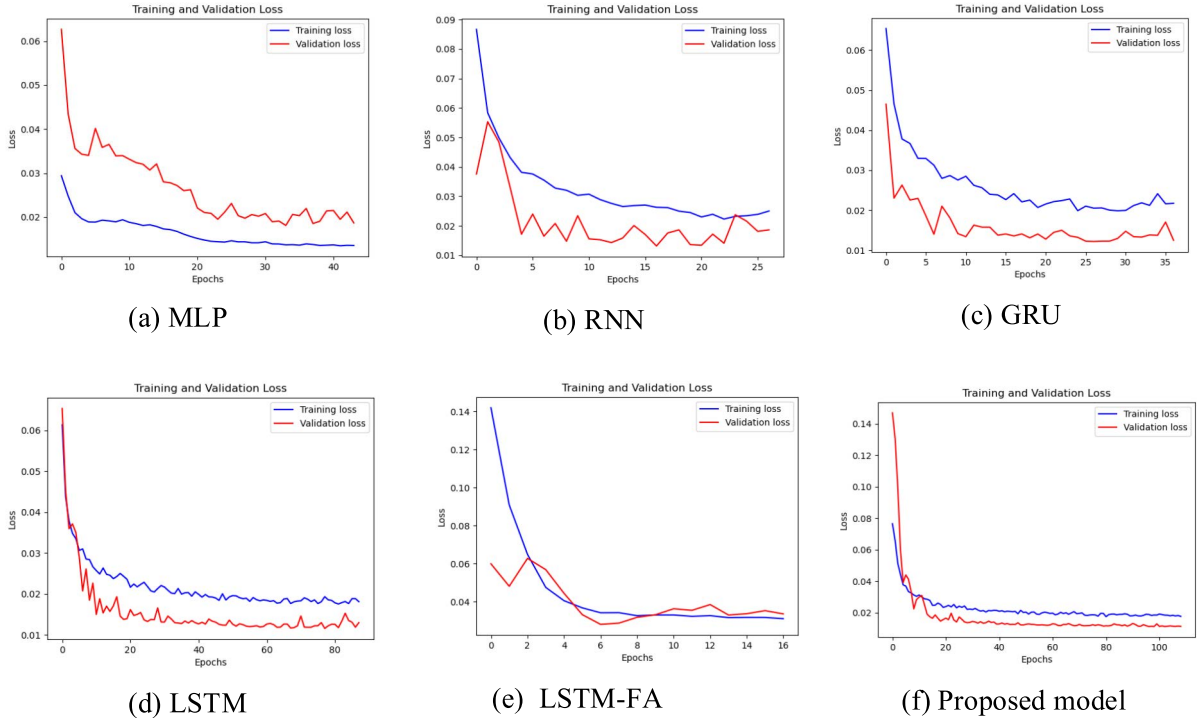


FIGURE 6. Plots of loss functions for six methods on the stock 000513. The blue line represents the training loss, and the red line represents the test loss. From the loss function plots, we can see that the proposed model performs the best with rapid decline and little fluctuation.

