Journal on Big Data
DOI:10.32604/jbd.2020.010958
Article

Tech Science Press

Workload Allocation Based on User Mobility in Mobile Edge Computing

Tengfei Yang[1,2], Xiaojun Shi[3], Yangyang Li[1,*], Binbin Huang[4], Haiyong Xie[1,5] and Yanting Shen[4]

[1]National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (NEL-PSRPC), Beijing, China
[2]National Computer Network Emergency Response Technical Team Coordination Center of China, Beijing, China
[3]Department of Science and Technology, China Electronics Technology Group Corporation, Beijing, China
[4]School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China
[5]University of Science and Technology of China, Hefei, China
[*]Corresponding Author: Yangyang Li. Email: liyangyang@live.com

Received: 10 April 2020; Accepted: 31 August 2020

Abstract: Mobile Edge Computing (MEC) has become the most possible network architecture to realize the vision of interconnection of all things. By offloading compute-intensive or latency-sensitive applications to nearby small cell base stations (sBSs), the execution latency and device power consumption can be reduced on resource-constrained mobile devices. However, computation delay of Mobile Edge Network (MEN) tasks are neglected while the unloading decision-making is studied in depth. In this paper, we propose a workload allocation scheme which combines the task allocation optimization of mobile edge network with the actual user behavior activities to predict the task allocation of single user. We obtain the next possible location through the user's past location information, and receive the next access server according to the grid matrix. Furthermore, the next time task sequence is calculated on the base of the historical time task sequence, and the server is chosen to preload the task. In the experiments, the results demonstrate a high accuracy of our proposed model.

Keywords: Edge computing; workload allocation; social-LSTM; context-sensitive; user mobility

1 Introduction

With the development of cloud computing, mobile terminals have been out of the shortcomings of large size and slow speed. However, with the rapid growth of mobile applications, such as real-time online games, image/video processing applications, vehicle network systems, etc., the network remain a heavy burden. These applications usually require the center of cloud computing to respond in real time, which could introduce a large amount of traffic and computational workload and result in high delay. Therefore, the concept of mobile edge computing is formally proposed to reduce network congestion by processing related tasks on cellular base stations, which are closer to the end user for rapid deployment of applications and other customer services. There are many existing works conducted to study the issue of energy optimization and reduced latency. Zhang [1] proposed a multi-device computing unloading framework, and developed a three-stage unloading scheme. Commonly, a task offload scheduling scheme based on the time-varying channel state of wireless offload is proposed in [2], which attempts to maximize the use of wireless channels and user buffers to reduce energy consumption. Liu et al. [3] used a one-dimensional search algorithm to solve the unloading decision according to the application buffer waiting queue state to minimize the single user execution delay. Mao optimized the task offload scheduling and transmission power allocation problem jointly, which reduced the offload delay [1]. And then Mao also proposed a single-user dynamic offloading scheme to minimize the execution delay of the energy harvesting device. In [4], a lightweight approximation method is proposed, which achieves a good balance between complexity and delay minimization. According to the recent activities of the experimental objects, a hybrid prediction model is proposed to predict the next position [5]. In [6], the long sequence of LSTM is proposed to predict the services to be used in the next phase. Awassizadeh [7] predicts the user's activity in a small area by training the user's behavioral trajectory.

In this paper, with the base station allocation problem for a wide range of mobile edge networks is narrowed down to single users, we propose a service prediction method based on user behavior and attribute it to two prediction models: 1) User geographic location prediction model; 2) User application prediction model. The models are treated as natural language processing problems by a context-aware method. By applying Social-LSTM method to derive the matrix weight values of user's behavior, we obtain the next possible location based on user's past location information and the next server access time based on the grid matrix. According to the historical time task sequence, the next task sequence is obtained, which is selected to preload the task. Extensive simulations are conducted to evaluate the efficiency of the presented scheme, and the results shows our proposed models have a high accuracy.

2 Context Analysis Based on User Behavior

2.1 Preliminaries

2.1.1 Problem Definition

Mobile edge network consists of multiple sBSs, covering specific areas such as airports, shopping malls, libraries, stadiums, etc. These sBSs combine into a real-time compute and storage services. While an sBS is processing the certain real-time tasks, mobile users may move in the MEN from one sBS to another. In this case, it is necessary to assign the task to the most appropriate sBS. With the movement of a user, as the server switches, the third-party service needs to be reloaded. Normally, it will cause a non-negligible delay. Therefore, it is feasible to optimize the MEC delay by task preloading.

