Federated Learning for Energy-Efficient Task Computing in Wireless Networks

Sihua Wang*, Mingzhe Chen†‡, Walid Saad§, and Changchuan Yin*
*Beijing Laboratory of Advanced Information Network, Beijing University of Posts and Telecommunications, Beijing, China
†Department of Electrical Engineering, Princeton University, Princeton, NJ, USA
‡The Future Network of Intelligence Institute, The Chinese University of Hong Kong, Shenzhen, China
§Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA
Emails: sihuawang@bupt.edu.cn, mingzhec@princeton.edu, walids@vt.edu, ccyin@bupt.edu.cn

***Abstract*—In this paper, the problem of minimizing energy consumption for task computation and transmission in a cellular network with mobile edge computing (MEC) capabilities is studied. In the considered network, each user needs to process a computational task at each time slot. A part of the task can be transmitted to a base station (BS) that can use its powerful computational ability to process the tasks offloaded from its users. Since the data size of each user's computational task varies over time, the BSs must dynamically adjust the resource allocation scheme to meet the users' needs. This problem is posed as an optimization problem whose goal is to minimize the energy consumption for task computing and transmission via adjusting user association scheme as well as their task and power allocation scheme. To solve this problem, a support vector machine (SVM)-based federated learning (FL) is proposed to determine the user association proactively. Given the user association, the BS can collect the information related to the computational tasks of its associated users using which, the transmit power and task allocation of each user will be optimized and the energy consumption of each user is also minimized. The proposed SVM-based FL method enables the BS and users to cooperatively build a global SVM model that can determine all users' association without any transmission of users' historical association and computational task offloading. Simulations using real data on city cellular traffic from the OMNILab at Shanghai Jiao Tong University show that the proposed algorithm can reduce the users' energy consumption by up to 20.1% compared to the conventional centralized SVM method.**

***Index Terms*—Task computing, user association, support vector machine based federated learning.**

I. Introduction

Emerging applications such as virtual reality and interactive online games require large computational capability. Nonetheless, mobile devices may not be able to perform these novel applications due to the limited computational capability [1]. One promising method is to deploy computational resources at the wireless base stations (BSs) for processing computational tasks offloaded from mobile devices hence improving the computational capability of mobile devices [2]. However, deploying computational resources over cellular networks faces many challenges such as high computing latency, cooperative edge computing, and computational resource allocation.

A number of existing works studied important problems related to task offloading and computational resource optimization such as in [3]–[6]. The authors in [3] developed a centralized sorting approach to maximize the number of users served by a cloud while guaranteeing a target task processing delay. In [4], the authors developed a task offloading scheme to minimize the energy consumption. The workers in [5] proposed a binary computation offloading scheme to maximize the sum computation rate of all users. In [6], the authors studied a multi-user task offloading problem. However, the existing works in [3]–[6] do not consider a scenario in which each user requests computational tasks that can be of different data size. As the data size of the requested computational task varies, each BS must dynamically adjust its user association as well as task and power allocation to minimize users' energy consumption. Moveover, traditional methods such as convex optimization require, as input, the information related to the data size of the computational task to optimize the user association as well as task and power allocation. Since each computational task is offloaded from a user, the BSs cannot collect the information related to the computational tasks before the user association is determined. In consequence, without the information of the computational task, traditional optimization methods cannot be used for optimizing the user association as well as task and power allocation. One promising solution is to use machine learning algorithms [7] for the predictions of optimal user association. In particular, machine learning algorithms can train a learning model to find a relationship between the future optimal user association and the computational task that each user processes at current slot. Based on the predicted optimal user association, the BS can collect the data size of the computational task requested by its associated users and thus optimizing task and power allocation for the users.

The main contribution of this paper is a novel framework to dynamically minimize energy consumption for wireless users that request computational tasks that can be of different data size over time. We consider a cellular network with mobile edge computing (MEC) capabilities, in which the BSs must determine the optimal user association as well as task and power allocation so as to provide computational service to users. This joint user association as well as task and power allocation problem is formulated as an optimization problem whose goal is to minimize the energy consumption for task computation and transmission. To solve this optimization problem, a support vector machine (SVM)-based federated learning (FL) [8] [9] is proposed to determine the user association proactively. The proposed SVM-based FL algorithm allows the BS and users to cooperatively train a global SVM model that can predict the optimal user association without any transmission of historical user association results and the data size of the task requested by each user at different time slots. Given the predicted user

This work was supported in part by the National Natural Science Foundation of China under Grants 61671086, 61629101, and 61871041, by Beijing Natural Science Foundation and Municipal Education Committee Joint Funding Project under Grant KZ201911232046, by the 111 Project under Grant B17007, and by the U.S. National Science Foundation under Grant CNS-1739642.

association, the optimization problem of task and transmit power allocation can be simplified and solved by using a gradient descent algorithm. Simulations using real data on city cellular traffic from the OMNILab at Shanghai Jiao Tong University show that the proposed algorithm can reduce the users' energy consumption by up to 20.1% compared to the conventional centralized SVM method. *To the best of our knowledge, this is the first work that studies the use of support vector machine (SVM)-based FL to dynamically determine user association so as to minimize the energy consumption under a delay constraint for task processing.*

The rest of this paper is organized as follows. The system model and the problem formulation are described in Section II. Then, Section III discusses the proposed learning framework. In Section IV, numerical results are presented and discussed. Finally, conclusions are drawn in Section V.

