# Content Popularity Prediction via Federated Learning in Cache-Enabled Wireless Networks

YAN Yuna[1], LIU Ying[2], NI Tao[2], LIN Wensheng[1], LI Lixin[1]

(1. Northwestern Polytechnical University, Xi'an 710072, China;
2. Shanghai Satellite Engineering Research Institute, Shanghai 200240, China)

DOI: 10.12142/ZTECOM.202302004

https://kns.cnki.net/kcms/detail/34.1294.TN.20230524.1802.004.html, published online May 26, 2023

Manuscript received: 2023-03-02

**Abstract:** With the rapid development of networks, users are increasingly seeking richer and high-quality content experience, and there is an urgent need to develop efficient content caching strategies to improve the content distribution efficiency of caching. Therefore, it will be an effective solution to combine content popularity prediction based on machine learning (ML) and content caching to enable the network to predict and analyze popular content. However, the data sets which contain users' private data cause the risk of privacy leakage. In this paper, to address this challenge, we propose a privacy-preserving algorithm based on federated learning (FL) and long short-term memory (LSTM), which is referred to as FL-LSTM, to predict content popularity. Simulation results demonstrate that the performance of the proposed algorithm is close to the centralized LSTM and better than other benchmark algorithms in terms of privacy protection. Meanwhile, the caching policy in this paper raises about 14.3% of the content hit rate.

**Keywords:** content popularity prediction; privacy protection; federated learning; long short-term memory

**Citation** (Format 1): YAN Y N, LIU Y, NI T, et al. Content popularity prediction via federated learning in cache-enabled wireless networks [J]. ZTE Communications, 2023, 21(2): 18 - 24. DOI: 10.12142/ZTECOM.202302004

**Citation** (Format 2): Y. N. Yan, Y. Liu, T. Ni, et al., "Content popularity prediction via federated learning in cache-enabled wireless networks," *ZTE Communications*, vol. 21, no. 2, pp. 18 - 24, Jun. 2023. doi: 10.12142/ZTECOM.202302004.

## 1 Introduction

Due to the explosive development of smart devices in networks, data traffic has increased unprecedentedly in recent years. With the limited communication resources, backhaul link congestion will occur in the peak period at times, which leads to poor quality of experience (QoE)[1]. Content caching is considered to be a promising solution to improving the QoE of users. For a traditional approach, almost all contents are placed on the cloud server. However, since a large number of popular files are easy to be repeatedly requested by users, the popular files can be cached in advance at local base stations (BSs), which not only guarantees the hit rate of content, but also is helpful to reduce the users' waiting time, alleviates the pressure on the core network and relieves traffic congestion[2 - 4].

In the past, traditional content-caching strategies, such as Least Recently Used (LRU)[5] and Least Frequently Used (LFU)[6], were used in the deployment phase. However, different users have different content preferences and these preferences are often time-varying, so the fixed content deployment cannot take full advantage of the network caching. Therefore, in order to further improve the performance of network caching, using machine learning (ML) to accurately predict popular files in the future attracts the interest of researchers during the deployment of caching content files[7]. WON and KIM[8] proposed a preference prediction neural network model based on DeepFM to predict the user's preference for movies, which improved the prediction accuracy by considering the interaction of low-order and high-order features of the input data. LI et al.[9] proposed proactive edge caching for device-to-device (D2D) assisted wireless networks. In this paper, the authors adopt bidirectional long short-term memory (LSTM) networks, graph convolutional networks and attention mechanisms to learn user preferences. JIANG et al.[10] proposed LSTM to predict the users' content request distribution, thereby achieving higher accuracy and better

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62001387, in part by the Young Elite Scientists Sponsorship Program by the China Association for Science and Technology (CAST) under Grant No. 2022QNRC001, and in part by Shanghai Academy of Spaceflight Technology (SAST) under Grant No. SAST2022052.

versatility.

The above methods can be classified as centralized ML, where the original training data sets need to be uploaded to the central server. However, the prediction of content popularity often involves personal information (e.g., home addresses, shopping, etc.) as training samples, which results in the risk of privacy leakage. As a privacy-preserving distributed learning framework, federated learning (FL) was proposed to tackle this challenge by training a global statistical model without accessing users' private raw data[11-12]. Recently, FL has been put forward to solve the challenging problems in wireless networks[13-14]. In particular, an FL-based approach was provided by FAROOQ et al.[15] to build a flood forecasting model. Specifically, the local training parameters are aggregated to build the global model. By transferring the training parameters instead of sending huge data sets, the leakage of the data privacy will be greatly decreased. WANG et al.[16] proposed an efficient content popularity prediction of privacy-preserving (CPPPP) scheme based on federated learning and Wasserstein generative adversarial network (WGAN), which achieves a high cache hit ratio. In this system, the server aggregated the users' updates using federated averaging, and each user performed training on its local data using WGAN, which could achieve high cache efficiency and protect the privacy of users. Therefore, considering privacy protection, FL can also be applied to the content popularity prediction in cache-enabled wireless networks.

