Journal of Quantum Computing
DOI:10.32604/jqc.2021.016346
Article

Tech Science Press

A Prediction Method of Trend-Type Capacity Index Based on Recurrent Neural Network

Wenxiao Wang[1,*], Xiaoyu Li[1,*], Yin Ding[1], Feizhou Wu[2] and Shan Yang[3]

[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, China

[2]SI-TECH Information Technology Company Limited, Beijing, China

[3]Department of Chemistry, Physics and Atmospheric Sciences, Jackson State University, Jackson, USA

[*]Corresponding Authors: Wenxiao Wang. Email: 2018091609006@std.uestc.edu.cn; Xiaoyu Li. Email: xiaoyuuestc@uestc.edu.cn

Received: 31 December 2020; Accepted: 13 May 2021

Abstract: Due to the increase in the types of business and equipment in telecommunications companies, the performance index data collected in the operation and maintenance process varies greatly. The diversity of index data makes it very difficult to perform high-precision capacity prediction. In order to improve the forecasting efficiency of related indexes, this paper designs a classification method of capacity index data, which divides the capacity index data into trend type, periodic type and irregular type. Then for the prediction of trend data, it proposes a capacity index prediction model based on Recurrent Neural Network (RNN), denoted as RNN-LSTM-LSTM. This model includes a basic RNN, two Long Short-Term Memory (LSTM) networks and two Fully Connected layers. The experimental results show that, compared with the traditional Holt-Winters, Autoregressive Integrated Moving Average (ARIMA) and Back Propagation (BP) neural network prediction model, the mean square error (MSE) of the proposed RNN-LSTM-LSTM model are reduced by 11.82% and 20.34% on the order storage and data migration, which has greatly improved the efficiency of trend-type capacity index prediction.

Keywords: Recurrent Neural Network (RNN); Long Short-Term Memory (LSTM) network; capacity prediction

1 Introduction

Driven by the technologies such as 5G, big data, and edge computing, the volume and types of business of telecommunications companies continue to increase. Traditional telecom operation and maintenance rely on experience or expert advice to adjust the configuration of servers, which has been difficult to adapt to the rapid growth on the number of servers today. With the rapid development of artificial intelligence and data science, telecom operation and maintenance are constantly optimized and upgraded. By analyzing the massive performance monitoring data generated by a large number of servers, the future performance change trend of servers can be predicted, which can provide reference for the system or operation and maintenance personnel to manage relevant resources. In addition, due to the limitation of enterprise resources, high-precision capacity prediction can reduce operating costs and improve system stability. Capacity refers to the amount of resources pre-allocated to specific application systems, such as CPU, memory, disk, network bandwidth, etc., and its configuration is related to the fluency of application system operation [1]. It has become an important content of capacity management to predict the future resource capacity, which can help allocate resources reasonably and reduce resource redundancy.

Aiming at the forecast of resource utilization, many forecasting schemes have been proposed. In [2], it proposes a prediction method based on the linear regression technology to predict future CPU usage. A multiple linear regression model is proposed in [3], which is used to predict the future capacity demand of telecom enterprises' cloud computing data centers. It is proposed in [4] to use Second exponential smoothing method as the basic model, and then use the linear regression model as an integrated model to fit multiple sets of predicted values to predict the future request volume of cloud servers. In [5], it proposes a prediction method based on the ARIMA. With the development of neural network technology, it has been applied to the field of resource prediction. In [6], it uses BP neural network to predict the load of server nodes. In [7], it proposes a hybrid model that first used ARIMA to predict the linear body of the server CPU utilization time series, and then used the BP neural network model to correct the nonlinear residuals, and finally superimposed the entire time series as the forecast result. In [8], it uses the RNN method to accurately predict short-term CPU utilization. In [9] and [10], it proposes to use LSTM neural network to predict the load of the system. In [11], it uses a multivariate LSTM model to predict resource usage in cloud workloads, and uses a method to analyze and compare the LSTM model and the Bi-directional Long Short-Term Memory (BiLSTM) model. In [12], it proposes a combined model, adding LSTM network on the basis of BP neural network model to correct the residual error and predict the cloud computing resource load.

Among the time series prediction schemes put by researchers, some research methods can show high accuracy on the index data of a certain business. However, the diversity of telecom business and the differences in deployed equipment make these methods less effective for all devices. In order to solve this problem, this paper first designed a classification method of capacity index data types, which divides the data into trend type, periodic type and irregular type [13]. Aiming at the trend-type data, it proposes a model based on recurrent neural network.