II. System Model and Problem Formulation

Consider a cellular network that consists of a set $\mathcal{N}$ of N BSs serving a set $\mathcal{M}$ of M users over both uplink and downlink. In this model, the users are associated with the BSs via wireless cellular links and each BS is equipped with computational resources to provide communication and computation services to the users. In the network, the uplink is used to transmit the computational task that each user offloads to the BS while the downlink is used to transmit the computing result of the offloaded task. We assume that the size of the data that user m needs to process at time slot t is $z_{m,t}$ and user m can transmit a fraction of the data to its associated BS for data processing.

A. Transmission Model

Let B be the total bandwidth of BS n, which is assumed to be equal for all BSs. We assume that each BS will allocate its bandwidth equally among its associated users. In the downlink, the rate of data transmission from BS n to user m is given by:

$$c_{mn,t}^{\mathrm{D}}\left(\boldsymbol{u}_t^n\right)=\frac{B}{\left|\boldsymbol{u}_t^n\right|}\log_2\left(1+\frac{P_{\mathrm{B}}h_{mn}}{\sum\limits_{k\in\mathcal{N},k\neq n}P_{\mathrm{B}}h_{mk}+\sigma_N^2}\right),\qquad(1)$$

where $\boldsymbol{u}_t^n=[u_{1,t}^n,\ldots,u_{M,t}^n]$ with $u_{m,t}^n=1$ indicating that user m connects to BS n at time slot t, otherwise, we have $u_{m,t}^n=0$. Thus, $\left|\boldsymbol{u}_t^n\right|$ is the module of $\boldsymbol{u}_t^n$, which represents the number of users that need to transmit data to BS n at time slot t. P_{B} is the transmit power of each BS n, which is assumed to be equal for all BSs. $h_{mn}=g_{mn}d_{mn}^{-\delta}$ is the channel gain between user m and BS n where g_{mn} is a Rayleigh fading parameter, d_{mn} is the distance between user m and BS n, and δ is the path loss exponent. σ_N^2 represents the variance of the additive white Gaussian noise. For uplink, we assume that all users occupy the same channel with bandwidth B^{U}. Then, the rate of data transmission from user m to BS n at time slot t is given by:

$$c_{mn,t}^{\mathrm{U}}\left(p_{m,t}\right)=B^{\mathrm{U}}\log_2\left(1+\frac{p_{m,t}h_{mn}}{\sum\limits_{k\in\mathcal{M},k\neq m}p_{k,t}h_{kn}+\sigma_N^2}\right),\qquad(2)$$

where $p_{m,t}$ is the transmit power of user m at time slot t.

Based on (1) and (2), the downlink and uplink transmission delay between user m and BS n can be given by:

$$l_{mn,t}^{\mathrm{D}}\left(\beta_{m,t},\boldsymbol{u}_t^n\right)=\frac{\beta_{m,t}z_{m,t}}{c_{mn,t}^{\mathrm{D}}\left(\boldsymbol{u}_t^n\right)},\qquad(3)$$

$$l_{mn,t}^{\mathrm{U}}\left(\beta_{m,t},p_{m,t}\right)=\frac{\beta_{m,t}z_{m,t}}{c_{mn,t}^{\mathrm{U}}\left(p_{m,t}\right)},\qquad(4)$$

where $\beta_{m,t}z_{m,t}$ is the fraction of the data that user m transmits to BS n for data processing at time slot t with $\beta_{m,t}$ being a variable that determines the size of the data that each user m needs to transmit to the BS.

B. Computing Model

Given the data size $\beta_{m,t}z_{m,t}$, the time used by BS n to process the task offloaded from user m can be given by:

$$l_{mn,t}^{\mathrm{CB}}\left(\beta_{m,t}\right)=\frac{\omega^{\mathrm{B}}\beta_{m,t}z_{m,t}}{f^{\mathrm{B}}},\qquad(5)$$

where f^{B} is the frequency of the central processing unit (CPU) clock of each BS n, which is assumed to be equal for all of the BSs. ω^{B} represents the number of CPU cycles required for computing data (per bit). We also assume that the BSs have enough computational resources to process the data transmitted from their associated users. Similarly, the computing time that user m uses to process the size of data $\left(1-\beta_{m,t}\right)z_{m,t}$ locally will be:

$$l_{mn,t}^{\mathrm{CU}}\left(\beta_{m,t}\right)=\frac{\omega_m^{\mathrm{U}}\left(1-\beta_{m,t}\right)z_{m,t}}{f_m^{\mathrm{U}}},\qquad(6)$$

where f_m^{U} is the frequency of the CPU clock of user m and ω_m^{U} is the number of CPU cycles required for computing the data (per bit) of user m.

C. Energy Consumption Model

In our model, the energy consumption of each user consists of three components: a) Device operation energy consumption, b) Data transmission energy consumption, and c) Data computing energy consumption. The energy consumption of user m at time slot t can be given by [10]:

$$\begin{aligned}&e_{m,t}\left(\beta_{m,t},p_{m,t}\right)\\&=O_m+\varsigma\left(f_m^{\mathrm{U}}\right)^2\left(1-\beta_{m,t}\right)z_{m,t}+p_{m,t}l_{mn,t}^{\mathrm{U}}(\beta_{m,t},p_{m,t}),\end{aligned}\qquad(7)$$

where O_m is the energy needed for device operation and ς is the energy consumption coefficient depending on the chip of user m's device. In (7), $\varsigma\left(f_m^{\mathrm{U}}\right)^2\left(1-\beta_{m,t}\right)z_{m,t}$ is the energy consumption of user m used for computing the size of data $\left(1-\beta_{m,t}\right)z_{m,t}$ at its own device and $p_m l_{mn,t}^{\mathrm{U}}\left(\beta_{m,t},p_{m,t}\right)$ represents the energy consumption of data transmission from user m to BS n.