Motivated by the above discussions, we propose a privacy-preserving algorithm for content popularity prediction named FL-LSTM, which combines LSTM with FL. Due to the unique design structure, LSTM is suitable for processing and predicting time series, such as content popularity prediction[10]. According to the aggregation mechanism of FL[11], the global content popularity prediction model will be built based on local training parameters. Thus, the FL-LSTM algorithm can inherently improve security performance and obtain reliable prediction performance.

The main contributions of this work are summarized as follows:

• We investigate a content popularity prediction problem in cache-enabled wireless networks, and aim at minimizing the mean-square error (MSE) and maximizing the cache hit rate. Considering the significance of privacy-preserving, a novel content popularity prediction algorithm FL-LSTM based on LSTM and FL is proposed. The algorithm avoids the direct transmissions of raw user data, which preserves user privacy.

• By utilizing a real-world dataset, the simulation results demonstrate that the proposed algorithm achieves similar performance to the centralized LSTM and better prediction ability than other state-of-the-art schemes.

The remainder of this paper is organized as follows. In Section 2, the communication system model and problem formulation are introduced. The privacy-preserving FL-LSTM algorithm is proposed in Section 3. In Section 4, the simulation results and experiment result analysis are shown. Finally, conclusions are drawn in Section 5.

## 2 System Model and Problem Formulation

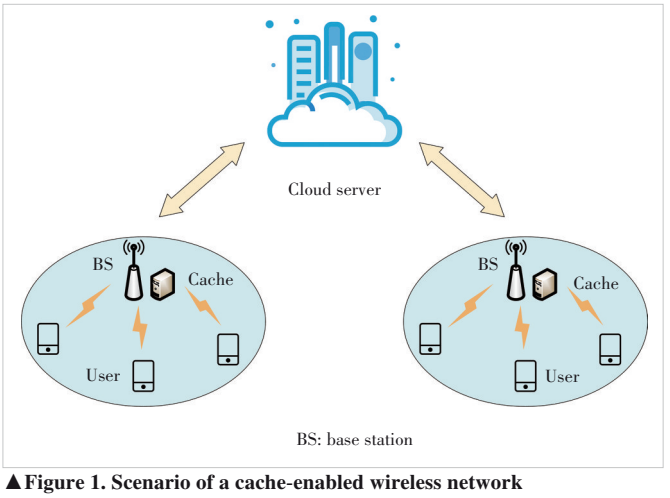
### 2.1 System Model

In this paper, we consider a cache-enabled wireless network as illustrated in Fig. 1, which consists of a cloud server, multiple BSs and the users served by each associated BS. There are $B$ BSs in this specific region and the set of BSs is denoted by $\mathcal{B}=\{1,2\cdots,b,\cdots,B\}$. The content library is denoted by $\mathcal{F}=\{1,2\cdots,f,\cdots,F\}$, where we assume the requested files have the same size. The cloud server $\mathcal{O}$ contains the whole content library and each BS can only store a limited number of files. To simplify the model, we assume that each BS is equipped with a cache $C_m$ of an equal size, where $C_m=n$ represents BS $b$ can only cache $n$ files from the cloud server.

It is assumed that the contents are requested and fetched during the discrete time periods and the set of time periods is expressed as $\mathcal{T}=\{1,2\cdots,t,\cdots,T\}$. In each time slot $t$: 1) According to users' previous request information, the local BSs and the cloud server jointly build the content popularity prediction model; 2) based on the prediction results, the related BS will cache the relevant contents from $\mathcal{O}$ in advance; 3) when the requested file is stored in the local BS, the associated users will directly obtain the requested contents; 4) otherwise, the requested file is fetched from the cloud server.

### 2.2 Problem Formulation

Based on the network discussed above, the basic framework and operation process of the communication system are introduced. It is obviously known that content popularity prediction is perceived as the key to the success of the system. For a specific file $f$, the popularity will change over time, and its popularity sequence is expressed as $\mathcal{P}_{b,f}=\left\{p_{b,f}^1,p_{b,f}^2,\cdots,p_{b,f}^t\right\}$, $p_{b,f}^t\in[0,1]$. Therefore, the popularity pre-




▲Figure 1. Scenario of a cache-enabled wireless network

diction of a file is transformed into a time series prediction problem, and the real and predicted values are expressed as $p_{b,f}^{t}$ and $\tilde{p}_{b,f}^{t}$, respectively. Moreover, the MSE is adopted to evaluate the accuracy of the prediction as follows:

$$\mathrm{MSE} = \frac{1}{T}\sum_{t=1}^{T}\left|\tilde{p}_{b,f}^{t} - p_{b,f}^{t}\right|^{2}. \tag{1}$$

