Load Predicting Model of Mobile Cloud Computing Based on Glowworm Swarm Optimization LSTM Network

https://doi.org/10.3991/ijim.v14i05.13361

P. Sudhakaran (✉)
SRM TRP Engineering College, Trichy, India
sudhakaran0523@gmail.com

Subbiah Swaminathan
Saveetha School of Engineering, Chennai, India

D. Yuvaraj
Cihan University-Duhok, Kurdistan region, Iraq

S. Shanmuga Priya
M.I.E.T Engineering College, Trichy, India

Abstract—Focusing on the issue of host load estimating in mobile cloud computing, the Long Short-Term Memory networks (LSTM)is introduced, which is appropriate for the intricate and long-term arrangement information of the cloud condition and a heap determining calculation dependent on Glowworm Swarm Optimization LSTM neural system is proposed. Specifically, we build a mobile cloud load forecasting model using LSTM neural network, and the Glowworm Swarm Optimization Algorithm (GSO) is used to search for the optimal LSTM parameters based on the research and analysis of host load data in the mobile cloud computing data center. Finally, the simulation experiments are implemented and similar prediction algorithms are compared. The experimental results show that the prediction algorithms proposed in this paper are in prediction accuracy higher than equivalent prediction algorithms.

Keywords—Mobile cloud; LSTM Network; Glowworm Swarm Optimization Algorithm; Load Forecasting.

1 Introduction

As a mix of distributed computing and portable net-work innovation [1], Versatile distributed computing has been granted permission for individual computerized or printed copies of all or part of this work. The use of the study hall is free of charge given that duplicates are not created or circulated for the profit or business advantage and that duplicates carry this notice and a complete reference on the main page. Usually duplicating, republishing, presenting on servers or redistributing to records re-

quires earlier explicit permission and cost. Increasingly more consideration from specialists and researchers as of late. Portable distributed computing exploits the incredible registering and capacity abilities of distributed computing, just as the portability and adaptability of versatile terminals, carrying extraordinary comfort to individuals' lives [2]. On the portable cloud stage, the versatile terminal will present the calculation assignment of use programs with a lot of computation and information handling to the remote cloud server farm. At that point, it will acquire the calculation result from the server farm, shortening the reaction time of the application, in this manner improving the client experience of the versatile application and boosting the advantage of the specialist co-op. However, with the increase of mobile applications, more and more mobile terminals use mobile cloud computing services, which causes the power consumption fluctuation of cloud data centers to be more severe, and a more granular resource allocation strategy is essential. Therefore, to reduce energy consumption, how to predict and allocate resources in cloud data centers has become a research hotspot in recent years. The cloud load prediction technology is one of the technological means of reducing the energy consumption of the data center, which grasps the rule of load change and predicts the host load value of the next period by analyzing the historical data of the data center. Due to time delay in the process from assign to use of the host resources, allocating the resources needed after the load change could break up the Service Level Agreement (SLA), especially when the load increases rapidly. Host load forecasting technology is a crucial energy optimization method in the cloud environment. It can be used to reasonably schedule data center resources, dynamically shut down some idle physical machines, and reduce energy consumption without breaking the SLA and affecting the operation of the data centers. Therefore, the host load forecasting technology has become one of the supporting technologies for mobile cloud computing. At present, more researches are focused on the problem of host load prediction in the grid-computing environment [3]. Different from grid computing, the host load in mobile cloud computer grid computing [4]. Therefore, it is very difficult for load Forecasting of traditional grid computing to achieve satisfactory results in mobile cloud computing.

Existing load prediction techniques generally could be divided into two basic categories: first, the traditional statistical analysis method, which obtains the laws of data through statistical analysis of a mass of data, such as autoregressive analysis algorithm (Lee et al., 2016); Second is the artificial intelligence analysis method, basing on the historical data, predicts the values of the load in the required time horizon. Such as the Recurrent Neural Network (RNN) [5]. With the increasing requirement of prediction accuracy and prediction time, the single prediction algorithm has been unable to meet the needs. Some hybrid prediction algorithm, which combines multiple prediction algorithms, has become the trend to solve the load prediction problem. This is because the hybrid prediction algorithm can generally get better performance, such as integrating Particle Swarm Optimization (PSO) with Fuzzy Clustering Neural Network [6].