In Section 2, it introduces the basic knowledge of recurrent neural networks. In Section 3, it proposes the classification method of capacity index and the prediction model based on the recurrent neural network. In Section 4, it analyzes the experimental results to verify the feasibility of the model. In Section 5, it summarizes the full text and puts forward a vision for future work.

2 Recurrent Neural Network

2.1 RNN Model

RNN is a special type of neural network with internal sub-connections in the field of deep learning, which can learn complex vector-to-vector mapping. RNN has input layer, hidden layer and output layer [14]. The state of the hidden layer is not only related to the current input layer, but also related to the state of the hidden layer at the previous time step, so the RNN has memory ability. The structure of a simple RNN expanded three times in the time dimension is shown in Fig. 1.

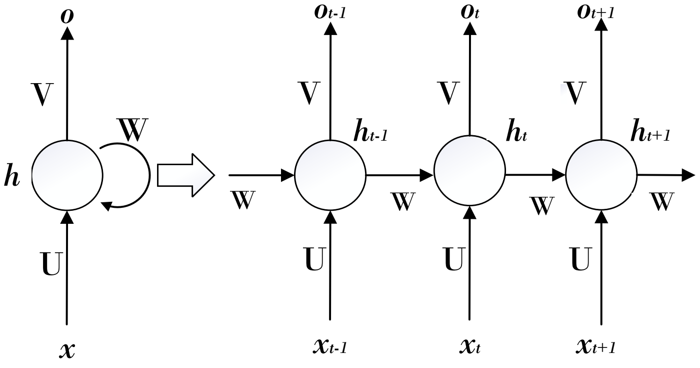

Figure 1: The unfolded RNN structure

In Fig. 1, t represents the moment, x_t represents the input at time t. Given input sequence $\mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \ldots \mathrm{x}_\mathrm{t}]$, the hidden layer state sequence $h = [h_1, h_2, \ldots h_t]$ can be calculated by Eq. (1), where U represents the weight matrix from the input layer to the hidden layer, W represents the weight matrix from the hidden layer at the last time step to the hidden layer at the current time step, f represents the activation function and b_h represents the bias.

$$h_t = f(U \cdot x_t + W \cdot h_{t-1} + b_h) \tag{1}$$

Then according to Eq. (2), the output sequence $o = [o_1, o_2, \ldots o_t]$ can be calculated, where V represents the weight matrix, g represents the activation function and b_o represents the bias from the hidden layer to the output layer.

$$o_t = g(V \cdot h_t + b_o) \tag{2}$$

In the training process, the Back-Propagation Through Time (BPTT) algorithm is often used to train RNN, whose essence is the BP algorithm. However, due to some characteristics of the network itself, the problem of vanishing gradient or exploding gradient is likely to occur when the BPTT algorithm is used to train the network with a long sequence.

2.2 LSTM Model

The LSTM network is a special form of the RNN network, which avoids the vanishing gradient and exploding gradient problems through clever design, and can learn long-term information. Compared with the simple RNN network, LSTM adds a transmission state and three control gate units: forget gate, input gate and output gate [14]. The LSTM unit structure is shown in Fig. 2.

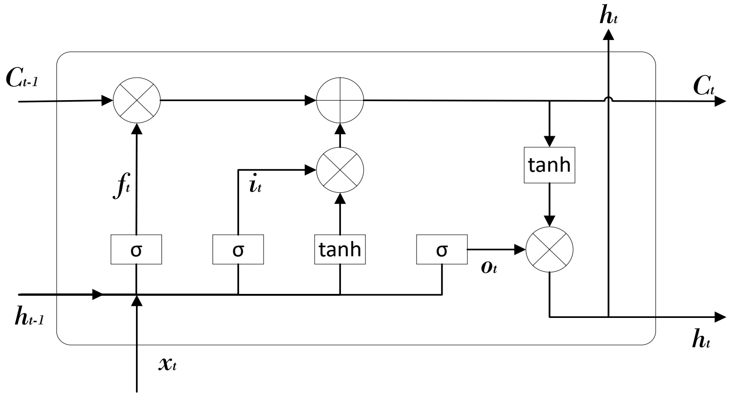

Figure 2: The LSTM unit structure

The forget gate can control what information to keep and what to forget. The forget gate receives the output information from the hidden layer of the previous time step and the current sample input, as shown in Eq. (3), and passes it to the sigmoid function after operation. The output value of the sigmoid function is in the range of [0,1], which means how much information is allowed to pass, 0 means no information is allowed to pass, and 1 means all. The information of memory cells in the hidden layer can be forgotten by the Hadamard product of f_t and cell state C_{t-1}, where x_t represents the input of the current network, h_{t-1} represents the output of the previous network, f_t represents the output value, W_f represents the weight matrix and b_f represents the bias of the forget gate.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3}$$

