IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 April 2023, accepted 11 May 2023, date of publication 15 May 2023, date of current version 23 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3276628

APPLIED RESEARCH

Time Series Prediction Based on LSTM-Attention-LSTM Model

XIANYUN WEN AND WEIBANG LI, (Member, IEEE)
The Key Laboratory for Computer Systems of State Ethnic Affairs Commission, Southwest Minzu University, Chengdu 610041, China
School of Computer Science and Engineering, Southwest Minzu University, Chengdu 610041, China

Corresponding author: Weibang Li (wbli2003@163.com)

This work was supported in part by the Southwest Minzu University for Nationalities 2022 Central Universities Basic Research Business Fund Special Funds for Excellent Student Training under Project 2022NYXXS070, and in part by the Sichuan Provincial Social Science Research Fund under Project SC20B127.

ABSTRACT Time series forecasting uses data from the past periods of time to predict future information, which is of great significance in many applications. Existing time series forecasting methods still have problems such as low accuracy when dealing with some non-stationary multivariate time series data forecasting. Aiming at the shortcomings of existing methods, in this paper we propose a new time series forecasting model LSTM-attention-LSTM. The model uses two LSTM models as the encoder and decoder, and introduces an attention mechanism between the encoder and decoder. The model has two distinctive features: first, by using the attention mechanism to calculate the interrelationship between sequence data, it overcomes the disadvantage of the coder-and-decoder model in that the decoder cannot obtain sufficiently long input sequences; second, it is suitable for sequence forecasting with long time steps. In this paper we validate the proposed model based on several real data sets, and the results show that the LSTM-attention-LSTM model is more accurate than some currently dominant models in prediction. The experiment also assessed the effect of the attention mechanism at different time steps by varying the time step.

INDEX TERMS Time series forecasting, long short-term memory networks, encoder and decoder model, attention mechanisms.

I. INTRODUCTION

With the rapid development of technologies such as Big Data and the Internet of Things, massive amounts of data have been accumulated in the fields of industrial engineering, financial technology and natural sciences, among which time series data (data arriving in the order of their time stamps) is an important component. The use of these time series data to predict its future state over time has a wide range of applications, such as in the financial sector for stock price forecasting and cash flow forecasting, in the transportation sector for traffic flow forecasting [1], in the retail sector for business revenue forecasting, etc. It is also widely used in the meteorological and tourism sectors to help decision makers make important decisions based on data support. Time series forecasting is widely used in real life, and its essence is mainly to project the value of the time series at time T+1 from the observations at the previous T moments. Although time series forecasting has important applications, it can be challenging to perform. As time series data is often generated with noise and missing values, the quality of the data will be greatly compromised, which directly affects the prediction accuracy of the data. In addition, inadequate forecasting models and insufficient data, for example, can also affect the effectiveness of time series forecasting. Considering the important value of time series forecasting, a number of scholars have conducted research on this issue.

There are currently a number of methods for forecasting time series data, which can be divided into two categories: traditional forecasting methods and machine learning based forecasting methods. Typical traditional methods include autoregressive (AR) [2] models, moving average (MA) [3] models, autoregressive moving average (ARMA) [4] models and differential autoregressive moving average (ARIMA) [5]

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

models. However, these traditional methods usually have certain limitations in forecasting. These methods tend to be relatively stringent in terms of data requirements, as they can only deal with some smooth data, but in real life time series data are basically unstable, so the accuracy of the traditional methods for forecasting data is often not high. With the development of artificial intelligence technology, some scholars began to shift their attention to neural networks and proposed some machine learning-based methods to consider deep learning techniques for solving time series prediction problems. Since recurrent neural networks (RNNs) [6] have achieved good results in speech recognition and text generation, researchers have tried to apply RNNs to time series prediction with some results. However, RNNs have a fatal drawback, that is, RNNs are prone to gradient disappearance and gradient explosion during the training process, so some new networks have been derived, such as long short-term memory networks (LSTM) [7] and gated recurrent unit networks (GRU) [8]. Although these models solve the problem of gradient disappearance and explosion, the prediction accuracy is not yet very high, and a variety of variants of LSTMs have been derived, such as stacked-LSTM [9] and bi-directional LSTM (BiLSTM) [10]. These variants do work better than the normal LSTM, but there is no research to show which of these variants are actually applicable to which time series. In this paper, we use the above-mentioned variants of LSTM to construct models to compare the prediction of time series data and to evaluate the prediction effectiveness of these models. The models are validated on different real data sets and the experimental results show the shortcomings of the existing methods.

In view of this situation, this paper proposes a new hybrid model, which is based on an LSTM network with an encoder-decoder structure [11] and an attention-mechanism [12]. The main contributions of this paper are 3-fold.

(1) An LSTM-attention-LSTM prediction model is proposed, which introduces an attention mechanism into the encoder-decoder structure.

(2) The new model is applied to time series prediction and compared with the current mainstream models. The experimental results show that our proposed model has more obvious advantages in terms of prediction accuracy.

(3) Experiments based on real data sets reveal the applicability of the existing mainstream models and the proposed model under different time step conditions.

II. RELATED WORK

Time series data are generally classified into two types, smooth and non-smooth, depending on the trend of the data. A smooth time series is one in which the mean and variance of the series do not vary systematically, while a non-smooth time series is generally one that contains characteristics such as trends, seasonality or periodicity.

For smooth time series, simple forecasting is often possible using traditional models such as autoregressive models (AR), moving average models (MA) and autoregressive moving average (ARMA) models. For non-stationary time series, the traditional approach is to use a differential autoregressive moving average model (ARIMA), the main difference between this model and other traditional models being that the non-stationary series is transformed into a stationary series by means of multiple differencing. However, these traditional methods have high prediction errors, do not take full advantage of the dynamics between multiple variables, and are less efficient when dealing with larger data sets, resulting in longer run times.