At present, many mature algorithms have been designed for load prediction through a single prediction algorithm. Sheng et al. designed a Bayesian model-based prediction method to predict the average load in the long-term time interval and the

average load in the future continuous time [4]. Xu et al. used Secret Markov model data clustering approach for cloud load prediction problems in the cloud host and used the genetic algorithm to optimize the Elman Network to predict the future load [7]. Fan et al. Designed a short-term building load prediction algorithm based on machine learning for short-term building load prediction [8]. Peng et al. modeling with adaptive neural fuzzy inference system to solve short-term power load, and constructed short-term power load prediction model based on factors such as weather and date [9]. The single prediction algorithm has shortcomings in preying gradient problems in RNN when predicting long-time Series data. The LSTM solves these problems by designing the control gate structure, which enables LSTM to predict for long-time series data effectively. At present, LSTM has been successfully applied to handwriting recognition [16], speech recognition [17], etc.

2 The GSO-LSTM Algorithm

Since the mobile cloud data center's host load is non-linear and unpredictable, and it is difficult to build an accurate model and to predict the load curve. The LSTM model can make full use of the advantages in dealing with complex nonlinear problems and the long-term time-series data. Further, the improved particle swarm optimization algorithm GSO is used to optimize the LSTM network parameters and improve the accuracy of the host load forecasting.

2.1 LSTM network

Host load types are primarily CPU, memory, network bandwidth and disk in the mobile cloud computing data center. Let the host load vector is L = (cpu, mem, hd, bw), where the four factors represent the load of CPU, memory, disk and network bandwidth respectively. The historical load sequence Ht= [Lt−d, Lt−d+1, · · ·, Lt−1] means loads of the d most recent intervals, where t is the current moment, d is the window value of Ht and will determines the accuracy of the load prediction.

The Recurrent Neural Network (RNN) is a recurrent neural network that uses sequence data as input, and all cyclic units form a closed-loop by chain connection, which possesses the strong ability to deal with nonlinear data. The difference between RNN and general neural networks is that the information transmission in RNN has a directed loop so that the state of a neuron at the previous moment will be passed to the next moment. However, RNN is prone to the vanishing gradient and exploding gradient problems when establishing a long-term dependency model [15]. LSTM is an improved recurrent neural network (RNN) that can effective processing long sequence data and avoid the defects of traditional RNN by using a unique 'gate' structure. Therefore, LSTM has a strong advantage in predicting long sequence data. The LSTM consists of multiple homogeneous cells (cyclic units) that store past information by updating state information, using the input, output, and forget gates to control the weight of the historical information. The unit structure of the LSTM-based mobile cloud host load prediction model at the moment t is shown in Figure 1. Where Ct−1

refers to the neuron status of the previous moment, Ht−1 is the output of the neurons at the last moment. The forget gates' inputs include the output Ht−1 and the load.

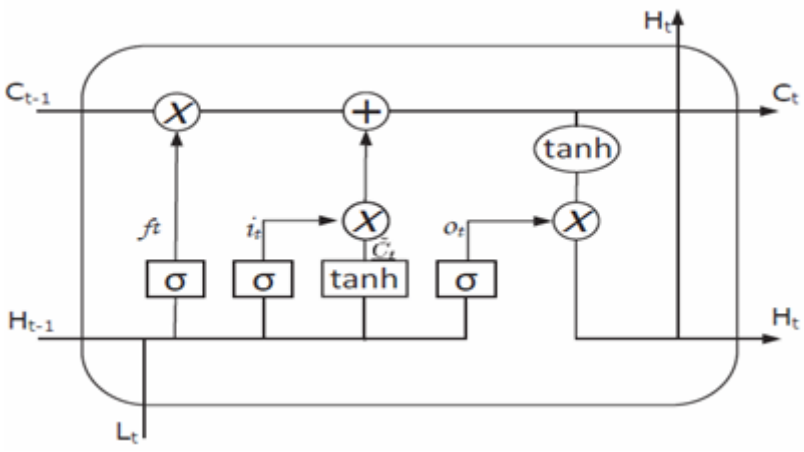

Fig. 1. The unit structure diagram of the LSTM based mobile cloud host load prediction model