The input gate controls how much current input data x_t flows into the memory cell. After forgetting the information in the memory cell, the information stored in the memory cell needs to be updated next. At this time, the input gate determines which parts of the new information are put into the cell state, which mainly includes two parts, as shown in Eq. (4). One part is the sigmoid layer, which can prevent the irrelevant content of the current input from entering the memory cell. The other part is the tanh layer, which generates a candidate value that will be added to the cell state, and the output value of the tanh function is in the range of [–1,1]. In Eq. (4), i_t, W_i and b_i represent the value, weight matrix and bias of the input gate respectively; $\tilde{C}_t$, W_c and b_c represent the candidate value, weight matrix and bias, respectively.

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \end{cases} \quad (4)$$

The new cell state information is jointly determined by the forget gate and the input gate, as shown in Eq. (5). It is determined by the Hadamard product of f_t and cell state C_{t-1} and the memory cell.

$$C_t \ = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (5)$$

In the output gate, the memory unit C_t has influence on the current output value h_t, as shown in Eq. (6). The input passes through a sigmoid layer to obtain a value o_t in the range of [0,1], which controls how much the memory cell outputs. Then, the final output h_t is determined by the Hadamard product of o_t and the value of state value C_t after tanh activation. In Eq. (6), W_o and b_o respectively represent the weight matrix and bias of the output gate.

$$\begin{cases} o_t \ = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t \odot \tanh(C_t) \end{cases} \quad (6)$$

3 Trend-Type Capacity Index Data Prediction Model Based on Recurrent Neural Network

This section elaborates on the trend-type capacity index data prediction model based on recurrent neural network. First, the classification method of index data designed in this paper is introduced; Then, the trend-type capacity index data prediction model based on recurrent neural network is introduced.

3.1 Classification Method of Index Data Types

3.1.1 Dynamic Time Warping

Dynamic Time Warping (DTW) confirms the optimal path between two time series by minimizing the cumulative distance between the original sequence $x(i), i \in [1, M]$ and the series to be aligned $y(j),\ j \in [1, N]$ [15]. Therefore, the cumulative distance between two time series can be used to measure the similarity of them. $d_{i,j}$ is used to represent the distance set between two points in the series, and its definition is shown in Eq. (7).

$$d_{i,j} = (x(i) - y(j))^2 \quad (7)$$

The sum of the cumulative distance sets is represented by D, where $D_{i,j}$ represents the minimum cumulative distance from the origin $(1,1)$ to the point (i, j). The calculation method is shown in Eq. (8), where $i = 2, \dots, N$; $j = 2, \dots, M$.

$$D_{i,j} = d_{i,j} + min \begin{cases} D_{i,j-1} \\ D_{i-1,j} \\ D_{i-1,j-1} \end{cases} \quad (8)$$

And the initial conditions are shown in Eq. (9).

$$\begin{cases} D_{1,1} = d_{1,1} \\ D_{1,j} = \sum_{a=1}^{j} d_{1,a} \quad j = 1, \dots, M \\ D_{i,1} = \sum_{b=1}^{i} d_{b,1} \ i = 1, \dots, N \end{cases} \quad (9)$$

After the preliminary investigation, it is learned that many Key Performance Indicators of businesses show a daily trend. In this paper, the time series $value$ with a length of l days is selected to compare the similarity between the time series of two adjacent days. By calculating the DTW distance of two adjacent days, the distance series $dist\_list$ of length $l - 1$ is obtained, which can represent the similarity of them. Then compare the $dist\_list$ with the preset DTW threshold to determine whether the time series has a certain daily trend.

3.1.2 Classification Method of Data

The index data types are divided into trend type, periodic type and irregular type by using data characteristics such as average value, standard deviation, and DTW value. The process of data type division is shown in Fig. 3 and the specific design is as follows:

(1) Data preprocessing. First, clean the collected index data of the target device, then fill in the empty values, and finally obtain the index series C in the specified format.

