Content Popularity Prediction in Fog Radio Access Networks: A Federated Learning Based Approach

Yuting Wu[1,2,3], Yanxiang Jiang[1,2,3,*], Mehdi Bennis[4], Fuchun Zheng[1,5], Xiqi Gao[1], and Xiaohu You[1]
[1]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China.
[2]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China
[3]Key Laboratory of Wireless Sensor Network & Communication, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, 865 Changning Road, Shanghai 200050, China
[4]Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland.
[5]School of Electronic and Information Engineering, Harbin Institute of Technology, Shenzhen 518055, China
[*]E-mail: yxjiang@seu.edu.cn

***Abstract*—In this paper, the content popularity prediction problem in fog radio access networks (F-RANs) is investigated. In order to obtain accurate prediction with low complexity, we propose a novel context-aware popularity prediction policy based on *federated learning*. Firstly, user preference learning is applied by considering that users prefer to request the contents they are interested in. Then, users' context information is utilized to cluster users efficiently by adaptive context space partitioning. After that, we formulate a popularity prediction optimization problem to learn the local model parameters using the stochastic variance reduced gradient (SVRG) algorithm. Finally, federated learning based model integration is proposed to construct the global popularity prediction model based on local models by combining the distributed approximate Newton (DANE) algorithm with SVRG. Our proposed popularity prediction policy not only predicts content popularity accurately, but also significantly reduces computational complexity. Simulation results show that our proposed policy increases the cache hit rate by up to 21.5 $\%$ compared to the traditional policies.**

***Index Terms*—F-RAN, popularity prediction, user preference learning, context-aware, federated learning.**

I. INTRODUCTION

With the unprecedented rapid proliferation of intelligent devices, wireless networks are confronted with a myriad of challenges, and notably data traffic pressure on the fronthaul wireless links. To cope with this, fog radio access network (F-RAN) has emerged as a promising solution to alleviate the traffic burden on fronthaul links by caching popular contents in fog access points (F-APs). In F-RANs, F-APs with limited caching and computing resources are densely deployed at network edges to serve users' requests [1]. Due to the caching capacity constraints, F-APs need to predict future content popularity accurately in order to prefetch the most popular contents during off-peak traffic periods and improve caching efficiency.

Traditional caching policies, such as least recently used (LRU), least frequently used (LFU), are widely used in wired networks, but their efficiency is limited since content popularity is not considered [2]. Recently, improving caching efficiency by predicting content popularity has gained significant interest. An auto-regressive (AR) model based content popularity prediction algorithm was proposed in [3] and the model parameters were learned by least square estimates. Similarly, an auto regressive and moving average (ARMA) model was presented and proved to outperform the LFU caching scheme in [4]. In [5], a deep learning based content popularity prediction scheme was proposed. In [6], the authors proposed to learn popularity prediction model for each content class from the historical popularity series through training a simplified bidirectional long short-term memory (Bi-LSTM) network. In [7], a multilevel probabilistic model was proposed and Bayesian learning was utilized to obtain the model parameters. In [8], a user preference model was proposed to predict content popularity and track the popularity change based on the user preference and the features of the requested content. However, these existing works except for [7] and [8] can not predict the popularity of newly-added or unseen contents whose statistical data is not available in advance. In [7], the knowledge of the prior distribution of the parameters is required. In [8], the authors propose a caching policy instead of predicting popularity through user preference learning. Moreover, most of the existing works ignore the complexity of the content prediction policies.

Motivated by the aforementioned discussions, we propose a context-aware popularity prediction policy based on federated learning. The inputs of the prediction model are the averaged popularity scores for contents of the clustered users, which are preprocessed by user preference learning and adaptive context space partitioning. Then, the popularity prediction model is learned automatically by training the model with historical popularity and the preprocessed inputs. Furthermore, federated learning based model integration is proposed to construct the global model based on local models in a distributed manner. Specifically, our proposed popularity prediction policy is efficient in predicting the popularity of newly-added contents, while reducing the computational complexity and communication overhead.

The remainder of this paper is organized as follows. In Section II, the system model is presented. The proposed popularity prediction policy is described in Section III. Simulation results are shown in Section IV. Final conclusions are drawn in Section V.