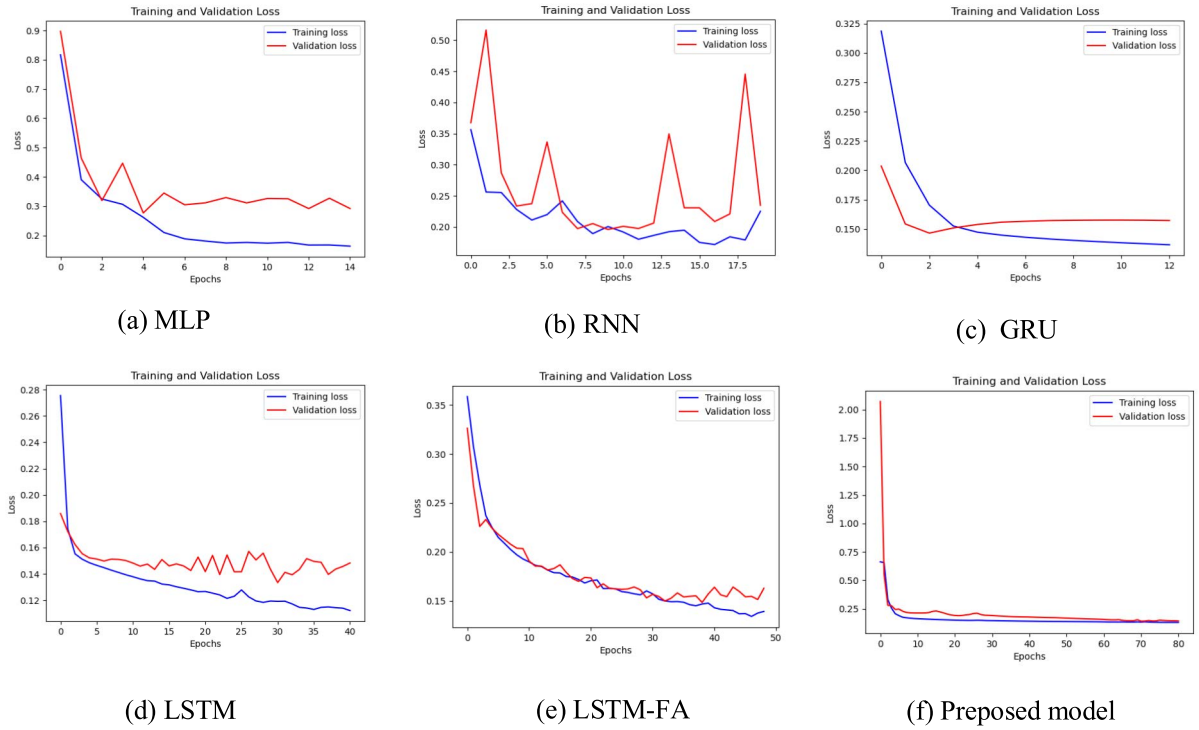


FIGURE 7. Plots of loss functions for six methods on the Weather forecast. The blue line represents the training loss, and the red line represents the test loss. From the loss function plots, we can see that the performance of the proposed model is the most stable. And the two curves are stable and very close, indicating that the model captures the main features required during the training process.

main reason for the improved performance of M-FA-LSTM. The experimental results show that only combining the three modules can significantly improve the final prediction performance.

TABLE 7. Comparison of FA-LSTM, MLP-LSTM, and our model.The best results are highlighted in bold.

	Stock 000513			Stock 600396			Weather forecast			Solar energy		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
MLP-LSTM	0.62522	0.8453	1.75%	0.1154	0.1422	4.43%	1.9249	2.0517	129.66%	389.8281	453.8535	140.43%
FA-LSTM	0.7293	0.9558	2.04%	0.1381	0.1699	5.32%	2.2318	2.4647	146.16%	386.4272	451.4381	131.10%
M-FA-LSTM	**0.4109**	**0.6293**	**1.15%**	**0.0942**	**0.1265**	**3.60%**	**1.2718**	**1.4579**	**89.44%**	**289.6937**	**403.7671**	**73.12%**

IV. CONCLUSION

In this paper, we propose an LSTM model based on MLP and feed-forward attention mechanism (M-FA-LSTM) to predict multivariate time series. MLP has the ability to map the original features to the latent space, so the model can reorganize the feature information in the dimensional space. In addition, assigning attention weights through feed-forward attention layers also furthers the ability of important feature information extraction, so the proposed model can retain more feature information than LSTM. Most of the existing methods usually employ an attention mechanism to extract relevant information between time steps of recurrent neural networks. However, our proposed model employs a feed-forward attention mechanism, which not only captures the multivariate feature information of the initial sequence well, but also makes full use of the related driving sequences. The idea of the model proposed in this paper can be extended based on other recurrent neural networks, and further enhance the ability to extract multiple features. We use four real datasets to experimentally verify the proposed model, and the results show that our model has better performance than the baselines in all three metrics. Meanwhile, ablation experiments are carried out to verify the necessity of each module in each model. In future work, we will consider other better-performing mapping layers (e.g., Convolutional Neural Network) to replace the MLP mapping layers to further improve the model performance.

REFERENCES

[1] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting," *IEEE Access*, vol. 8, pp. 71326–71338, 2020, doi: 10.1109/ACCESS.2020.2985763.

[2] S. M. Bhagavathi, A. Thavasimuthu, A. Murugesan, C. Latha, L. Raja, and R. Thavasimuthu, "Weather forecasting and prediction using hybrid C5.0 machine learning algorithm," *Int. J. Commun. Syst.*, vol. 34, Jul. 2021, Art. no. e4805.

[3] K. Wang, K. Li, L. Zhou, Y. Hu, Z. Cheng, J. Liu, and C. Chen, "Multiple convolutional neural networks for multivariate time series prediction," *Neurocomputing*, vol. 360, pp. 107–119, Sep. 2019, doi: 10.1016/j.neucom.2019.05.023.

[4] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2015, doi: 10.1111/jtsa.12194.

[5] J. Li, B. Wu, X. Sun, and Y. Wang, "Causal hidden Markov model for time series disease forecasting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12105–12114.