In this paper, content popularity is defined as the ratio of the number of requests for a file to the number of requests for all files within a time slot. If a file is frequently requested by users, the higher popularity of the content is, the more likely it is to be accessed again in the next time slot. In time slot $t$, the content popularity of a file $f$ can be expressed as $q_{b,t}^{f} = \mathrm{req}_{t,f} \Big/ \sum_{i=1}^{f}\mathrm{req}_{t,f}$, where $\mathrm{req}_{t,f}$ represents the number of user requests for the file $f$ in a time slot $t$. Therefore, the popularity of the content library is denoted as $\mathcal{P}_{b,t} = \left\{p_{b,t}^{1}, p_{b,t}^{2}, \cdots, p_{b,t}^{f}\right\}$, $p_{b,t}^{f} \in [0,1]$ and $\sum_{f=1}^{F} p_{b,t}^{f} = 1$, where the order of files is in a descending order according to the popularity.

Due to the limited storage capacity of each BS $b$, after the prediction task is completed, each BS $b$ needs to sort the predicted popular files, select the contents that are more popular with users to cache in advance and replace the files with low popularity. If the contents are cached locally, this operation does not need to be repeated. For a certain discrete time slot $t$, the selected pre-cached files in local BS $b$are formulated as $\mathcal{A}_{b,c} = \left\{a_{b,c}^{1}, a_{b,c}^{2}, \cdots, a_{b,c}^{f}\right\}$, where $a_{b,c}^{f} = 1$ if the file $f$ is cached in BS $b$, otherwise $a_{b,c}^{f} = 0$ and $\sum_{f=1}^{F} a_{b,c}^{f} \leqslant n$.

Furthermore, when we measure the cache-enabled wireless network, the cache hit rate is considered as an important metric of caching performance. The hit rate of each BS during each time slot is defined as $h_b = \frac{1}{n}\sum_{f=1}^{F} a_{b,c}^{f} \times p_{b,t}^{f}$, referring to the probability that the precached file is popular content, which is used to represent the effectiveness of the content cache. Therefore, the hit rate of the network is averaged to be the total hit rate $\bar{h}$ during each training episode.

Due to the limited information collected by a single BS, combining several local BSs by the cloud server is necessary to obtain the whole popularity of the content library. However, it will also cause a privacy leakage issue to some extent. Last but not least, the objective of this paper is to predict the content popularity accurately and maximize the cache hit rate during each time slot while preserving users' privacy.

## 3 FL-LSTM for Content Popularity Prediction

### 3.1 Literature Overview

1) LSTM: LSTM was first proposed by HOCHREITER et al.[17]. Although the recurrent neural network (RNN) can be used to process and predict the sequence data, the processing and prediction effect of LSTM is better than that of the RNN as the time scale of the processing sequence increases, and the phenomena of "gradient vanishing" and "gradient explosion" of the RNN can be avoided through LSTM. Therefore, LSTM is selected as the benchmark algorithm to predict popular files. The illustration of the LSTM algorithm is shown in Fig. 2. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate, where the "gate" structure is an approach to data control. The formulas of the three gate structures are defined as follows. The forget gate $\boldsymbol{f}_t$ decides what kind of previous information will be forgotten, i.e.,

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_{fx} \cdot \boldsymbol{x}_t + \boldsymbol{W}_{fh} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right), \tag{2}$$

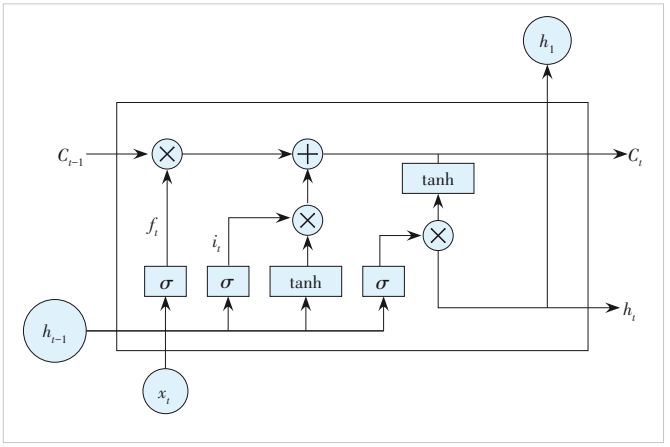
where $\boldsymbol{x}_t$ is the input vector of the current time and $\boldsymbol{h}_{t-1}$ is the hidden status of the previous time. $\boldsymbol{W}_{fx}$, $\boldsymbol{W}_{fh}$ and $\boldsymbol{b}_f$ are the input weight, recurrent weight and corresponding bias of the forget gate $\boldsymbol{f}_t$, respectively. The input gate $\boldsymbol{i}_t$ is used to select which information will be recorded, i.e.,