To compensate for the shortcomings of traditional models, machine learning has been widely applied to the prediction of non-stationary time series, such as support vector machines (SVM), clustering [13] and neural networks [14], etc. Based on support vector machine (SVM) theory and time series analysis, Li [15] et al. developed a dynamic prediction model of surface motion, which was used to predict future surface motion and achieved good results. Kai [16] et al. used a multilayer back propagation (BP) neural network and a generalized regression neural network (GRNN) to predict wind speed sequences by using a time series model to select the input variables for the neural network, and the results showed that the algorithm has some practical value. The recent rise of deep learning has received a lot of attention, with LSTM being widely used in the field of time series prediction. LSTM is a temporal recurrent neural network suitable for processing time-series data with time delays. For example, Felix [17] and others have used LSTMs to solve many time series problems that could not be solved by using fixed window feedforward networks. Another GRU model similar to LSTM is also widely used for time series forecasting, and Liu and Chen [18] et al. achieved higher prediction accuracy in dealing with non-stationary multivariate time series forecasting by mixing GRU and MGU models to propose a linear hybrid gate unit (MIXGU). In order to improve the effectiveness of the LSTM model, many scholars have improved the LSTM according to the characteristics of the data. Dai [19] et al. predicted the PM2.5 concentration in Beijing by combining the empirical modal decomposition algorithm (EMD) and the LSTM, which achieved higher prediction accuracy compared with a single prediction algorithm.

In summary, the prediction accuracy of currently available models has improved compared to traditional models and single prediction models, but the prediction error has not been reduced significantly, and the prediction accuracy tends to decrease for time series prediction with longer time steps. To address the shortcomings of existing methods, this paper proposes an LSTM-attention-LSTM model for time series forecasting.

III. PRELIMINARIES

A. TIME SERIES

Time series generally refer to the value pairs of different values corresponding to a certain time period or point in time. The value pairs have only two specific data, the time

element and the value element. The time element is a certain time period or point in time, while the numerical element can correspond to one variable or to multiple variables. A single variable is known as a univariate time series, while multiple variables are known as multivariate time series.

B. VARIABLE TIME SERIES FORECASTING

Assume that the input is a multivariate time series data $X = (X_1, X_2, X_3, \ldots, X_t)$ with N variables of length t, where $X_t \in R^N$ denotes the data values of all N variables at time t and $X_t^i \in R$ denotes the data value of the i-th variable at time t. The output is the corresponding predicted value $Y = (Y_{t+1}, Y_{t+2}, Y_{t+3}, \ldots, Y_{t+L})$, where $L$is the time step.

IV. LSTM-ATTENTION-LSTM MODEL

In this section, we first introduce the basic architecture of the LSTM model, then describe the encoder-decoder structure and the attention mechanism, and illustrate the advantages of the encoder-decoder structure. Finally, the LSTM-attention-LSTM, the time series prediction model of this paper, is proposed and an example is given to illustrate the necessity of incorporating the attention mechanism.

A. LONG AND SHORT-TERM MEMORY NETWORK

The control process of the LSTM is similar to that of the RNN in that it processes the data as it flows through the cells during forward propagation, but the RNN is prone to gradient explosion or disappearance during training. The proposed LSTM is a good solution to this problem because the core of the LSTM lies in the cell transitions and the ''gate'' structure [20], which uses sigmoid functions with outputs either close to 0 or close to 1. This allows the gradient to pass well through the LSTM well and reduces the probability of gradient explosion or disappearance. The structure of the LSTM [21] is shown in Figure 1.

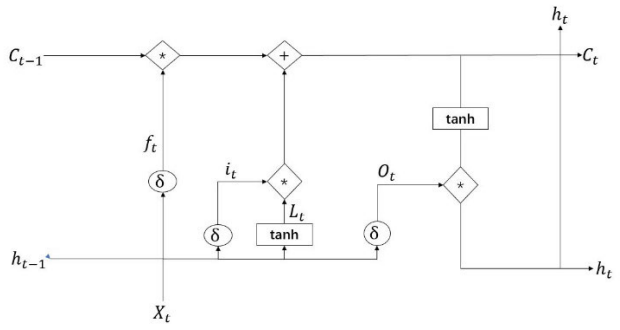

FIGURE 1. Long and short-term memory network LSTM model.

The structure of the LSTM consists of three main gates: an oblivion gate, an input gate and an output gate [19]. One of these forgetting gates can be expressed as follows:

$$f_t = \delta(W_f[h_{t-1}, X_t] + b_f) \tag{1}$$

where δ is the sigmoid activation function, X_t refers to the vector of inputs, h_{t-1} denotes the hidden vector of the previous layer, and W_f and b_f denote the weights and bias values of the inputs, respectively [22]. The input gate can be expressed as shown below:

$$i_t = \delta(W_i[h_{t-1}, X_t] + b_i) \tag{2}$$

The output gate controls the output of the cell state values and can be expressed as follows:

$$O_t = \delta(W_o[h_{t-1}, X_t] + b_o) \tag{3}$$

In addition to the three gates, a candidate memory cell can be represented as follows:

$$L_t = \tanh(W_c[h_{t-1}, X_t] + b_c) \tag{4}$$

An updated representation of the current cell state is as the following:

$$C_t = f_t \times C_{t-1} + i_t \times L_t \tag{5}$$

The final output of the current hidden state and its use as input to the next layer of the LSTM can be represented as shown below:

$$h_t = O_t \times \tanh(C_t) \tag{6}$$

Therefore, it is easy to see from the LSTM formula and the structure diagram that the LSTM is very good at handling gradient explosion or disappearance, so the LSTM has some advantages in time series prediction.

B. ENCODER-DECODER MODEL

The code-and-decode model was first proposed to solve the seq2seq problem, with the aim of translating text or answering input questions. Since then, researchers have tried to apply the model to time series prediction and have achieved good results. The code-and-decode model is shown in Figure 2.