In short, the purpose of this paper is to get the next possible location information through the user's past location information and to derive the server accessed at the next moment according to the grid matrix. Obtain the task sequence for the next time period according to the historical time task sequence and select the obtained server for task preloading to optimize the task execution delay of MEC.

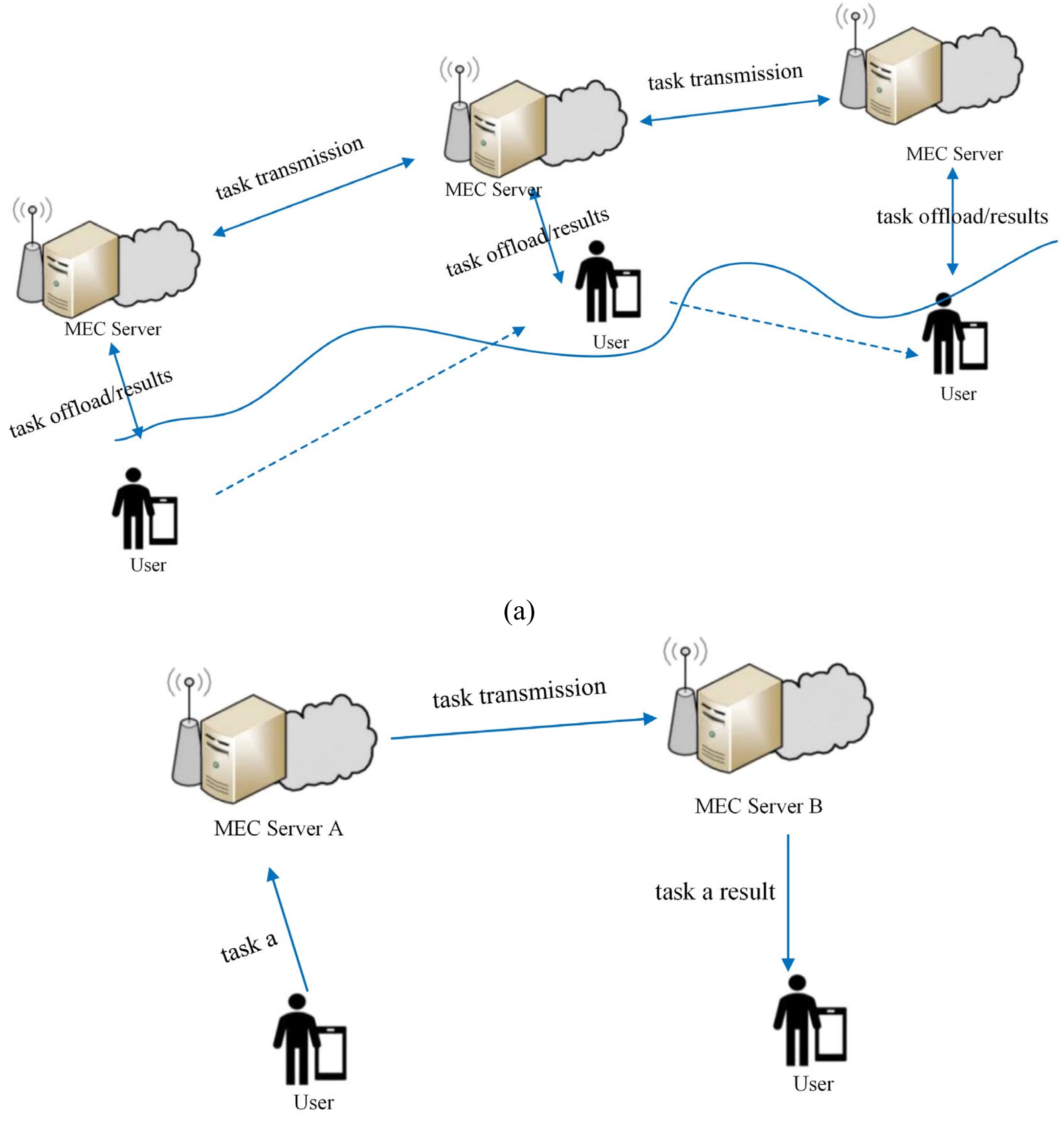
2.1.2 System Model

As illustrated in Fig. 1(a), we consider a task allocation system with a single user and multiple MEC servers in this paper. We assume that all of the task requests of the user are handled by edge servers with sufficient computing power. So we can see that, there are a certain number of edge servers on the movement path of a user, which have a fixed and known location and a limited range of service. Users always choose the nearest edge server for task uploading and downloading for accepting results. Moreover, the tasks offloading to the edge servers are independent from others, and the execution order of the tasks is also irrelevant.

