Big Data Research 27 (2022) 100296

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

NGCU: A New RNN Model for Time-Series Data Prediction

Jingyang Wang [b], Xiaolei Li [b], Jiazheng Li [b], Qiuhong Sun [b], Haiyao Wang [a,*]

[a] School of Ocean Mechatronics, Xiamen Ocean Vocational College, Xiamen, 361100, China
[b] School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050018, China

ARTICLE INFO

Article history:
Received 3 June 2021
Received in revised form 16 September 2021
Accepted 3 December 2021
Available online 9 December 2021

Keywords:
NGCU
RNN
Time-series
Prediction

ABSTRACT

With the rapid development of machine learning, a possibility is provided for high-precision prediction of time-series. This paper proposes a new unit which is called New Gate Control Unit (NGCU) based on Recurrent Neural Networks (RNN). The proposal of NGCU is mainly used for prediction of time-series data. NGCU alleviates the problems of gradient disappearance and explosion of traditional RNN. Compared with Long Short-Term Memory (LSTM) and Gate Recurrent Unit (GRU), NGCU improves not only the computational complexity of gating unit but also the sensitivity of model learning. To verify the accuracy, efficiency and feasibility of NGCU, this paper uses RNN, LSTM and GRU to conduct comparative experiments, and uses three different data of air quality, Hang Seng Index, and gold future price to prove the generalization of NGCU. Mean Absolute Error (MAE), Mean Squared Error (MSE), Explained Variance Score (EVS), R^2 and training time are used to evaluate experimental results. Among the three different data prediction results, the R^2 of NGCU is 0.9736, 0.9872, and 0.9231, respectively. And NGCU's MAE, MSE, EVS are also the best. Compared with LSTM and GRU, NGCU has the least training time, which is 323.5261s, 53.3257s, and 43.4814s respectively.

1. Introduction

Recurrent Neural Network (RNN) is derived from the Hopfield network proposed by Saratha Sathasivam in 1982 [1]. The traditional machine learning algorithm relies heavily on manual feature extraction, which makes the feature extraction bottleneck in image recognition, speech recognition, and natural language processing [2,3]. And the fully connected neural network also has problems such as too many parameters and inability to use the time-series information in the data. With the more effective structure of RNN being put forward, the deep expression ability of sequential information and semantic information in data mining is fully utilized. RNN has achieved breakthroughs in speech recognition [4–6], machine translation, and timing analysis [7–9].

The main purpose of time-series analysis is to predict the future value according to the existing historical data. RNN is one of the most important neural networks in time-series analysis. Compared with ordinary neural networks, RNN is mutually independent. The calculation results of each hidden layer of RNN are related to the current input and the previously hidden layer results. RNN has the characteristics of memory previous results. However, due to the long-term dependence of the data on the RNN, the problems of gradient disappearance and explosion appear in the model training [10].

LSTM can well alleviate the problems of gradient disappearance and explosion in the training process of RNN [11,12]. LSTM uses gate control technology to determine the deletion and preservation of historical data and input data through forget gate, input gate, and output gate. An important variant of LSTM is the GRU constructed by Cho [13], which combines the forget gate and input gate into an update gate. Compared with LSTM, the number of GRU parameters is reduced, and the structure of the model is simpler.

LSTM has many parameters, the calculation of LSTM is too complicated. Therefore, the training time of LSTM is long [14–17]. Compared with LSTM, GRU reduces the training time due to the decrease in the number of parameters [18,19]. The NGCU proposed in this paper aims to ensure that the model can alleviate the gradient disappearance and explosion problems of RNN to improve prediction accuracy and reduce model training time. NGCU has made some innovative advances in learning sensitivity, time-series prediction accuracy, and training time. In terms of model learning sensitivity, compared with RNN, LSTM and GRU, NGCU introduces the anti-saturation conversion module for the first time, which greatly improved the model's sensitivity to data learning. In terms of the accuracy of time-series prediction, compared with RNN, LSTM and GRU, the forget gate of NGCU adds learning from historical data, so it can more accurately grasp the overall trend of

[*] Corresponding author.
E-mail address: wanghaiyao@xmoc.edu.cn (H. Wang).

https://doi.org/10.1016/j.bdr.2021.100296

data, and the model's prediction accuracy of data is correspondingly improved. In terms of training time, compared with LSTM and GRU, NGCU has a more concise structure and fewer parameters, so training time is reduced. To verify the accuracy, efficiency and feasibility of NGCU, in this paper, RNN, LSTM and GRU are used to conduct comparative experiments, and three different data of air quality, Hang Seng Index, and gold future price are used to prove the generalization of NGCU. The prediction results are evaluated by MAE, MSE, EVS, R^2, and model training time. Experimental results show that the training time of NGCU is lower than that of LSTM and GRU, and the MAE, MSE, EVS, R^2 of NGCU are better than the other three models. In summary, the main contributions of this article are as follows:

(1) Through analysis of RNN, LSTM, GRU, and time-series data, this paper proposes a new unit which is called New Gate Control Unit (NGCU) based on RNN. The proposal of NGCU is mainly applied to the prediction of time-series data.
(2) By introducing gate technology and adding anti-oversaturation conversion modules, NGCU is more sensitive to data feature learning and can effectively alleviate the gradient disappearance and explosion problems of RNN caused by long-term data dependence. And NGCU is verified that the computational complexity of NGCU is reduced because the parameter number of NGCU is less than LSTM and GRU.
(3) Through comparative experiments, the training time of the model is greatly reduced. Experiments show that the evaluation results of time-series prediction model based on NGCU are better than those of the other three models.

2. Related work

The models based on traditional regression model prediction mainly include: mean regression, Autoregressive Integrated Moving Average model (ARIMA), and Support Vector Machine (SVM). With the rapid development of the neural network, the machine learning method based on the neural network provides a possibility for time-series prediction.

To study the dynamic law of surface movement caused by coal mining activities, Li et al. proposed a time-series analysis based on SVM and established a dynamic prediction model of surface movement [20]. Wu et al. proposed corn futures price prediction based on ARIMA time-series and SVM [21]. Due to the non-linear characteristics of some time-series data, it is difficult for the traditional regression models to achieve higher prediction accuracy.