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_{ix} \cdot \boldsymbol{x}_t + \boldsymbol{W}_{ih} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right), \tag{3}$$

where $\boldsymbol{W}_{ix}$, $\boldsymbol{W}_{ih}$ and $\boldsymbol{b}_i$ are the input weight, recurrent weight and corresponding bias of the input gate $\boldsymbol{i}_t$, respectively. $\boldsymbol{x}_t$ and $\boldsymbol{h}_{t-1}$ are weighted to update the value of the input gate through the sigmoid function $\sigma(\cdot)$. The candidate memory cell state $\tilde{\boldsymbol{C}}_t$ is updated by Eq. (4), i.e.,

$$\tilde{\boldsymbol{C}}_t = \tanh\left(\boldsymbol{W}_{cx} \cdot \boldsymbol{x}_t + \boldsymbol{W}_{ch} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right), \tag{4}$$

where $\boldsymbol{W}_{cx}$, $\boldsymbol{W}_{ch}$ and $\boldsymbol{b}_c$ are the input weight, recurrent weight and corresponding bias of the candidate memory cell, respectively. And the tanh function $\tanh(\cdot)$ can control the range of its value to $[-1, 1]$. The new memory cell $\boldsymbol{C}_t$ controls the input and forget mechanism, which updates the state of the unit at




▲Figure 2. Illustration of long short-term memory (LSTM) algorithm

the previous time based on the output of the forget gate and the input gate, i.e.,

$$\boldsymbol{C}_t = \boldsymbol{f}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{C}}_t, \tag{5}$$

The output gate $\boldsymbol{o}_t$ decides what value should be output by Eqs. (4) – (7), and then the output of the hidden layer $\boldsymbol{h}_t$ is obtained by Eq. (7), i.e.,

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_{ox} \cdot \boldsymbol{x}_t + \boldsymbol{W}_{oh} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right), \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{C}_t\right), \tag{7}$$

where $\boldsymbol{W}_{ox}$, $\boldsymbol{W}_{oh}$ and $\boldsymbol{b}_o$ are the input weight, recurrent weight and corresponding bias of the output gate $\boldsymbol{o}_t$, respectively.

2) Fedavg: The Fedavg algorithm was proposed by KHAI et al.[18] and has several advantages in privacy protection and distributed training. In addition, users do not need to upload all the data, but only upload the local parameters needed by the model, which greatly reduces the overall communication overhead. In this algorithm, the cloud server starts FL training by sharing global model parameters with the base station. Then, each base station selects samples from the local data subset to perform the stochastic gradient descent (SGD) in order to update the local model and share the updated model weight with the cloud server. After that, the cloud server aggregates all the updated local model weight parameters and averages them to generate the global model. Compared with the centralized machine learning, the algorithm has some differences. The algorithm flow is presented as follows:

(a) At the beginning of training, the global model parameters $\boldsymbol{W}^o$ in the cloud server are initialized and then sent to the local BSs as $\boldsymbol{W}_t^b$.

(b) The BS $b$ trains the local dataset and updates $\boldsymbol{W}_t^b$ to $\boldsymbol{W}_{t+1}^b$ after the training epochs, i.e., $\boldsymbol{W}_{t+1}^b \leftarrow \text{LocalUpdate}\left(b, w_t^b\right)$.

(c) The cloud server aggregates each BS's $\boldsymbol{W}_{t+1}^b$ and then generates a new global model $\boldsymbol{W}_{t+1}^o$. This formula of aggregation can be described as:

$$\boldsymbol{W}_{t+1}^o = \frac{1}{B}\sum_{b=1}^{B} \boldsymbol{W}_{t+1}^b. \tag{8}$$

(d) Afterwards, the $\boldsymbol{W}_{t+1}^o$ will be broadcasted to all the BSs and the next round of training is started.

### 3.2 Proposed FL-LSTM Algorithm

Based on the aforementioned LSTM and Fedavg algorithms, we develop the FL-LSTM algorithm. The algorithm can predict the content popularity accurately while preserving the users' privacy. The illustration of the FL-LSTM algorithm is shown in Fig. 3. Then we will introduce the details of this algorithm.

Firstly, the initial LSTM network is adopted on the cloud server as the global model, and each BS will build the local LSTM network with the initial parameters $\boldsymbol{W}^o$ from the global model.