Lt at current moment. Finally, get ft through the leftmost activation function sigmoid, which is calculated as follows:

$$ft = \sigma (Wf [Ht\text{-}1, Lt] + bf) \quad (1)$$

Wf is the forget gates' weight matrix, bf denotes the deviation matrix, σ represents the activation function sigmoid. Ft indicates the probability that the last layer of neurons been forgotten, with a range of [0,1], where 1 is fully reserved, and 0 is completely discarded. The input gates determine which new input in Lt can be stored in the neuron, which is mainly divided into it and ~ Ct. The input gates are calculated as follows:

$$it = \sigma (Wi [Ht\text{-}1, Lt] + bi) \quad (2)$$

$$\sim Ct = \tanh (Wc [Ht\text{-}1, Lt] + bc) \quad (3)$$

Wi and Wc are the input gates' weight matrices, bi and bc are the deviation matrices. it .~ Ct represents the proportion of load information that needs to be retained at the moment, which added to the retained neuron state at the previous moment to generate the updated neuron state information Ct. The output gates are used to control the output of the neuron state and transfer the state to the next neuron. The process is as follows:

$$ot = \sigma (Wo. [Ht\text{-}1, Lt] + bo) \quad (4)$$

$$Ht = ottanh (Ct) \quad (5)$$

Where Wo and bo are the weight matrix and the deviation matrix of the output gates layer respectively, Ht is the out- put of the current layer, and the calculation of the last layer obtains the final load prediction value.

The parameter optimization algorithm based on glowworms warm optimization: In the LSTM-based cloud hostload prediction model, the value of will limit the performance and accuracy of the LSTM-based load prediction model. However, the LSTM model does not handle the parameter selection problem well, resulting in poor regression prediction ability of the model. The commonly used parameter selection method is to manually adjust the value of the specified parameter and fix the remaining parameters at the same time, adjusting the value until the parameter is which is not only time consuming, but also easy to fall into local optimum. Therefore, this paper proposes an optimization parameter algorithm based on the new GSO Particle Swarm Optimization Algorithm.

The Glowworm Swarm Optimization (GSO) algorithm is a swarm intelligence algorithm that is proposed by mimicking the natural phenomena of Glowworm's clustering behavior. The GSO algorithm was first proposed by Krishnan and Chose in 2005 and has been successfully applied in various practical fields [19, 20]. When glowworms seek food or look for a spouse, they interact through fluorescein. In GSO, the location of each Glowworm represents a solution, and all of them constitute the solution space. The initial position of the glowworm group is randomly assigned in the search space, and then each Glowworm moves to the neighbor that has greater fluorescein than its own according to the probability mechanism. The greater the luciferin value is, the stronger the attraction to the Glowworm. After multiple iterations, most of the Glowworm tend to gather at the same location, which has the highest fluorescein concentration, the optimal solution for the GSO model.

Let L(O) = (batchsize, 10, epochs, hiddensize) denotes the initial position information of glowworm i, that is, the initial parameter combination of the LSTM load prediction model. Set the parameters of the GSO algorithm, including the number of iterations and the size of the population, and then execute the GSO optimization algorithm. The final GSO output is the optimal parameter combination of the LSTM model. The GSO Parameter Optimization Algorithm can be divided into seven steps:

1. **Initialization:** Randomly place n glowworm in the feasible domain, each glowworm has a fluorescein value of lo, the dynamic decision domain is ro, the initialization step- value is s, Neighborhood threshold is nt, the update rate and the disappearance rate of the fluorescein is Lambda and p.
2. **Update the fluorescein value of glowworm i:**

$$l_i(t) = (1 - p)\, li(t-1) + \lambda J(Li(t)) \tag{6}$$

   Where li(t) indicates the fluorescein of glowworm i, p and)., represent the fluorescein decay constant and the enhancement constant respectively. J(L;(t)) refers to the target function of the glowworm at position Si.

3. **Looking for the neighbors of glowworm:** A glowworm with larger fluorescein is selected as its neighbor in its local decision domain radial range ri(t).

4. Determine the direction of movement of the glowworm z:

$$P_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in Ni(t)} l_k(t) - l_i(t)} \tag{7}$$

Where Ni(t) represents the neighbor set of glowworm i. Glowwormi select a neighbor location to move based on the transition probability P; j (t).

5. Update the location of glowworm i:

$$L_i(t+1) = L_i(t) + s \cdot \frac{L_j(t) - L_i(t)}{||L_j(t) - L_i(t)||} \tag{8}$$

6. Update dynamic decision domain:

$$r_d^i(t+1) = \min\{r_s, \max\{0, r_d^i(t) + \beta(n_t - |Ni(t)|)\}\}$$

Where rs indicates the radiation range of the fluorescein, β represents the refresh rate of the local decision domain, and nt is the parameter that controls the number of neighbors.