Compared with traditional regression models, time-series prediction based on neural network models has stronger nonlinear fitting capabilities. Chimmula et al. used the LSTM to predict the time-series of COVID-19 transmission in Canada [22]. Livieris et al. proposed a CNN-LSTM model for gold price time-series prediction [23]. Preeti et al. used LSTM to predict non-stationary financial finance [24]. All the above scholars apply LSTM to the related feature prediction of time-series, and the above scholars' research also shows that the time-series prediction model based on LSTM has a higher accuracy than the traditional regression model. However, due to the too many LSTM parameters, the model training time is long. Zhou et al. proposed an air pollutant concentration prediction model based on the GRU. The experimental results showed that GRU had a higher prediction accuracy, and this model was effective for the time-series prediction of air pollutants [25]. According to the researches, the prediction model based on gate technology can improve the prediction accuracy and reduce the training time.

Hu et al. proposed a time-series prediction model based on a variant LSTM, but this variant did not simplify the complexity of the model structure and had a large number of parameters. So, the data training time of the model was not improved [26]. Jun et al. proposed a multi-stage attention network for a multivariate time-series prediction model which introduced a data conversion module based on LSTM to increase the sensitivity of model learning and improve the accuracy of prediction [27]. But the problem of the long training time of this model was not effectively solved. Zu et al. proposed a short-term wind power prediction method based on wavelet packet decomposition and improved GRU, which modified the activation function of GRU, such as introducing SELU as the activation function of output. But the training time of this model was not reduced [28].

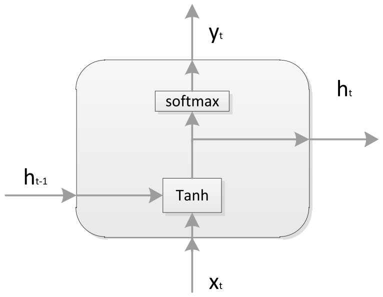

Fig. 1. RNN structure.

3. NGCU

3.1. Model comparison and introduction

3.1.1. RNN

The main purpose of the RNN is to process and predict sequence data. The model structure is shown in Fig. 1, where x_t is the input of the training sample at time t in time-series. h_t is the hidden state of the model at time t in the time-series. y_t is the output of the model at time t in the time-series.

When the inputs are $x_1, x_2, \ldots\ldots, x_t$, the corresponding hidden states are $h_1, h_2, \ldots\ldots, h_t$, and the outputs are $y_1, y_2, \ldots\ldots, y_t$. For example, the calculation process of the classic RNN can be expressed as formula (1)(2):

$$h_t = f(Ux_t + Wh_{t-1} + b) \tag{1}$$

$$y_t = softmax(Vh_t + c) \tag{2}$$

U, W and V are the weight parameter, b and c are bias, and $f()$ is the activation function, which is generally the *tanh* function.

Although RNN can theoretically solve the training of sequence data well, when the data sequence is very long, it is prone to gradient disappearance and explosion problems. Therefore, the RNN generally cannot be directly used in the application fields (such as speech recognition, handwritten books, machine translation and NLP).

3.1.2. LSTM

The structure of LSTM is shown in Fig. 2. LSTM is composed of forget gate, input gate and output gate. The three gates jointly control the storage and deletion of data information [29,30].

$$f_t = \sigma(W_{fh} \cdot h_{t-1} + W_{fx} \cdot x_t + b_f) \tag{3}$$

As shown in formula (3). The forget gate f_t controls how much data information from the previous moment can be retained to the current moment. Where x_t is the input vector of the t-th time step, that is, the t-th component of the input sequence x, and x_t is linearly transformed by multiplying with the weight matrix W_{fx}, h_{t-1} saves the information of the previous time step $t-1$, and h_{t-1} performs linear transformation by multiplying with the

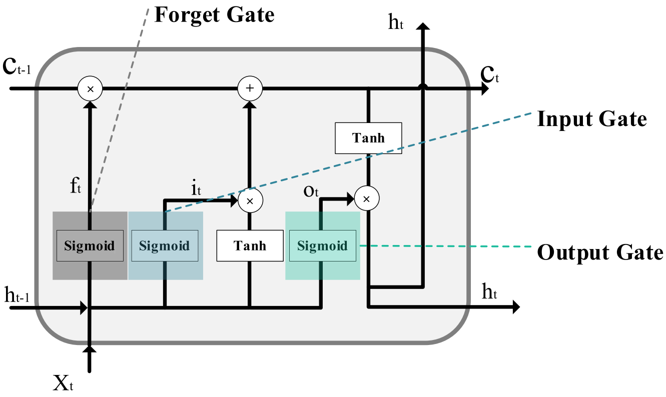

Fig. 2. LSTM structure.

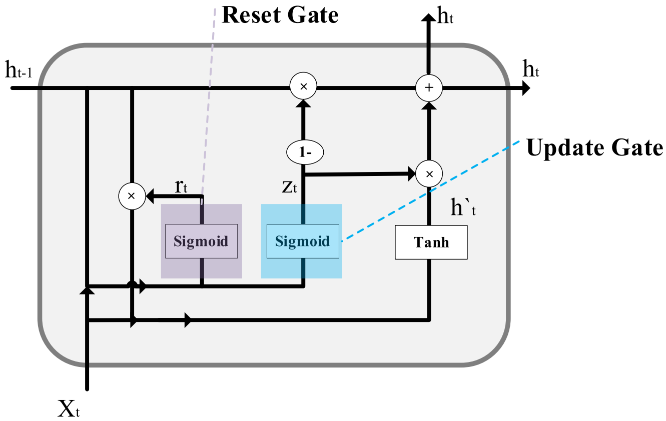

Fig. 3. GRU structure.

weight matrix ${}_{W_{fh}}$. b_f represents the bias. These three parts of information are added together, and then calculated by the Sigmoid activation function.

$$i_t = \sigma(W_{ih} \cdot h_{t-1} + W_{ix} \cdot x_t + b_i) \tag{4}$$

As shown in formula (4). The input gate i_t controls how much data information input at the current moment can be retained to the current moment.

$$o_t = \sigma(W_{oh} \cdot h_{t-1} + W_{ox} \cdot x_t + b_o) \tag{5}$$

$$c_t = f_t \times c_{t-1} + i_t \times tanh(W_o \times [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t \times tanh(c_t) \tag{7}$$

As shown in formula (5). The output gate ${}_{ot}$ controls how much data information at the current moment can be passed to the next moment. As shown in formula (6), c_t is the data information retained from the beginning to the current moment. As shown in formula (7), h_t controls how much data information retained from the beginning to the current moment can be transferred to the next moment.