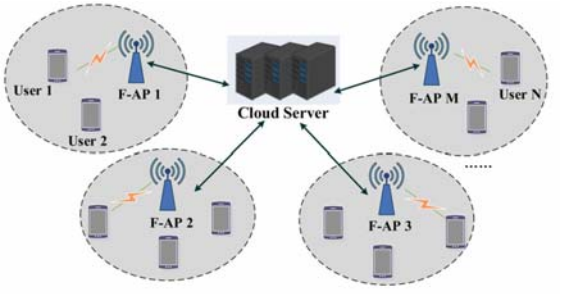

Fig. 1. Illustration of the scenario in F-RAN.

II. SYSTEM MODEL

As shown in Fig. 1, we consider the F-RAN in a specific region consisting of M F-APs and N users. Let $\mathcal{Q} = \{q_1, q_2, \ldots, q_m, \ldots, q_M\}$ denote the set of F-APs. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n, \ldots, u_N\}$ denote the set of users in the region. Every F-AP has its own coverage area, limited storage capacity and computing ability. Mobile users are associated with an F-AP and can be served by it when located in its coverage area. Let $\mathcal{C} = \{c_1, c_2, \ldots, c_i, \ldots, c_I\}$ denote the content library, where I denotes the cumulative number of contents for the time being. If u_n sends a request for c_i that is stored in its associated F-AP q_m, u_n can directly fetch c_i from its local cache. Otherwise, q_m needs to fetch the content from neighboring F-APs or the cloud server. Assume that all the contents have the same size, and the coverage areas of different F-APs do not ovelap with each other.

The popularity of contents in F-APs is determined by the requests sended by the associated users. If individual caching strategy in a single F-AP is considered, the local popularity prediction model of the F-AP is required; if cooperative caching strategy among several F-APs is considered, the global popularity prediction model in the larger coverage area of these F-APs is required. Let $\hat{p}_i$ and p_i denote the predicted and real popularity of c_i, respectively. There exists deviation between $\hat{p}_i$ and p_i. Therefore, the mean-square error (MSE) is utilized to measure the accuracy of the prediction as follows:

$$\text{MSE} = \frac{1}{I}\sum_{i=1}^{I} |\hat{p}_i - p_i|^2 \tag{1}$$

After prediction, the F-APs cache the most popular contents according to the predicted popularity. If a request from users is served by the cache of F-APs instead of fetching the content from the cloud server through fronthaul links, a cache hit event occurs. Cache hit rate is defined as the ratio of the cache hits to the overall number of requests. It is utilized to evaluate the caching performance.

The objective of this paper is to find a content popularity prediction policy to predict future popularity accurately and maximize the cache hit rate.

III. PROPOSED POPULARITY PREDICTION POLICY

In order to maximize the cache hit rate, we propose a novel content popularity prediction policy, which includes offline user preference learning, adaptive context space partitioning, popularity prediction model construction and federated learning based model integration. The proposed policy can predict content popularity accurately and cache popular contents based on the prediction.

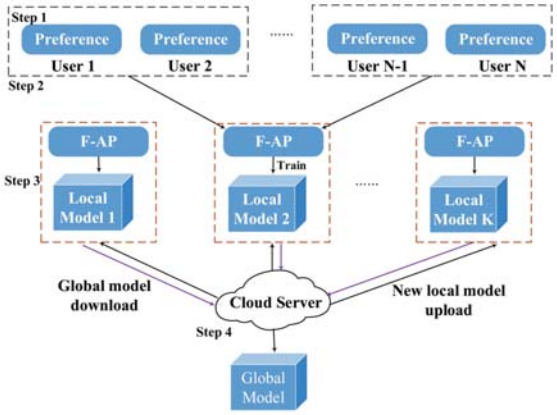

Fig. 2. Illustration of the proposed popularity prediction policy.

A. Policy Description

The procedure of the proposed popularity prediction policy is shown as the following four steps.

(1) *Offline user preference learning*: The popularity of contents will change constantly due to the instability of the user preference. Therefore, user preference learning is the first step of the proposed policy. User preference learning is conducted independently by training the local data in the user equipment (UE) in an offline manner. If the user preference is similar with the content feature, the probability for the user requesting the content will be high, leading to a higher popularity score for the content [9].

(2) *Adaptive context space partitioning*: After offline user preference learning, users send the preference and context information to their associated F-APs. Context information of users includes gender, age, personality, occupation, location, equipment, etc. Considering the privacy issue, the context information transmitted to the F-APs needs to be carefully selected or encoded. Then, the F-APs can take the corelation between user preference and context information into account, since users having similar context are more likely to have similar preference. As a consequence, adaptive context space partitioning is applied as the second step in the proposed policy to cluster users efficiently [10].