7. Parameter optimization: Set the Mean Absolute Percentage Error (MAPE) of the LSTM load prediction model as the objective function. The calculation formula of MAPE is shown in 10.

$$MAPE = \frac{1}{n} \sum_{i=1}^{N} \frac{|f_i - y_i|}{y_i} \tag{9}$$

Where fi and yi represent the predicted values and the actual values, respectively. The smaller the MAPE value of glowworm i, the better the parameter combination. As mentioned above, after multiple iterations, the algorithm will converge to a point, the glowworm with the largest fluorescein, that is, the position of the glowworm is the optimal value of the LSTM load prediction model. The pseudo-code of the GSO-based parameter optimization algorithm is shown in the algorithm 1.

Algorithm 1: The GSO-based parameter optimization algorithm

```
  Input: The range of the model parameter combination and
the parameter value of the GSO algorithm.
  Output: The optimal value of model parameter combina-
tion.
  Initialize GSO;
  for k = 1; k ≤ iterationNum do
  Clustering (GSO);
  for i = 1; i ≤ N do
    li =updateLuci(); //Update fluorescein  according to
formula 6.
```

```
    Ni = findNeighbor (); //Looking for the neighbors of
glowworm i.
    j=maxProbably(i); //Select a transfer neighbor based
on the transition probability.
    UpdateLocation(i); //Update the location of glowworm
i.
    UpdateLuci(i);//Update the fluorescein value of glow-
worm i.
  end
  if MAPE is satisfied then
  break;
  end
  end
```

3 The GSO-LSTM Based Host Load Prediction Model

As mentioned in the last section, LSTM has a strong adaptive advantage in the prediction of time series models. It is combined with the benefits of GSO in parameter optimization, which can achieve an excellent load forecasting model for cloud computing and improve the accuracy of load sequences regression analysis.

3.1 Mobile cloud host load analysis

The key load types for mobile cloud data centers are CPU, memory, network bandwidth, and disk. On the one hand, since the demander in the mobile cloud computing system are the mobile terminals, the load of the mobile cloud computing data center has certain randomness and uncertainty. On the other hand, due to mobile users are more affected by time factors in the use of cloud computing services, the host load of the data center has a certain regularity, which can analyze and predict future host load through the mining of the historical data. Assuming that the historical load sequence H of the mobile cloud computing data center is a d dimension vector that representing the load from moment $t - d$ to moment $t - 1$. The input vector dimension d is determined by the autocorrelation analysis of the cloud's historical load. Autocorrelation is a characteristic of data that shows the degree of similarity between the values of the same variables at different points in time. The autocorrelation's strengths is an essential parameter of the LSTM iteration unit in Figure 1. We performed an autocorrelation analysis of the data center load and output the autocorrelation of the cloud host memory load, as shown in Figure 2.

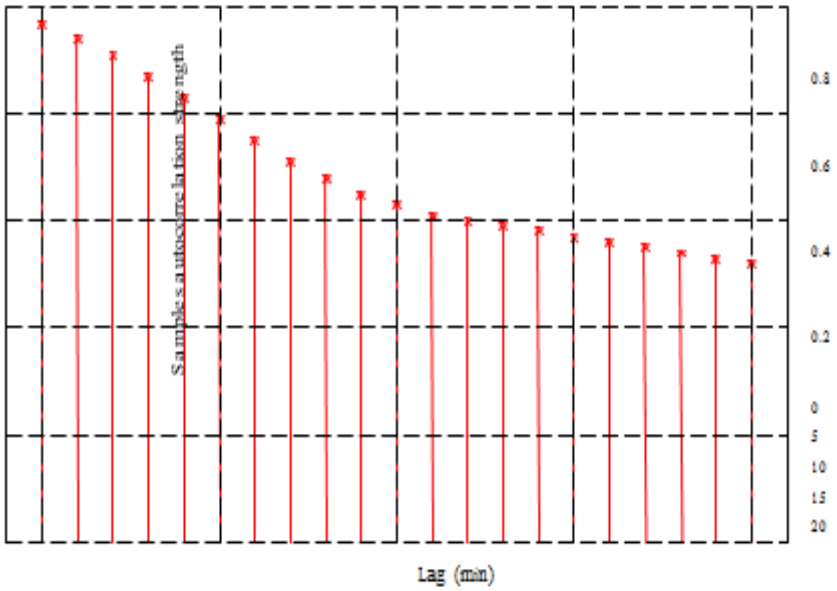

Fig. 2. The auto correlation of cloud host memory

As shown in Figure 2, the correlation of the host memory load data is linear. Setting an autocorrelation threshold of r, the dimension d of the input variable can be derived from the autocorrelation analysis results. For example, assuming r = 0.6, The input vector dimensions should be d =13.Therefore, the load prediction model of this paper selects the historical series of loadHt as the input of the GSO- LSTM model and calculates the output through the hidden layer and output layer of the GSO-LSTM to obtain the predicted value Lt of current moment load. Thereby achieving the purpose of cloud host load forecasting.