D. Problem Formulation

Having defined the system model, next, we formulate an optimization problem whose goal is to minimize the energy consumption of each user while satisfying the task processing delay requirement. Since the BSs can have continuous power supply, we do not consider the energy consumption of the BSs in our optimization problem. The minimization problem of the energy consumption for all users involves determining the BS that each user associates with, the size of the data that must be transmitted to the BS, and the uplink transmit power of each user for data transmission, which is given by:

$$\min_{\boldsymbol{\beta}_t,\boldsymbol{U}_t,\boldsymbol{p}_t}\sum_{m=1}^{M}e_{m,t}\left(\beta_{m,t},p_{m,t}\right)\qquad(8)$$

$$\text{s. t. } 0 \leqslant \beta_{m,t} \leqslant 1, \quad \forall m \in \mathcal{M}, \tag{8a}$$
$$0 \leqslant p_{m,t} \leqslant P_{\max}, \quad \forall m \in \mathcal{M}, \tag{8b}$$
$$\max\left\{l_{mn,t}^{\mathrm{D}}+l_{mn,t}^{\mathrm{U}}+l_{mn,t}^{\mathrm{CB}}, l_{mn,t}^{\mathrm{CU}}\right\} \leqslant \gamma, \ \forall m \in \mathcal{M}, n \in \mathcal{N}, \tag{8c}$$

where γ is the delay requirement of user m, which is assumed to be equal for all users. $\boldsymbol{\beta}_t = [\beta_{1,t}, \ldots, \beta_{M,t}]$, $\boldsymbol{U}_t = [\boldsymbol{u}_t^1, \ldots, \boldsymbol{u}_t^N]$, and $\boldsymbol{p}_t = [p_{1,t}, \ldots, p_{M,t}]$. $l_{mn,t}^{\mathrm{D}}$, $l_{mn,t}^{\mathrm{U}}$, $l_{mn,t}^{\mathrm{CB}}$, and $l_{mn,t}^{\mathrm{CU}}$ are simplified notations for $l_{mn,t}^{\mathrm{D}}(\beta_{m,t}, \boldsymbol{u}_t^n)$, $l_{mn,t}^{\mathrm{U}}(\beta_{m,t}, p_{m,t})$, $l_{mn,t}^{\mathrm{CB}}(\beta_{m,t})$, and $l_{mn,t}^{\mathrm{CU}}(\beta_{m,t})$. (8a) indicates that the data requested by each user can be cooperatively processed by both BSs and users. (8b) is a constraint on the maximum transmit power of each user m. (8c) is the delay requirement of user m. As the data size of the requested computational task varies, the BSs must dynamically adjust each user's association as well as task and power allocation to minimize each user's energy consumption. The problem in (8) is challenging to solve by conventional optimization algorithms since they require the information related to the users' computational task requests, which cannot be obtained before the user association is determined. Hence, we need a machine learning approach that can predict the user association. Based on the predicted optimal user association, the BS can collect the data size of the computational task requested by its associated users and hence, optimize task and power allocation for the users. User association can be considered as a binary classification problem and SVM methods are good at solving such problems, and, hence, we propose an SVM-based machine learning approach for predicting user association. However, centralized SVM methods require each user to transmit its historical user association and computational task requests to a central controller for training, which results in unnecessary network traffic and high energy consumption. Thus, we propose an SVM-based FL algorithm to determine the user association proactively so as to minimize the energy consumption. The proposed algorithm enables each user to use its local dataset to collaboratively train a global SVM model that can determine user association for all users while keeping the training data locally. Based on the proactive user association, the optimization problem in (8) can be simplified and solved by a gradient descent algorithm.

III. Machine Learning For Minimizing the Energy Consumption

In this section, an SVM-based FL algorithm is proposed for proactively determining the user association. Then, given the user association, we introduce a gradient descent algorithm to optimize task allocation and transmit power of each user so as to solve problem (8).

A. SVM-based FL for User Association

We propose an SVM-based FL algorithm for determining users' future association as shown in Fig. 1. In the proposed algorithm, the local SVM model trained at each user is used to build a relationship between each user's association and the data size of the task that the user must process at current time slot. Then, the BS generates a covariance matrix that can measure the difference among the local SVM models hence improving the local SVM model of each user. After that, each user obtains a global SVM model that can predict the optimal user

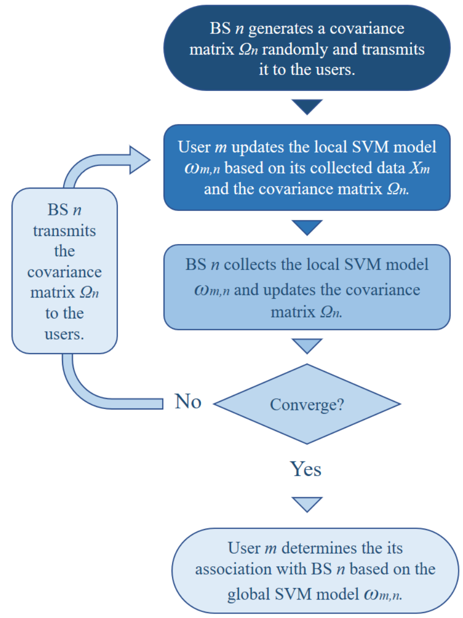

Fig. 1. The learning procedure of the SVM-based FL algorithm.

association. In this section, we first introduce the components of the SVM-based FL. Then, the training method of SVM-based FL algorithm is presented.