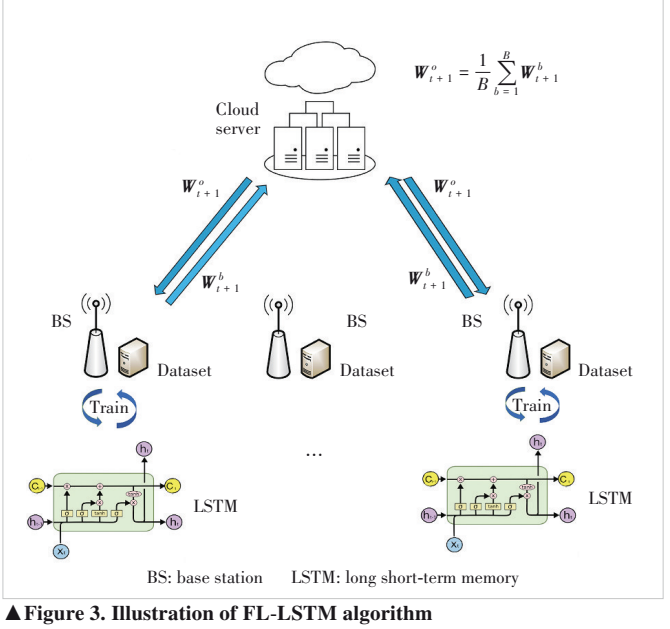
After that, to avoid sharing the raw data directly, the predicted model based on LSTM will be trained locally. The raw data $\mathcal{D}$ are divided uniformly and posted on each BS, with $d_b$ representing the local dataset of BS $b$. The previous $k$ moments are selected for prediction and the network can be characterized as follows: the input time series $\mathcal{X} = \{x_1, x_2, \cdots, x_t\}$ are defined as the historical popularity, through the vector $\mathcal{X}$ to predict the output vector $\mathcal{Y}$. The output time series $\mathcal{Y} = \{y_1, y_2, \cdots, y_t\}$ are defined as the predicted popularity, and $\mathcal{H} = \{h_1, h_2, \cdots, h_t\}$ means the information of the hidden layer.

The local weight parameters are collectively referred to as $\boldsymbol{W}_t^b$. The small BS adopts the SGD optimizer, and the weight parameters will be updated to $\boldsymbol{W}_{t+1}^b$ according to Eqs. (2) – (7).

According to the aggregation mechanism in Eq. (8), the cloud server aggregates the updated parameters $\boldsymbol{W}_{t+1}^b$ which are uploaded by all local BSs to generate a new global model. Subsequently, the parameters of the global model $\boldsymbol{W}_{t+1}^o$ are downloaded to each BS and then the next round of training will be started. Model training and parameter updating are repeated until the algorithm is terminated when the maximum number of iterations $T$ is reached. The objective of this algorithm is to minimize the MSE denoted in Eq. (1). Specifically, the proposed FL-LSTM algorithm is summarized in Algorithm 1.

**Algorithm 1.** Content popularity prediction based on FL-LSTM

1: Initialize the system: cloud server $\mathcal{O}$, local BSs $\mathcal{B}$.
2: Initialize the LSTM network by Eqs. (2) – (7): $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{H}$; the global weight $\boldsymbol{W}^o$; the local weight $\boldsymbol{W}^b$ and the local sampled batch size $k$.
3: Initialize the round index $t$ and the local training epoch index $n$.




▲Figure 3. Illustration of FL-LSTM algorithm

4: **for** round=1,⋯, $t$ ,⋯, $T$ **do**
5: At time-step $t$, the global model $\boldsymbol{W}_t^o$ is broadcasted to each BS as $\boldsymbol{W}_t^b$.
6: (For each BS $b$, start the local training.)
7: **for** epoch=1,⋯, $n$,⋯, $n_{\max}$ **do**
8: Samples a batch $\{x_i, \tilde{y}_i\}_{i=1}^k$ from $d_b$;
9: Update the local LSTM network by the loss functions in Eq. (1):
10: $\boldsymbol{W}_{n+1}^b \leftarrow \mathrm{SGD}\left(\nabla_\omega \frac{1}{k}\sum_{i=1}^{\kappa}(y_i - \tilde{y}_i)^2\right)$
11: **end for**
12: The cloud server aggregates each BS's $\boldsymbol{W}_{t+1}^b$ and updates $\boldsymbol{W}_t^o$ to $\boldsymbol{W}_{t+1}^o$ by Eq. (8).
13: $t \leftarrow t + 1$.
14: **end for**

## 4 Simulations and Discussions

### 4.1 Datasets

The MovieLens 1M Dataset[19] is used to evaluate the performance of the proposed FL-LSTM algorithm in this paper. The dataset contains 1 000 209 ratings of approximately 3 900 movies made by 6 040 users. Each sample in the data set includes a user ID, movie ID, user rating, and time stamp when commenting. We assume that the number of ratings by users can reflect the popularity of relevant movies. The process of dataset construction is as follows: Based on this assumption, we choose the ten movies with the most ratings as the prediction objects and divide one hundred discrete time slots based on the provided time stamp. Next, we count the number of ratings by users according to each time slot and consider it as the number of requests. Then, the request times are normalized to calculate the content popularity of the corresponding movie file. Finally, according to the content popularity of each film in one hundred discrete time slots, the LSTM model is used to predict the content popularity of each film in the next moment. By ranking the results, we can predict which movies will be popular at the next moment.