3.2 The host load prediction algorithm based on the GSO-LSTM model

For the cloud host load prediction problem, this paper proposes a host load prediction algorithm based on the GSO- LSTM model. The flowchart is shown in Figure 3. The algorithm is mainly divided into three stages, namely, data preprocessing, model training, and load prediction. First, in the data preprocessing step, normalizing the host load data of the mobile cloud data center with the logical reduction function. The logical reduction function is shown as follow:

$$L' = \frac{1}{1 + e^{-L}} \tag{10}$$

Where L′ is the normalized input load. Second, in the model training phase, the proposed LSTM model is trained using the host load data of the data center. After the host load data is normalized, inputting these sample data into the LSTM model to

optimizing the parameters of the LSTM model. To improve the optimizing efficiency for the LSTM model, the parameter optimization model GSO- LSTM based on the GSO algorithm is used to find the optimal parameter combination (batch size, ε, epochs, hidden size) of LSTM model. Third, in the load prediction phase, the load sequence consisting of the host load values of the last d intervals of the current moment t is used as the input of the GSO-LSTM network model, calculating the host load value Lt of the present moment t. The algorithm 2 gives the pseudo-code for the GSO-LSTM host load prediction model.

Algorithm 2: The host load prediction algorithm based on the GSO-LSTM model.

```
Input: The load of the cloud data center host, historical load vector at the current moment.
Output: The prediction value of the cloud host load at the current moment.
1. Dataset ds = new Dataset (); //Input the load of the cloud data center host.
2. ds. Normalization (); //Normalization.
3. ds. autocor Analy (d); //Autocorrelation analysis and calculate the input vector dimension d.
4. initialize (); //Initialize parameters in the GSO algorithm.
5. Use Algorithm 1 to Solve Parameter Combination of GSO-LSTM Model;
6. for i = 1; i ≤ N do
7. g LSTM. train (ds. get (i)); //Using Sample Data to Train GSO-LSTM Model.
8. end
9. Hash Mapload Prediction;
10. while N < i < max do
11. Li = predict (); //Calculation of Load Prediction Value.
12. load Prediction.add (Li);
13. i++;
14. end
```

4 Experiments and Results

The host load data set of the cloud computing data center published by Google was selected for comparison experiments to check the model and algorithm proposed in this paper. The Google Cloud Dataset records various load data, including CPU, memory, and disk space for each server in the data center. The dataset version is an updated version in 2014 with a dataset size of 60G. To facilitate the evaluation of the experimental error, the experiment was evaluated using the Mean Absolute Error MAE and the Mean Absolute Percentage Error MAPE.

4.1 Experimental setup

Figure 1 shows the autocorrelation analysis of cloud computing host memory usage. The correlation between different samples is considerable. Therefore, we select the host memory load data for verification. We refer to Ref. [21], and the range of values for the parameters of the LSTM model is shown in Table 1.

Table 1. The parameter settings for the LST Model

Parameters	Ranges
The size of batch processing batch size	1~100
Learning rate ε	0.001~0.1
Number of iterations epochs	500~1000
The number of hidden units hidden size	10~30

We optimize the LSTM model parameters with the GSO network; the settings of GSO need to be assigned. The parameter settings for the optimization algorithm are shown in Table 2, which is similar to [22].To more accurately quantify the prediction performance of the GSO-LSTM algorithm and the prediction model, we choose the Support Vector Regression algorithm SVR [10] and the Autoregressive Integrated Moving Average model ARIMA [23] as the benchmark for a comparison experiment. The kernel function of the SVR algorithm selects the most commonly used Gaussian kernel function, and other parameters are the same as the settings in this paper. The ARIMA algorithm performs once difference to the load sequence, which is setting the times of difference calculation for time series as 1. According to the autocorrelation analysis of Figure 1, setting the number of autoregressive terms is $p = 3$, the size of the moving average window is $q = 2$, that is the parameters of the ARIMA algorithm is set to (3, 1, 2)