[6] F. J. J. dos Santos and H. de Arruda Camargo, "Forecasting in fuzzy time series by an extension of simple exponential smoothing," in *Proc. Ibero-Amer. Conf. Artif. Intell.*, vol. 8864, Nov. 2014, pp. 257–268, doi: 10.1007/978-3-319-12027-0_21.

[7] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, "An experimental review on deep learning architectures for time series forecasting," *Int. J. Neural Syst.*, vol. 31, no. 3, Mar. 2021, Art. no. 2130001, doi: 10.1142/S0129065721300011.

[8] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990, doi: 10.1016/0364-0213(90)90002-E.

[9] L. Vincent and N. Thome, "Shape and time distortion loss for training deep time series forecasting models," in *Proc. NeurIPS*, 2019, pp. 4189–4201.

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

[11] S. Hochreiter and J. J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[12] S. Elsworth and S. Güttel, "Time series forecasting using LSTM networks: A symbolic approach," 2020, *arXiv:2003.05672*.

[13] J. Zhao, F. Huang, J. Lv, Y. Duan, Z. Qin, G. Li, and G. Tian, "Do RNN and LSTM have long memory?" in *Proc. 37th Int. Conf. Mach. Learn.*, Nov. 2020, pp. 11365–11375.

[14] Y. Hu, F. O'Donncha, P. Palmes, M. Burke, R. Filgueira, and J. Grant, "A spatio-temporal LSTM model to forecast across multiple temporal and spatial scales," 2021, *arXiv:2108.11875*.

[15] Y. Cong, X. Zhao, K. Tang, G. Wang, Y. Hu, and Y. Jiao, "FA-LSTM: A novel toxic gas concentration prediction model in pollutant environment," *IEEE Access*, vol. 10, pp. 1591–1602, 2022, doi: 10.1109/ACCESS.2021.3133497.

[16] J. Yang, J. Qu, Q. Mi, and Q. Li, "A CNN-LSTM model for tailings dam risk prediction," *IEEE Access*, vol. 8, pp. 206491–206502, 2020, doi: 10.1109/ACCESS.2020.3037935.

[17] S. Merity, "Single headed attention RNN: Stop thinking with your head," 2019, *arXiv:1911.11423*.

[18] 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, doi: 10.24963/ijcai.2017/366.

[19] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6910–6920, Nov. 2021, doi: 10.1109/TITS.2020.2997352.

[20] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowl.-Based Syst.*, vol. 181, Oct. 2019, Art. no. 104785.

[21] S.-Y. Shih, F.-K. Sun, and H.-Y. Lee, "Temporal pattern attention for multivariate time series forecasting," *Mach. Learn.*, vol. 108, nos. 8–9, pp. 1421–1441, Sep. 2019, doi: 10.1007/s10994-019-05815-0.

[22] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," 2020, *arXiv:2012.07436*.

[23] W. Zheng, P. Zhao, K. Huang, and G. Chen, "Understanding the property of long term memory for the LSTM with attention mechanism," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, p. 2717, doi: 10.1145/3459637.3482399.

[24] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," 2015, *arXiv:1512.08756*.
[25] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," 2015, *arXiv:1503.08895*.
[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent., Conf. Track (ICLR)*, 2016, pp. 1–15.
[27] A. Kuzmiakova, G. Colas, and A. McKeehan, "Short-term memory solar energy forecasting at University of Illinois," Tech. Rep., 2017.
[28] C. Hsu, C. Chang, and C.-J. Lin, "A practical guide to support vector classication," *Bioinformatics*, vol. 1, pp. 1396–1400, Jan. 2003.
[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 56, pp. 1929–1958, 2014.
[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent., Conf. Track, (ICLR)*, 2015, pp. 1–15.

YUNTONG LIU received the B.S. degree from Hubei University, Wuhan, China, in 2020. She is currently pursuing the M.S. degree with Yunnan University, Kunming, China. Her research interests include data mining, time series forecasting, and deep learning.

CHUNNA ZHAO was born in Liaoning, China, in 1978. She received the B.S. degree from Liaoning Normal University, Dalian, China, in 2001, and the M.S. and Ph.D. degrees from Northeastern University, Shenyang, China, in 2004 and 2006, respectively.

She is currently a Professor with Yunnan University. Her current research interests include fractional systems and artificial intelligence prediction method.

YAQUN HUANG was born in Sichuan, China, in 1971. She received the B.S. and M.S. degrees from East China Normal University, Shanghai, China, in 1993 and 1996, respectively.

She is currently an Associate Professor with Yunnan University. Her current research interests include image processing and intelligent data analysis.