(3) *Popularity prediction model construction*: In the third step, a popularity prediction model is constructed based on the averaged popularity scores of users in the context subspaces after partitioning. The model parameters can be learned by training the historical data.

(4) *Federated learning based model integration*: Finally, the federated learning based model integration method is a complement for the proposed policy. If the local popularity

prediction models in F-APs have been obtained, a global model consisting of these F-APs can be generated by integrating the local models in a distributed manner [11]. By using this approach, computational complexity can be reduced and the bandwidth for transmitting local data can be saved.

The proposed popularity prediction policy is illustrated in Fig. 2. The details of the four steps of the policy will be presented below.

B. Offline User Preference Learning

Let $\boldsymbol{\chi}_i = [\chi_{i1}, \chi_{i2}, \ldots, \chi_{ij}, \ldots, \chi_{iJ}]^T$ denote the content feature vector of c_i, where J denotes the number of content categories. Users have different preference for these J content categories. Let $\boldsymbol{w}_n = [w_{n1}, w_{n2}, \ldots, w_{nj}, \ldots, w_{nJ}]^T$ denote the user preference vector of u_n. User preference can be learned independently without interaction with each other. Therefore, the user preference learning is flexible, which can be conducted at any idle time.

There are a large amount of records of user requests in UE to be used as training samples. Let $\mathcal{A}_n = \{(\boldsymbol{\chi}_i, y_{n,i}), i \in [1, I]\}$ denote the set of training samples for u_n, where $y_{n,i}$ denotes the binary request label. If u_n has requestd for c_i, $y_{n,i} = 1$; otherwise $y_{n,i} = 0$. Due to the offline property, the privacy of users can be protected since numerous user data is kept in private equipment instead of transmitting to the central server.

Sigmoid function is used to approximate the correspondence between the feature vector and request label of contents. For simplicity, let $h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i) = 1/(1 + e^{-\boldsymbol{w}_n^T \cdot \boldsymbol{\chi}_i})$. The probability function of u_n is formulated to represent the probability for $y_{n,i}$ as follows:

$$p_{\boldsymbol{w}_n}(y_{n,i}|\boldsymbol{\chi}_i) = h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i)^{y_{n,i}} (1 - h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i))^{1-y_{n,i}}. \quad (2)$$

Given enough samples, the likelihood function of u_n can be expressed as follows:

$$L(\boldsymbol{w}_n) = \prod_{(\boldsymbol{\chi}_i, y_{n,i}) \in \mathcal{A}_n} p_{\boldsymbol{w}_n}(y_{n,i}|\boldsymbol{\chi}_i). \quad (3)$$

Therefore, the cross entropy loss function is formulated as the negative log-likelihood function as follows:

$$F(\boldsymbol{w}_n) = \frac{1}{|\mathcal{A}_n|} \sum_{(\boldsymbol{\chi}_i, y_{n,i}) \in \mathcal{A}_n} \left[\ln\left(1 + e^{\boldsymbol{w}_n^T \cdot \boldsymbol{\chi}_i}\right) - y_{n,i} \cdot \boldsymbol{w}_n^T \cdot \boldsymbol{\chi}_i\right]. \quad (4)$$

Consequently, $\boldsymbol{w}_n$ can be obtained by minimizing $F(\boldsymbol{w}_n)$.

The "Follow The (Proximally) Regularized Leader" (FTRL-Proximal) algorithm is adopted to learn the user preference [8]. In addition, user preference needs to be updated dynamically when $F(\boldsymbol{w}_n)$ exceeds a certain threshold.