(2) Data type determination. Calculate the DTW value $value\_dtw$ and the coefficient of variation $value\_cv$ of C. The coefficient of variation is the ratio of the standard deviation to the average of the original data, which can be used to measure the degree of data dispersion and eliminate the influence of measurement scale and dimension. Data type was determined by comparing $value\_dtw$ and $value\_cv$ with the threshold $threshold\_dtw$, trend-type variation coefficient $threshold\_cv\_trend$ and periodic variation coefficient $threshold\_cv\_periodic$. The process is as follows:

Step1 If $value\_dtw < threshold\_dtw$ and $value\_cv < threshold\_cv\_trend$, it is a trend type, otherwise enter Step 2;

Step2 If $value\_dtw < threshold\_dtw$ and $value\_cv<threshold\_cv\_periodic$, it is a periodic type, otherwise it enters Step 3;

Step3 This type is irregular.

(3) Output the data type result. Output the data type judgment in (2).

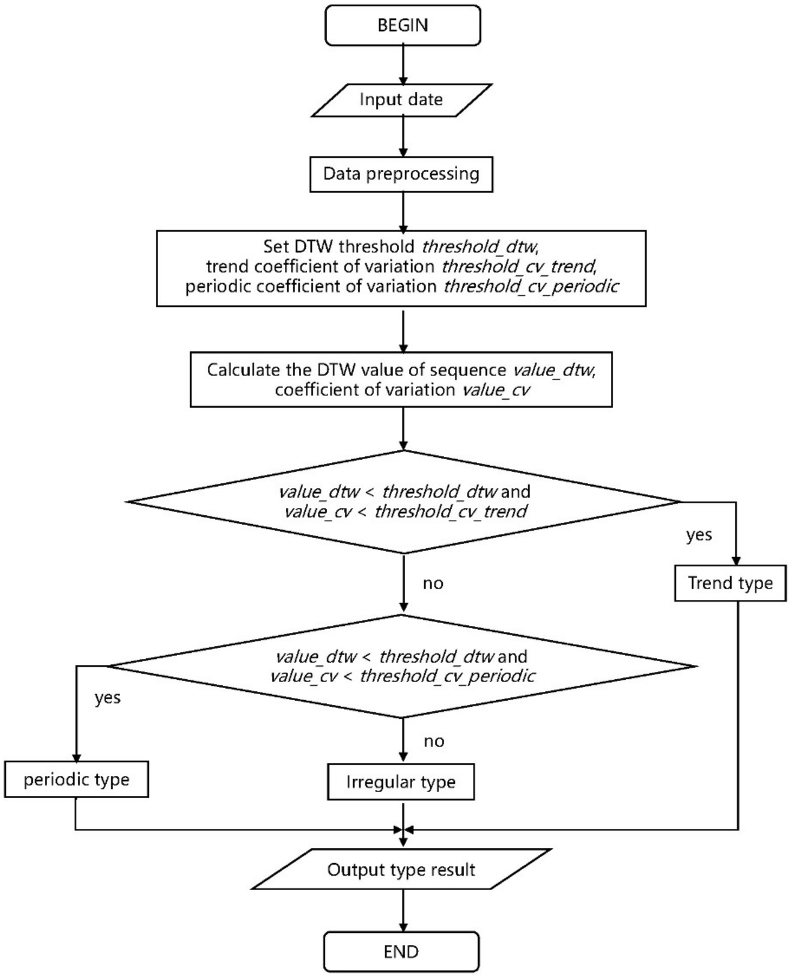
BEGIN
Input date
Data preprocessing
Set DTW threshold threshold_dtw,
trend coefficient of variation threshold_cv_trend,
periodic coefficient of variation threshold_cv_periodic
Calculate the DTW value of sequence value_dtw,
coefficient of variation value_cv
value_dtw < threshold_dtw and
value_cv < threshold_cv_trend
yes
Trend type
no
value_dtw < threshold_dtw and
value_cv < threshold_cv_periodic
yes
no
periodic type
Irregular type
Output type result
END

Figure 3: The data type division process

3.2 Trend-Type Capacity Index Data Prediction Model Based on Recurrent Neural Network

It first preprocesses the collected data, then uses the designed recurrent neural network model for training, and finally makes predictions. The specific design is as follows:

