This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

Federated Learning for Task and Resource Allocation in Wireless High Altitude Balloon Networks

Sihua Wang, *Student Member, IEEE*, Mingzhe Chen, *Member, IEEE,*
Changchuan Yin, *Senior Member, IEEE*, Walid Saad, *Fellow, IEEE,*
Choong Seon Hong, *Senior Member, IEEE*, Shuguang Cui, *Fellow*, *IEEE*,
and H. Vincent Poor, *Fellow*, *IEEE*

***Abstract*—In this paper, the problem of minimizing energy and time consumption for task computation and transmission in mobile edge computing-enabled balloon networks is investigated. In the considered network, high-altitude balloons (HABs), acting as flying wireless base stations, can use their powerful computational abilities to process the computational tasks offloaded from their associated users. Since the data size of each user's computational task varies over time, the HABs must dynamically adjust their resource allocation schemes to meet the users' needs. This problem is posed as an optimization problem whose goal is to minimize the energy and time consumption for task computation and transmission by adjusting the user association, service sequence, and task allocation schemes. To solve this problem, a support vector machine (SVM)-based federated learning (FL) algorithm is proposed to determine the user association proactively. The proposed SVM-based FL method enables HABs to cooperatively build an SVM model that can determine all user associations without any transmissions of either user historical associations or computational tasks to other HABs. Given the predictions of the optimal user association, the service sequence and task allocation of each user can be optimized so as to minimize the weighted sum of the energy and time consumption. Simulations with real city cellular traffic data show that the proposed algorithm can reduce the weighted sum of the energy and time consumption of all users by up to 15.4% compared to a conventional centralized method.**

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

S. Wang and C. Yin are with the Beijing Laboratory of Advanced Information Network, and the Beijing Key Laboratory of Network System Architecture and Convergence, Beijing University of Posts and Telecommunications, Beijing 100876, China. Email: sihuawang@bupt.edu.cn; ccyin@ieee.org.

M. Chen and H. V. Poor are with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ, 08544, USA. Emails: mingzhec@princeton.edu; poor@princeton.edu

W. Saad is with the Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24060, USA, Email: walids@vt.edu.

C. S. Hong is with the Department of Computer Science and Engineering, Kyung Hee University, South Korea, Email: cshong@khu.ac.kr.

S. Cui is with the Shenzhen Research Institute of Big Data (SRIBD) and the Future Network of Intelligence Institute (FNii), Chinese University of Hong Kong, Shenzhen, 518172, China, Email: robert.cui@gmail.com.

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800802, in part by the Natural Science Foundation of China under Grant 61629101 and Grant 61671086, in part by the Beijing Natural Science Foundation under Grant KZ201911232046, in part by Beijing Laboratory Funding under Grant 2020BJLAB01, in part by the 111 Project under Grant B17007, in part by the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2021-2015-0-00742) supervised by the IITP, in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by Shenzhen Outstanding Talents Training Fund, by Guangdong Research Project No. 2017ZT07X152, and in part by U.S. National Science Foundation Grant CCF-1908308.

I. Introduction

High altitude balloons (HABs) are attracting increasing attention for future wireless communication networks, owing to their low deployment expense and large coverage range [1]. In particular, HABs can be used for various services including broadband Internet access, digital video/audio request, and emergency response [2]. To provide these services to ground users, HABs, acting as relays between ground users and base stations (BSs) as done in [3]–[5], must transmit computational tasks that are generated by the ground users to terrestrial BSs or the cloud via wireless backhaul links. Since the wireless resources that can be used for relaying ground user data to far-away BSs is limited, it is impractical for HABs to transmit all of their computational tasks to the BSs or the cloud. In addition, long-haul transmission will incur significant delays [6]. To reduce the task transmission delay and enable the HABs to process computational tasks locally, one can deploy mobile edge computing (MEC) at each HAB [7]. In particular, MEC-enabled HABs can directly process the computational tasks offloaded from the ground users without the need to transmit them to far-away BSs. However, deploying MEC at HABs faces many challenges such as energy efficiency of processing computational tasks, user association, and computational task allocation.

A. Related Works

A number of existing works [8]–[11] have studied important problems related to task offloading and computational resource optimization. In [8], the authors studied the use of MEC-enabled unmanned aerial vehicles (UAVs) to service ground users. The authors in [9] minimized the sum transmit power of UAVs and users via jointly optimizing users' transmit power and task allocation in a MEC network. In [10], the authors maximized the computation rate under energy constraints. The authors in [11] studied the deployment of UAVs so as to maximize the number of service users. However, the existing works in [8]–[11] did not consider the use of HABs to service ground users. Compared to UAVs with limited flight time [8]–[11], HABs can be tethered and equipped with powerful computing resources and, hence, they can continuously hover to serve ground users. Meanwhile, HABs can be deployed in the stratosphere to reduce the energy cost for hovering [12]. Meanwhile, these existing works [8]–[11] did not consider scenarios in which the data size of computational tasks requested by each user changes over time. As the data size of each computational task varies, each HAB must dynamically adjust user association, service sequence, and task allocation to minimize the ground users' energy and time consumption. For this purpose, each HAB must collect the users' computational task information. However, each computational task processed by HABs is offloaded from a ground user and, hence, each HAB must first determine user association so as to collect the users' computational task information and adjust service sequence as well as task allocation. In addition, each HAB can only collect the information related to the computational tasks of its associated users instead of the computational information from all users. Therefore, given only the computational task information of a subset of users, each HAB must use traditional iterative or distributed optimization methods, such as Lagrangian dual decomposition [13] or game theory [14], to find the globally optimal user association, thus resulting in additional overhead and delay for computational task processing. Moreover, if such known techniques are used, as the data size of each computational task varies, the HABs must rerun their iterative or distribution optimization algorithm to cope with this change thus increasing the time needed to minimize the energy and time consumption of ground users. To tackle this challenge, each HAB needs to predict the user association based on the historical information of the computational tasks. One promising solution is to use machine learning algorithms for the prediction 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 is currently executing. Based on the predicted optimal user association, the HABs can optimize service sequence and task allocation hence minimizing the energy and time consumption of each user.