C. Adaptive Context Space Partitioning

User preference may differ based on their context information [12]. For example, the movie types favored by boys and girls, young and old, are usually different. Therefore, we propose to partition the context space uniformly into parts of similar contexts for further popularity prediction. In addition, popularity scores for contents can be learned independently

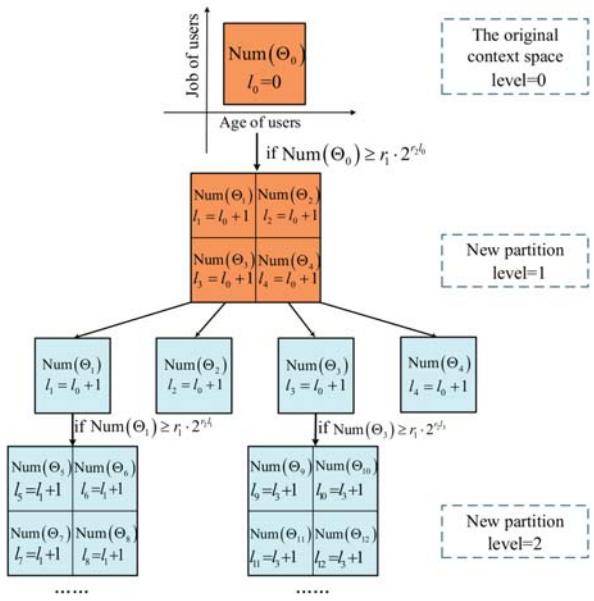

Fig. 3. Illustration of adaptive context space partitioning (D = 2).

in each context subspace. After partitioning, the users in the same context subspace having similar preference can be treated as a group. By using this approach, the number of items to be kept track is significantly reduced. As a consequence, the computational burden is maintained to a manageable extent.

Let $\boldsymbol{\zeta}_n = [\zeta_{n,1}, \zeta_{n,2}, \ldots \zeta_{n,D}]^T \in [0, 1]^D$ denote the context vector of u_n, where D denotes the dimension of context space. The process of adaptive context space partitioning is illustrated in Fig. 3 and Algorithm 1. The original context space Θ_0 is constructed and normalized as $[0, 1]^D$, and its level $l_0 = 0$. All the users considered are included in Θ_0. Consider Θ_i to be partitioned. Let Num(Θ_i) denote the number of users in Θ_i. Let l_i denote the level of Θ_i. If Num(Θ_i) exceeds a certain threshold, Θ_i is splitted into 2^D subspaces. As one of these child subspaces, Θ_j has an increased level of $l_j = l_i + 1$. The threshold determines the rate at which the context space is partitioned. Therefore, it needs to be carefully designed. We design the threshold to have the form $r_1 \cdot 2^{r_2 l_i}$, where $r_1 > 0$ and $r_2 > 0$ are hyper-parameters in Algorithm 1 [13]. Due to this splitting process, a subspace of level l_i has length 2^{-l_i} along each axis. For clarity of description, we rename the final obtained subspaces as $\Theta_1, \Theta_2, \ldots, \Theta_s, \ldots \Theta_S$, where S denotes the cumulative number of context subspaces. Consequently, users are clustered according to the partitoning results. In addition, the context space should be reconstructed and partitioned when the users in the specific region migrate massively.

Since user preference has been obtained, sigmoid function $h_{\boldsymbol{w}_n}(\boldsymbol{\chi}_i)$ can be used as mapping function $f(\boldsymbol{w}_n, \boldsymbol{\chi}_i)$ to map the correlation of user preference and content feature to popularity scores. The average popularity score $x_{s,i}$ of Θ_s for c_i is obtained by averaging the popularity scores of all the

Algorithm 1 Adaptive Context Space Partitioning

Input: $\boldsymbol{\zeta}_n = [\zeta_{n,1}, \zeta_{n,2}, \dots \zeta_{n,D}]^T$, D, r_1, r_2, ψ =number of the total subspaces
Output: $\Theta_1, \Theta_2, \dots, \Theta_s, \dots \Theta_S$

```
initialize i = 0, l_0 = 0, ψ = 1
while i < ψ do
    if Num(Θ_i) ≥ r_1 · 2^{r_2 l_i} then
        split Θ_i into 2^D subspaces Θ_j
        ψ = ψ + 2^D
        l_j = l_i + 1
    end if
    i = i + 1
end while
```

users in Θ_s:

$$x_{s,i} = \frac{1}{\mathrm{Num}(\Theta_s)} \sum_{\boldsymbol{\zeta}_n \in \Theta_s} f\left(\boldsymbol{w}_n, \boldsymbol{\chi}_i\right). \tag{5}$$

Let $\boldsymbol{x}_i = [x_{1,i}, x_{2,i}, \dots, x_{S,i}]^T$ denote the popularity scores for c_i of all the context subspaces after partitioning. The popularity scores are used as the inputs of the popularity prediction model in the following subsection.