Table 2. The parameter settings for the optimization algorithm

Parameters	Ranges
Fluorescein decay constant ρ	0.4
Fluorescein enhancement constant λ	0.6
Refreshing Rate of Local Decision Domain β	0.08
Moving step s	0.03
Parameters controlling the number of neighbors nt	5
Glowworm population scale	20
Number of iterations	300
Fluorescein radiation range rs	5

4.2 Evaluation

We choose the SVR and the ARIMA algorithm to compare with the proposed scheme, in which the kernel function of SVR uses the Gaussian kernel function. Figure 4 is a comparison of the predicted results of the three algorithms with the actual

load data. As shown in Figure 4, the prediction results of the GSO-LSTM prediction algorithm proposed in this paper are closer to the actual data.

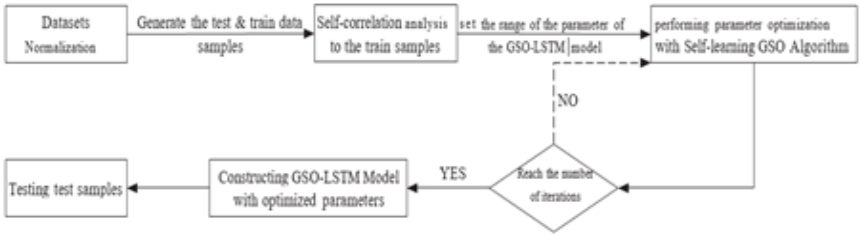

Fig. 3. The flowchart of the host load prediction algorithm based on GSO-LSTM model

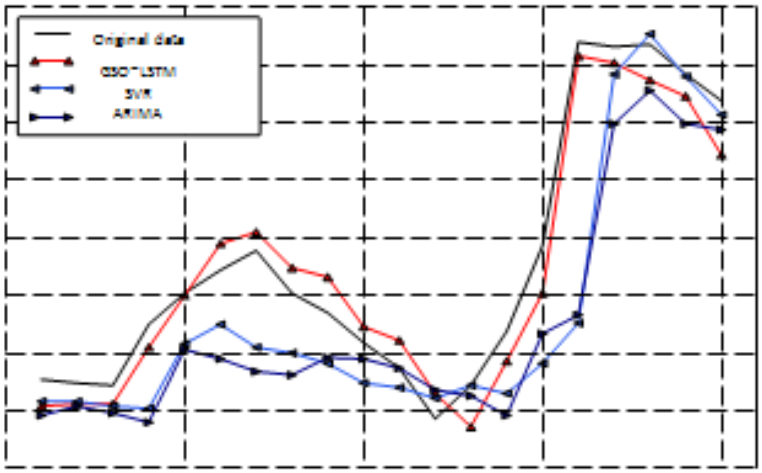

Fig. 4. The comparison of SVR, ARIMA, and GSO-LSTM

This is because the GSO parameter optimization algorithm can avoid the particles falling into local optimum and obtain better optimization results; that is, the GSO-LSTM converges faster. In contrast, the prediction error of the SVR and the ARIMA algorithm is relatively large, and there is a certain hysteresis. The prediction result of the SVR algorithm is slightly better than the ARIMA algorithm. Error is an important indicator to evaluate the pros and cons of a prediction algorithm. In this paper, the MAE and MAPE are selected as evaluation indicators for assessing the GSO-LSTM algorithm. Figure 4 shows the comparison of the two evaluation indicators for the SVR, the ARIMA, and the GSO-LSTM. As shown in Figure 5, the prediction algorithm GSO-LSTM proposed in this paper is lower than the SVR and ARIMA algorithms for both of the MAE and MAPE. It means that the proposed model and algorithm for host load prediction performance is better than the similar prediction algorithm.

5 Conclusion

Aiming at the host load prediction problem in the mobile cloud computing environment, we proposed a GSO-LSTM based host load prediction model, which uses the advantages of LSTM in processing long-term sequence prediction problems and optimizes the LSTM model parameters with adaptive learning parameter optimization algorithm GSO. Finally, the model and algorithm proposed in this paper are verified by using the host load data of Google Cloud Data Center. At the same time, two classic prediction algorithms, the SVR algorithm, and the ARIMA algorithm are compared with the GSO-LSTM that presented in this paper. The experimental results show that the prediction errors of the GSO-LSTM are lower than the other two algorithms, and the accuracy is better.