Recently, a number of existing works such as in [15]–[17] used machine learning algorithms to solve resource optimization problems related to MEC. The work in [15] developed a deep learning method to optimize the user association scheme. In [16], the authors minimized the task processing delay using deep reinforcement learning. The authors in [17] designed a communication resource allocation scheme using reinforcement learning. However, most of these works [15]–[17] used centralized learning algorithms that require each distributed node to transmit its local dataset to a central controller for training. However, it is impractical to send all local datasets to a central controller in MEC-enabled HAB networks since the transmission of local datasets can lead to significant energy consumption. To address this challenge, one can use federated learning (FL) [18] that enables distributed devices to collaboratively train a machine learning model via sharing trained parameters with other devices instead of massive dataset. In [19], the problem of joint transmission power and resource allocation is solved by FL to reduce the queuing delay of all users. The work in [20] introduced an energy-efficient strategy for transmission and computation resource allocation under delay constraints. In [21], the authors proposed an FL algorithm to optimize resource allocation scheme in mobile edge computing based networks. However, the existing works in [19]–[21] directly averaged the learning models without the optimization of the parameters, thus degrading the FL performance. Therefore, it is necessary to develop a novel FL algorithm that can capture the relationships among HABs' user association schemes and can be implemented over HABs.

The main contribution of this paper is a novel framework for dynamically optimizing the energy and time consumption of wireless users in a MEC-enabled HAB network based on accurate predictions of the user association. Our key contributions include:

- We consider a MEC-enabled HAB network, in which the users request computational tasks that can be of different data size over time. To provide computing services to their users, the HABs must dynamically determine the optimal user association, service sequence, and task allocation. This joint user association, service sequence and task allocation problem is formulated as an optimization problem whose goal is to minimize the weighted sum of the energy and time consumption of all users.
- To solve this optimization problem, an SVM-based FL algorithm is proposed to determine the user association proactively. The proposed SVM-based FL algorithm allows the HABs to cooperatively train an optimal SVM model that can predict the optimal user association without any transmissions of historical user association results nor of the data size of the task requested by each user.
- Given the predicted user association, the original, non-convex problem is divided into two sub-optimization problems: service sequence optimization problem and task allocation optimization problem, which are solved iteratively. In particular, given the task allocation vector, we derive a closed-form expression for the optimal service sequence. Given the optimal service sequence, the task allocation optimization problem can be transformed to a piecewise linear problem which can be solved by linear programming.

Simulations using real data show that the proposed algorithm can reduce the weighted sum of the energy and time consumption of all users by up to 16.1% compared to the conventional centralized SVM method. To 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 weighted sum of the energy and time consumption in a MEC-enabled HAB network*.

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 to predict user association. The optimization of service sequence and task allocation are determined in Section VI. In Section V, numerical results are presented and discussed. Finally, conclusions are drawn in Section VI.

II. System Model and Problem Formulation

Consider a MEC-enabled HAB network that consists of a set $\mathcal{N}$ of N HABs serving a set $\mathcal{M}$ of M users over both uplink and downlink in a given geographical area. In this model, the users are associated with the HABs via wireless links and each HAB is equipped with computational resources to provide communication and computing services to the users. For example, HABs can be equipped with computational resources for analyzing the optimal route from the current

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

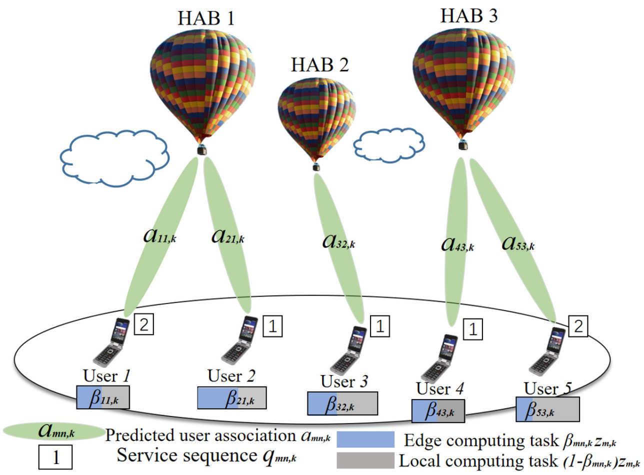


Fig. 1. An illustration of the considered MEC-enabled HAB network model.