As is shown in Fig. 1(b), the user accesses edge A when uploading task a. At the next moment, the user moves to edge B, and the task has been processed. Then the result is returned to the user via edge B. Therefore, we focus on the user's geographic location changes to predict the application which will be generated in the next phase, so that we can deploy the services in the edge server where the task will be transferred in advance. After transferred, the task can be processed without additional waiting time. Furthermore, the task execution result will be returned to the user by the closet server, which reduces the potential loss caused by the transmission of task results.

2.2 User Behavior Analysis

With the development of the Internet and mobile terminals, more and more mobile applications request the geographic location rights to record the user's motion track. It is well known that people usually have certain regularities in the daily activities, e.g., students always move between classrooms-cafeterias-dormitories, and office workers always appear near the company-a transit station-home. As the users are always active in a certain area, so we can take advantage of such features to analyze the next geographical location he may go.



(a)

(b)

Figure 1: (a) A task allocation system with a single user and multiple MEC servers, and (b) Task assignment of a single user

In reality, each user is a unique individual and will have his own habits. People do not use an application only in a certain geographic location, there is no connection between geographic location and the application. The repetitive behavior of using an app is more reflected in certain times of one day. In general, the application preference is obvious in one day, in other words, the user will use an application for a specific period of time. Therefore, with the task sequence divided by time T, we can predict the application which is most likely to be used in the next time period by the algorithm.

2.3 Context Prediction Based on User Behavior

When people interact with their surroundings, they often use simple information such as expressions and gestures. And by receiving such information, people can infer the meaning and react through the former. However, the computer cannot give a good conceptual model of context information. Researchers mostly define the context by enumeration. Schilit classified contexts as: computing context, user context, and physical context [8]. Xu et al. supplements the definition of context in mainly four aspects: computation, user, physics, and historical context [9–11].

And the context can refer to any association between entities, e.g., the influence of temperature and humidity on people, the general direction of the action track, the eyes, gestures and other related information [12,13]. So it can be seen that most of the cases can be attributed to contextual problems. For specific situations, we just need to establish an appropriate contact model for analysis [14,15].

According to the analysis in Section 2.2, the geographical location and the use of services have certain contextual characteristics. Thus, we can turn the task assignment problem into a context-sensitive issue, which selected in this paper is divided into two categories: 1) The geographical location of the user for one day; 2) The sequence of tasks used by the user for one day.

2.4 Social-LSTM

As a variant of RNN, LSTM is widely used in time-related prediction and sequence generation, such as speech recognition, scene analysis, etc. However, simple LSTM cannot predict the implicit relationship between different modes. Then Social-LSTM is proposed in [16], which a "social" collection layer is introduced based on traditional LSTM. The architecture connects LSTM corresponding to nearby sequences, which allows LSTM of the spatially near-end sequences to share their hidden state with each other. In each time step, LSTM unit receives the aggregated hidden state information from the neighboring LSTM unit to find the hidden mode in the track.

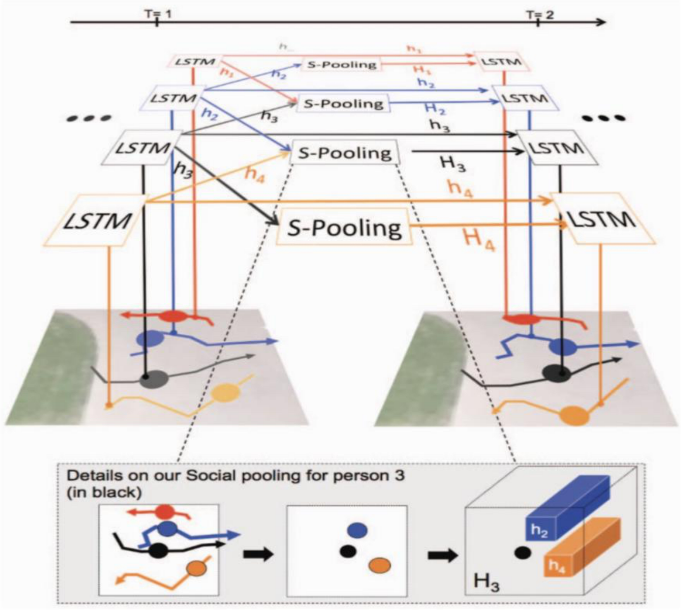


Figure 2: Social-LSTM structure diagram

3 Task Allocation Model

In this paper, the Social-LSTM is utilized to cascade LSTM and build a context time series model. Considering that there is no close relationship between the application and geographical location, the task assignment in MEC web is divided into two models: (1) Base station prediction model and (2) Task prediction model. Then, the two models are combined to get an accurate task assignment.