D. Popularity Prediction Model Construction

Without loss of generality, we consider the local popularity prediction model in a typical F-AP. Let $\boldsymbol{a} = [a_1, a_2, \dots, a_s, \dots, a_S]^T$ denote the model parameters. By using $\boldsymbol{x}_i$ as the input of the prediction model, $\hat{p}_i$ can be expressed as follows:

$$\hat{p}_i = \boldsymbol{a}^T \boldsymbol{x}_i = a_1 x_{1,i} + a_2 x_{2,i} + \cdots + a_S x_{S,i}. \tag{6}$$

Least square method is applied to learn $\boldsymbol{a}$. Let $\mathcal{B} = \{(p_i, \boldsymbol{x}_i), i \in [1, I]\}$ denote the training samples in the considered F-AP. The model parameters $\boldsymbol{a}$ can be obtained by minimizing MSE as follows:

$$\begin{array}{ll} \min & \frac{1}{L} \sum_{(p_i, \boldsymbol{x}_i) \in \mathcal{B}} \left| p_i - \boldsymbol{a}^T \boldsymbol{x}_i \right|^2 \\ \text{s.t.} & 0 < a_s < 1, \forall s \in [1, S] \end{array}, \tag{7}$$

where L denotes the number of training samples in the F-AP. The model is trained with historical popularity and historical popularity scores in context subspaces.

The elements of $\boldsymbol{a}$ in the prediction model represent the active levels of the users in the corresponding context subspaces [14]. For example, the active levels for teenagers and mid-aged are totally different, because young people use mobile phones more frequently. It implies that their contributions to the network traffic differ a lot. Stochastic variance reduced gradient (SVRG) algorithm is adopted to solve the optimization problem in (7) [15]. It is a variant of stochastic gradient descent (SGD) with explicit variance reduction, which can achieve faster convergence. Let $v_i(\boldsymbol{a}) = \left| p_i - \boldsymbol{a}^T \boldsymbol{x}_i \right|^2$, $v(\boldsymbol{a}) = \frac{1}{L} \sum_{(p_i, \boldsymbol{x}_i) \in \mathcal{B}} v_i(\boldsymbol{a})$. The detailed procedure of SVRG is shown in Algorithm 2. Specifically, the algorithm operates in two nested loops. In the outer loop, it computes the gradient

Algorithm 2 Stochastic Variance Reduced Gradient (SVRG)

Input: $\boldsymbol{x}_i = [x_{1,i}, x_{2,i}, \dots, x_{S,i}]^T, \forall i \in [1, I]$
Output: $\boldsymbol{a} = [a_1, a_2, \dots, a_s, \dots, a_S]^T$

```
1: initialize φ =number of stochastic steps per epoch, h =stepsize, a^0
2: for j = 0, 1, 2, ··· do
3:     compute and store                          ▷ Full pass through data
4:     ▽v(a^j) = (1/L) Σ_{(p_i, x_i) ∈ B} ▽v_i(a^j)
5:     set a_1 = a^j
6:     for t = 1 to φ do
7:         pick (p_i, x_i) ∈ B uniformly at random
8:         calculate g_t according to (8)
9:         a_{t+1} = a_t − h g_t                  ▷ Stochastic update
10:    end for
11:    a^{j+1} = a_{φ+1}
12: end for
```

value of the entire function (Line 4 in Algorithm 2). In the inner loop, the gradient in iteration t is calculated as follows:

$$\boldsymbol{g}_t = \bigtriangledown v_i(\boldsymbol{a}_t) - (\bigtriangledown v_i\left(\boldsymbol{a}^j\right) - \bigtriangledown v\left(\boldsymbol{a}^j\right)), \tag{8}$$

where $(\bigtriangledown v_i\left(\boldsymbol{a}^j\right) - \bigtriangledown v\left(\boldsymbol{a}^j\right))$ is regarded as the bias of the gradient estimate $\bigtriangledown v_i(\boldsymbol{a}_t)$. Namely, the algorithm modifies the gradient $\bigtriangledown v_i(\boldsymbol{a}_t)$ based on $\boldsymbol{a}_t$ in iteration t. By adjusting the number of stochastic steps, the tradeoff between convergence rate and computational complexity can be flexibly balanced.