TABLE I
LIST OF NOTATIONS

Notation	Description
N	Number of HABs
M	Number of users
H	Hight of each HAB
A	Attenuation due to clouds and rain
B	Bandwidth of each HAB
C	Speed of light
f_c	Carrier frequency
G_H	Antenna gain of each HAB
G_m	Antenna gain of user m
P_{B}	Transmit power of each HAB
P_{U}	Transmit power of each user
$z_{m,k}$	Data size of each computational task
$x^{\mathrm{X}}_{m,k}, x^{\mathrm{Y}}_{m,k}$	Coordinates of user m
$\boldsymbol{a}_{n,k}$	User association vector
$\boldsymbol{\beta}_{n,k}$	Task allocation vector
$\boldsymbol{q}_{n,k}$	Service sequence vector
$l^{\mathrm{U}}_{mn,k}$	Uplink transmission delay
$l^{\mathrm{D}}_{mn,k}$	Downlink transmission delay
$l^{\mathrm{B}}_{mn,k}$	Edge computing time
$l^{\mathrm{L}}_{mn,k}$	Local computing time
$l^{\mathrm{S}}_{mn,k}$	Access delay
$t_{mn,k}$	Total processing delay
$e_{m,k}$	Energy consumption of user m
$\boldsymbol{w}_{mn}$	SVM model
Ω_m	Structure matrix
η	Learning rate

location to the destination of each ground vehicle so as to provide navigation service to ground vehicles [22]. In this network, the uplink is used to transmit the computational task that each user offloads to the HAB while the downlink is used to transmit the computing result of the offloading task. We assume that the size of each task that user m needs to process in each time instant k is $z_{m,k}$, which will be changed as time elapses. Table I provides a summary of the notations used throughout this paper.

A. Transmission Model

In the considered scenario, the millimeter wave (mmWave) frequency bands are used to provide high data rate services for ground users so as to satisfy the delay requirement of computational tasks [23]. A time division multiple access (TDMA) scheme is adopted to support directional transmissions over the mmWave band as done in [24]. Note that, the channel gains of the mmWave links depend on the instantaneous large scale and small scale fading. For HAB-ground user transmission links (air-to-ground transmission links), the large scale fading is the free space path loss and attenuation due to rain and clouds [25]. Small scale fading is modeled as Ricean fading due to the presence of line-of-sight rays from the HAB to most of the locations in the HAB service area [26]. The channel gains $g_{mn,k}$ and $h_{mn,k}$ between HAB n and user m over uplink and downlink during each time instant k are given by [12]

$$g_{mn,k} = \left(\frac{C}{4\pi r_{mn} f_c}\right)^2 G_H(\Psi_{mn}) G_m A(d_{mn}) \varphi_{n,k}, \quad (1)$$

$$h_{mn,k} = \left(\frac{C}{4\pi r_{mn} f_c}\right)^2 G_H(\Psi_{mn}) G_m A(d_{mn}) \varphi_{m,k}, \quad (2)$$

where C is the speed of light, f_c is the carrier frequency, and r_{mn} is the distance between HAB n and user m; $G_H(\Psi_{mn}) = \cos(\Psi_{mn})^{\rho} \frac{32\log 2}{2(2\arccos(\sqrt[\rho]{0.5}))^2}$ is the gain seen at an angle Ψ_{mn} between user m and HAB n's boresight axis with ρ being the roll-off factor of the antenna. G_m is the antenna gain of user m. $A(r_{mn}) = 10^{\left(\frac{3\chi r_{mn}}{10H}\right)}$ is the attenuation due to clouds and rain with H being the height of a HAB and χ being the attenuation through the cloud and rain in dB/km. $\varphi_{n,k}$ and $\varphi_{m,k}$ represent the small scale Ricean gain during time instant k for HAB n and user m, respectively. Since a directional antenna is adopted at each HAB, the connectivity between a given HAB and a given user can be available for data transmission only if the directional antenna is directed towards the user and hence, interference is negligible. Given a bandwidth B for each HAB, the rates of data transmission for uplink and downlink between user m and HAB n during time instant k will be

$$u_{mn,k}\left(a_{mn,k}\right) = a_{mn,k} B \log_2\left(1+\frac{P_{\mathrm{U}} g_{mn,k}}{BN_0}\right), \quad (3)$$

$$d_{mn,k}\left(a_{mn,k}\right) = a_{mn,k} B \log_2\left(1+\frac{P_{\mathrm{B}} h_{mn,k}}{BN_0}\right), \quad (4)$$

where $a_{mn,k}$ is the index of the user association with $a_{mn,k} = 1$ indicating that user m connects to HAB n at time instant k; otherwise, we have $a_{mn,k} = 0$. P_{B} and P_{U} are the transmit power of each HAB and a user, which are assumed to be equal for all HABs and users, respectively. N_0 represents the noise power spectral density. The uplink and downlink transmission delay between user m and HAB n at time instant k can be given by

$$l^{\mathrm{U}}_{mn,k}\left(\beta_{mn,k}, a_{mn,k}\right) = \frac{\beta_{mn,k} z_{m,k}}{u_{mn,k}\left(a_{mn,k}\right)}, \quad (5)$$

$$l^{\mathrm{D}}_{mn,k}\left(\beta_{mn,t}, a_{mn,k}\right) = \frac{\beta_{mn,k} z_{m,k}}{d_{mn,k}\left(a_{mn,k}\right)}, \quad (6)$$

where $\beta_{mn,k} z_{m,k}$ is the fraction of the task that user m transmits to HAB n for processing in each time instant k with $\beta_{mn,k} \in [0,1]$ being the task allocation parameter. In (5) and (6), we assume that the size of computational task that each user needs the HAB to process and the size of the computational result are equal for simplification.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

B. Computational Model

In the considered model, edge computing refers the process of each HAB computing the partial computational tasks offloaded from the users while local computing refers to the process that each user computes the remaining computational task itself. Next, we introduce the models of edge computing and local computing in detail.