3.1.3. GRU

GRU has only two gates. GRU combines the input gate and forget gate in LSTM into one, which is called the update gate. As shown in Fig. 3, the update gate ${}_{zt}$ controls the amount of data that the front memory information can continue to retain to the current moment, as shown in formula (8). The reset gate ${}_{rt}$ controls how much past information should be forgotten, as shown in formula (9).

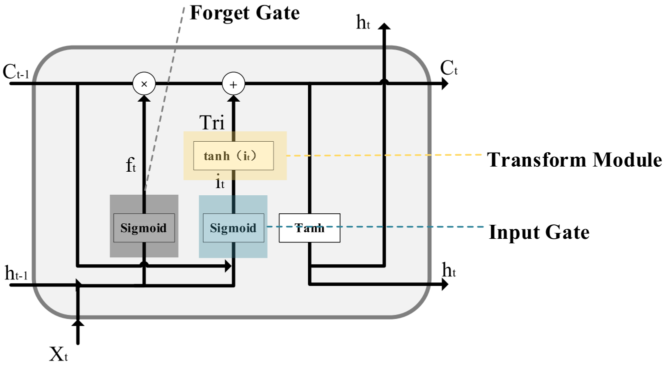

Fig. 4. NGCU structure.

$$z_t = \sigma(W_{zh} \cdot h_{t-1} + W_{zx} \cdot x_t + b_z) \tag{8}$$

$$r_t = \sigma(W_{rh} \cdot h_{t-1} + W_{rx} \cdot x_t + b_r) \tag{9}$$

$$h'_t = tanh(W_{xh}x_t + r_t W_{hh}h_{t-1}) \tag{10}$$

As shown in formula (10). The input information x_t and the previous time step information h_{t-1} are linearly transformed, and different matrices W are right-multiplied respectively. Then the reset gate r_t and $W_{hh}h_{t-1}$ are multiplied together. Finally, the new information h'_t of the current time is calculated through the *tanh* activation function.

$$h_t = z_t h_{t-1} + (1 - z_t)h'_t \tag{11}$$

As shown in formula (11). The product of z_t and h_{t-1} represents the final data information retained at the previous time step. This product plus the information retained from the current memory to the final memory is equal to the content h_t output by the final gated loop unit.

3.1.4. NGCU

To improve the sensitivity of NGCU learning, the anti-oversaturation conversion module *Tri* is introduced. And NGCU introduces the entire data stream information c_{t-1} up to the previous moment in the calculation process of the input gate.

$$f_t = \sigma(W_{fh} \cdot h_{t-1} + W_{fx} \cdot x_t + b_f) \tag{12}$$

As shown in formula (12), the forget gate f_t mainly determines how much data from the previous moment can be retained to the current moment, where x_t is the input vector of the t-th time step, and x_t multiply with the weight matrix W to perform linear transformation. h_{t-1} stores the data of the previous time step $t-1$, and h_{t-1} performs linear transformation by multiplying with the weight matrix W. The two parts of data information are added, and then calculated by the Sigmoid activation function, so the numerical result f_t obtained is between 0 and 1.

$$i_t = \sigma(W_{ih} \cdot h_{t-1} + W_{ix} \cdot x_t + c_{t-1} + b_i) \tag{13}$$

As the formula (13) shows, input gate i_t mainly determines how much data can be retained to the current time. Compared with the forget gate, the input gate i_t of NGCU introduces the whole data flow information c_{t-1} until the previous time, so the input gate has a memory effect on the retention of the current time data. The input gate in NGCU uses the Sigmoid activation function. The sigmoid activation function is shown in Fig. 5. When x is less than -3 or greater than 3, the value of the sigmoid activation function would fall into the supersaturated zone. Therefore, when the input data enters the supersaturated zone of the input gate, the value

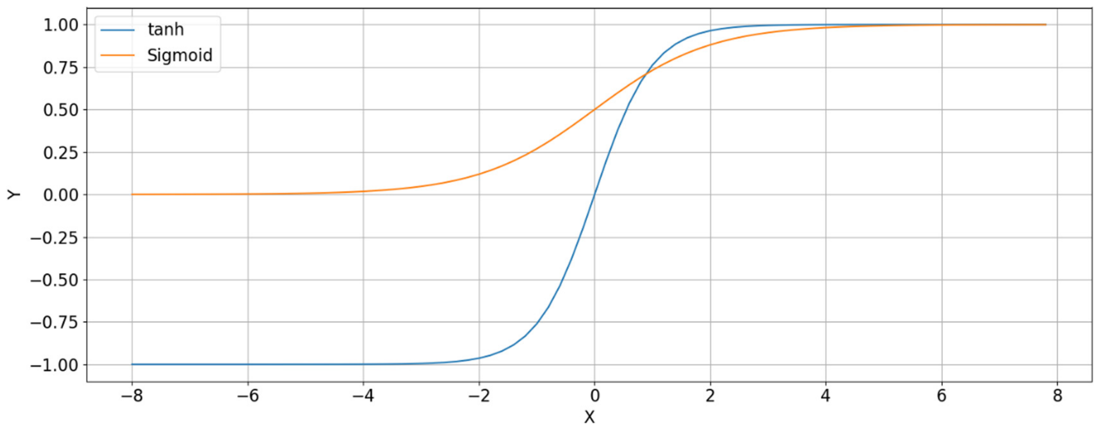

Fig. 5. tanh and Sigmoid function.

Table 1
LSTM, GRU, NGCU model parameters.

LSTM	GRU	NGCU
Three gates 8 weight parameters 4 bias parameters	Two gates 6 weight parameters 3 bias parameters	Two gates 4 weight parameters 2 bias parameters

does not change significantly, resulting in a decrease in the sensitivity of learning.

$$Tri = tanh\,(i_t) \tag{14}$$

The calculation formula of anti-supersaturation conversion module *Tri* is shown in Formula (14) and the *tanh* function is shown in Fig. 5. The sigmoid activation function is in the range of (0, 1). When the data passes the input gate, the value generated by the conversion module *Tri*, so the value is more significant.

$$c_t = f_t \times c_{t-1} + Tri \tag{15}$$

As shown in formula (15), c_t is the data information retained from the beginning to the current moment. c_t controls how much data information at the current moment can be retained to the next moment by *tanh* function.

$$h_t = tanh(c_t) \tag{16}$$

As shown in formula (16), h_t is the data information retained at the current moment.