1. Data preprocessing. First, clean the collected index data of the target device, then fill in the empty value, and finally get the index series C in the specified format.
2. Build a trend-type capacity index data prediction model based on recurrent neural network. Compared with a single-layer network, a multi-layer recurrent neural network can learn more hidden information. After preliminary experiment comparison, the three-layer model is more effective, so this model is designed as a three-layer network. First, the processed performance index series are input to the SimpleRNN network, which is used to mine the law of performance index changing with time. The series result of SimpleRNN is input into the two connected LSTM layers. To prevent overfitting, the model adds a Dropout layer after each LSTM layer. After the last LSTM layer, two fully connected layers are added. And there is Rectified Linear Units activation function after each fully connected layer. Through the fully connected layers, the impact of historical data at each time step on future time point data is summarized. Finally, a real value is output as the predicting value.
3. Train the model. Divide the historical data into three parts: training set, validation set and test set. Input the training set data into the trend-type capacity data prediction model based on recurrent neural network for training, and finally evaluate the trained model.

4 Experiment and Result Analysis

In this section, it first introduces the overview of the experimental data. Then, the experimental operating environment, experimental tools, experimental parameter settings and evaluation indicators are introduced. And then, the trend-based capacity data prediction model based on recurrent neural network proposed is implemented and compare it with other models in related research. Finally, the experimental results are compared and analyzed.

4.1 Experimental Data

CPU, as one of the important indexes to measure the performance of host, is the most demanding resource and the main reason for the shortage of host resources [16]. Therefore, this paper selects CPU resource as the object of capacity prediction. In this experiment, the performance monitoring data are from the Customer Relationship Management System (CRM) system in one province provided by Beijing SI-TECH Information Technology Co., Ltd., China (http://www.si-tech.com.cn). Two businesses, data migration and order storage, which occupy a large number of equipment, are selected from branch businesses with the characteristics of daily trend data. Then among these business servers, 4 months 'CPU index data of a randomly selected server are regarded as experimental data. And due to the balance of server load, this is reasonable. Index data is collected every 6 min, and each server has about 31,200 CPU index data. Examples of data collected by the monitoring center are shown in Tab. 1. Since frequent forecasting will take up a lot of resources, the company requires hourly granularity to make forecasts in actual production, so the data needs to be processed before forecasting.

Table 1: The samples of data collected by monitoring center

EQ_ID	KPI_ID	KPI_VALUE	COLL_TIME
4194	PM-H-01-010-11	36.667	2019-04-30 09:33:32
4194	PM-H-01-010-11	29	2019-04-30 09:39:05
4194	PM-H-01-010-11	24.667	2019-04-30 09:39:32
4194	PM-H-01-010-11	26.333	2019-04-30 09:45:31

4.2 Experimental Environment and Parameter Configuration

The computer used in the experiment is with 16GB memory and 3.20GHz Inter® Core™ i7-8700 processor. The experiment is designed to predict the CPU utilization in the t+1 hour through the CPU utilization in the previous t hours. The original data were divided into training set and test set according to the ratio of 0.8:0.2. Experiments were carried out on 8, 16, 32, 64, 128 neurons in each layer of the designed neural network, and the number of neurons with the best effect was selected as the number of neurons in the final model. The relevant parameters are set to epochs = 100, batch_size = 60 and time_step = 48, and Adaptive Moment Estimation (Adam) is selected as the optimizer.

The accuracy of the prediction is measured by Mean Squared Error (MSE). MSE is the average square of the difference between the estimate and the true value. It is used to measure the deviation between the predicted value and the true value. MSE is defined as Eq. (10), where $true_t$ represents the true value of the data at time t, $predicted_t$ represents the predicted value of the data at time t, and m represents the total number of samples.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(true_t - predicted_t)^2 \quad (10)$$

4.3 Prediction Model Based on Recurrent Neural Network

4.3.1 Prediction Result Analysis

In order to verify whether the trend-type capacity index data prediction model based on the recurrent neural network has a good effect, the SimpleRNN, LSTM, and two-layer LSTM models are used as the control groups for experiments.