1) Edge computing model: Given the data size $\beta_{mn,k}z_{m,k}$ of the task that is offloaded from user m, the time used for HAB n to process the task can be given by

$$l_{mn,k}^{\mathrm{B}}\left(\beta_{mn,k}\right)=\frac{\omega^{\mathrm{B}}\beta_{mn,k}z_{m,k}}{f^{\mathrm{B}}}, \tag{7}$$

where f^{B} is the frequency of the central processing unit (CPU) clock of each HAB n. ω^{B} is the number of CPU cycles required for computing data (per bit).

2) Local computing model: Given the data size $(1-\beta_{mn,k})z_{m,k}$ of the task that is computed locally, the time used for user m to process the task can be given by

$$l_{mn,k}^{\mathrm{L}}\left(\beta_{mn,k}\right)=\frac{\omega_m^{\mathrm{U}}\left(1-\beta_{mn,k}\right)z_{m,k}}{f_m^{\mathrm{U}}}, \tag{8}$$

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.

Since users and HABs can process their computational tasks simultaneously, the total time used for the task computation is determined by the maximum time between the local computing time and edge computing time. Thus, based on (5)–(8), the time needed by user m and HAB n to cooperatively process the computational task of user m can be given by

$$\begin{aligned}l_{mn,k}(\beta_{mn,k},a_{mn,k})=\max\{&l_{mn,k}^{\mathrm{D}}\left(\beta_{mn,k},a_{mn,k}\right),\\&l_{mn,k}^{\mathrm{U}}(\beta_{mn,k},a_{mn,k})+l_{mn,k}^{\mathrm{B}}\left(\beta_{mn,k}\right)+l_{mn,k}^{\mathrm{L}}\left(\beta_{mn,k}\right)\},\end{aligned} \tag{9}$$

where $l_{mn,k}^{\mathrm{L}}\left(\beta_{mn,k}\right)$ represents the local computing time and $l_{mn,k}^{\mathrm{U}}(\beta_{mn,k},a_{mn,k})+l_{mn,k}^{\mathrm{B}}(\beta_{mn,k})+l_{mn,k}^{\mathrm{D}}(\beta_{mn,k},a_{mn,k})$ represents the edge computing time.

Since TDMA is used in the considered model, each user must wait for service, thus incurring a wireless access delay. For a given user m that is associated with HAB n, the access delay can be given by

$$l_{mn,k}^{\mathrm{S}}(q_{mn,k})=\sum_{m'\in\mathcal{Q}_m}l_{m'n,k}(a_{m'n,k},\beta_{m'n,k}), \tag{10}$$

where $q_{mn,k}$ is a service sequence variable that satisfies $1\leqslant q_{mn,k}\leqslant|\boldsymbol{a}_{n,k}|$. $|\boldsymbol{a}_{n,k}|$ is the module of $\boldsymbol{a}_{n,k}$ and represents the number of users that are associated with HAB n. $\mathcal{Q}_m=\{m'\in\mathcal{M}\,|q_{m'n,k}<q_{mn,k}\}$ is the set of users that are served by HAB n before user m. Given the access delay and processing delay of each user, the total delay for user m to process a computational task can be given by

$$t_{m,k}(\beta_{mn,k},a_{mn,k},q_{mn,k})=l_{mn,k}^{\mathrm{S}}(q_{mn,k})+l_{mn,k}(\beta_{mn,k},a_{mn,k}). \tag{11}$$

C. Energy Consumption Model

In our model, the energy consumption of each user consists of three components: a) Device operation, b) Data transmission, and c) Data computation. Here, the energy consumption of device operation is the energy consumption caused by the users using their devices for any applications. The energy consumption of user m can be given by [27]

$$\begin{aligned}&e_{m,k}\left(\beta_{mn,k},a_{mn,k}\right)\\&=O_m+\varsigma_m\left(f_m^{\mathrm{U}}\right)^2\left(1-\beta_{mn,k}\right)z_{m,k}+P_{\mathrm{U}}l_{mn,k}^{\mathrm{U}}\left(\beta_{mn,k},a_{mn,k}\right),\end{aligned} \tag{12}$$

where O_m is the energy consumption of device operation and ς_m is the energy consumption coefficient depending on the chip of user m's device. In (12), $\varsigma_m\left(f_m^{\mathrm{U}}\right)^2\left(1-\beta_{mn,k}\right)z_{m,k}$ is the energy consumption of user m computing the size of task $\left(1-\beta_{mn,k}\right)z_{m,k}$ at its own device and $P_{\mathrm{U}}l_{mn,k}^{\mathrm{U}}\left(\beta_{mn,k},a_{mn,k}\right)$ represents the energy consumption of task transmission from user m to HAB n.

Similarly, the energy consumption of each HAB n for computing the task offloaded by user m can be given by

$$\begin{aligned}&e_{mn,k}\left(\beta_{mn,k},a_{mn,k}\right)\\&=O_n+\varsigma\left(f^{\mathrm{B}}\right)^2\beta_{mn,k}z_{m,k}+P_{\mathrm{B}}l_{mn,k}^{\mathrm{D}}\left(\beta_{mn,k},a_{mn,k}\right),\end{aligned} \tag{13}$$

where O_n is the energy consumption of hover for HAB n and ς is the energy consumption coefficient depending on the chip of HAB's device. In (13), $\varsigma\left(f^{\mathrm{B}}\right)^2\beta_{mn,k}z_{m,k}$ is the energy consumption of HAB n computing the data size $\beta_{mn,k}z_{m,k}$ of task that is offloaded from user m and $P_{\mathrm{B}}l_{mn,k}^{\mathrm{D}}\left(\beta_{mn,k},a_{mn,k}\right)$ represents the energy consumption of task transmission from HAB n to user m.