1) Components of the SVM-based FL: An SVM-based FL algorithm consists of five components: a) agents, b) user input, c) user output, d) BS input, and e) BS output, which are defined as follows:

- *Agents*: The agents in our system are the users and the BSs. An SVM-based FL algorithm is used to determine its optimal association with one of the BSs. For example, SVM algorithm n is used to determine the user association with BS n. However, it cannot be used to determine user association with other BSs. To determine the user association with N BSs individually, each user needs to perform N SVM-based FL algorithms. Due to the same components of each SVM-based algorithm, we only introduce the components of one SVM-based FL algorithm.
- *User Input*: The user input of the SVM-based FL algorithm is a matrix $\boldsymbol{\Omega}_n$ and local data samples $\boldsymbol{X}_m$ where $\boldsymbol{\Omega}_n \in \mathbb{R}^{M \times M}$ received from BS n is used to capture the relationships among the users' local SVM models and $\boldsymbol{X}_m$ includes historical user association as well as the data size of the task requested by each user at different time slots. Here, $\boldsymbol{X}_m = \{(\boldsymbol{x}_{m,1}, u_{m,1}^n), \ldots, (\boldsymbol{x}_{m,K_m}, u_{m,K_m}^n)\}$ with K_m being the number of the data samples of user m and for each data sample $(\boldsymbol{x}_{m,k_m}, u_{m,k_m}^n)$, $\boldsymbol{x}_{m,k_m} = [x_{m,k_m}^{\mathrm{X}}, x_{m,k_m}^{\mathrm{Y}}, z_{m,k_m}, \gamma]^{\mathrm{T}}$ with x_{m,k_m}^{X} and x_{m,k_m}^{Y} being the location of user m at current time slot.
- *User Output*: The output of training local SVM model is a vector $\boldsymbol{w}_{m,n}$ that represents the parameters related to the local SVM model and determines user m's association with BS n.
- *BS Input*: The SVM-based FL algorithm for BS n takes a matrix $\boldsymbol{W}_n = [\boldsymbol{w}_{1,n}, \ldots, \boldsymbol{w}_{M,n}]$ as input, where $\boldsymbol{w}_{m,n}$ is received from user m.
- *BS Output*: The output of the proposed algorithm for BS n is a structure matrix $\boldsymbol{\Omega}_n$ that is used to measure the

difference among the local SVM models of the users so as to generate the global SVM model hence enabling each user to accurately determine its association with BS n.

2) Training of SVM-based FL: The aim of training the SVM-based FL algorithm is to construct a global SVM model that can accurately determine all users' association with BS n, which is done in a way to solve [11]:

$$\min_{\boldsymbol{W}_n,\boldsymbol{\Omega}_n}\sum_{m=1}^{M}\sum_{k_m=1}^{K_m}\left\{l_{m,n}((\boldsymbol{w}_{m,n})^{\mathrm{T}}\boldsymbol{x}_{m,k_m},u_{m,k_m}^n)+\mathcal{R}(\boldsymbol{W}_n,\boldsymbol{\Omega}_n)\right\}, \tag{9}$$

$$\text{s. t. } \boldsymbol{\Omega}_n \succeq 0, \tag{9a}$$

$$\mathrm{tr}(\boldsymbol{\Omega}_n)=1, \tag{9b}$$

where $l_{m,n}((\boldsymbol{w}_{m,n})^{\mathrm{T}}\boldsymbol{x}_{m,k_m},u_{m,k_m}^n)=(u_{m,k_m}^n-(\boldsymbol{w}_{m,n})^{\mathrm{T}}\boldsymbol{x}_{m,k_m})^2$ is the loss functions that measures a squared error between the predict user association and the optimal user association. $\mathcal{R}(\boldsymbol{W}_n,\boldsymbol{\Omega}_n)=\lambda_1\left\|\boldsymbol{W}_n\right\|_F^2+\lambda_2\mathrm{tr}(\boldsymbol{W}_n(\boldsymbol{\Omega}_n)^{-1}(\boldsymbol{W}_n)^{\mathrm{T}})$ with $\lambda_1,\lambda_2>0$ is used to build a global SVM model where $\left\|\boldsymbol{W}_n\right\|_F^2$ penalizes the complexity of $\boldsymbol{W}_n$ to simplify the trained local SVM models and $\mathrm{tr}(\boldsymbol{W}_n(\boldsymbol{\Omega}_n)^{-1}(\boldsymbol{W}_n)^{\mathrm{T}}$ captures the relationships among local SVM models. In (9a), $\boldsymbol{\Omega}_n\succeq 0$ implies that matrix $\boldsymbol{\Omega}_n$ is positive semidefinite.

To solve the optimization problem in (9), we observe the following: a) Given $\boldsymbol{\Omega}_n$, updating $\boldsymbol{W}_n$ depends on the data pair $(\boldsymbol{x}_{m,k_m},u_{m,k_m}^n)$ which is recorded by the distributed users and b) Given $\boldsymbol{W}_n$, optimizing $\boldsymbol{\Omega}_n$ only depends on $\boldsymbol{W}_n$ and neither on data $\boldsymbol{x}_{m,k_m}$ nor u_{m,k_m}^n.