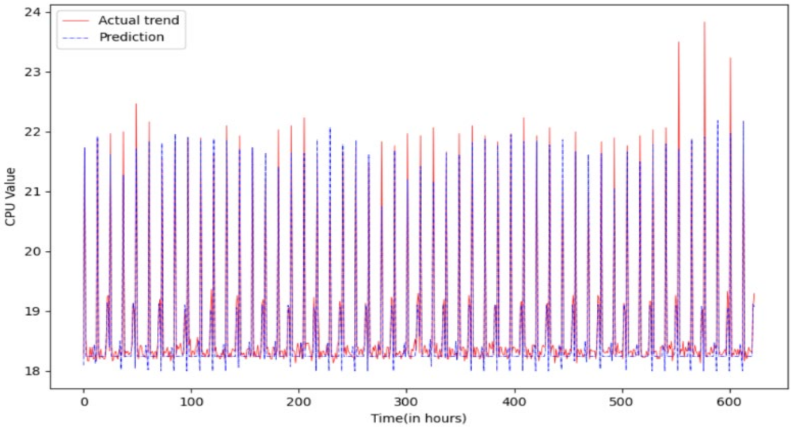

Figure 4: Two-layer LSTM prediction results

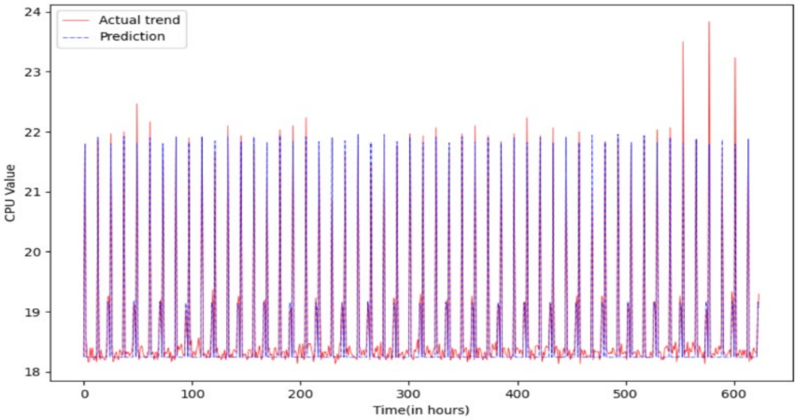

Figure 5: RNN-LSTM-LSTM prediction results

Fig. 4 shows the test results using the two-layer LSTM network model. Fig. 5 shows the test results using the recurrent neural network model designed in this paper. The horizontal axis in Fig. 4 and 5 are the time axis. There are 624 time slices in total, and each time slice represents 1 hour. The vertical axis represents the CPU index value in the range of [0,100]. The two curves in the figure are the actual and predicted values of the CPU indicators. Analyzing the CPU index value of 30 days, we can see that the business server CPU index value is relatively large at night (22:00 to 2:00 the next day) and noon (11:00 to 14:00). Although the two-layer LSTM network can reflect the change trend of the CPU index value, it does not fit the true curve at the peaks and troughs. And the model designed in this paper can better track the change trend of the CPU index value and achieve a more accurate prediction.

Table 2: The results of 4 kinds of network prediction model

Model	Data migration MSE	Order storage MSE
SimpleRNN	0.073421	0.219227
LSTM	0.105500	0.220745
Two-layer LSTM	0.050944	0.203115
RNN-LSTM-LSTM	0.033414	0.189295

Tab. 2 respectively lists the average MSE values predicted by four models on the data set of data migration and order storage business. It shows that for data migration services, the MSE of the RNN-LSTM-LSTM model is at least 34.41% lower than other models. For the order business, the MSE of the RNN-LSTM-LSTM model is also at least 6.80% lower than other models.

In order to further verify the capacity of the capacity prediction model, it was compared with the traditional prediction methods ARIMA, Holt-Winters algorithm and BP neural network. The average MSE obtained is shown in Tab. 3.

Table 3: The results of traditional prediction models

Model	Data migration MSE	Order storage MSE
Holt-Winters	0.518302	0.227169
ARIMA	0.183369	0.214663
BP neural network	0.041947	0.253612
RNN-LSTM-LSTM	0.033414	0.189295

It can be seen from Tab. 3 that for data migration and order storage business, the MSE values of the predicted results of the RNN-LSTM-LSTM model are at least 20.34% and 11.82% lower than those of the three traditional models.

From the above experimental results, it can be seen that the prediction accuracy of the RNN-LSTM-LSTM network model is higher and the error is smaller, indicating that the model can better predict trend-type capacity index data.