D. Problem Formulation

Next, we formulate our optimization problem whose goal is to minimize the weighted sum of the energy and time consumption of each user. This minimization problem involves determining user association, service sequence, and the size of the data that must be transmitted to the HAB, as per the below formulation

$$\min_{\boldsymbol{A}_k,\boldsymbol{Q}_k,\boldsymbol{\beta}_k}\sum_{k=1}^{K}\sum_{m=1}^{M}(\gamma_{\mathrm{E}}e_{m,k}(\beta_{mn,k},a_{mn,k})+\gamma_{\mathrm{T}}t_{m,k}(\beta_{mn,k},a_{mn,k},q_{mn,k})) \tag{14}$$

$$\text{s. t. } a_{mn,k}\in\{0,1\},\forall n\in\mathcal{N},\forall m\in\mathcal{M}, \tag{14a}$$

$$\sum_{n\in\mathcal{N}}a_{mn,k}\leqslant 1,\forall m\in\mathcal{M}, \tag{14b}$$

$$1\leqslant q_{mn,k}\leqslant|\boldsymbol{a}_{n,k}|\,,q_{mn,k}\in\mathbb{Z}^+,\forall m\in\mathcal{M},\forall n\in\mathcal{N}, \tag{14c}$$

$$q_{mn,k}\neq q_{m'n,k},\forall m\neq m',m,m'\in\mathcal{M},\forall n\in\mathcal{N}, \tag{14d}$$

$$0\leqslant\beta_{mn,k}\leqslant 1,\forall m\in\mathcal{M},\forall n\in\mathcal{N}, \tag{14e}$$

$$\sum_{m=1}^{M}e_{mn,k}\left(\beta_{mn,k},a_{mn,k}\right)\leqslant E_k,\forall n\in\mathcal{N}, \tag{14f}$$

where $\boldsymbol{A}_k=[\boldsymbol{a}_{1,k},\ldots,\boldsymbol{a}_{N,k}]$ with $\boldsymbol{a}_{n,k}=[a_{1n,k},\ldots,a_{Mn,k}]$, $\boldsymbol{Q}_k=[\boldsymbol{q}_{1,k},\ldots,\boldsymbol{q}_{N,k}]$ with $\boldsymbol{q}_{n,k}=[q_{1n,k},\ldots,q_{mn,k}]$, and $\boldsymbol{\beta}_k=[\boldsymbol{\beta}_{1,k},\ldots,\boldsymbol{\beta}_{N,k}]$ with $\boldsymbol{\beta}_{n,k}=[\beta_{1n,k},\ldots,\beta_{Mn,k}]$. In (14), γ_{E} and γ_{T} are used to adjust the weights of energy consumption and time consumption according to the optimization purpose with $\gamma_{\mathrm{E}}+\gamma_{\mathrm{T}}=1$. (14a) and (14b) ensure that each user can connect to only one HAB for task processing. (14c) and (14d) guarantee that each HAB can only process one computational task at each time instant. (14e) indicates that the data requested by each user can be cooperatively processed by both a HAB and the user itself. (14f) is the energy constraint of HAB n at time instant k. The problem in (14) is challenging to solve by conventional optimization algorithms due to the following reasons. First, each HAB must collect the information related to the computational task requested by each user so as to minimize the energy and time consumption of ground users. However, each computational task is generated by a ground user and, hence, each HAB can only collect the information related to the computational tasks of its associated users instead of all users' computational information. When using optimization techniques, since each HAB only knows the computational task information of its associated users, it must use traditional iterative methods to find the globally optimal user association thus increasing the delay for processing computational task. Second, as the data size of each computational task varies, the HABs must re-execute the iterative methods which leads to additional delays and overhead. Thus, iterative algorithms cannot minimize the time consumption for processing the computational tasks of the users. To tackle this challenge, we need a machine learning approach that enables each HAB to generate a common learning model via using its collected information to predict the optimal user association. Based on the predicted optimal user association, each HAB can collect the data size of the computational task from its associated users thus optimizing service sequence and task allocation for the users. Note that, the user association index $a_{mn,k}$ is a binary variable and hence, the proposed user association problem is a single class classification problem. The SVM method is good at solving such single class classification problems since the parameters of SVM are optimized by convex quadratic programming that can find the optimal classification model with known data samples [28]. Thus, we propose an SVM-based FL algorithm to determine the user association proactively so as to minimize the energy and time consumption. The proposed algorithm enables each HAB to use its local dataset to collaboratively train an optimal SVM model that can determine user association. Based on the proactive user association, the optimization problem in (14) can be simplified and solved.

III. Federated Learning For Proactive User Association

Next, we introduce the training process of the SVM-based FL model for predicting user association. Using the proposed algorithm, each HAB first trains an SVM model locally using its locally collected data so as to build a relationship between each user's future association and the data size of the task that the user must process currently. Then, each HAB exchanges the trained SVM model with other HABs to aggregate the trained local SVM models and improve the SVM model locally so as to collaboratively perform a prediction for each user without training data exchange.