3.1 Server Settings

In this paper, we assume that the locations of all servers are fixed. Hence in the process of servers, we regard the upper and lower boundaries of the user's latitude and longitude as the user's active area. According to the fixed spacing, we divide it to into multiple square areas. And we treat the latitude and longitude of the center point of the square areas as the coordinates of the servers, the edge servers serve specific users at these points.

3.2 Data Definition

Data Definition of Base Station Prediction Model. Based on Section 3.1, we mapped the geographical trajectory of the user to the range of base station in this paper, and the behavior of base stations is predicted. The data is denoted as:

$$p = (time, sBS) \tag{1}$$

$$s_p^{u,d} = (p_1, p_2, p_3, \cdots, p_n) \tag{2}$$

where p is the base station that the user accesses at time t, and $s_p^{u,d}$ is all *(time, sBS)* sequences which belong to the user u on day d.

Data Definition of Task Prediction Model. In this paper, based on the time pattern of the tasks, we divide the application used by the user in one day into time slots. By default, the user uses only one application in a certain time slot. We predict the task according to the time slot sequences.

Let r^u be the application set which is used by the user u. And at a certain time t, within the range of duration T, we regard the application sequence appears in turn as $(app_i, app_j, app_n, app_m)$. Then we select the application with the longest application service time as the application uploaded. The data sequence of task prediction model can be expressed as:

$$a = (time, app) \tag{3}$$

$$s_a^{u,d} = (a_1, a_2, \cdots, a_k) \tag{4}$$

where a indicates the application record which is used at the time t after dividing by duration T, and $s_a^{u,d}$ is all the sequences *(time, task)* of user u on day d.

3.3 Embedding Layer

Word2vec is a neural network–based approach that comes in very handy in text mining analysis. So in this paper, the Word2Vec algorithm is applied to convert the input data into a vector representation, which is a text-word vector transformation method that takes context into account.

Embedding layer Of Base Station Prediction Model. We regard the access base station sequence s_p^u of the user u for several days as an article and consider the access base station sequence $s_p^{u,d}$ for one day as a long sentence. The base station accessed by the user at a certain moment is seen as a word. After the vectorization of the Word2Vec algorithm, the vectorized representation of the time and base station can be obtained respectively as:

$$\overrightarrow{time} = [t_1, t_2, \cdots, t_l] \tag{5}$$

$$\overrightarrow{sBS} = [s_1, s_2, \cdots, s_l] \tag{6}$$

By connecting the time, latitude and longitude vectors, the vector of the user u at a certain moment can be obtained as:

$$\vec{p} = \left[\overrightarrow{time}, \overrightarrow{sBS}\right] \tag{7}$$

Similarly, the access base sequence of the user *u* for a certain day can be expressed as:

$$s_p^{u,d} = (\overrightarrow{p_1}, \overrightarrow{p_2}, \overrightarrow{p_3}, \cdots, \overrightarrow{p_n}) \tag{8}$$

Embedding layer of Task Prediction Model. We regard the application record s_a^u of the user *u* for several days as an article and consider the application sequence $s_a^{u,d}$ for one day as a long sentence. The two-tuple of applications which used by the user *u* at a certain moment is seen as a word. After the vectorization of the Word2Vec algorithm, the vectorized representation of the time and task can be obtained respectively as:

$$\overrightarrow{time} = [t_1, t_2, \cdots, t_l] \tag{9}$$

$$\overrightarrow{app} = [app_1, app_2, \cdots, app_l] \tag{10}$$

By connecting the time, latitude and longitude vectors, the vector of the user *u* at a certain moment can be obtained as:

$$\vec{a} = \left\lfloor \overrightarrow{time}, \overrightarrow{app} \right\rfloor \tag{11}$$

Similarly, the access base sequence of the user *u* for a certain day can be expressed as:

$$s_a^{u,d} = (\overrightarrow{a_1}, \overrightarrow{a_2}, \overrightarrow{a_3}, \cdots, \overrightarrow{a_k}) \tag{12}$$

3.4 Social-LSTM Layer

In this paper, we focus on the single user. We assume that there is a certain factor relationship between the user's geographic trajectory and the used task sequence, and that there is a certain regularity at a certain time on the relationship. Therefore, we train the behaviors from the single user for several days.