The structure of NGCU is shown in Fig. 4. NGCU is composed of forget gate and input gate. Compared with LSTM, NGCU removes the output gate. Compared with GRU, the model structure is simpler. The parameters of LSTM, GRU and NGCU are shown in Table 1. Compared with LSTM, the number of W parameters is reduced from 8 to 4, and the number of b parameters is reduced from 4 to 2. Compared with GRU, the number of W parameters is reduced from 6 to 4, and the number of b parameters is reduced from 3 to 2.

3.2. Model formula derivation

The simple data flow of the whole network is shown in Fig. 6, a given input sequence $< x_1, x_2, x_3, x_4, \ldots, x_k >$ is the coding of x_t. There are two parametric matrices in the entire network $[W_{fh}, W_{fx}]$ and $[W_{ih}, W_{ix}]$, the computational formula of $\mathrm{h_t}$ is shown in (17): c_0 and h_1 are hidden layer initialization states, which are generally zero vectors.

$$h_t = tanh\,(c_t) \tag{17}$$

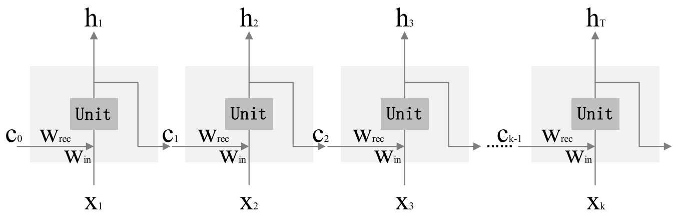

Fig. 6. Data flow.

Taking $W = [W_{fh}, W_{fx}, W_{ih}, W_{ix}]$, L_t is the loss corresponding to the code h_t, which stands for the function of W. L is the total loss. The W derivation of L is shown in Formula (18):

$$\frac{\partial L}{\partial W} = \sum_{t=1}^{T} \frac{\partial L_t}{\partial W} \tag{18}$$

The RNN uses formula (19) to update the W parameters.

$$W = W - \frac{\partial L_t}{\partial W} \tag{19}$$

Where $\frac{\partial L_t}{\partial W}$ can be written into formula (20):

$$\frac{\partial L_k}{\partial W} = \frac{\partial L_k}{\partial h_k}\frac{\partial h_k}{\partial c_k}\cdots\frac{\partial c_2}{\partial c_1}\frac{\partial c_1}{\partial W} \tag{20}$$

The $\frac{\partial L_k}{\partial W}$ can be simplified as given by formula (21):

$$\frac{\partial L_k}{\partial W} = \frac{\partial L_k}{\partial h_k}\frac{\partial h_k}{\partial c_k}\cdots\frac{\partial c_2}{\partial c_1}\frac{\partial c_1}{\partial W} = \frac{\partial L_k}{\partial h_k}\left(\prod_{t=2}^{k}\frac{\partial c_t}{\partial c_{t-1}}\right)\frac{\partial c_1}{\partial W} \tag{21}$$

Where c_t can be acquired by the formula (22):

$$c_t = f_t \times c_{t-1} + Tri \tag{22}$$

Taking the derivative of c_t, we can get the formula (23):

$$\begin{aligned}\frac{\partial c_t}{\partial c_{t-1}} &= f_t + (tanh\,(\sigma\,(W_{ih}\cdot h_{t-1} + W_{ix}\cdot x_t + c_{t-1} + b_i))^2 - 1)\sigma' \\ &\quad \times (W_{ih}\cdot h_{t-1} + W_{ix}\cdot x_t + c_{t-1} + b_i)\end{aligned} \tag{23}$$

The total loss can be expressed as formula (24):

$$\begin{aligned}\frac{\partial L_k}{\partial W} = \frac{\partial L_k}{\partial h_k}\Bigg(\prod_{t=2}^{k}&(f_t + (1 - tanh\,(\sigma\,(W_{ih}\cdot h_{t-1} \\ &+ W_{ix}\cdot x_t + c_{t-1} + b_i))^2)\sigma'(W_{ih}\cdot h_{t-1} \\ &+ W_{ix}\cdot x_t + c_{t-1} + b_i))\Bigg)\frac{\partial c_1}{\partial W}\end{aligned} \tag{24}$$

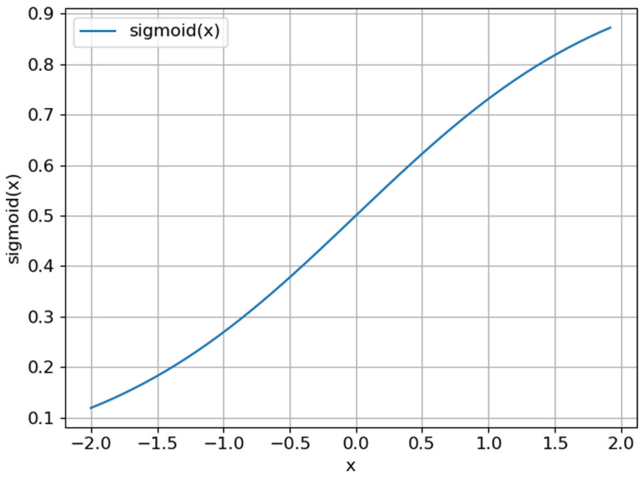

Fig. 7. $\sigma(x)$ function.

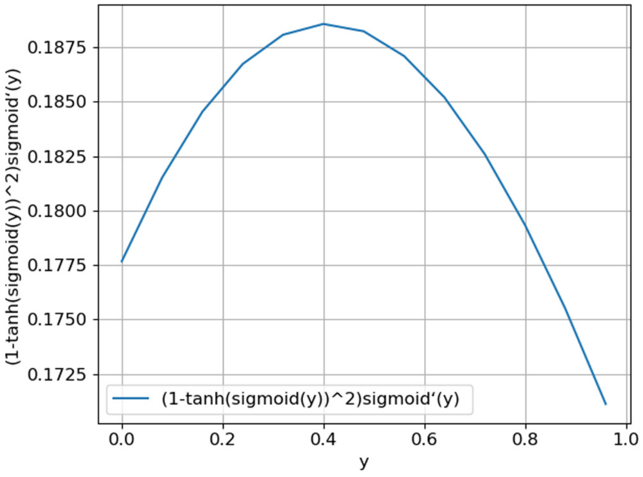

Fig. 8. $(1 - tanh\,(\sigma(y))^2)\sigma'(y)$ function.