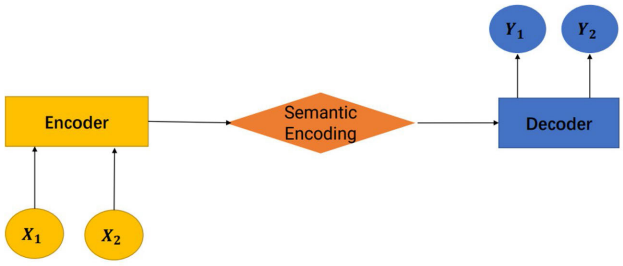

FIGURE 2. Encoder-decoder model.

Encoding is the process of converting the input sequence into a vector of fixed length, in the middle of which there will be a process of semantic encoding and finally decoding, and decoding is the process of converting the previously fixed vector back into the output sequence. The encoder and decoder are not fixed, the models inside can be replaced at will, common ones are LSTM, RNN and CNN etc.

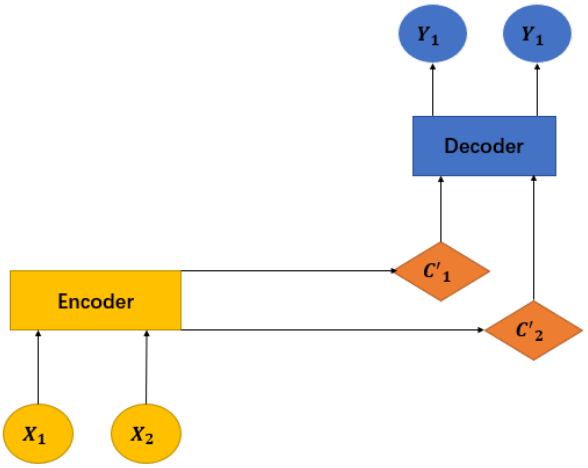

FIGURE 3. Encoder-attention-decoder model.

C. ATTENTION MECHANISM

The code-and-decode model was first proposed to solve the seq2seq problem, with the aim of translating text or answering input questions. It has since been applied to time series prediction with promising results. The encoder-decoder model is more effective than a single model, but it has some limitations. The model has only one link between the encoder and decoder models, and this link is a fixed-length vector, so if the input sequence is long, the previous input sequence is forgotten by the later input sequence. In addition, this fixed-length vector cannot fully represent the whole sequence. To solve this problem, the attention mechanism is added to the encoder-decoder model. The encoder-decoder model after the addition of the attention mechanism is shown in Figure 3.

It is easy to see from Figure 3 that with the addition of the attention mechanism, the model no longer feeds the output sequence of the encoder into a fixed vector, but has multiple vectors to share the work, thus solving the problem of inattention in the model.

D. LSTM-ATTENTION-LSTM MODEL

Based on the above LSTM model, encoder-decoder model and attention mechanism mentioned above, this paper proposes a new model for predicting time series. Considering the success of the above models in text recognition and translation, this paper considers the fusion of all three models to take full advantage of multiple models. The structure of the model is shown in Figure 4.

The LSTM-attention-LSTM model structure consists of an input layer, an encoding layer, an attention layer, a decoding layer and an output layer, where the input layer first inputs the pre-processed [sample, time step, feature] style 3D time series data $X = (X_1, X_2, X_3 \ldots\ldots X_t)$, which is encoded by the encoding layer, followed by the attention layer to calculate the attention weights, and then the decoding layer to decode the data, and finally the output layer to calculate

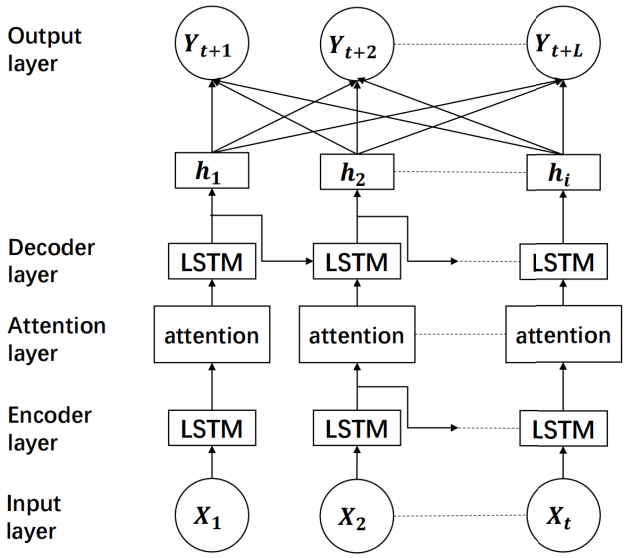

FIGURE 4. Structure of LSTM-attention-LSTM model.

the predicted sequence data of the next time step $Y = (Y_{t+1}, Y_{t+2}, Y_{t+3}, \ldots, Y_{t+L})$, where t is the length of the input sequence and L is the time step.

The inputs to the model include a time series X, a time step L and a training count n. The intermediate layers are processed once through the coding, attention and decoding layers, and the final outputs are the predicted time series Y, the root mean square error (RMSE) and the mean absolute percentage error (MAPE). Detailed descriptions of the coding, attention and decoding layers are given in the following subsections.

1) CODING LAYER

The LSTM-attention-LSTM model first encodes the input 3D vector $X(X_1, X_2, X_3 \ldots\ldots X_t)$, the encoder is composed of an LSTM, and the encoding layer is calculated as follows:

$$h_i = \mathrm{f}(X_i, h_{i-1}) \quad 1 \le i \le t \tag{7}$$

where h_i denotes the hidden state at moment i calculated by the coding layer, h_{i-1} denotes the hidden state at moment $i-1$, t denotes the time step, and f denotes the calculation function of the input gate, forgetting gate and output gate in the LSTM model, which are calculated in Equations (1)-(6).

2) ATTENTION LAYER

After the coding layer, the output vectors are discarded and batch normalized. Batch normalization uses the mean and standard deviation of the small batches to continuously adjust the intermediate output of the model so that the intermediate output of each layer is more stable. Both are used to prevent the model from overfitting. The next layer is the attention layer, where the attention mechanism calculates the attention weights α_{ij} for moment i of sequence j. The output vector C'_i is then obtained from the attention weights.