The concept of step size is introduced in Social-LSTM, so that the best performance of the model can be obtained. In this paper, we set the step size to W, which means that the next vector is predicted through the input W consecutive vectors. Then we can get the input set of the training model for the user as [x, y], x and y represent the input and predicted output of the training model, respectively. Taking the base station prediction model of the user as an example, during the session of the training model of the user u on a certain day, the training input and output can be expressed as:

$$x = (\overrightarrow{p_l}, \overrightarrow{p_{l+1}}, \overrightarrow{p_{l+2}}, \cdots, \overrightarrow{p_{l+W-1}}) \tag{13}$$

$$y = \left\lfloor \overrightarrow{p_{l+W}} \right\rfloor \tag{14}$$

In the training process, the output y is obtained by the input x through the forward propagation network. And the optimal model is obtained by using the optimization function to adjust the training weight matrix through the back-propagation network by the difference between the predicted value and y.

We use the tanh and sigmoid functions as the activation function. And cross entropy is utilized as the loss function. The sigmoid function is used to avoid the problem of the reduction of the learning rate of mean-square error (MSE). Since the dimension of the word vector transformed by the Word2Vec algorithm is large, the Adadelta method is adopted to optimize the overall algorithm.

3.5 Data Integration

In this paper, the resource allocation problem of the user is divided into two prediction models.

User Location Prediction. During the process of predicting the geographical location, the user inputs a series of time, latitude and longitude information of a certain day. In the algorithm, the geographic location is mapped to the base station for model training, and the next time to be predicted is input. The base station information accessed by the next time is determined by the range of each base station divided by the user.

User task prediction. During the process of predicting a preload task of a user, it needs to input a combination of task sequences that appear in a series of time T in a certain day. By inputting a certain time T to be predicted, the task vector that appears at the beginning of a certain time T is obtained and converted into the corresponding task name through the vector.

Based on the base stations that will be accessed at a certain moment, we predict the services that may be loaded in the next period of time. In dealing with the relationship (base station, task), we will input the two models through time, so that we can get which service the base station needs to preload at a certain moment.

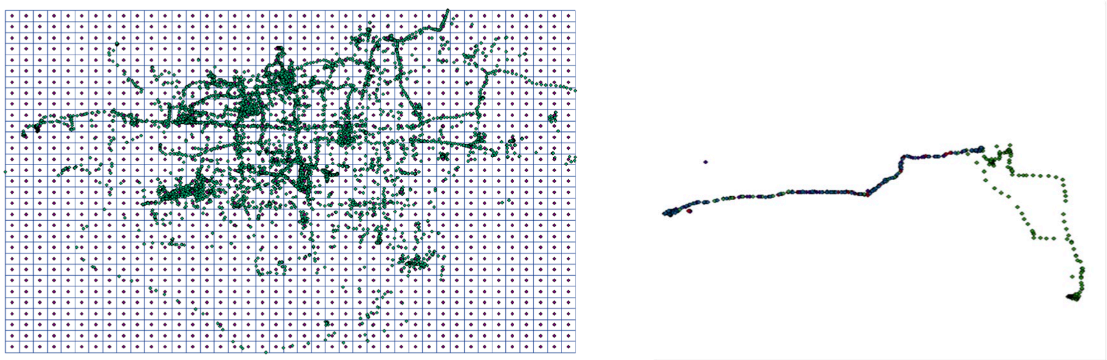
4 Experiment Results and Evaluation

4.1 Runtime Experiment

In order to prove the efficiency of our Social-LSTM model, our experiment ran in servers with 2*INTEL XEON E5-2630 V4 2.2GHz, 2*32G RECC DDR4/1600MHz to test, INTEL 500G SSD, and with 4x NVIDIA TITAN X 12G to train the model. Experiment codes are written by python3.5 in JetBrains PyCharm 2017.1.3.