Taking x be equal to formula (25):

$$x = W_{fh} \cdot h_{t-1} + W_{fx} \cdot x_t + b_f \tag{25}$$

Taking y be equal to formula (26):

$$y = W_{ih} \cdot h_{t-1} + W_{ix} \cdot x_t + c_{t-1} + b_i \tag{26}$$

Taking $z(x, y)$ be equal to formula (27):

$$z(x, y) = \sigma\,(x) + (1 - tanh\,(\sigma(y))^2)\sigma'(y) \tag{27}$$

The function $\sigma(x)$ is shown in Fig. 7. The function $(1 - tanh\,(\sigma(y))^2)\sigma'(y)$ is shown in Fig. 8, value of $\sigma(x)$ is in the range (0,1), value of $\sigma(y)$ is in the range (0,1), so function value of $(1 - tanh\,(\sigma(y))^2)\sigma'(y)$ is in the range (0.1720,0.1880). The function gradient z is shown in Fig. 9. It can be seen from the Fig. 9 that the value range of function gradient $z(x, y)$ is more reasonable. Therefore, NGCU can greatly alleviate the gradient disappearance and explosion problems.

4. Experiments

4.1. Experimental environment

This experiment is conducted on the Windows 10 operating system, and the hardware equipment is Intel(R) Core (TM) i510300H CPU @2.50 GHz, RAM 8.00 GB and NVIDIA GeForce GTX1650Ti. The programming language used in the entire experiment is Python 3.7.0, and the compiler is PyCharm 2020 1.3 ×64. To make the experimental framework more concise and clearer, and improve the efficiency of the experiment, Keras 2.1.0 and TensorFlow 1.14.0 are used in this experiment as the deep learning framework.

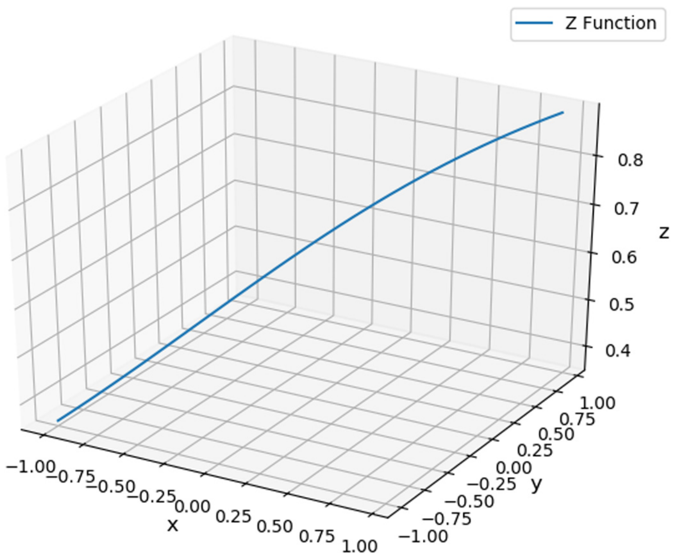

Fig. 9. $z(x, y)$ function.

4.2. Data sources

The air quality data is obtained from http://data.epmap.org/ and the API interface (https://www.nowapi.com/api/weather.history). Air quality data is stored in CSV files in chronological order.

Hang Seng Index data set is obtained from the Wind database, and Hang Seng Index data is stored in CSV files in chronological order.

The gold price data set is obtained from the Wind database, and gold price data is stored in CSV files in chronological order.

4.3. Data introduction

In order to ensure the generalization of NGCU predictions, three different data sets are selected in this experiment. The three data sets are different in terms of the total amount of data, the number of input items, the size of data eigenvalue and the stability of data. Compared with the other two data sets, the air quality data set has a larger amount data and higher data stability. Compared with the air quality data set, the Hang Seng Index data set has a smaller amount of data and input items of data. Compared with the other two data sets, the gold future price data set has the smallest amount of data and the worst stability of data.

As shown in Table 2, there are 32856 pieces of air quality data sets with 11 input items, which are AQI, CO, NO_2, O_3, PM10, PM2.5, SO_2, temperature, humidity, weather, winp. The predicted value of the AQI is output. Compared with the other two data sets, there are more data and more input items. In addition, the air quality data set is in the range of -12-1411, with an average value of 44.9534 and a standard deviation of 60.2430. Compared with the other two data sets, data stability is high.

There are 6980 pieces of Hang Seng Index data sets with 7 input items, which are the previous closing price, opening price, highest price, lowest price, closing price, rise and fall, and amount of rise and fall. The predicted value of the futures closing price is output. Compared with air quality data, the total amount of data is reduced, and the number of input items is reduced to 7. Hang Seng

Table 2
Data characteristics.

Data characteristics	Air quality data	Hang Seng Index data	Gold future data
Total data	32856	6980	3081
input items	11	7	10
Predictive value	AQI	Hang Seng Index	Closing price
Numerical range	-12-1411	-2061.2300-33484.0800	6.0412-1244366
average	44.9534	12057.7051	32753.1905
standard deviation	60.2430	9649.1170	82683.2800

Table 3
Weather data conversion.

Weather data	Quantification
Haze	1
Fog	2
Sunny	3
Cloudy	4
Overcast	5
Light rain, Rain, Light snow, Sleet, Snow	6
Moderate rain, Moderate snow	7
Shower, Thunder shower, Hail, Heavy rain, Snow shower, Heavy snow, Blizzard	8

Index data set has a value range of -2061.23-33484.08, with an average value of 12057.7051 and a standard deviation of 9649.117. Compared with air quality data, the stability of the data is poor.

There are 3081 pieces of gold future price data sets with 10 input items, which are futures settlement price, futures closing price, futures trading volume, futures holdings, AU99.99, USD_CNY, U.S. dollar index, Dow Jones Industrial Average, Nasdaq Index. The predicted value of the futures closing price is output. Compared with the other two data sets, the total amount of data is the lowest. And the value range of the gold price data set is 6.0412-1244366, the average is 32753.1905, and the standard deviation is 82683.28. Compared with the other two data sets, the stability of the data is the worst.