6 References

[1] K.-m. kim, H. Kim and K. Kim, "Design of an Intrusion Detection System for Unknown-attacks based on Bio-inspired Algorithms," in Computer Security Symposium 2015.

[2] T. Li and N.-f. Xiao, "Novel heuristic dual-ant clustering algorithm for network intrusion outlier's detection," Optic - International Journal for Light and Electron Optics, vol. 126, no. 4, pp. 494-497, February 2015. https://doi.org/10.1016/j.ijleo.2014.08.036

[3] I. Butun, S. Morgera and R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 266-282, 17 May 2013. https://doi.org/10.1109/surv.2013.050113.00191

[4] V. T. Alaparthy and S. Morgera, "Modelling an Intrusion Detection System based on Adaptive Immunology," in Wireless Telecommunications Symposium, Chandler, 2018.

[5] P. Pongle and G. Chavan, "Real time intrusion and wormhole attack detection in internet of things," vol. 121, no. 9, 2015. https://doi.org/10.5120/21565-4589

[6] M. Zeeshan, H. Javed and S. Ullah, "Discrete R-Contiguos Bit Matching Mechanism Appropriateness for Anomaly Detection in Wireless Sensor Networks," International Journal of Communication Networks and Information Security, vol. 9, no. 2, August 2017.

[7] Ahamed, A., C. Eswaran, and R. Kannan. "Lossy image compression based on vector quantization using artificial bee colony and genetic algorithms." Advanced Science Letters 24, no. 2 (2018): 1134-1137. https://doi.org/10.1166/asl.2018.10702

[8] P. Sherubha, N. Mohanasundaram," An Efficient Intrusion Detection and Authentication Mechanism for Detecting Clone Attack in Wireless Sensor Networks', Jour of Adv Research in Dynamical & Control Systems, Vol. 11, No. 5, 2019.

[9] P. Sherubha, "An Efficient Network Threat Detection and Classification Method using ANP-MVPS Algorithm in Wireless Sensor Networks", International Journal of Innovative Technology and Exploring Engineering, 2019. https://doi.org/10.35940/ijitee.k3958.0981119

[10] P. Deshpande, S. Sharma, S. Peddoju, and S. Junaid, "HIDS: A host-based intrusion detection system for cloud computing environment," IJSAEM, pp. 1–10, 2014.

[11] J. B. Hong and D. S. Kim, "Assessing the effectiveness of moving target defenses using security models," IEEE Transactions on Dependable and Secure Computing, vol. 13, no. 2, pp. 163–177, 2016. https://doi.org/10.1109/tdsc.2015.2443790
[12] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Optimal load distribution for the detection of VM-based DDoS attacks in the cloud," IEEE Transactions on Services Computing, 2017. https://doi.org/10.1109/tsc.2017.2694426
[13] IEEE Transactions on Services Computing, 2017.
[14] W. Peng, F. Li, C.-T. Huang, and X. Zou, "A moving-target defense strategy for cloud-based services with heterogeneous and dynamic attack surfaces," in IEEE ICC, 2014, pp. 804–809. https://doi.org/10.1109/icc.2014.6883418
[15] M. Idhammad, K. Afdel, M. Belouch, "Semi-supervised machine learning approach for DDoS detection," Applied Intelligence, no. 3,pp. 1-16, 2018. https://doi.org/10.1007/s10489-018-1141-2
[16] Z.Tüske,R.Schlüter,and H.Ney,"Investigationon lstmrecurrentn-gram language models forspeech recognition.,"in*Interspeech*,pp. 3358–3362,2018.
[17] R. Kyusakov, J. Eliasson, J. Delsing, J. van Deventer, and J. Gustafsson, "Integration of wireless sensor and actuator nodes with IT infrastructure using service-oriented architecture," IEEE Trans. on Ind. Informat., vol. 9, no. 1, pp. 43-51, Feb. 2013. https://doi.org/10.1109/tii.2012.2198655
[18] N. Katenka, E. Levina, and G. Michailidis, "Local vote decision fusion for target detection in wireless sensor networks," IEEE Trans. on Signal Process., vol. 56, no. 1, pp. 329–338, Jan. 2008. https://doi.org/10.1109/tsp.2007.900165
[19] G.Y. Keung, B. Li, and Q. Zhang, "The intrusion detection in mobile sensor network," IEEE/ACM Trans. on Networking, vol. 20, no. 4, pp. 1152-1161, Aug. 2012. https://doi.org/10.1109/tnet.2012.2186151
[20] Z. Shen, Y. Chang, H. Jiang, Y. Wang, and Z. Yan, "A Generic Framework for Optimal Mobile Sensor Redeployment," IEEE Trans.On Veh.Technol., vol.59, no.8, pp. 4043-4057, Oct. 2010. https://doi.org/10.1109/tvt.2010.2062203
[21] S. J. Li, and H. Shen, "Minimizing the Maximum Sensor Movement for Barrier Coverage in the Plane," in Proc. IEEE INFOCOM, Apr. 26-May. 1 2015, pp.244-252 https://doi.org/10.1109/infocom.2015.7218388
[22] M. Rout, and R. Roy. "Self-Deployment of Randomly Scattered Mobile Sensors to Achieve Barrier Coverage." IEEE Sens. J., vol.16, no.18, pp. 6819-6820, Sep. 2016. https://doi.org/10.1109/jsen.2016.2590572
[23] Z. B. Wang, J. L. Liao, Q. Cao, H. R. Qi, and Z. Wang, "Achieving k-Barrier Coverage in Hybrid Directional Sensor Networks," IEEE Trans.on Mobile Comput., vol. 13, no. 7, pp.1443-1455, Jul. 2014. https://doi.org/10.1109/tmc.2013.118
[24] Ahamed, A. Mohamed Uvaze, C. Eswaran, and R. Kannan. "CBIR system based on prediction errors." J. Inf. Sci. Eng 33, no. 2 (2017): 347-365.