A. Components of the SVM-based FL

An SVM-based FL algorithm consists of four components: a) agents, b) input, c) output, d) SVM model, which are defined as follows:

- *Agents*: The agents are the HABs. Since each SVM-based FL algorithm typically performs prediction for just one user, each HAB must implement M SVM-based FL algorithms to determine the optimal user association for all users. Hereinafter, we introduce an SVM-based FL algorithm that HAB n used for the prediction of user m's future association.
- *Input*: The input of the SVM-based FL algorithm that is implemented by HAB n for predicting user m's future association is defined by $\boldsymbol{X}_{mn}$ that includes user m's hisrotical association and the data size of its requested task at historical time instants. Here, $\boldsymbol{X}_{mn} = \{(\boldsymbol{x}_{m,1}, a_{mn,2}), \ldots, (\boldsymbol{x}_{m,K}, a_{mn,K+1})\}$ where $K+1$ is the number of the time instants in which the data size of the computation task of each user m collected by HAB n and $\boldsymbol{x}_{m,k} = [x^{\mathrm{X}}_{m,k}, x^{\mathrm{Y}}_{m,k}, z_{m,k}]^{\mathrm{T}}$ with $x^{\mathrm{X}}_{m,k}$ and $x^{\mathrm{Y}}_{m,k}$ being the location of user m at current time instant, $a_{mn,k+1}$ is the index of the user association between user m and HAB n at time instant $k+1$.
- *Output*: The output of the proposed algorithm performed by HAB n for predicting user m's future association at time instant k is $a_{mn,k+1}$ that represents the user association between HAB n and user m at time instant $k+1$.
- *SVM model*: For each user m, we define an SVM model represented by a vector $\boldsymbol{w}_{mn}$ and a matrix $\boldsymbol{\Omega}_m \in \mathbb{R}^{N \times N}$ where $\boldsymbol{w}_{mn}$ is used to approximate the prediction function between the input $\boldsymbol{x}_{m,k}$ and the output $a_{mn,k+1}$ thus building the relationship between the future user association and the data size of the task that user m needs to process currently. $\boldsymbol{\Omega}_m$ is used to measure the difference between the SVM model generated by HAB n and the SVM models that are generated by other HABs. In fact, optimizing $\boldsymbol{\Omega}_m$ can improve the prediction performance of HAB n's local SVM model.

B. Training of SVM-based FL

Next, we introduce the training procedure of the SVM-based FL algorithm for the prediction of user m's future association. In particular, the training of the SVM-based FL algorithm is done in a way to solve [29]

$$\min_{\boldsymbol{W}_m, \boldsymbol{\Omega}_m} \sum_{n=1}^{N} \sum_{k=1}^{K} \{ l_n(\boldsymbol{w}_{mn}, (\boldsymbol{x}_{m,k}, a_{mn,k+1})) + \mathcal{R}(\boldsymbol{W}_m, \boldsymbol{\Omega}_m) \}, \tag{15}$$

$$\text{s. t.} \quad \boldsymbol{\Omega}_m \succeq 0, \tag{15a}$$

$$\mathrm{tr}(\boldsymbol{\Omega}_m) = 1, \tag{15b}$$

where $l_n((\boldsymbol{w}_{mn})^{\mathrm{T}}\boldsymbol{x}_{m,k}, a_{mn,k+1}) = (a_{mn,k+1} - (\boldsymbol{w}_{mn})^{\mathrm{T}}\boldsymbol{x}_{m,k})^2$ is a loss function that measures a squared error between the predicted user association and the target user association. $\mathcal{R}(\boldsymbol{W}_m, \boldsymbol{\Omega}_m) = \lambda_1 \|\boldsymbol{W}_m\|_F^2 + \lambda_2 \mathrm{tr}(\boldsymbol{W}_m (\boldsymbol{\Omega}_m)^{-1} (\boldsymbol{W}_m)^{\mathrm{T}})$ with $\lambda_1, \lambda_2 > 0$ is used to collaboratively build an SVM-based FL model for user m where $\|\boldsymbol{W}\|_F^2$ is used to perform L_2 regularization on each local model, and $\mathrm{tr}(\boldsymbol{W}_m (\boldsymbol{\Omega}_m)^{-1} (\boldsymbol{W}_m)^{\mathrm{T}})$ captures the relationship among the SVM models so as to improve the performance of SVM models that are used to determine user m's association. In (15a), $\boldsymbol{\Omega}_m \succeq 0$ implies that matrix $\boldsymbol{\Omega}_m$ is positive semidefinite and $\mathrm{tr}(\boldsymbol{\Omega}_m) = 1$ guarantees problem (15) is convex with respect to $\boldsymbol{\Omega}_m$.

To solve the optimization problem in (15), we observe the following: a) Given $\boldsymbol{\Omega}_m$, updating $\boldsymbol{W}_m$ depends on the data pair $(\boldsymbol{x}_{m,k}, a_{mn,k+1})$ which is collected by HAB n and b) Given $\boldsymbol{W}_m$, optimizing $\boldsymbol{\Omega}_m$ only depends on $\boldsymbol{W}_m$ and not on data pair $(\boldsymbol{x}_{m,k}, a_{mn,k+1})$. Based on these observations, it is natural to divide the training process of the proposed algorithm into two stages: a) $\boldsymbol{W}_m$ training stage in which HAB n updates $\boldsymbol{w}_{mn}$ using its local collected data and b) $\boldsymbol{\Omega}_m$ training stage in which HAB n first transmits $\boldsymbol{w}_{mn}$ to other HABs so as to generate $\boldsymbol{W}_m$ and then, calculates $\boldsymbol{\Omega}_m$ using its generated $\boldsymbol{W}_m$.