Based on these observations, it is natural to divide the training process of the proposed algorithm into two stages: a) A local SVM model training stage in which each user updates $\boldsymbol{W}_n$ using its local collected data and b) A global SVM model training stage in which BS n updates $\boldsymbol{\Omega}_n$ using its received local SVM models $\boldsymbol{W}_n$. Next, we introduce the two stages of the training process.

- *Local SVM Model Training Stage (at users)*: For the local SVM model training stage, users update $\boldsymbol{W}_n$ cooperatively based on the local dataset $\boldsymbol{X}_m$ and $\boldsymbol{\Omega}_n$ that is received from BS n. Given $\boldsymbol{\Omega}_n$, the optimization problem in (9) can be rewritten as:

$$\min_{\boldsymbol{W}_n}\sum_{k_m=1}^{K_m}\left\{l_{m,n}((\boldsymbol{w}_{m,n})^{\mathrm{T}}\boldsymbol{x}_{m,k_m},u_{m,k_m}^n)+\mathcal{R}(\boldsymbol{W}_n|\boldsymbol{\Omega}_n)\right\}, \tag{10}$$

  We calculate the gradients of (10) with respect to $\boldsymbol{W}_n$ and obtain [12]:

$$\boldsymbol{W}_n=\sum_{k_m=1}^{K_m}\boldsymbol{x}_{m,k_m}\boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{\Omega}_n(\lambda_1\boldsymbol{\Omega}_n+\lambda_2\boldsymbol{I}_m)^{-1}, \tag{11}$$

  where $\boldsymbol{I}_m\in\mathbb{R}^{M\times M}$ is an identity matrix and $\boldsymbol{e}_m$ is column m of $\boldsymbol{I}_m$. Here, column m of $\boldsymbol{W}_n$ is the output of the local SVM model of user m. Thus, for each user m, the update process of vector $\boldsymbol{w}_{m,n}$ is given by:

$$\boldsymbol{w}_{m,n}=\sum_{k_m=1}^{K_m}\boldsymbol{x}_{m,k_m}\boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{\Omega}_n(\lambda_1\boldsymbol{\Omega}_n+\lambda_2\boldsymbol{I}_m)^{-1}\boldsymbol{e}_m. \tag{12}$$

- *Global SVM model training stage (at a BS)*: For global SVM model training stage, BS n first collects $\boldsymbol{w}_{m,n}$ from each user m, and, hence, BS n will have $\boldsymbol{W}_n=[\boldsymbol{w}_{1,n},\ldots,\boldsymbol{w}_{M,n}]$. Based on $\boldsymbol{W}_n$, BS n calculates a structure matrix $\boldsymbol{\Omega}_n$ to measure the difference of the local SVM models among users to build a global SVM model that can analyze the relationship between user association and the data size of the task that each user needs to process so as to determine user association. Given $\boldsymbol{W}_n$, (9) can be rewritten as:

$$\min_{\boldsymbol{\Omega}_n}\mathrm{tr}(\boldsymbol{W}_n(\boldsymbol{\Omega}_n)^{-1}(\boldsymbol{W}_n)^{\mathrm{T}}), \tag{13}$$

$$\text{s. t. } \boldsymbol{\Omega}_n\succeq 0, \tag{13a}$$

$$\mathrm{tr}(\boldsymbol{\Omega}_n)=1. \tag{13b}$$

  From (13), we can see that compared to the standard FL algorithms [13] that directly average the learning parameters $\boldsymbol{W}_n$, the proposed FL algorithm uses a matrix $\boldsymbol{\Omega}_n$ to find the relationship among all users' association and, hence, improving the FL prediction performance. Given (13) and (13b), we have:

$$\begin{aligned}\mathrm{tr}(\boldsymbol{W}_n(\boldsymbol{\Omega}_n)^{-1}(\boldsymbol{W}_n)^{\mathrm{T}})&=\mathrm{tr}(\boldsymbol{W}_n(\boldsymbol{\Omega}_n)^{-1}(\boldsymbol{W}_n)^{\mathrm{T}})\mathrm{tr}(\boldsymbol{\Omega}_n)\\&\geqslant(\mathrm{tr}(\boldsymbol{\Omega}_n)^{-\frac{1}{2}}((\boldsymbol{W}_n)^{\mathrm{T}}\boldsymbol{W}_n)^{\frac{1}{2}}(\boldsymbol{\Omega}_n)^{\frac{1}{2}})^2\\&=(\mathrm{tr}((\boldsymbol{W}_n)^{\mathrm{T}}\boldsymbol{W}_n)^{\frac{1}{2}})^2,\end{aligned} \tag{14}$$

  where the inequality holds because of the Cauchy-Schwarz inequality for the Frobenius norm. Given (14), we have:

$$\boldsymbol{\Omega}_n=\frac{((\boldsymbol{W}_n)^{\mathrm{T}}\boldsymbol{W}_n)^{1/2}}{\mathrm{tr}(((\boldsymbol{W}_n)^{\mathrm{T}}\boldsymbol{W}_n)^{1/2})}. \tag{15}$$

  At each learning step, user m will update its local SVM model based on $\boldsymbol{X}_m$ and BS n will update $\boldsymbol{\Omega}_n$ based on $\boldsymbol{W}_n$. As the proposed algorithm converges, we can find the optimal $\boldsymbol{W}_n$ and $\boldsymbol{\Omega}_n$ to solve problem (9). The entire process of training the proposed SVM-based FL algorithm is shown in Algorithm 1. Note that, in our model, the energy consumption for training the SVM-based FL model is negligible [14]. This is because the proposed learning model can be used to predict the optimal user association in a sustainable period once the machine learning model completes the training process.