E. Federated Learning Based Model Integration

The global popularity prediction model aggregated by K F-APs is considered. Let $\{q_1, q_2, \cdots, q_m, \cdots, q_K\}$ denote the set of considered F-APs. In order to learn the global model, the cloud server needs to solve the optimization problem as follows:

$$\begin{array}{ll} \min & \frac{1}{L^*} \sum_{(p_i, \boldsymbol{x}_i) \in \mathcal{B}^*} \left| p_i - \boldsymbol{a}^T \boldsymbol{x}_i \right|^2 \\ \text{s.t.} & 0 < a_s < 1, \forall s \in [1, S] \end{array}, \tag{9}$$

where L^* denotes the number of training samples of the K F-APs in the cloud server, $\mathcal{B}^*$ denotes the corresponding set of training samples. Due the expanded coverage area and the increased number of associated users compared with the local model, it is obvious that $L^* \gg L$. As a consequence, the computational load will be extremely high. In addition, the training data in the distributed F-APs needs to be transmitted to the cloud server, which will cost the bandwidth resources. To cope with these issues, federated learning based model integration is proposed to generate the global model based on local models in a distributed manner.

The model integration based on federated learning is conducted as follows: Firstly, the F-APs accepts the global model parameters broadcasted by the cloud server. Then, each F-AP computes an update to the current global model by using its local data. Finally, all the F-APs communicate these updates to the cloud server to aggregate a new global model. By using this approach, bandwidth resources can be saved, because only the model parameters need to be transmitted to the cloud

server instead of transmitting all the training data. Moreover, the computational complexity is significantly reduced, since repeated calculation is avoided.

Let $\mathcal{B}_m$ denote the set of training samples in q_m. The MSE in q_m can be expressed as $V_m(\boldsymbol{a}) = \frac{1}{L_m}\sum_{(p_i,\boldsymbol{x}_i)\in\mathcal{B}_m} v_i(\boldsymbol{a})$, where L_m denotes the number of training samples in q_m. Therefore, the optimization problem in (9) can be converted to the following equivalent form:

$$\begin{array}{ll}\min & \sum_{m=1}^{K} \frac{L_m}{L^*} V_m(\boldsymbol{a}) \\ \text{s.t.} & 0 < a_s < 1, \forall s \in [1, S]\end{array}. \tag{10}$$

It indicates that the global optimization is determined by local optimizations. The most intuitive solution is to make each F-AP minimize its local function, and then average the results of all the F-APs. However, it is impractical unless the local model parameters are all the same.

The distributed approximate Newton (DANE) [15] algorithm is applied to remedy the above method by modifying the local problems before each aggregation step. A feasible solution is to perturb the local function $V_m(\boldsymbol{a})$ of q_m in each iteration j with the disturbance term: $-(\boldsymbol{b}_m^j)^T\boldsymbol{a}$. It means that in each iteration j, the F-APs parallelly solve the optimization problem as follows:

$$\boldsymbol{a}_m^{(j)} = \arg\min_{\boldsymbol{a}\in\mathbb{R}^S} \left\{ V_m(\boldsymbol{a}) - (\boldsymbol{b}_m^j)^T\boldsymbol{a} \right\}, \tag{11}$$

where $\boldsymbol{a}_m^{(j)}$ denotes the vector of model parameters of q_m in iteration j. According to [15], the value of $\boldsymbol{b}_m^j$ can be calculated as $(\bigtriangledown V_m(\boldsymbol{a}^j) - \bigtriangledown v(\boldsymbol{a}^j))$.

As described in the previous subsection, (11) can be solved with SVRG. Consequently, federated learning based model integration is proposed to incorporate DANE and SVRG into the popularity prediction model. By using this approach, the global model can be learned by distributed computation. Moreover, the convergence rate can be improved by adopting SVRG. The detailed description is shown in Algorithm 3. Let $\boldsymbol{g}_m^t$ denote the gradient in iteration t in q_m. In the actual update loop, $\boldsymbol{g}_m^t$ is calculated by applying (8) to solving (11) as follows:

$$\boldsymbol{g}_m^t = \bigtriangledown v_i\left(\boldsymbol{a}_m^{(t)}\right) - \bigtriangledown v_i\left(\boldsymbol{a}^j\right) + \bigtriangledown v\left(\boldsymbol{a}^j\right). \tag{12}$$

According to the above descriptions, the proposed popularity prediction policy not only predicts popularity of existing or newly-added contents with a high accuracy, but also significantly reduces the computational complexity and communication overhead.