Algorithm 1 LSTM-attention-LSTM

Input: Time-series $X = (X_1, X_2, X_3, \ldots, X_t)$, Time-step L, Train-epoch n
Output: Predict time-series $Y = (Y_{t+1}, Y_{t+2}, Y_{t+3}, \ldots, Y_{t+L})$,
root mean square error RMSE, mean absolute percentage error MAPE
1 for t to n do:
2 for i to L do:
3 for j to L do:
4 encoder layer calculate encoding vector $h_i \leftarrow X_i$
5 $f_i = \delta(W_f[h_{i-1}X_i] + b_f)$
6 $i_i = \delta(W_i[h_{i-1}X_i] + b_i)$
7 $L_i = \tanh(W_c[h_{i-1}X_i] + b_c)$
8 $C_i = f_i \times C_{i-1} + i_i \times L_i$
9 $O_i = \delta(W_o[h_{i-1}X_i] + b_o)$
10 $h_i = O_i \times \tanh(C_i)$
11 dropout layer
12 BatchNormal layer
13 attention layer calculate $C'_i \leftarrow h_i$
14 $e_{ij} = \mathrm{a}\left(h_{i-1}, h_j\right)$
15 $\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} = \mathrm{softmax}(e_{ij})$
16 $C'_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$
17 decoder layer calculate decoding vector $y_i \leftarrow C'_i$
18 dropout layer
19 BatchNormal layer
20 fully connected layer $Y = \mathrm{relu}(y_i)$
21 end
22 end
23 end
24 calculate RMSE MAPE
25 return Y RMSE MAPE

With the introduction of the attention mechanism, the different degree of influence of the sequence at a given time on the prediction of the sequence at the current time will be reflected in the time series prediction. Suppose we first input a time series $X(X_1, X_2, X_3, X_4)$ with a time step of 4. At first the sequence X is computed by the input gate, forgetting gate and output gate in the coding layer to derive the output vector h_i at moment i. Then we need to first compute the value of the attentional allocation probability distribution for this sequence at each moment. The calculation of the value of the attention allocation probability distribution is shown in Figure 5.

Since this paper uses the LSTM as a decoder, when predicting the value y_i at moment i, we are able to know the output value H_i of the hidden layer node obtained by the decoder before y_i. Thus, we can use the state H_i of the hidden layer node at moment i to compare it with the hidden layer state h_j at moment j obtained by the encoder in the input sequence, i.e.,

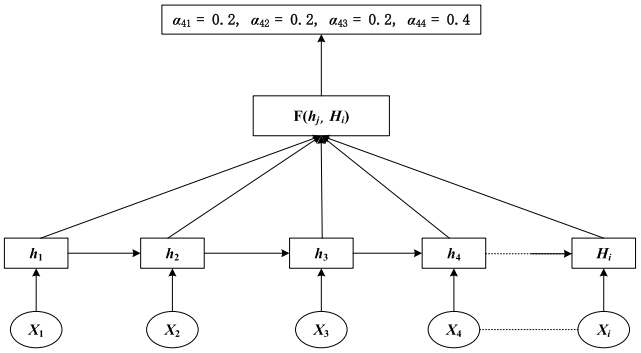

FIGURE 5. Diagram for calculating the probability of attention distribution.

by the function $\mathrm{F}(h_j, H_i)$ to calculate the attention weight α_{ij} for each moment of the sequence. The formula for calculating the attention weight α_{ij} is shown below.

$$e_{ij} = \mathrm{a}\left(h_{i-1}, h_j\right) \tag{8}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} = \mathrm{softmax}(e_{ij}) \tag{9}$$

where α_{ij} denotes the attentional weight and e_{ij} denotes the number of attentional correlations from moment j to moment i [23]. The role of the non-linear function a is to compare the hidden state h_j corresponding to the input sequence X_j at moment j in the encoder with the hidden state h_{i-1} at the previous moment of the generated sequence y_i in the decoder, so as to calculate the degree of match between each input sequence X_j and the generated sequence y_i. The higher the degree of match, the larger will be e_{ij} and α_{ij}, which means that the more attention the output at moment i is allocated to the input at moment j, then the more i will be influenced by j. Thus, the attention mechanism is still an obvious optimization for the codec model.

For example, we compute the attention weights α_{4j} for the sequence $X(X_1, X_2, X_3, X_4)$: $\{(X_1, 0.2), (X_2, 0.2), (X_3, 0.2), (X_4, 0.4)\}$, which means the attentional size assigned by the attention allocation model to each time point for the predicted time point sequence X_4. Similarly, each time point sequence in the target sequence also learns the attention allocation probability information of its corresponding original sequence, so that when generating each time point sequence y_i, the previously fixed semantic vector C is replaced with C'_i, which constantly changes according to the currently generated sequence. Thus, for the probability distribution $\{(X_1, 0.2), (X_2, 0.2), (X_3, 0.2), (X_4, 0.4)\}$, the information corresponding to the time point sequence X_4 is as follows:

$$C'_4 = \mathrm{g}(0.2 \times h_1, 0.2 \times h_2, 0.2 \times h_3, 0.4 \times h_4) \tag{10}$$

where h_1, h_2, h_3 and h_4 are the output values of the encoding layer at each time point and the g function represents the transformation function of the encoder that integrates the sequence at each time point into a time step time series, which

can be expressed as:

$$C_i' = \sum_{j=1}^{t} \alpha_{ij} h_j \tag{11}$$

where the g function is essentially a weighted summation function, C_i' is the output vector of the attention layer at moment i, h_j is the hidden state at moment j, α_{ij} denotes the attention weight from moment j to moment i, and t is the time step.