B. Optimization of Task Allocation and Power

Based on the FL model, each user will determine its user association, and hence, $\boldsymbol{U}_t$ can be determined. Given $\boldsymbol{U}_t$, the optimization problem in (9) can be simplified as:

$$\min_{\boldsymbol{\beta_t},\boldsymbol{p}_t}\sum_{m=1}^{M}e_{m,t}\left(\beta_{m,t},p_{m,t}\right) \tag{16}$$

$$\text{s. t. } 0\leqslant\beta_{m,t}\leqslant 1,\quad\forall m\in\mathcal{M}, \tag{16a}$$

$$0\leqslant p_{m,t}\leqslant P_{\max},\quad\forall m\in\mathcal{M}, \tag{16b}$$

$$\max\left\{l_{mn,t}^{\mathrm{D}}+l_{mn,t}^{\mathrm{U}}+l_{mn,t}^{\mathrm{CB}},l_{mn,t}^{\mathrm{CU}}\right\}\leqslant\gamma,\ \forall m\in\mathcal{M},n\in\mathcal{N}. \tag{16c}$$

To solve the optimization problem in (16), we first capture the relationship between $\beta_{m,t}$ and $p_{m,t}$ to simplify the problem in (16). Then we solve (16) using a gradient descent algorithm.

The delay requirement of user m in (16c) can be rewritten as:

$$l_{mn,t}^{\mathrm{D}}\left(\beta_{m,t}\right)+l_{mn,t}^{\mathrm{U}}\left(\beta_{m,t},p_{m,t}\right)+l_{mn,t}^{\mathrm{CB}}\left(\beta_{m,t}\right)\leqslant\gamma, \tag{17}$$

$$l_{mn,t}^{\mathrm{CU}}\left(\beta_{m,t}\right)\leqslant\gamma. \tag{18}$$

Note that the energy consumption for data transmission, $p_m l_{mn,t}^{\mathrm{U}}\left(\beta_{m,t},p_{m,t}\right)$ in (7) is a monotonically increasing function of $p_{m,t}$. Hence, to minimize the energy consumption for

Algorithm 1 Support Vector Machine-Based Federated Learning Framework

Input: Data $\boldsymbol{X}_m$ from $m = 1, \cdots, M$ users, stored on one of M users.
Initialize: $\boldsymbol{\Omega}_n$ is initially generated randomly via a uniform distribution.
for iterations $i = 0, 1, \cdots$ **do**
  for $m \in \{1, 2, \cdots, M\}$ in parallel over M users **do**
    For each user, calculating and returning $\boldsymbol{w}_{m,n}$ based on local dataset and $\boldsymbol{\Omega}_n$ in (12).
    Send updates $\boldsymbol{w}_{m,n}$ to BS n.
  end for
  BS n collects $\boldsymbol{W}_n$ from M users.
  Update $\boldsymbol{\Omega}_n$ centrally based on $\boldsymbol{W}_n$ in (15).
end for
Output: $\boldsymbol{W}_n := [\boldsymbol{w}_{1,n}, \boldsymbol{w}_{2,n}, \cdots, \boldsymbol{w}_{M,n}]$.

data transmission, user m must minimize the transmit power $p_{m,t}$, which leads to a decrease in the transmission rate and an increase in the transmission delay $l^{\mathrm{U}}_{mn,t}(\beta_{m,t}, p_{m,t})$. Hence, (18) holds the equality as $e_{m,t}(p_{m,t})$ is minimized. (18) can be rewritten as:

$$\frac{\beta_{m,t} z_{m,t}}{c^{\mathrm{D}}_{mn,t}} + \frac{\beta_{m,t} z_{m,t}}{c^{\mathrm{U}}_{mn,t}(p_{m,t})} + \frac{\omega^{\mathrm{B}} \beta_{m,t} z_{m,t}}{f^{\mathrm{B}}} = \gamma. \tag{19}$$

Substituting (3), (4), and (5) into (19), the relationship between $\beta_{m,t}$ and $p_{m,t}$ is given by:

$$\beta_{m,t}(p_{m,t}) = \frac{\gamma / z_{m,t}}{\frac{1}{c^{\mathrm{D}}_{mn,t}} + \frac{1}{c^{\mathrm{U}}_{mn,t}(p_{m,t})} + \frac{\omega^{\mathrm{B}}}{f^{\mathrm{B}}}}. \tag{20}$$