4.2 Application Dataset

We use the UbiqLog [9,12], data set for experimental work. UbiqLog runs on a 35-user smartphone for about 2 months, collecting their calls, SMS headers, app usage, WiFi and Bluetooth devices nearby, geolocation, and Google Play API sports. The user location record of UbiqLog dataset and the activity track chart of user34 are shown in Fig. 3. In the experiment, a total of 5 user training data are selected as the model, which are user 5, 18, 19, 24, and 34.



(a) (b)

Figure 3: (a) User location record of UbiqLog dataset, and (b) Activity track chart

4.3 Experimental Evaluation

Since we predict the server that needs to be preloaded based on the user's behavior, the accuracy rate is selected as the evaluation index of the experiment:

$$Accuracy(x,y)=\frac{\sum_i y_i \subseteq x_i}{|x|},|x|=|y| \tag{15}$$

In the data verification, if the predicted data appears once in the total test, then it is a hit, and the hit ratio is the accuracy of predicted model.

4.4 Experimental Results

In this paper, Social-LSTM is used to train the model. When different LSTM numbers are selected, the average accuracy of the base station prediction model can be seen from the following Fig. 4. The trend is first increased and then decreased. Therefore, three LSTM cascades are selected to train the model in the experiment.

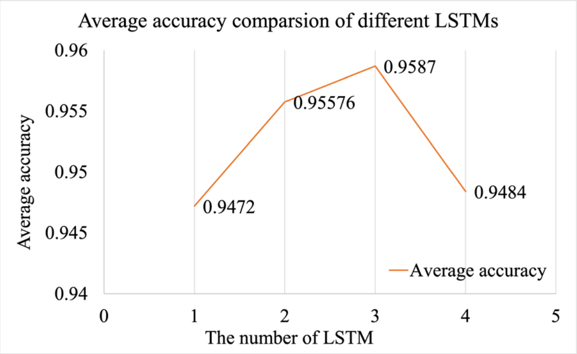
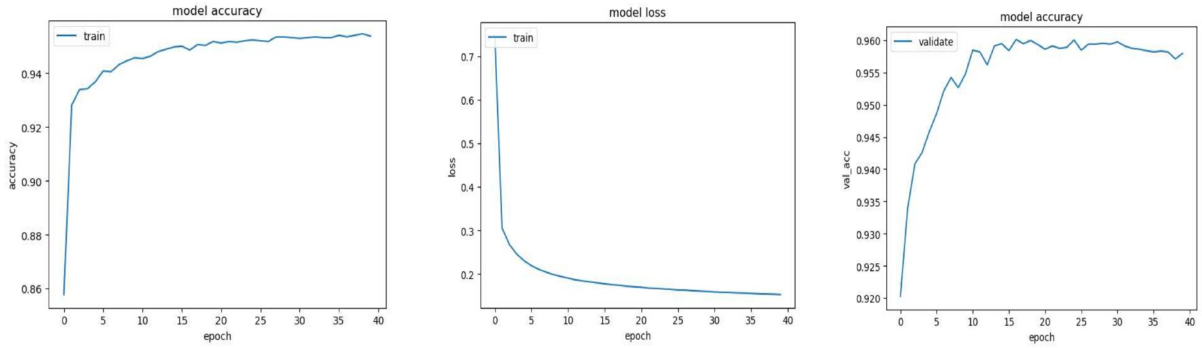
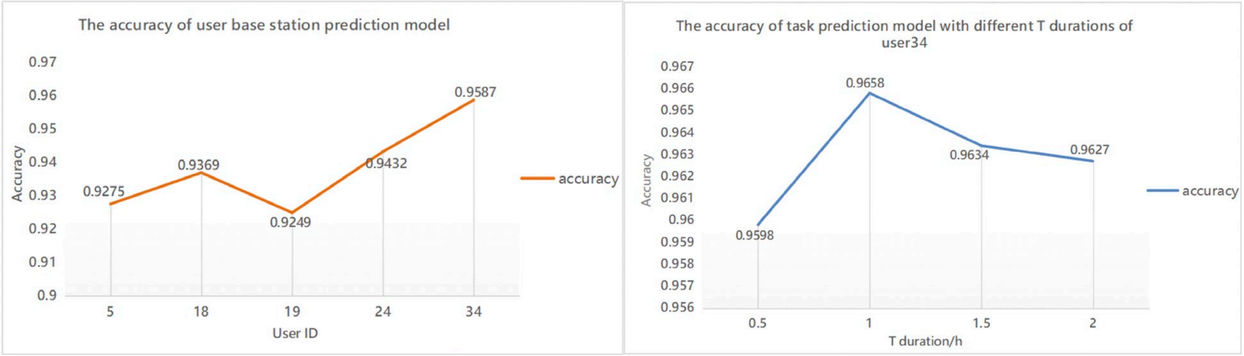