4.4. Data pre-processing

After the data is obtained, there are often problems such as missing data, repeated data, and non-numeric data. To improve the availability and stability of data, it is necessary to further process the original data. In this experiment, the missing data is filled with the average values of the previous and subsequent data [31], as shown in (26).

$$V_t = \frac{V_{t-1} + V_{t+1}}{2} \tag{28}$$

Where V_t is the missing value at time t, V_{t-1} is the previous value, and V_{t+1} is subsequent value.

For repeat data, the first item is kept and the duplicates are deleted in this data.

For the non-numeric data, such as the '°C' part of the input item of temperature should be deleted, and the '%' part of humidity should be deleted too. We perform quantitative conversion on non-quantized feature values, such as data conversion on weather, as shown in Table 3.

When the original index values are directly used for calculation and analysis, the role of the index with higher value is highlighted in the comprehensive analysis, and the role of the index with lower value is relatively weakened. Therefore, it is necessary to perform data normalization processing on the extracted feature value to ensure that each feature value is treated equally by the prediction model, to reduce the influence of outliers on the model, and to ensure that the model converges faster. In this experiment, z-score normalization is selected to process the original data. The formula is shown in formula (29):

Table 4
The parameters based on RNN, LSTM, GRU and NGCU.

Data set	Parameters	Value
Air quality data	The units of the RNN, LSTM, GRU and NGCU layer	16
	The activation of the RNN, LSTM, GRU and NGCU layer	tanh
	learning rate	0.001
	loss	mae
	epochs	100
	batch size	128
	time step	24
Hang Seng Index data	The units of the RNN, LSTM, GRU and NGCU layer	64
	The activation of the RNN, LSTM, GRU and NGCU layer	tanh
	learning rate	0.001
	loss	mae
	epochs	200
	batch size	64
	time step	3
Gold future price	The units of the RNN, LSTM, GRU and NGCU layer	10
	The activation of the RNN, LSTM, GRU and NGCU layer	tanh
	learning rate	0.001
	loss	mae
	epochs	100
	batch size	64
	time step	5

$$\mathrm{x}^* = \frac{\mathrm{x} - \overline{\mathrm{x}}}{\sigma} \tag{29}$$

Where $\overline{x}$ is the mean value of the original data, σ is the standard deviation of the original data, and x^* is the value after the standardization is completed.

4.5. Model parameters setting

The difference in model parameters affects the overall prediction results of the model. In this experiment, RNN, LSTM and GRU are used as the basic comparison model, and the model parameters are set consistently in the same data set to ensure the accuracy of model comparison. The parameters in RNN, LSTM, GRU, and NGCU are shown in Table 4.

4.6. Experimental results

In this experiment, MAE, MSE, EVS, and R^2 are used to make evaluation indexes of the prediction results. The evaluation indexes of Air quality prediction are shown in Table 5. The comparison between the AQI predicted values of the four models and real values is shown in Fig. 10. Compared with RNN, LSTM and GRU, the MAE of NGCU reduces by 0.1865, 0.3509 and 0.1330. Compared with RNN, LSTM and GRU, the MSE of NGCU reduces by 10.0704, 11.1478 and 7.4879. Compared with RNN, LSTM and GRU,

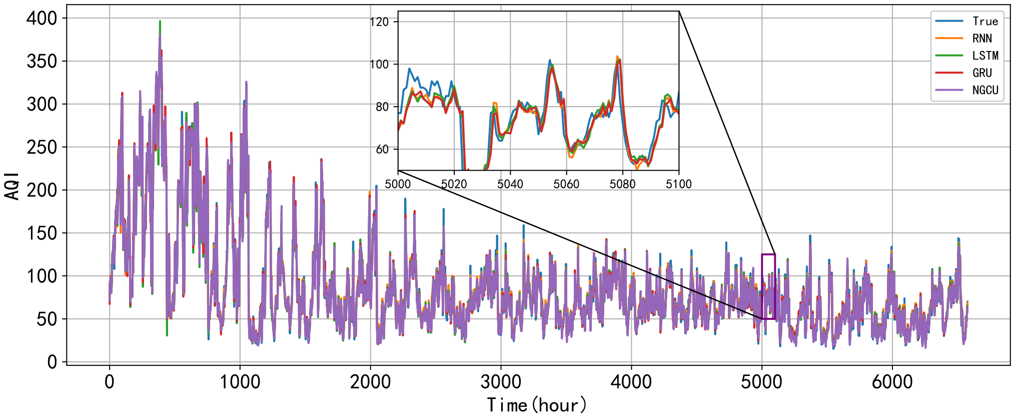

Fig. 10. Comparison of four models of air quality predicted values and true values.

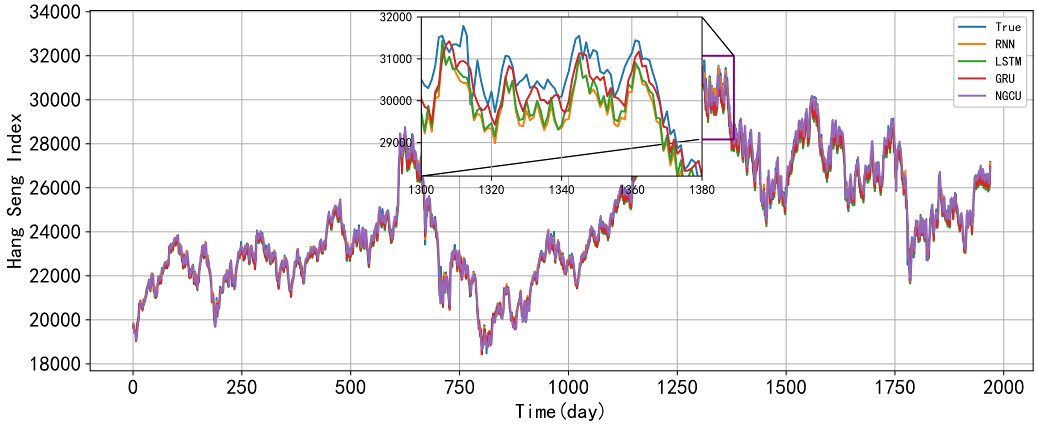

Fig. 11. Comparison of four models of Hang Seng Index predicted values and true values.

Table 5
Comparison of evaluation indexes of four air quality prediction models.

Model	MAE	MSE	EVS	R^2	Time
RNN	6.3359	96.8515	0.9707	0.9705	271.5667
LSTM	6.5003	97.9289	0.9704	0.9702	409.8045
GRU	6.2824	94.2690	0.9716	0.9713	349.2728
NGCU	6.1494	86.7811	0.9737	0.9736	323.5261

Table 6
Comparison of evaluation indexes of four Hang Seng Index prediction models.