Given the relationship between $\beta_{m,t}$ and $p_{m,t}$, the optimization problem in (16) can be rewritten as follows:

$$\min_{\boldsymbol{p}_t} \sum_{m=1}^{M} e_{m,t}(p_{m,t}) \tag{21}$$

$$\text{s. t. } \frac{1}{c^{\mathrm{U}}_{mn,t}(p_{m,t})} \leqslant \frac{\gamma}{z_{m,t}} - \frac{1}{c^{\mathrm{D}}_{mn,t}} - \frac{\omega^{\mathrm{B}}}{f^{\mathrm{B}}}, \;\; \forall m \in \mathcal{M}, \tag{21a}$$

$$0 \leqslant p_{m,t} \leqslant P_{\max}, \quad \forall m \in \mathcal{M}, \tag{21b}$$

$$\max\{l^{\mathrm{D}}_{mn,t} + l^{\mathrm{U}}_{mn,t} + l^{\mathrm{CB}}_{mn,t}, l^{\mathrm{CU}}_{mn,t}\} \leqslant \gamma, \; \forall m \in \mathcal{M}, n \in \mathcal{N}. \tag{21c}$$

However, it is still difficult to optimize $p_{m,t}$ since the problem in (21) remains non-convex. Therefore, a gradient descent algorithm is used to obtain a suboptimal solution [15]. The Lagrange function of problem (21) will be:

$$\mathcal{L}(p_{m,t}) = \sum_{m=1}^{M} e_{m,t}(p_{m,t}) + \sum_{m=1}^{M} \mu_m \left(\gamma - l^{\mathrm{CU}}_{mn,t}(p_{m,t})\right), \tag{22}$$

with $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_M]$. We calculate the suboptimal values of $p_{m,t}$ using the gradient method:

$$p_{m,t} = (p_{m,t} - \tau \frac{\partial L(p_{m,t}, \mu_m)}{p_{m,t}}) \Big|_{0}^{P_{\max}}, \tag{23}$$

where $e\,|_b^a = \min\{\max\{e, b\}, a\}$ and τ is a dynamically chosen step-size sequence.

Given $p_{m,t}$, the value of $\boldsymbol{\mu}$ can be determined by the gradient method. The update procedure is:

$$\mu_m = \mu_m + \varDelta \left(\sum_{m=1}^{M} p_{m,t} - P_{\max} \right)^{+}, \tag{24}$$

where $[x]^+ = \max\{x, 0\}$ and $\varDelta$ is a dynamically chosen step-size sequence.

TABLE I
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
N	3	δ	4
M	8	σ_N^2	-95 dBm
B^{U}	3 MHz	ω^{U}	1500
B	10 MHz	ω^{B}	1500
P_{B}	5 W	f^{U}	0.5 GHz
$P_{\max}$	1 W	f^{B}	100 GHz

IV. Simulation Results and Analysis

In our simulations, a circular network area having a radius $r = 500$ m is considered with $M = 8$ uniformly distributed users and $N = 3$ uniformly distributed BSs. Without loss of generality, the channel gain follows a Rayleigh distribution with unit variance. The values of other parameters are defined in Table I. All statistical results are averaged over 5000 independent runs. Real data used to train the proposed algorithm is obtained from the OMNILab at Shanghai Jiao Tong University [16]. We consider the data size of cellular traffic in the dataset as the data size of each user's computational task. The optimal user associations used for training the SVM model to minimize the energy consumption of all users are obtained by exhaustive search. In simulations, we propose two baseline algorithms named SVM-based local learning and SVM-based global learning, respectively. The SVM-based local learning enables each user to train its local SVM model individually while the SVM-based global learning requires each user to transmit its local dataset to the BSs for training purpose.

Fig. 2 shows how the sum energy consumption changes as the number of users varies. From Fig. 2, we can see that the sum energy consumption increases as the number of users increases. This stems from the fact that, as the number of users increases, the number of tasks that users need to process increases, which increases the sum energy consumption for data transmission and local computation. Meanwhile, as the number of users increases, the uplink interference increases and hence, each user must increase its uplink transmit power to satisfy the delay requirement. Fig. 2 also shows that the proposed algorithm reduces the sum energy consumption by up to 35.5% compared to the solution with random user association, task and power allocation for a network with 12 users. This is because the proposed algorithm enables each user to dynamically adjust its association as well as task and power allocation as the data size of the computational task changes. Moreover, the proposed algorithm can achieve up to 20.1% gain in terms of energy consumption compared to the conventional centralized SVM method. This gain stems from the fact that the proposed algorithm enables each user to build the global SVM model cooperatively without transmitting the local training data samples to the BS hence reducing energy consumption for local data transmission.

In Fig. 3, we show how the accuracy rate changes as the number of data samples varies. In this figure, the accuracy rate is the probability with which the considered algorithms accurately predict the optimal user association. Clearly, as the number of data samples increases, the accuracy rate of all algorithms increases. This is due to the fact that, as the number of data samples increases, the probability of underfitting decreases and hence, the accuracy rate of all considered algorithms increases. Fig. 3 also shows that the proposed algorithm achieves only a

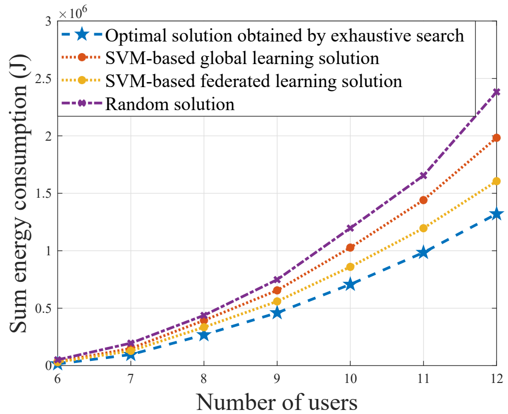

Fig. 2. Sum energy consumption as the number of users varies.