Figure 4: The comparison of average accuracy of different number of LSTM



(a) (b) (c)

Figure 5: a) The base station training accuracy of $user_{34}$, b) The base station loss function of $user_{34}$, and c) The accuracy curve for verification set of $user_{34}$



(a) (b)

Figure 6: (a) The accuracy of user base station prediction model, and (b) The accuracy of task prediction model with different T durations of $user_{34}$.

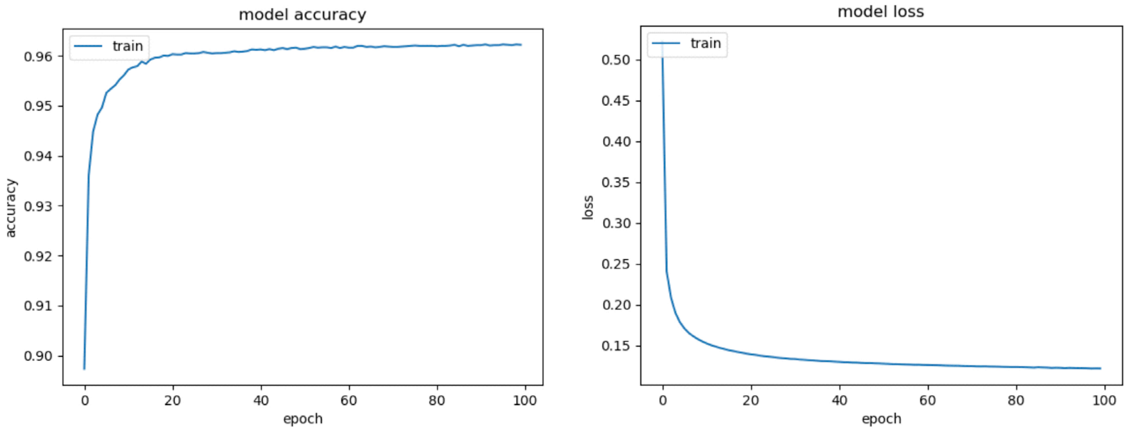
Taking the user 34 as the specimen, the experiment selects 80% of the data as the training set, 20% of the data as the verification set, and the empirical value parameter is selected for the experiment.

Among them, L2 regularization parameter is 0.0005, dropout parameter is 0.8, learning rate is 0.005, learning rate decay is 0.95, and batch size is 16. The model accuracy and loss are obtained as the following Figs. 5(a) and 5(b), respectively. It can be seen that as the epoch grows, the loss function value gradually decreases, and the training accuracy finally converges to 96%. Under this parameter condition, the accuracy curve of the test set is shown in Fig. 5(c).

The accuracy of the prediction model of the five user base stations selected in the experiment is as following Fig. 6(a). It can be seen that the prediction model of user base station shows a high precision.

In the task prediction model, the selection of the duration T is important. For different T, the experimental accuracy on $user_{34}$ is shown as Fig. 6(b). It can be seen that when the $user_{34}$ is selected, the accuracy of the task prediction model is the highest, reaching 96.58%. The increase of T will lead to the reduction of training data, so the regularity is not so strong. For each user, there will be a suitable duration, which can be interpreted as the time rule of the user using the application.

As above, the training accuracy and loss function of $user_{34}$ is given in Fig. 7.



(a) (b)

Figure 7: (a) Task training accuracy of $user_{34}$, (b) Task training loss function of $user_{34}$

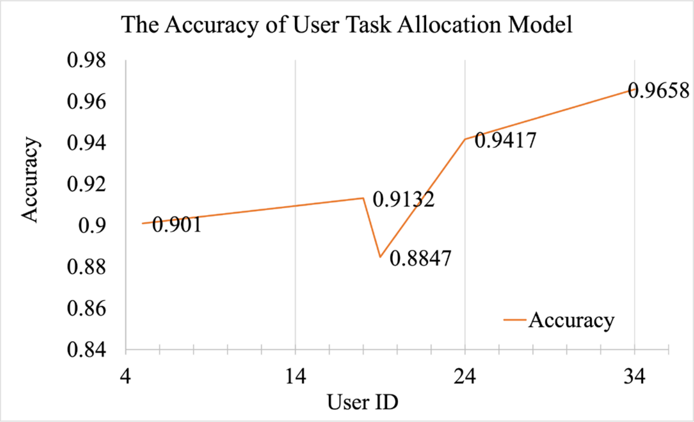


Figure 8: The accuracy of user task allocation model

For different users, the optimal division of T duration and data processing are performed separately, and the task assignment model of 5 users is obtained. The accuracy rate is shown as following Fig. 8.