5 Conclusion and Future Work

This paper first designs a classification method of index data types, which can be divided into trend type, periodic type and irregular type. Aiming at the prediction of trend data, a capacity index prediction model based on recurrent neural network is proposed. Combined with the CRM system data provided by Beijing SI-TEQI Information Technology Co., Ltd., the model was validated by the experiment. The experimental results show that the MSE of the RNN-LSTM-LSTM model proposed in this paper are at least 11.82% and 20.34% lower than those of the traditional Holt-Winters, ARIMA and BP neural network prediction models respectively in terms of the accuracy of the two trend-type business prediction, which indicate the feasibility of the model proposed in this paper. The next-step work will continue to improve the prediction accuracy of periodic and irregular data and combine the three types of prediction models for practical application.

Acknowledgement: The study was supported by Research on Big Data Technology for New Generation Internet Operators (H04W180609) and the second batch of Sichuan Science and Technology Service Industry Development Fund Projects in 2018 (18KJFWSF0388).

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. Q. Zhou, "Exploration of software system capacity management," *Financial Computer of China*, vol. 25, no. 11, pp. 60–63, 2013.

[2] F. Fahimeh, L. Pasi and P. Juha, "Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers," in *Proc. SEAA*, Santander, Spain, pp. 357–364, 2013.

[3] F. S. Yue and Q. Wei. "Research on cloud computing capacity estimation based on multiple linear regression," *Information & Communications*, vol. 1, no. 2, pp. 1–3, 2016.

[4] J. T. Shi, J. X. Sun and H. F. Wu, "Ensemble prediction model based on exponential smoothing for cloud server request quantity," *Computer Engineering and Design*, vol. 41, no. 2, pp. 432–439, 2020.

[5] J. Q. Jiang, "Prediction of Telecom capacity expansion index based on ARIMA," *Kexue Yu Xinxihua*, vol. 5, no. 6, pp. 28–29, 2020.

[6] B. B. Zhang, N. J. Chen and D. D. Hu, "Virtual machine deployment strategy by load prediction based on BP neural network," *Journal of Huazhong University of Science and Technology-Nature Science*, vol. 40, no. S1, pp. 120–123, 2012.

[7] J. N. Wang, Y. M. Yan and J. Guo, "Research on the prediction model of CPU utilization based on ARIMA-BP neural network," *MATEC Web of Conferences*, vol. 65, no. 1, pp. 1–11, 2016.

[8] M. Duggan, K. Mason, J. Duggan, E. Howley and E. Barrett, "Predicting host CPU utilization in cloud computing using recurrent neural networks," in *Proc. ICITST*, Cambridge, UK, pp. 67–72, 2017.

[9] L. Fu, "Time series-oriented load prediction using deep LSTM," *Modern Computer*, vol. 8, no. 9, pp. 25–28, 2020.

[10] F. Schmidt, M. Niepert and F. Huici, "Representation learning for resource usage prediction," in *Proc. SysML Conf.*, Palo Alto, USA, pp. 1–10, 2018.

[11] S. Gupta, D. A. Dinesh, "Resource usage prediction of cloud workloads using deep bidirectional long short term memory networks," in *Proc. ANTS*, Bhubaneswar, India, pp. 1–6, 2017.

[12] D. L. Chen, W. J. Lin and L. L. Huang, "Load forecasting for cloud computing resource based on BPNN-LSTM composite model," *Journal of Beijing University of Posts and Telecommunications (Social Sciences Edition)*, vol. 22, no. 1, pp. 53–60, 2020.

[13] H. M. Yang, Z. S. Pan and W. Bai, "Overview of time series forecasting methods," *Computer Science*, vol. 46, no. 1, pp. 21–28, 2019.

[14] L. Yang, Y. X. Yang, J. L. Wang and Y. L. Liu, "Research on recurrent neural network," *Journal of Computer Applications*, vol. 38, no. z2, pp. 1–6, 2018.

[15] M. Morel, C. Achard, R. Kulpa and S. Dubuisson, "Time-series averaging using constrained dynamic time warping with tolerance," *Pattern Recognition*, vol. 74, no. 1, pp. 77–89, 2018.

[16] M. Duggan, K. Mason, J. Duggan, E. Howley and E. Barrett, "Predicting host CPU utilization in cloud computing using recurrent neural networks," in *Proc. ICITST*, Cambridge, UK, pp. 67–72, 2017.