Algorithm 3 Federated learning based model integration

Input: $\boldsymbol{x}_i = [x_{1,i}, x_{2,i}, \ldots, x_{S,i}]^T$, φ =number of stochastic steps per epoch, h =stepsize,
Output: $\boldsymbol{a} = [a_1, a_2, \ldots, a_s, \ldots, a_S]^T$

```
initialize φ =number of stochastic steps per epoch, h =stepsize, a^0
for j = 0, 1, 2, ··· do                                   ▷ Overall iterations
    compute and store
    ▽v(a^j) = (1/L*) Σ_{(p_i,x_i)∈B*} ▽v_i(a^j)
    for m = 1, 2, ··· , K do                              ▷ Distributed loop
        (in parallel over nodes)
        initialize a_m^(1) = a^j
        for t = 1 to φ do                                 ▷ Actual update loop
            sample (p_i, x_i) ∈ B_m uniformly at random
            calculate g_m^t according to (12)
            a_m^(t+1) = a_m^(t) − h g_m^t
        end for
    end for
    a^{j+1} = a^j + (1/K) Σ_{m=1}^{K} (a_m^(φ+1) − a^j)   ▷ Aggregate
end for
```

IV. SIMULATION RESULTS

To evaluate the performance of the proposed popularity prediction policy, we perform simulations based on the data extracted from the MovieLens Dataset [16]. The MovieLens 100K Dataset contains 100000 ratings of 943 users on 1682 movies. Each user has rated at least 20 movies. Each data set entry consists of an anonymous user ID, a movie ID, a rating (1-5) and a timestamp. We assume that the ratings correspond to the number of requests from users. In addition, demographic information about the users is provided in the dataset, including their gender, age, occupation and Zip-code. For numerical evaluations, we select gender and age as context information. Besides, the genres of the contained movies are provided, which can be used as the content feature.

In Fig. 4, we show the logarithmic root mean-square error (RMSE) of our proposed popularity prediction policy and the AR model based policy with different number of considered contents [3]. It can be observed that the RMSE value of the proposed policy gradually decreases as the number of contents increasing. The reason is that the proposed policy can better learn the relationship between popularity and content features with more training samples. It can also be observed that the RMSE of the proposed policy is smaller than the AR model based policy. The reason is that the inputs of these two policies are different. The input of the proposed policy captures the internal characteristics of popularity after a series of data preprocessing, whereas the input in the AR model based policy is processed roughly.

In Fig. 5, we show the cache hit rates of our proposed policy and four benchmark policies, including the AR model based policy, LRU, LFU, and Random Caching (RC). Assume that the caching policy for the proposed policy and the AR based policy is to cache the most popular contents preferentially according to prediction. It can be observed that the cache hit rates of all the considered policies increase gradually as the storage capacity increasing. It can also be observed that the proposed policy achieves better performance than the four benchmarks with higher cache hit rate. The LRU and LFU are liable to suffer performance degradation due to their neglect of content popularity. Due to the improved prediction accuracy as shown in Fig. 4, the proposed policy achieves higher cache

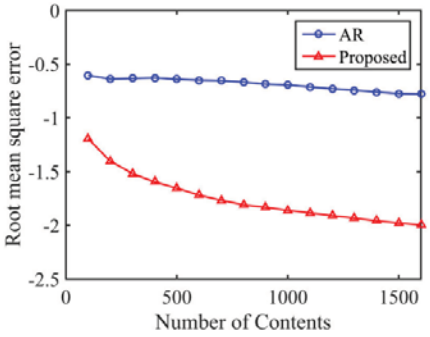

Fig. 4. Root mean-square error versus number of contents.

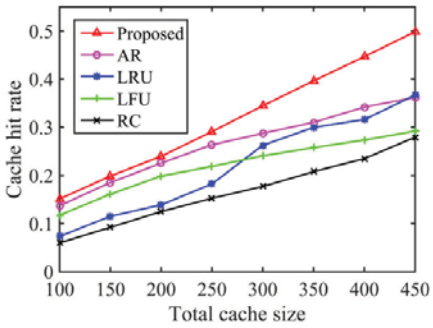

Fig. 5. Cache hit rates versus total cache size.