Model	MAE	MSE	EVS	R^2	Time
RNN	264.2367	110548.7920	0.9898	0.9857	32.2799
LSTM	346.7847	193895.7178	0.9867	0.9773	73.1288
GRU	284.3403	141484.6643	0.9878	0.9835	63.9057
NGCU	254.4748	109564.9144	0.9903	0.9872	53.3257

the EVS of NGCU increases by 0.003,0.0033 and 0.0021. Compared with RNN, LSTM and GRU, the R^2 of NGCU increases by 0.0031 0.0034 and 0.0023. Compared with RNN, LSTM and GRU, the running time of NGCU reduces by 86.2784s and 25.7467s. Comparison of four models of air quality price predicted values and true values are shown in Fig. 10.

The prediction and evaluation indexes of the next trading day of Hang Seng Index are shown in Table 6. The real value, RNN prediction value, LSTM prediction value, GRU prediction value, and NGCU prediction value of the next trading day of Hang Seng Index are shown in Fig. 11. Compared with RNN, LSTM and GRU, the MAE of NGCU decrease by 9.7619, 92.3099 and 29.8655. Moreover, the MSE, EVS and R^2 of NGCU are better than those of other models. In terms of model training time, compared with LSTM, NGCU is reduced by 37.13%. Compared with GRU, NGCU decrease by 19.84%.

For the next trading day of gold future prices, as shown in Table 7, Compared with RNN, LSTM, and GRU, the R^2 of NGCU reduces by 4.57%, 6.38% and 8.2%, respectively. Comparison of four models of gold future price predicted values and true values are shown in Fig. 12.

Table 7
Comparison of evaluation indexes of four gold future price prediction models.

Model	MAE	MSE	EVS	R^2	Time
RNN	10.7934	277.8392	0.9192	0.8827	31.3629
LSTM	11.1624	313.8497	0.9125	0.8677	55.2909
GRU	11.9535	348.5716	0.9024	0.8531	44.6871
NGCU	8.7322	182.4184	0.9511	0.9231	43.4814

4.7. Discussion

In this paper, three different data sets are selected. For air quality data sets with a large amount and relatively stable data, the prediction results of NGCU are the best in terms of MAE, MSE, EVS, and R compared with the prediction results of RNN, LSTM, and GRU. For the Hang Seng Index data set with small data volume and low data stability, the prediction results of NGCU are the best in MAE, MSE, EVS, and R compared with the prediction results of RNN, LSTM, and GRU. For the gold future data set with the smallest amount of data and poor data stability, the prediction results of NGCU are the best in terms of MAE, MSE, EVS, and R compared with the prediction results of RNN, LSTM, and GRU. Obviously, the forget gate, input gate, and anti-saturation conversion

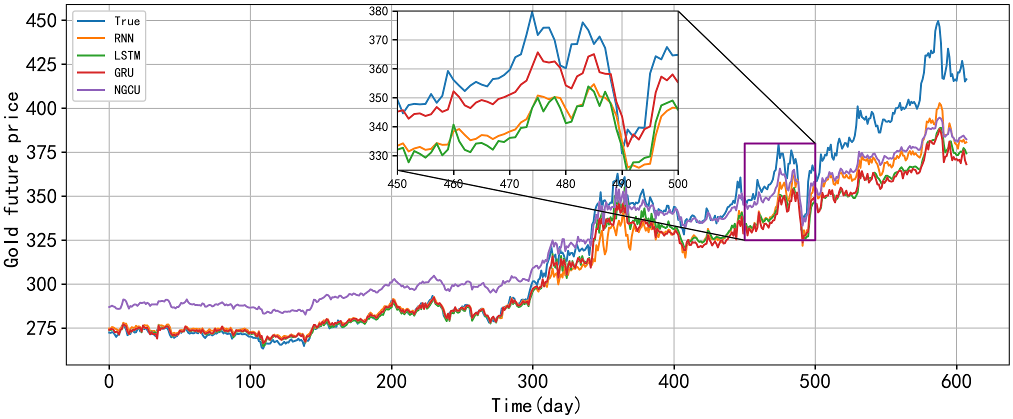

Fig. 12. Comparison of four models of gold future price predicted values and true values.

module of NGCU play a role in screening and learning historical data and current time, and the prediction accuracy of the model is also improved.

Compared with GRU, LSTM has made greater progress in training time in three different data set experiments due to the reduced complexity of NGCU. For example, when the three models predict the air quality-related eigenvalues, the training time of NGCU is 86.2784s less than that of LSTM, and the training time of NGCU is 25.7467s less than that of GRU. And in small data and unstable data, NGCU training time is also reduced. There are two reasons for the reduction in complexity: 1. In terms of gating units, compared with LSTM, the gating units are reduced from three to two, so the NGCU model is more concise. 2. In terms of parameter configuration, compared with LSTM, the number of W parameters is reduced from 8 to 4, and the number of b parameters is reduced from 4 to 2. Compared with GRU, the number of W parameters is reduced from 6 to 4, and the number of b parameters is reduced from 3 to 2, so the computational complexity of NGCU is greatly reduced. Compared with LSTM and GRU, NGCU model has the greatest advantages of simple structure and fast training speed.

4.8. Conclusions

Through the research of RNN, LSTM, GRU and time-series prediction, this paper proposes a new gate control unit. The proposal of NGCU is mainly used for time-series data prediction. Compared with traditional RNN, NGCU greatly alleviates the problems of gradient disappearance and explosion caused by long-term data dependence. In this paper, RNN, LSTM and GRU are introduced as the basic comparison model, and three different data characteristics are selected as the experimental data set. The experimental results show that the R^2 predicted by NGCU on the relevant feature values of the air quality data set, the Hang Seng Index data set, and the gold futures price data set reached 0.9736, 0.9872, and 0.9231 respectively, which are superior to RNN, LSTM, and GRU. Compared with LSTM and GRU, NGCU consumes the shortest training time in three different data prediction processes, which are 323.5261s, 53.3257s, 43.4814s, respectively. In terms of the MAE of air quality, Hang Seng Index, and gold future price prediction, the MAE of NGCU is 6.1494, 254.4748, and 8.7322 respectively, which is the smallest prediction error compared with RNN, LSTM, and GRU. Therefore, NGCU has made some innovative advancements in learning sensitivity, time-series prediction accuracy, and training time. Therefore, NGCU can alleviate the problems of gradient disappearance and explosion caused by long-term data dependence of RNN.