7 Authors

P. Sudhakaran received his B.E. degree specialized in Computer Science and Engineering from Bharathidasan University, Tiruchirappalli, M.E. degree specialized in the field of Computer Science and Engineering from Anna University, Chennai, Tamilnadu, India, M.B.A degree specialized in the field of Information Technology from Sikkim Manipal University, Karnataka, India. Then he received his Doctoral - Ph.D. in Computer Science Engineering in the year 2016 from Anna University,

Chennai, India. He has nearly 19 years of experience in Teaching both UG and PG program. He is presently working as a professor in the department of Computer Science and Engineering, at SRM TRP Engineering College in Tiruchirappalli, Tamilnadu, India. He has published over more than 10 Research papers in different International Journals and Conferences, more in the area of opinion mining and data mining, Information Retrieval, Web mining.

Subbiah Swaminathan currently working as a professor, department of Computer science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai. He Completed doctorate in computer science and engineering in the area of cloud computing security issues. Previously worked as principal in various affiliated engineering colleges. His area of research includes cloud computing, data security, computer network and data mining. He is a life member of ISTE and CSI. subbussp2007@gmail.com

D. Yuvaraj have completed his B E (CSE) at J.J.College of Engineering and Technology, Bharathidasan University, Trichy, Tamilnadu, India. He received his M.Tech (CS&IT) degree from Manonmanium Sundaranar University, Tirunelveli, Tamilnadu in 2004. He has awarded Ph.D degree in Information and Communication Engineering (computer science and engineering) from Anna university, Chennai in 2017. He has nearly 19 years of experience in Teaching both UG and PG program. He is presently working as an Assistant professor in Department of computer science, Cihan university – Duhok, Kurdistan Region, Iraq. His field of interest is in image processing, Data Mining, Image retrieval, Information retrieval and Artificial intelligence. He has published more than 40 papers in National journals and international journals. yuva.r.d@gmail.com

S. Shanmuga Priya is working as an Asst. Professor in Department of Computer Science, M.I.E.T Engineering College, Trichy, India. She is currently pursuing her research in Anna University, Chennai, India. She has completed her undergraduate degree in Information Technology from Bharathidasan University Trichy in 2004, and completed her Masters in Information Technology from Sathyabama University Chennai in 2006. She has awarded Ph.D degree in Information and Communication Engineering (computer science and engineering) from Anna university, Chennai in 2019.She has 14 years of teaching experience. Her areas of interest include Mobile computing, Wireless Networks and Cloud Computing. She has a number of research papers in the field of Cloud Computing, Security and image processing. shanmugapriya.s@miet.edu

Article submitted 2020-01-14. Resubmitted 2020-02-23. Final acceptance 2020-02-23. Final version published as submitted by the authors.