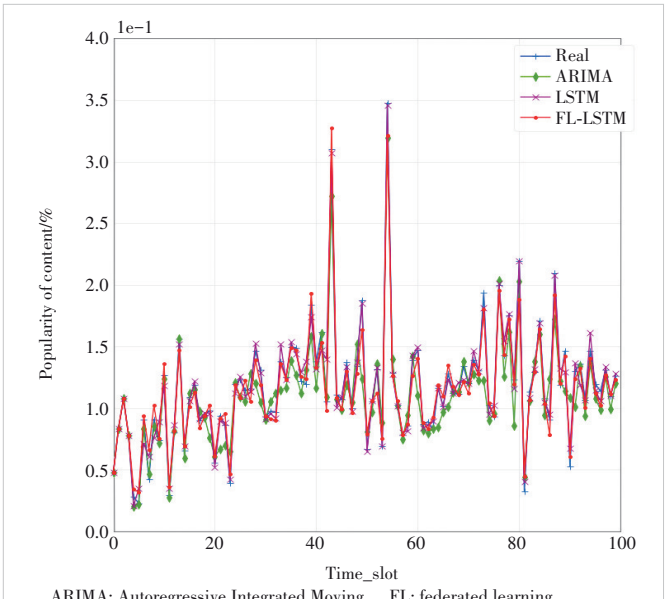
### 4.2 Simulation Setup

The real MovieLens 1M Dataset was used to construct the content popularity prediction datasets. The top 10 rated movies are selected as the forecast object with their trends for 100 days. In the time series prediction problem, we set $k$ = 10, which means that we use the previous 10 time-slots data to predict the popularity of the next moment. In addition, the number of BSs is set as $B$ = 5 and the dataset is divided uniformly and handed out to each BS.

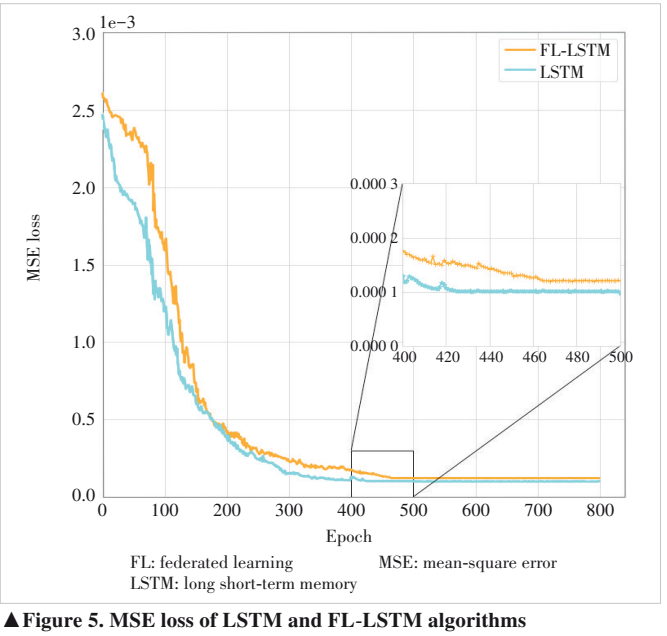
### 4.3 Performance Comparison

In Fig. 4, the performance of different algorithms is compared under the task of predicting the content popularity for 100 days. The LSTM algorithm and the Autoregressive Integrated Moving Average Model (ARIMA) algorithm[20] are selected as the benchmark algorithms. As shown in Fig. 4, the LSTM and FL-LSTM algorithms have similar prediction results, and their accuracy is significantly better than the ARIMA algorithm. The ARIMA algorithm depends on the statistical characteristics of the data and the performance is limited by parameter estimation, so it is difficult to achieve high accuracy. In contrast, the LSTM and the FL-LSTM algorithms have the same core prediction network, which reflects obviously the superiority of performance for the time series prediction problem.