3) DECODING LAYER

The C_i' obtained by the attention mechanism will be input to the next decoding layer as an input sequence, and y_i will be calculated by the decoding layer, expressed as follows:

$$f_i = \delta\left(W_f\left[y_{i-1}, h_i, C_i'\right] + b_f\right) \tag{12}$$
$$i_i = \delta(W_i[y_{i-1}, h_i, C_i'] + b_i) \tag{13}$$
$$L_i = \tanh(W_c[y_{i-1}, h_i, C_i'] + b_c) \tag{14}$$
$$C_i = f_i \times C_{i-1} + i_i \times L_i \tag{15}$$
$$O_i = \delta(W_o[y_{i-1}, h_i, C_i'] + b_o) \tag{16}$$
$$y_i = O_i \times \tanh(C_i) \tag{17}$$

where f_i, i_i and O_i denote the forgetting gate, input gate and output gate respectively, L_i and C_i denote the update of the cell state, C_{i-1} is the cell state at moment i-1, δ is the sigmoid activation function. W_f, W_i, W_c and W_o are the weights of the inputs, while b_f, b_i, b_c and b_o denote the bias values of the inputs respectively. y_{i-1} is the predicted output value at moment i-1 in the decoder, h_i is the hidden state at moment i in the decoder, and C_i' is the output vector of the attention layer at moment i. The predicted output value y_i at moment $i$is obtained from the decoder output, and y_i is then calculated by the relu activation function of the fully connected layer to obtain the final prediction sequence $Y(Y_1, Y_2, Y_3, Y_4)$.

V. EXPERIMENTAL EVALUATION

In this summary we first present the evaluation metrics and information about the dataset, then describe how we pre-processed the dataset and detailed the relevant parameter settings for the experiments, and finally analyze and evaluate the obtained experimental results.

A. EVALUATION INDICATORS

There are several common evaluation metrics for time series forecasting, such as root mean square error (RMSE) [24], mean absolute percentage error (MAPE) [25], and coefficient of determination (R^2) [26], and so on. In this paper, we use the more commonly used root mean square error (RMSE) and mean absolute percentage error (MAPE), which are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(Y_{real,i} - Y_{pre,i})^2} \tag{18}$$
$$\text{MAPE} = \frac{100\%}{T}\sum_{i=1}^{T}\left|\frac{Y_{pre,i} - Y_{real,i}}{Y_{real,i}}\right| \tag{19}$$

where $Y_{real,i}$ is the true value of the i-th data in the sequence, $Y_{pre,i}$ is the predicted value of the i-th data in the sequence and T is the length of the sequence. Therefore when the RMSE and MAPE values are smaller, it means that the model is better at predicting.

B. DATASETS

In this experiment, we used five open-source datasets, namely the Beijing air quality dataset (Air) [27], the electricity consumption dataset of a household in Paris, France (Electricity) [28], the daily stock dataset (Stock) [29], the daily gold price dataset (Gold) [30] and the Chengdu PM2.5 concentration dataset (CDPM2.5) [31].

1) AIR

The first dataset AIR is an air quality dataset derived from weather and air pollution indices collected by the US Embassy in Beijing over a five-year period from 2010 to 2014, with a one-hour collection period, i.e. data collected every hour. The dataset includes time, PM2.5 concentration, dew point, temperature, atmospheric pressure, wind direction, wind speed, cumulative hourly snowfall and cumulative hourly rainfall, a typical multivariate time series dataset.

2) ELECTRICITY

The second dataset ELECTRICITY is also a multivariate time series dataset covering the electricity consumption of a household in Paris, France, from December 2006 to November 2010, with a 1-minute collection period. The dataset includes time, active power per minute, reactive power per minute, average voltage per minute, average current intensity per minute, active electricity in the kitchen, active electricity in the laundry and active electricity in the electric water heater and air conditioner.

3) STOCK

The third dataset STOCK is a stock dataset that counts daily stock movements from 26 April 2010 to 24 April 2020, with a 1-day collection period. The dataset includes the stock opening price, closing price, all day high, all day low, total and code.

4) GOLD

The fourth dataset GOLD is a daily gold price dataset which captures the daily gold price from 1 January 2014 to 5 August 2022, with a 1-day collection period. The dataset includes the daily closing price of gold, the opening price, the highest price for the day, the lowest price for the day, the total number of trades for the day and the daily rise and fall.

5) CDPM2.5

The fifth dataset is a record of PM2.5 data for Chengdu from 2010 to 2015, collected over a 1-day period. The dataset also includes PM2.5 concentration, dew point,

temperature, humidity, pressure, combined wind direction, cumulative wind speed, precipitation and cumulative precipitation.

For this experiment we will take a selection of these for experimental analysis and the information on the five data sets is shown in Table 1.

TABLE 1. Datasets.

Dataset name	Data size	Variables
AIR	8760	8
ELECTRICITY	6120	7
STOCK	2426	6
GOLD	2227	6
CDPM2.5	8760	9

C. DATA PRE-PROCESSING

For the processing of the datasets, all five datasets were converted into [sample, time step, feature] style 3D data for later input, we also processed the outliers and missing values in them, using mean fill for the missing values and direct deletion for the outliers.

In AIR, as the wind direction is not predictive of the data, we remove this variable and use the mean fill method as there are missing values in this dataset. Similarly, in ELECTRICITY the kitchen active energy and the electric water heater and air conditioner active energy are also non-compliant data and we remove them as well. In STOCK we remove the non-compliant coded variables. In GOLD, we also removed the variables that did not meet the requirements of our experiment. In CDPM2.5, these variables were removed as the PM2.5 and combined wind data were anomalous, while precipitation and cumulative precipitation were zero and had no effect on the prediction of the time series, so the final five data sets for the experiment were 7, 5, 5, 5 and 5 variables respectively.