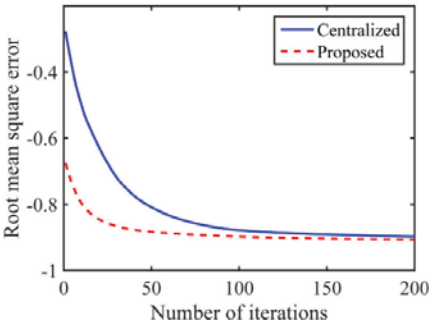

Fig. 6. Root mean-square error versus number of iterations.

hit rate than the AR based policy by up to 21.5%. Moreover, the proposed policy can predict the popularity of newly-added contents by leveraging user preference, which also leads to better caching performance.

In Fig. 6, we show the logarithmic RMSE of our proposed policy and the centralized policy as the number of iterations varies when generating the global model. It can be observed that our proposed policy achieves faster convergence than the centralized policy. The reason is that our proposed policy generates the global model by integraing the existing local models based on federated learning. The global model can be initialized by the local models, instead of random generation. Consequently, a large part of repeated computation is avoided. While the centralized policy has to recalculate the training samples after data transmission, which is resource-wasting.

V. CONCLUSION

In this paper, we have proposed a novel popularity prediction policy based on user preference learning, adaptive context space partitioning and federated learning. Our proposed policy can predict content popularity accurately, even when the contents have no statistical data in advance. The reason is that the relation between content popularity and user preference is considered. Specifically, federated learning based model integration is efficient in reducing computational complexity and communication overhead, which makes our proposed policy more practical. Simulation results have shown that our proposed policy achieves better predicting and caching performance than traditional policies.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under Grant 61971129, the Natural Science Foundation of Jiangsu Province under Grant BK20181264, the National Key R&D Program of China under Grant 2018YFB1801103, the Research Fund of the State Key Laboratory of Integrated Services Networks (Xidian University) under Grant ISN19-10, and the Research Fund of the Key Laboratory of Wireless Sensor Network & Communication (Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences) under Grant 2017002.

REFERENCES

[1] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 137–12 151, Dec. 2018.

[2] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[3] Y. Liu, T. Zhi, H. Xi, X. Duan, and H. Zhang, "A novel content popularity prediction algorithm based on auto regressive model in information-centric IoT," *IEEE Access*, vol. 7, pp. 27 555–27 564, 2019.

[4] N. B. Hassine, R. Milocco, and P. Minet, "ARMA based popularity prediction for caching in content delivery networks," in *2017 Wireless Days*, March. 2017, pp. 113–120.

[5] W. Liu, J. Zhang, Z. Liang, L. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2018.

[6] H. Feng, Y. Jiang, D. Niyato, F. Zheng, and X. You, "Content popularity prediction via deep learning in cache-enabled fog radio access networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6.

[7] S. Mehrizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "A Bayesian Poisson–Gaussian Process model for popularity learning in edge-caching networks," *IEEE Access*, vol. 7, pp. 92 341–92 354, 2019.

[8] Y. Jiang, M. Ma, M. Bennis, F. Zheng, and X. You, "User preference learning-based edge caching for fog radio access network," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1268–1283, Feb. 2019.

[9] A. Zohourian, H. Sajedi, and A. Yavary, "Popularity prediction of images and videos on instagram," in *2018 4th International Conference on Web Research (ICWR)*, Apr. 2018, pp. 111–117.

[10] Q. Chen, W. Wang, and Z. Zhang, "Clustered popularity prediction for content caching," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May. 2019, pp. 1–6.

[11] J. Park, S. Samarakoon, M. Bennis, and M. Debbah, "Wireless network intelligence at the edge," *CoRR*, vol. abs/1812.02858, 2018. [Online]. Available: http://arxiv.org/abs/1812.02858

[12] S. Müller, O. Atan, M. van der Schaar, and A. Klein, "Context-aware proactive content caching with service differentiation in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 2, pp. 1024–1036, Feb. 2017.

[13] S. Li, X. Jie, M. V. D. Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Transactions on Multimedia*, vol. 18, no. 12, pp. 2503–2516, 2016.

[14] W. Jing, X. Wen, Z. Lu, and H. Zhang, "User-centric delay-aware joint caching and user association optimization in cache-enabled wireless networks," *IEEE Access*, vol. 7, pp. 74 961–74 972, 2019.

[15] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, vol. abs/1610.02527, 2016. [Online]. Available: http://arxiv.org/abs/1610.02527

[16] F. M. Harper and J. A. Konstan, *The MovieLens Datasets: History and Context*, 2015.