- $\boldsymbol{W}_m$ *training stage*: In this stage, HAB n updates $\boldsymbol{w}_{mn}$ based on the local dataset $\boldsymbol{X}_{mn}$ and $\boldsymbol{\Omega}_m$ that is calculated at last iteration. We first introduce the use of quadratic approximation to divide the optimization problem in (15) into distributed subproblems and then, the distributed subproblems that are solved by each HAB is presented. Given $\boldsymbol{\Omega}_m$, the dual problem of (15) can be rewritten as

$$\min_{\boldsymbol{\alpha}_m}\left\{D(\boldsymbol{\alpha}_m)=\sum_{n=1}^{N}\sum_{k=1}^{K}l_n^*(-\alpha_{mn,k})+R^*(\boldsymbol{X}_m\boldsymbol{\alpha}_m|\boldsymbol{\Omega}_m)\right\}, \tag{16}$$

where $l_n^*(-\alpha_{mn,k})=\max(-\alpha_{mn,k}\boldsymbol{w}_{mn}\boldsymbol{x}_{m,k}-l_n(\boldsymbol{w}_{mn}\boldsymbol{x}_{m,k}))$ and $R^*(\boldsymbol{X}_m\boldsymbol{\alpha}_m|\boldsymbol{\Omega}_m)=\max(\boldsymbol{X}_m\boldsymbol{\alpha}_m\boldsymbol{W}_m-\mathcal{R}(\boldsymbol{W}_m|\boldsymbol{\Omega}_m))$. In (16), $\boldsymbol{X}_m = \mathrm{Diag}[\boldsymbol{X}_{m1},\ldots,\boldsymbol{X}_{mN}]$ and $\boldsymbol{\alpha}_m = [\boldsymbol{\alpha}_{m1},\ldots,\boldsymbol{\alpha}_{mN}]$ where $\boldsymbol{\alpha}_{mn} = [\alpha_{mn,1},\ldots,\alpha_{mn,K}]$ with $\alpha_{mn,k}$ being the dual variable for the data sample $(\boldsymbol{x}_{m,k}, a_{mn,k+1})$. Note that, given dual variables $\boldsymbol{\alpha}_{mn}$, the primal variables $\boldsymbol{w}_{mn}$ can be found via $\boldsymbol{W}_m(\boldsymbol{\alpha}_m) = \nabla R^*(\boldsymbol{X}_m\boldsymbol{\alpha}_m|\boldsymbol{\Omega}_m)$ where $\boldsymbol{w}_{mn}$ is column n of $\boldsymbol{W}_m(\boldsymbol{\alpha}_m)$. To solve (16) in a distributed manner, we first define a local dual problem for each user. These local dual problems are used to approximate (16). Using a quadratic approximation, the local dual problem of each user m is

$$\begin{aligned}\min_{\Delta\boldsymbol{\alpha}_{mn}}\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}\,|\boldsymbol{\Omega}_m)=&\sum_{k=1}^{K}l_n^*(-\alpha_{mn,k}-\Delta\alpha_{mn,k})\\ &+\langle\boldsymbol{w}_{mn}(\boldsymbol{\alpha}_{mn}),\boldsymbol{X}_{mn}\Delta\boldsymbol{\alpha}_{mn}\rangle+\frac{\sigma}{2\mu_1}\|\boldsymbol{X}_{mn}\Delta\boldsymbol{\alpha}_{mn}\|^2\\ &+R^*(\boldsymbol{X}_n\boldsymbol{\alpha}_{mn}|\boldsymbol{\Omega}_m),\end{aligned} \tag{17}$$

where $\sigma=\max\limits_{\boldsymbol{\alpha}_{mn}\in\mathbb{R}^K}\frac{\|\boldsymbol{X}_m\boldsymbol{\alpha}_{mn}\|^2}{\sum_{n=1}^{N}\|\boldsymbol{X}_{mn}\boldsymbol{\alpha}_{mn}\|^2}\in(0,1)$ measures the correlation between HAB m's dataset and other HABs' datasets. $\Delta\boldsymbol{\alpha}_{mn} = [\Delta\alpha_{mn,1},\cdots,\Delta\alpha_{mn,K}]$ represents the difference between $\boldsymbol{\alpha}_m$ in (16) and $\boldsymbol{\alpha}_{mn}$ in (17). From (17), we can see that, to solve the local dual problem, we only need to use the data collected by each HAB n. Hence, the problem in (16) can be approximated by (17) and solved by each HAB in a distributed manner. To prove the convergence of the distributed algorithm, we first need to quantify the gap between $D(\boldsymbol{\alpha}_m)$ in (16) and $\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}\,|\boldsymbol{\Omega}_m)$ in (17), as shown in the following lemma whose proof follows directly from [29, Lemma 4].

Lemma 1. Given global dual variable $\boldsymbol{\alpha}_m$, $\Delta\boldsymbol{\alpha}_{mn}$, and learning rate η, the gap between $D(\boldsymbol{\alpha}_m)$ and $\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}\,|\boldsymbol{\Omega}_m)$ can be given by

$$\begin{aligned}&D(\boldsymbol{\alpha}_m+\eta\Delta\boldsymbol{\alpha}_m)\\ &\leqslant(1-\eta)D(\boldsymbol{\alpha}_m)+\eta\sum_{n=1}^{N}\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}\,|\boldsymbol{\Omega}_m).\end{aligned} \tag{18}$$

From Lemma 1, we can see that, at each iteration, as the global dual variable $\boldsymbol{\alpha}_m$ changes to $\boldsymbol{\alpha}_m + \eta\Delta\boldsymbol{\alpha}_m$, the value of $D(\boldsymbol{\alpha}_m)$ is bounded by the value of $\sum_{n=1}^{N}\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn},\boldsymbol{\Omega}_m)$. Given the relationship between $D(\boldsymbol{\alpha}_m)$ and $\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}\,|\boldsymbol{\Omega}_m)$, we can prove that $D(\boldsymbol{\alpha}_m)$ always converges to the optimal solution $D(\boldsymbol{\alpha}_m^*)$ using the results in [30, Lemma 7].