In conducting the experiments, for Air and CDPM2.5 we took the first 11 months of the year as the training set and the remaining month as the test set, with the variables wind speed and humidity as the final data to be predicted for Air and CDPM2.5 respectively. For ELECTRICITY we took the first 6000 data as the training set and the remaining as the test set, with the variable average voltage per minute as the final predictor. For STOCK we take 2376 data as the training set, the remaining data as the test set, and the closing price as the predictor variable. The Gold data set was treated with the first 2160 data as the training set and the closing price as the predictor variable. After slicing the data set, we also need to perform a maximum-minimum normalization process on the data set, in order to prevent the results of the experiment from being affected by any large discrepancies in the data, so in the final prediction session we also need to back-normalize the data to the original value.

D. EXPERIMENTAL SETTINGS

In terms of the experimental setup, the five datasets were set up differently for the five types of datasets due to their different acquisition periods, and the difference is mainly reflected in the time step settings. Since the period of AIR is 1 hour, we set the time step for AIR and CDPM2.5 to 24 and the predicted length to 24; similarly for ELECTRICITY, the collection period is 1 minute, so we set the time step to 30 and the predicted length to 30; STOCK and GOLD both have a collection period of 1 day, so the time step is set to 24 and the predicted length to 24.

In terms of model parameters, we set the epoch to 50 times, the batch-size to 128, the hidden layer in the LSTM to 100 layers, and the activation function to relu function. A dropout function is added after each LSTM model, and the dropout parameter is set to 0.3 to prevent overfitting. The data are then batch normalized, with the aim of making the whole neural network more stable at the intermediate output of each layer. During the training process, we chose the mse loss function, the Adam algorithm for the optimization algorithm, the learning rate was set to 0.001, and the learning rate decayed by 1e-5 per round.

E. EXPERIMENTAL RESULTS AND ANALYSIS

The newly constructed model is compared with some of the more popular models, namely CNN, CNN-LSTM, LSTM, stacked-LSTM, BiLSTM, encoder-decoder-LSTM and the LSTM-attention-LSTM model with the addition of the attention mechanism proposed in this paper.

We first compare the performance of each model in terms of RMSE and MAPE across the five datasets, where we chose to sum the RMSE for each time to obtain the final RMSE, and the final MAPE was averaged over the results. To reflect the stability of the models, we conducted five experiments on the same model and then averaged the obtained RMSE and MAPE to obtain the final RMSE and MAPE, the results of which are shown in Table 2.

Table 2 shows that our model has the smallest RMSE and MAPE values in all five datasets and significantly outperforms several other currently popular models, especially in the two datasets of Electricity and Gold, where it has a significant effect, reducing the RMSE by 0.224 and 0.298 respectively compared to the lowest RMSE in the other models, and for MAPE It is also 0.75% and 7.04% lower than the lowest MAPE in the other models. Although the RMSE and MAPE in Stock were only 0.006 and 1.76% lower than the lowest RMSE and MAPE in the other models, overall, the improvement in the LSTM-attention-LSTM was still a significant improvement in the prediction of the time series. the poor prediction performance of the Encoder-decoder-LSTM model This is because the complexity of the model is higher compared to the other models, ignoring the fact that time series data with longer time steps tend to forget the previous data, while our proposed LSTM-attention-LSTM

TABLE 2. Multivariate time series forecasting results for five data sets.

Methods	CNN		CNN-LSTM		LSTM		Stacked-LSTM		BiLSTM		Encoder-decoder-LSTM		LSTM-attention-LSTM	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
AIR	1.282	49.85%	1.234	50.09%	1.264	52.17%	1.291	52.74%	1.530	61.83%	1.315	58.86%	**1.194**	**49.62%**
ELECTRICITY	2.792	9.54%	0.749	2.68%	0.937	3.31%	1.484	5.15%	1.675	5.79%	0.907	3.21%	**0.525**	**1.93%**
STOCK	0.305	60.30%	0.030	12.28%	0.128	30.07%	0.334	61.64%	0.139	29.56%	0.470	79.52%	**0.024**	**10.52%**
GOLD	4.878	77.47%	0.507	26.93%	1.506	54.17%	3.205	63.90%	1.238	46.19%	0.690	38.65%	**0.209**	**19.89%**
CDPM2.5	0.233	11.52%	0.219	11.11%	0.178	9.03%	0.349	17.72%	0.551	27.62%	0.170	8.39%	**0.165**	**8.16%**

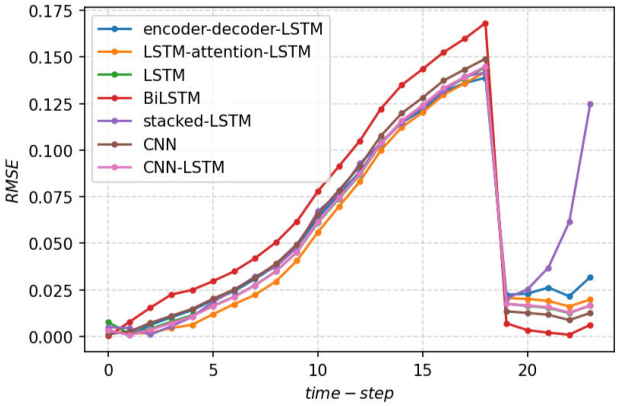

FIGURE 6. RMSE distribution (AIR).

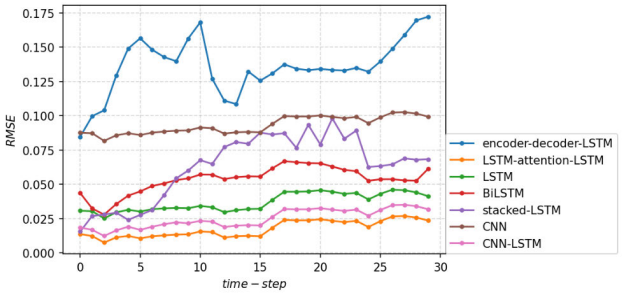

FIGURE 7. RMSE distribution (ELECTRICITY).