Fig. 5 indicates the convergence of the MSE loss with the



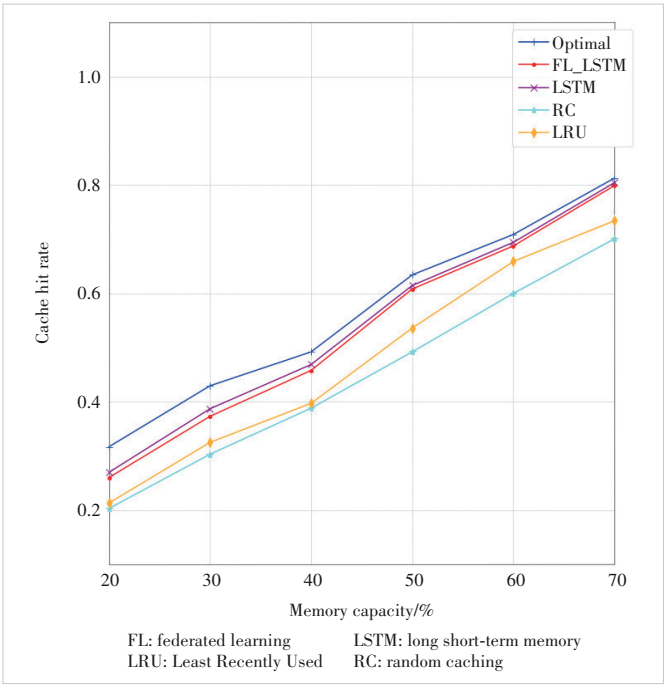

▲Figure 4. Content popularity prediction by different algorithms




▲Figure 5. MSE loss of LSTM and FL-LSTM algorithms

LSTM algorithm and the FL-LSTM algorithm, where the MSE loss is related to the accuracy of the prediction. In the case of setting the same simulation parameters, both of these algorithms can complete the convergence within the range of 400 – 500 training rounds. At the same time, the value of MSE is very small, and both can be below 0.000 2, which proves the superiority of the prediction performance. In addition, in the same simulation environment, the convergence of the FL-LSTM is slightly slower than the LSTM algorithm and the loss is similar, i.e., only 4.2% between the two algorithms. The main reasons why the federated learning scheme is slightly inferior to the centralized one are as follows: 1) The user data distribution of each BS is different, which also leads to a certain delay in updating the global model during the distributed FL-LSTM training; 2) when weight aggregation is carried out in federated learning, fractional parts of weight parameters are generally truncated to improve uploading efficiency. Therefore, there will be a certain degree of numerical precision loss in the process of average weighted sum. For the sake of privacy protection, it is acceptable to sacrifice a little bit of performance.

Fig. 6 shows how the total hit rate changes as the memory capacity of the BSs increases. Cache capacity is defined as the ratio of the number of files cached to the total number of files in a file set. There is a significant upward trend when increasing the memory capacity. The reason is that the users' requested contents are more likely to be accurately predicted and cached in the base station. However, for the LRU and the random caching (RC), although it takes into account historical popularity information, the content popularity will not be predicted in advance, which will suffer more inaccurate caching and cause a lower hit rate by at least a 14.3% difference. The optimal performance is obtained under the real content popularity, which is an ideal situation. In addition, compared with traditional cache algorithms, although the FL-LSTM algorithm proposed in this paper requires additional resources for model prediction, the time required to execute model prediction is relatively small. From Fig. 6, it can also be observed that when the memory capacity increases, the hit rate of the FL-LSTM algorithm will gradually approach the optimal value, which is only a 2.3% performance loss at 70% capacity. Therefore, this shows the superiority of the algorithm proposed in this paper.




▲**Figure 6. Total hit rates versus memory capacity**

## 5 Conclusions

In this paper, we investigate the content popularity prediction problem in cache-enabled wireless networks. Meanwhile, a novel prediction algorithm FL-LSTM based on LSTM and FL is proposed for privacy preservation. The proposed algorithm can not only predict the content popularity accurately but also protect the users' privacy information. Moreover, the performance of the FL-LSTM is validated on the real-world dataset compared to other algorithms. Simulation results demonstrate that the performance of the proposed algorithm just declined slightly, only 4.2% compared with the centralized LSTM and is better than other state-of-the-art schemes while the privacy can be well preserved.

## References

[1] WANG C M, HE Y, YU F R, et al. Integration of networking, caching, and computing in wireless systems: a survey, some research issues, and challenges [J]. IEEE communications surveys & tutorials, 2018, 20(1): 7 – 38. DOI: 10.1109/COMST.2017.2758763

[2] NDIKUMANA A, TRAN N H, HO T M, et al. Joint communication, computation, caching, and control in big data multi-access edge computing [J]. IEEE transactions on mobile computing, 2020, 19(6): 1359 – 1374. DOI: 10.1109/TMC.2019.2908403

[3] PASCHOS G S, IOSIFIDIS G, TAO M X, et al. The role of caching in future communication systems and networks [J]. IEEE journal on selected areas in communications, 2018, 36(6): 1111 – 1125. DOI: 10.1109/JSAC.2018.2844939

[4] WEI Y F, YU F R, SONG M, et al. Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor-critic deep reinforcement learning [J]. IEEE Internet of Things journal, 2019, 6(2): 2061 – 2073. DOI: 10.1109/JIOT.2018.2878435

[5] AHMED M, TRAVERSO S, GIACCONE P, et al. Analyzing the performance of LRU caches under non-stationary traffic patterns [EB/OL]. (2013-01-21)[2023-03-01]. https://arxiv.org/abs/1301.4909

[6] JALEEL A, THEOBALD K B, STEELY S C, et al. High performance cache replacement using re-reference interval prediction (RRIP) [C]//37th annual international symposium on computer architecture. ACM, 2010: 60 – 71. DOI: 10.1145/1815961.1815971

[7] LI L X, XU Y, YIN J Y, et al. Deep reinforcement learning approaches for content caching in cache-enabled D2D networks [J]. IEEE Internet of Things journal, 2020, 7(1): 544 – 557. DOI: 10.1109/JIOT.2019.2951509