As we can see, the accuracy of both models is above 90%. In the base station prediction model, if the noise data is rare, there is no way to identify regular activity in the overall training, which will cause a change of the accuracy of the model on the test set. In the task prediction model, we use a half-point idea to segment *T*. For users with strong regularity, such as $user_{34}$, the accuracy is high. In general, these two models are applicable to MEC and can provide a more reasonable solution for task assignment under MEC network.

5 Conclusion

This paper chooses the mobility of users in MEC as the focus of cut-in. Considering the uniqueness of the user, the user's behavior pattern is identified first. By predicting the base station and tasks at the next moment, the server is notified to perform task preloading to reduce the task transmission delay and improve the user experience then. Simulation results indicate that the prediction results show a high hit rate in the case of adopting appropriate parameters and steps, which also shows that the model provides a solution for user resource allocation in the mobile edge computing environment. However, the activity behavior of mobile users is usually changeable. It is impossible to know whether the model in this paper is applicable to all scenarios accurately. In the future, we plan to consider the impact of different influencing factors on the environment.

Funding Statement: This work is supported by the CETC Joint Advanced Research Foundation (No. 6141B08020101) and Major Special Science and Technology Project of Hainan Province (No. ZDKJ2019008).

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

References

[1] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li *et al.,* "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2017.

[2] W. Labidi, M. Sarkiss and M. Kamoun, "Joint multi-user resource scheduling and computation offloading in small cell networks," in *Proc. 2015 IEEE 11th Inter. Conf. on Wireless and Mobile Computing, Networking and Communications*, 2015.

[3] Y. Mao, J. Zhang and K. B. Letaief, "Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems," in *Proc. of the Wireless Communications & Networking Conf.*, 2017.

[4] J. Yang, B. Jiang, Z. Lv and K. K. Raymond Choo, "A task scheduling algorithm considering game theory designed for energy management in cloud computing," *Future Generation Computer Systems*, vol. 105, pp. 985–992, 2017.

[5] H. Jeung, Q. Liu, H. T. Shen and X. Zhou, "A hybrid prediction model for moving objects," in *Proc. IEEE Inter. Conf. on Data Engineering*, 2008.

[6] Z. Tong, "Research on marginal service loading optimization based on user behavior and location perception," Zhejiang University, 2017.

[7] R. Rawassizadeh, M. Tomitsch, K. Wac and A. M. Tjoa, "UbiqLog: A generic mobile phone-based life-log framework," *Personal & Ubiquitous Computing*, vol. 17, no. 4, pp. 621–637, 2013.

[8] B. Schilit, N. Adams and R. Want, "Context-aware computing applications," in *Proc. Workshop on Mobile Computing Systems & Applications*, 1994.

[9] X. U. Guang, "Pervasive/Ubiquitous computing," *Chinese Journal of Computers*, 2003.

[10] W. R. Shi, B. Zhou and X. U. Lei, "Pervasive computing: Human-centered computing," *Computer Applications*, 2005.

[11] S. Yu, J. Liu, X. Zhang, S. Wu, "Social-aware based secure relay selection in relay-assisted D2D communications," *Computers, Materials & Continua*, vol. 58, no. 2, pp. 505–516, 2019.

[12] A. K. Dey, "Understanding and using context," *Personal & Ubiquitous Computing*, vol. 5, no. 1, pp. 4-7, 2001.

[13] R. Li, “Research on several key technologies of context-aware computing,” Ph.D. dissertation, Hunan University, 2007.

[14] O. Zhang and X. Wei, “Online magnetic flux leakage detection system for sucker rod defects based on LabVIEW programming,” *Computers, Materials & Continua*, vol. 58, no. 2, pp. 529–544, 2019.

[15] Y. Zhao, X. Yang and R. Li, “Design of feedback shift register of against power analysis attack,” *Computers, Materials & Continua*, vol. 58, no. 2, pp. 517–527, 2019.

[16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.