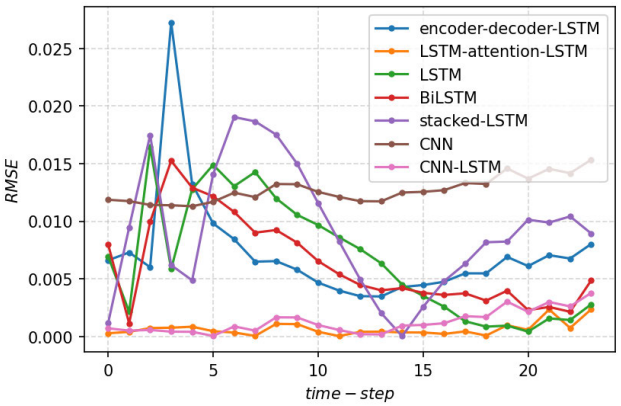

FIGURE 8. RMSE distribution (STOCK).

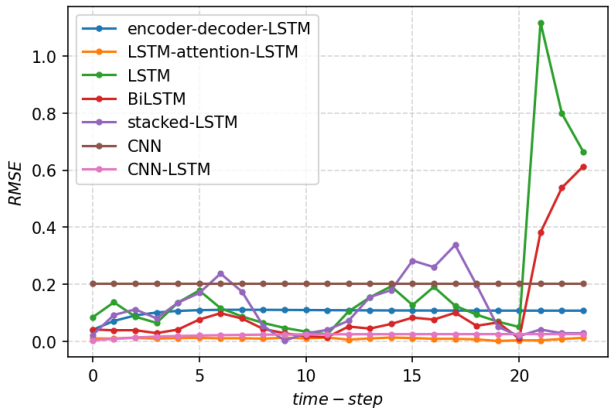

FIGURE 9. RMSE distribution (GOLD).

model solves this problem well, making the prediction effect of the model greatly improved.

Let's take a look at the specific distribution of RMSE for our model at each time point. Figures 6, 7, 8, 9 and 10 show the distribution of RMSE at each time point for the model proposed in this paper and several other mainstream models.

Comparing the RMSE of the model proposed in this paper with the RMSE distribution of several other mainstream models at each time point in Figures 6-10, it is easy to see that the RMSE of the LSTM-attention-LSTM is smaller than the other models at most time points, which indicates that the prediction accuracy of the model in this paper is higher than the other models. Also in Figure 6, our model has a large dropout in the later interstep, which is due to the fact that there are more missing values in AIR, and we treated the missing values as mean-filling at that time, so some of the data have a large dropout, and this happens in the figure. In the other three datasets without missing values, our model performs relatively better than in AIR, without a very large drop-off, and shows good robustness, which has implications for realistic practical applications such as weather prediction and traffic planning. The maximum and minimum RMSEs of LSTM-attention-LSTM in Electricity and CDPM2.5 differ only by about 0.02 and 0.005, and the maximum and minimum RMSEs in Stock and Gold are even more similar, while in AIR it is as high as about 0.13, which indicates that pre-processing of the dataset is crucial, and that the dataset pre-processing needs to be strengthened. Overall, our LSTM-attention-LSTM model performs much better than the more popular time series forecasting models currently available.

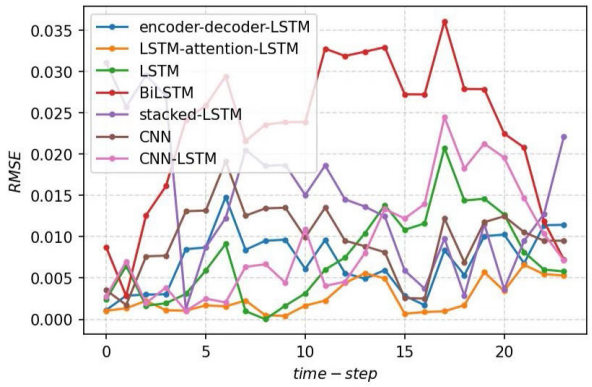

FIGURE 10. RMSE distribution (CDPM2.5).

On the other hand, in order to verify the applicability of the attention mechanism, i.e. whether it works well all the time, we tested this with another set of experiments. The model proposed in this paper was compared with the other models for a second time on the ELECTRICITY and STOCK datasets, but this time the time step for ELECTRICITY was changed to 10 compared to 30 and for Stock to 6 compared to 24. We compared the results from these two time steps. The results are shown in Tables 3 and 4.

TABLE 3. Rmse and Mape (electricity) of the model at different time steps.

Time-step	30-step		10-step	
Methods	RMSE	MAPE	RMSE	MAPE
LSTM	0.937	3.31%	0.158	1.88%
Stacked-LSTM	1.484	5.15%	0.184	2.09%
BiLSTM	1.675	5.79%	0.353	3.88%
Encoder-decoder-LSTM	0.907	3.21%	0.059	1.10%
LSTM-attention-LSTM	**0.525**	**1.93%**	**0.052**	**0.89%**
CNN	2.792	9.54%	0.203	2.34%
CNN-LSTM	0.749	2.68%	0.131	1.60%

TABLE 4. Rmse and Mape (stock) of the model at different time steps.

Time-step	24-step		6-step	
Methods	RMSE	MAPE	RMSE	MAPE
LSTM	0.128	30.07%	0.023	27.62%
Stacked-LSTM	0.334	61.64%	0.006	3.83%
BiLSTM	0.139	29.56%	0.015	17.97%
Encoder-decoder-LSTM	0.470	79.52%	0.011	13.86%
LSTM-attention-LSTM	**0.024**	**10.52%**	**0.005**	**3.48%**
CNN	0.305	60.30%	0.046	15.57%
CNN-LSTM	0.030	12.28%	0.036	14.02%

Tables 3 and 4 show that the model proposed in this paper still has advantages in the RMSE and MAPE metrics compared to Stacked-LSTM, BiLSTM and other models when the time step is reduced. This indicates that the model proposed in this paper is not only suitable for forecasting time series with larger step lengths, but also performs well in forecasting time series with shorter step lengths. From the above experiments, the LSTM-attention-LSTM model proposed in this paper has more obvious advantages in predicting time series with longer time steps.