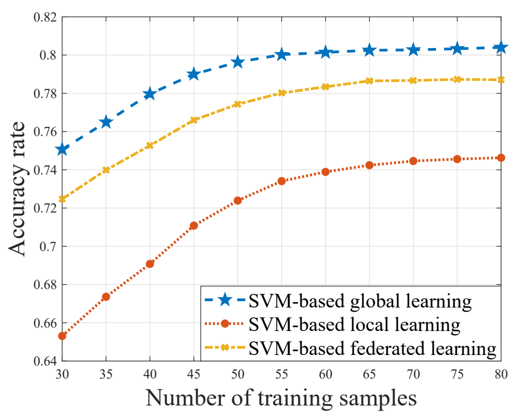

Fig. 3. Accuracy rate as the number of training samples varies.

3% accuracy gap compared to the SVM-based global learning. However, the SVM-based global learning algorithm requires each user to transmit all datasets to the BS for training purpose, which results in high energy consumption for data transmission.

Fig. 4 shows the number of iterations needed till convergence for all considered algorithms. In this figure, the loss function is used to measure the difference between the output values obtained by the trained model and actual values. From this figure, we can see that, as time elapses, the value of loss function for the considered algorithms decreases until convergence. Fig. 4 also shows that the proposed algorithm yields up to 72% gain in terms of the number of iterations needed to converge compared to SVM-based global learning. This implies that the proposed algorithm enables each user to train the learning model simultaneously and to generate the global SVM model, thus speeding up the convergence.

V. Conclusion

In this paper, we have studied the problem of minimizing energy consumption for task computation and transmission. We have formulated this problem as an optimization problem that seeks to minimize the total energy consumption while meeting the delay requirement of each user. To solve this problem, we have developed an SVM-based FL algorithm which enables each user to train the local SVM model using its own data and generate a global SVM model. The global SVM model can analyze the relationship between the future user association and the data size of the task that each user needs to process at current time slot so as to determine the user association proactively.

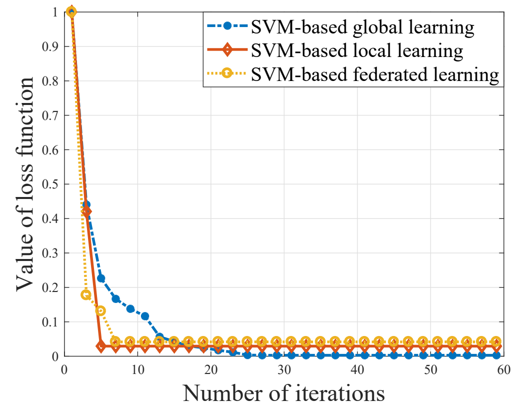

Fig. 4. Value of loss function as the total number of iterations varies.

Given the user association, the optimization problem can be simplified and solved by a gradient descent algorithm. Simulation results have shown that the proposed approach yields significant gains in terms of total energy consumption compared to conventional approaches.

References

[1] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE Network*, to appear, 2020.

[2] L. Zhang, T. Jiang, and K. Luo, “Dynamic spectrum allocation for the downlink of OFDMA-based hybrid-access cognitive femtocell networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1772-1781, Mar. 2016.

[3] L. Liu, Q. Fan, and R. Buyya, “A deadline-constrained multi-objective task scheduling algorithm in mobile cloud environments,” *IEEE Access*, vol. 6, pp. 55982-52996, Sept. 2018.

[4] S. Yu, R. Langar, X. Fu, L.Wang, and Z. Han, “Computation offloading with data caching enhancement for mobile edge computing,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11098-11112, Nov. 2018.

[5] S. Bi and Y. J. Zhang, “Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 4177-4190, Jun. 2018.

[6] X. Chen, L. Jiao, W. Li, and X. Fu, “Efficient multi-user computation offloading for mobile-edge cloud computing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795-2808, Oct. 2016.

[7] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, “Artificial neural networks-based machine learning for wireless networks: A tutorial,” *IEEE Communications Surveys and Tutorials*, to appear, 2019.

[8] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, “A joint learning and communications framework for federated learning over wireless networks,” *arXiv:1909.07972*, 2019.

[9] A. Ferdowsi and W. Saad, "Brainstorming generative adversarial networks (BGANs): Towards multi-agent generative models with distributed private datasets", *arXiv:2002.00306*, 2020.

[10] Y. Pan, C. Pan, Z. Yang, and M. Chen, “Resource allocation for D2D communications underlaying a NOMA-based cellular network,” *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 130-133, Feb. 2018.

[11] Y. Zhang and D. Y. Yeung, “A convex formulation for learning task relationships in multi-task learning,” in *Proc. of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, Catalina Island, USA. July. 2010.

[12] V. Smith, C. K. Chiang, M. Sanjabi, and A. Talwalkar, “Federated multi-task learning,” in *Proc. of Neural Information Processing Systems (NIPS)*, Long Beach, USA. Dec. 2017.

[13] M. Chen, H. V. Poor, W. Saad, and S. Cui, “Convergence time optimization for federated learning over wireless networks,” *arXiv:2001.07845*, 2020.

[14] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, “Energy efficient federated learning over wireless communication networks,” *arXiv:1911.02417*, 2019.

[15] S. Boyd and L. Vandenberghe, “Convex Optimization,” Combrage Univ Press, 2013.

[16] J. Long, “City Cellular Traffic Map (C2TM),” *Available Online: http://xiaming.me/city-cellular-traffic-map/*.