[8] WON D U, KIM H S. A prediction scheme for movie preference rating based on DeepFM model [C]//International Conference on Information Networking (ICOIN). IEEE, 2022: 385 – 390. DOI: 10.1109/ICOIN53446.2022.9687136
[9] LI D Y, ZHANG H X, DING H, et al. User preference learning-based proactive edge caching for D2D-assisted wireless networks [J]. IEEE Internet of Things journal, 2023, early access. DOI: 10.1109/JIOT.2023.3244621
[10] JIANG Y X, FENG H J, ZHENG F C, et al. Deep learning-based edge caching in fog radio access networks [J]. IEEE transactions on wireless communications, 2020, 19(12): 8442 – 8454. DOI: 10.1109/TWC.2020.3022907
[11] MCMAHAN H B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data [EB/OL]. (2016-02-17) [2023-03-01]. https://arxiv.org/abs/1602.05629
[12] SHI Y M, YANG K, JIANG T, et al. Communication-efficient edge AI: algorithms and systems [J]. IEEE communications surveys & tutorials, 2020, 22(4): 2167 – 2191. DOI: 10.1109/COMST.2020.3007787
[13] KHAN L U, PANDEY S R, TRAN N H, et al. Federated learning for edge networks: resource optimization and incentive mechanism [J]. IEEE communications magazine, 2020, 58(10): 88 – 93. DOI: 10.1109/MCOM.001.1900649
[14] LIM W Y B, LUONG N C, HOANG D T, et al. Federated learning in mobile edge networks: a comprehensive survey [J]. IEEE communications surveys & tutorials, 2020, 22(3): 2031 – 2063. DOI: 10.1109/COMST.2020.2986024
[15] FAROOQ M S, TEHSEEN R, QURESHI J N, et al. FFM: flood forecasting model using federated learning [J]. IEEE access, 2023, 11: 24472 – 24483. DOI: 10.1109/ACCESS.2023.3252896
[16] WANG K L, DENG N, LI X H. An efficient content popularity prediction of privacy preserving based on federated learning and Wasserstein GAN [J]. IEEE Internet of Things journal, 2023, 10(5): 3786 – 3798. DOI: 10.1109/JIOT.2022.3176360
[17] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural computation, 1997, 9(8): 1735 – 1780. DOI: 10.1162/neco.1997.9.8.1735
[18] DOAN K N, VAN NGUYEN T, QUEK T Q S, et al. Content-aware proactive caching for backhaul offloading in cellular network [J]. IEEE transactions on wireless communications, 2018, 17(5): 3128 – 3140. DOI: 10.1109/TWC.2018.2806971
[19] HARPER F M, KONSTAN J A. The MovieLens datasets [J]. ACM transactions on interactive intelligent systems, 2016, 5(4): 1 – 19. DOI: 10.1145/2827872
[20] GUO J Q, HE H W, SUN C. ARIMA-based road gradient and vehicle velocity prediction for hybrid electric vehicle energy management [J]. IEEE transactions on vehicular technology, 2019, 68(6): 5309 – 5320. DOI: 10.1109/TVT.2019.2912893

## Biographies

**YAN Yuna** is currently working toward her master's degree under the supervision of Prof. LI Lixin with the School of Electronics and Information, Northwestern Polytechnical University, China. Her research interests include federated learning, deep learning and semantic communications.

**LIU Ying** is an engineer of Shanghai Satellite Engineering Research Institute, China, mainly engaged in satellite system design and satellite communications.

**NI Tao** is a senior engineer of Shanghai Satellite Engineering Research Institute, China, mainly engaged in satellite system design and satellite communications.

**LIN Wensheng** received his BE degree in communication engineering and ME degree in electronic and communication engineering from Northwestern Polytechnical University, China in 2013 and 2016. He received his PhD degree in information science from the Japan Advanced Institute of Science and Technology in 2019. He is currently an associate professor with the School of Electronics and Information, Northwestern Polytechnical University. His research interests include network information theory, distributed source coding, and age of information.

**LI Lixin** (lilixin@nwpu.edu.cn) received his BS degree (Hons.), MS degree (Hons.), and PhD degree from Northwestern Polytechnical University (NPU), China in 2001, 2004, and 2008, respectively. From 2009 to 2011, he was a postdoctoral research fellow with NPU. In 2011, he joined the School of Electronics and Information, NPU, where he is currently a full professor and chair of Department of Communication Engineering. In 2017, he held the visiting scholar position with the University of Houston, USA. He holds 26 granted patents and has authored or coauthored five books and more than 200 peer-reviewed papers in many prestigious journals and conferences. His research interests include 5G/6G wireless networks, federated learning, game theory, and machine learning for wireless communications. He was the recipient of the 2016 NPU Outstanding Young Teacher Award, which is the highest research and education honors for young faculties in NPU. He was an exemplary reviewer for *IEEE Transactions on Communications* in 2020.