VI. CONCLUSION

In this paper, we first investigate LSTM, coding-decoding model and attention mechanism, and then propose a new time series prediction model LSTM-attention-LSTM, and apply our new model to five open source time series datasets for prediction. The experimental results show that the proposed time series prediction model is better than the mainstream time series prediction model represented by LSTM. The experimental results show that the LSTM-attention-LSTM model proposed in this paper is not only suitable for time series prediction with short time steps, but also has more obvious advantages in predicting time series with larger time steps.

REFERENCES

[1] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial–temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, LA, USA, 2018, pp. 1–8.
[2] A. I. McLeod and W. K. Li, "Diagnostic checking ARMA time series models using squared-residual autocorrelations," *J. Time Ser. Anal.*, vol. 4, no. 4, pp. 269–273, Jul. 1983.
[3] J. L. Torres, A. García, M. D. Blas, and A. De Francisco, "Forecast of hourly average wind speed with ARMA models in Navarre (Spain)," *Sol. Energy*, vol. 79, no. 1, pp. 65–77, Jul. 2005.
[4] C. ByoungSeon, *ARMA Model Identification*. New York, NY, USA: Springer, 2012, pp. 23–25.
[5] G. Box, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Wiley, 2015, pp. 89–91.
[6] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Phys. D, Nonlinear Phenomena*, vol. 404, no. 1, pp. 51–53, 2020.
[7] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.
[8] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," 2014, *arXiv:1409.1259*.
[9] M. Ullah, H. Ullah, S. D. Khan, and F. A. Cheikh, "Stacked LSTM network for human activity recognition using smartphone data," in *Proc. 8th Eur. Workshop Vis. Inf. Process. (EUVIP)*, Rome, Italy, Oct. 2019, pp. 175–180.
[10] R. Chandra, S. Goyal, and R. Gupta, "Evaluation of deep learning models for multi-step ahead time series prediction," *IEEE Access*, vol. 9, pp. 83105–83123, 2021.
[11] R. Ghanbari and K. Borna, "Multivariate time-series prediction using LSTM neural networks," in *Proc. 26th Int. Comput. Conf., Comput. Soc. Iran (CSICC)*, Tehran, Iran, Mar. 2021, pp. 1–5.
[12] V. Ashish, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, no. 1, pp. 105–109.
[13] L. Feng and Q. Jun, "Time series forecasting method based on cluster analysis and neural network," *Microelectron. Comput.*, vol. 23, no. 9, pp. 85–87, 2006.
[14] T. P. Vogels, K. Rajan, and L. F. Abbott, "Neural network dynamics," *Annu. Rev. Neurosci.*, vol. 28, pp. 357–376, Jul. 2005.
[15] P. Li, Z. Tan, L. Yan, and K. Deng, "Time series prediction of mining subsidence based on a SVM," *Mining Sci. Technol.*, vol. 21, no. 4, pp. 557–562, Jul. 2011.
[16] K. Cai, L. Tan, C. Li, and X. Tao, "Short-term wind speed prediction combining time series and neural network method," *Power Syst. Technol.*, vol. 32, no. 8, pp. 82–85, 2008.

[17] F. Gers, "Applying LSTM to time series predictable through time-window approaches," in *Neural Nets WIRN Vietri-01*. London, U.K.: Springer, 2002, pp. 193–200.
[18] J. Liu and S. Chen, "Non-stationary multivariate time series prediction with MIX gated unit," *J. Comput. Res. Develop.*, vol. 56, no. 8, p. 1642, 2019.
[19] S. Dai, Q. Chen, Z. Liu, and H. Dai, "Time series prediction based on EMD-LSTM model," *J. Shenzhen Univ. Sci. Eng.*, vol. 37, no. 3, pp. 221–230, 2020.
[20] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Netw.*, vol. 116, pp. 237–245, Aug. 2019.
[21] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[22] H. B. Ouyang, K. Huang, and H. J. Yan, "Financial time series forecasting based on LSTM neural network," *CMS*, vol. 28, no. 4, pp. 27–35, 2020.
[23] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liù, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[24] T. Ma, T. Ji, G. Yang, Y. Chen, W. Xu, and H. Liu, "Incidence trend prediction of hand-foot-mouth disease based on long short-term memory neural network," *J. Comput. Appl.*, vol. 41, no. 1, p. 265, 2021.
[25] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, Jun. 2016.
[26] L. Jin and S. Myers, "R2 around the world: New theory and new tests," *J. Financial Econ.*, vol. 79, no. 2, pp. 257–292, Feb. 2006.
[27] (Apr. 5, 2022). *Beijing $PM_{2.5}$ Dataset*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data
[28] (Apr. 5, 2022). *Individual Household Electric Power Consumption Dataset*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption
[29] (Apr. 5, 2022). *Daily Stock Dataset*. [Online]. Available: https://www.kaggle.com/datasets/dsadads/databases
[30] (Apr. 5, 2022). *Daily Gold Price Dataset*. [Online]. Available: https://www.kaggle.com/datasets/nisargchodavadiya/daily-gold-price-20152021-time-series
[31] (Apr. 8, 2023). *Chengdu $PM_{2.5}$ Dataset*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/PM2.5+Data+of+Five+Chinese+Cities

XIANYUN WEN was born in Ganzhou, China, in 1998. He received the bachelor's degree from Changzhou University. He is currently pursuing the master's degree with Southwest Minzu University. His main research interests include time series forecasting and big data.

WEIBANG LI (Member, IEEE) received the B.S. degree in computer science and technology from the Nanjing University of Aeronautics and Astronautics, in 2003, and the Ph.D. degree in computer science from Northwestern Polytechnical University, in 2017. He is currently a Lecturer with the School of Computer Science and Engineering, Southwest Minzu University. His current research interests include data quality management, big data analysis, and machine learning.