In this research, it is found that in different time-series data, NGCU's prediction of relevant feature values can still maintain a high prediction accuracy rate. However, the accuracy of NGCU's prediction of extreme values is not effectively improved. Further explore will be conducted in future on how to improve the accuracy of the model's prediction of extreme values.

Funding

This research was funded by Innovation Foundation for Postgraduate of Hebei Province, grant number CXZZSS2022082, the Scientific Research Project Foundation for High-level Talents of the Xiamen Ocean Vocational College under Grant KYG202102, and Foundation of Hebei University of Science and Technology under Grant 2019-ZDB02.

Declaration of competing interest

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

References

[1] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA 79 (8) (1982) 2554–2558.

[2] X. Wei, L. Zhang, H.Q. Yang, et al., Machine learning for pore-water pressure time-series prediction: application of recurrent neural networks, Geosci. Front. 12 (1) (2021) 453–467.

[3] J. Zheng, L. Zheng, A dictionary-based convolutional recurrent neural network model for sentiment analysis, in: CISCE, 2019.

[4] Z. Yu, Y. Dong, G. Chen, Advanced recurrent neural networks for automatic speech recognition, in: New Era for Robust Speech Recognition, 2017, https://doi.org/10.1007/978-3-319-64680-0_11.

[5] S. Agarwalla, Sarma K.K., Machine learning based sample extraction for automatic speech recognition using dialectal Assamese speech, Neural Netw. 78 (2016) 97–111.

[6] D. Soutner, J. Zelinka, L. Müller, On a hybrid NN/HMM speech recognition system with a RNN-based language model, in: International Conference on Speech and Computer, vol. 8773, 2014, pp. 315–321.

[7] K. Cho, B.V. Merrienboer, C. Gulcehre, et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation, Computer Science, 2014.

[8] M.K. Vathsala, Holi Ganga, RNN based machine translation and transliteration for Twitter data, Int. J. Speech Technol. 23 (2020) 499–504.

[9] A. Rius, I. Ruisánchez, M.P. Callao, et al., Reliability of analytical systems: use of control charts, time series models and recurrent neural networks (RNN), Chemom. Intell. Lab. Syst. 40 (1) (1998) 1–18.

[10] Q. Yin, R. Zhang, X.L. Shao, et al., CNN and RNN mixed model for image classification, MATEC Web Conf. 277 (8) (2019) 1–7.

[11] M. Sundermeyer, H. Ney, R. Schluter, From feedforward to recurrent LSTM neural networks for language modeling, IEEE/ACM Trans. Audio Speech Lang. Process. 23 (3) (2015) 517–529.

[12] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, Neural Comput. 12 (10) (2000) 2451–2471.

[13] K. Cho, B.V. Merrienboer, C. Gulcehre, et al., Learning phrase representations using RNN encoder-decoder for statistical machine translation, Comput. Sci. (2018), https://doi.org/10.3115/v1/D14-1179.

[14] M.D. Caux, F. Bernardini, J. Viterbo, Short-term forecasting in bitcoin time series using LSTM and GRU RNNs, in: Symposium on Knowledge Discovery, Mining and Learning, 2020.

[15] M.C. Sorkun, O.D. Incel, C. Paoli, Time series forecasting on multivariate solar radiation data using deep learning (LSTM), Turk. J. Electr. Eng. Comput. Sci. 28 (1) (2020) 211–223.

[16] Y. Baek, H.Y. Kim, A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module, Expert Syst. Appl. 113 (2018) 457–480.

[17] J. Wu, Z. Zhou, Y. Wang, et al., Multi-feature and multi-instance learning with anti-overfitting strategy for engagement intensity prediction, in: 2019 International Conference on Multimodal Interaction, 2019.

[18] J. Chung, C. Gulcehre, K.H. Cho, et al., Empirical evaluation of gated recurrent neural networks on sequence modeling, Computer Science, arXiv:1412.3555v1, 2014.

[19] S. Gao, S. Huang, S. Zhang, et al., Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation, J. Hydrol. (2020), https://doi.org/10.1016/j.jhydrol.2020.125188.

[20] P. Li, Z. Tan, Y. Lili, et al., Time series prediction of mining subsidence based on a SVM, Min. Sci. Technol. 21 (004) (2011) 557–562.

[21] S.C. Wu, F.J. Shao, R.C. Sun, Corn futures price forecast based on Arima time series and support vector machine, in: ICSCBD, 2018, pp. 47–55.

[22] V. Chimmula, L. Zhang, Time series forecasting of COVID-19 transmission in Canada using LSTM networks, Chaos Solitons Fractals (2020), https://doi.org/10.1016/j.chaos.2020.109864.

[23] I.E. Livieris, E. Pintelas, P. Pintelas, A CNN–LSTM model for gold price time-series forecasting, Neural Comput. Appl. 32 (2020) 17351–17360.

[24] Bala R. Preeti, B.P. Singh, Financial and non-stationary time series forecasting using LSTM recurrent neural network for short and long horizon, in: ICCCNT, 2019.

[25] X. Zhou, J. Xu, P. Zeng, et al., Asir pollutant concentration prediction based on GRU method, J. Phys. Conf. Ser. 1168 (3) (2019) 032058.

[26] J. Hu, X. Wang, Y. Zhang, et al., Time series prediction method based on variant LSTM recurrent neural network, Neural Process. Lett. 52 (2) (2020) 1485–1500.

[27] J. Hu, W. Zheng, Multistage attention network for multivariate time series prediction, Neurocomputing 383 (2020) 122–137.

[28] X.R. Zu, R.X. Song, Short-term wind power prediction method based on wavelet packet decomposition and improved GRU, J. Phys. Conf. Ser. 1087 (2) (2018) 022034.

[29] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: continual prediction with LSTM, Neural Comput. 12 (10) (2000) 2451–2471.

[30] F.A. Gers, D. Eck, J. Schmidhuber, Applying LSTM to time series predictable through time-window approaches, in: Optimization with Clifford Support Vector Machines and Applications, vol. 7, 2010, pp. 233–262.

[31] F. Karim, S. Majumdar, H. Darabi, et al., LSTM fully convolutional networks for time series classification, IEEE Access 6 (2017) 1662–1669.