Lemma 2. Given a random initial solution $\boldsymbol{\alpha}_m^{(0)}$, the gap between the optimal value of $D(\boldsymbol{\alpha}_m^*)$ and the value of $D\left(\boldsymbol{\alpha}_m^{(h+1)}\right)$ obtained after $h+1$ iterations can be given by

$$\begin{aligned}&\mathbb{E}\left[\left(D\left(\boldsymbol{\alpha}_m^{(h+1)}\right)-D(\boldsymbol{\alpha}_m^*)\right)\right]\\ &\quad\leqslant(1-s\eta(1-\Theta))^{h+1}\left(D\left(\boldsymbol{\alpha}_m^{(0)}\right)-D(\boldsymbol{\alpha}_m^*)\right).\end{aligned} \tag{19}$$

where $\Theta=\frac{\mathbb{E}\left[\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn}^*;\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}^{(h)}|\boldsymbol{\Omega}_m)-\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn}^{(h)};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}^{(h)}|\boldsymbol{\Omega}_m)\right]}{\mathcal{G}_n^\sigma(\boldsymbol{0};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}^{(h)}|\boldsymbol{\Omega}_m)-\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn}^*;\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}^{(h)}|\boldsymbol{\Omega}_m)}$ and $s\in(0,1)$ represents the normalized distance between the value of $\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn}^{(h)};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}^{(h)}\,|\boldsymbol{\Omega}_m)$ and the optimal value of $\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn}^*;\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}^{(h)}\,|\boldsymbol{\Omega}_m)$ with $\Delta\boldsymbol{\alpha}_{mn}^* = \arg\max\limits_{\Delta\boldsymbol{\alpha}_{mn}\in\mathbb{R}^K}\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn}^{(h)};\boldsymbol{w}_{mn},\boldsymbol{\alpha}_n^{(h)},\boldsymbol{\Omega}_m)$ being the optimal solution.

From Lemma 2, we observe that as the number of iterations increases, the gap between $D(\boldsymbol{\alpha}_m^{(h+1)})$ that is obtained by the proposed algorithm and the optimal value of $D(\boldsymbol{\alpha}_m^*)$ decreases. Thus, the proposed algorithm will find a global optimal SVM model at convergence. Moveover, Lemma 2 shows that, the convergence speed depends on Θ. As Θ decreases, the gap between $\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_{mn}^*;\boldsymbol{w}_{mn},\boldsymbol{\alpha}_{mn}^{(h)}\,|\boldsymbol{\Omega}_m)$ and $\mathcal{G}_n^\sigma(\Delta\boldsymbol{\alpha}_n;\boldsymbol{w}_{mn},\boldsymbol{\alpha}_n\,|\boldsymbol{\Omega}_m)$ decreases, and hence, the number of iterations needed for convergence decreases.

- $\boldsymbol{\Omega}_m$ *training stage*: In this stage, each HAB n first transmits $\boldsymbol{w}_{mn}$ to other HABs and generates $\boldsymbol{W}_m$. Based on $\boldsymbol{W}_m$, each HAB n calculates a structure matrix $\boldsymbol{\Omega}_m$ to measure the difference of the SVM models that HABs used to predict the association of user m thus improving the SVM model used to predict the association. Given $\boldsymbol{W}_m$, (15) can be rewritten as

$$\min_{\boldsymbol{\Omega}_m}\mathrm{tr}(\boldsymbol{W}_m(\boldsymbol{\Omega}_m)^{-1}(\boldsymbol{W}_m)^{\mathrm{T}}), \tag{20}$$

$$\text{s. t. } \boldsymbol{\Omega}_m\succeq 0, \qquad \mathrm{tr}(\boldsymbol{\Omega}_m)=1. \tag{20a}$$

From (20), we can see that, compared to the standard FL algorithm in [31] that directly averages the learning parameters $\boldsymbol{W}_m$, the proposed FL algorithm uses a matrix $\boldsymbol{\Omega}_m$ to find the relationship among all HABs' user association schemes. This approach can, in turn, improve the FL prediction performance. Given (20) and (20a), we have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

Algorithm 1 SVM Based FL Framework

Input: Historical imformation of user m $\boldsymbol{X}_m = \boldsymbol{X}_{m1}, \cdots, \boldsymbol{X}_{mN}$ where $\boldsymbol{X}_{mn}$ is stored on HAB n.
Initialize: $\boldsymbol{\Omega}_m$ is initially generated randomly via a uniform distribution. $\boldsymbol{\alpha}^{(0)} := \mathbf{0} \in \mathbb{R}^n$.
for iterations $i = 0, 1, \cdots$ **do**
    for each HAB **do**
        Calculate $\Delta\boldsymbol{\alpha}_{mn}$ of the local subproblem in (17).
        Update local variables $\boldsymbol{\alpha}_{mn} \leftarrow \boldsymbol{\alpha}_{mn} + \Delta\boldsymbol{\alpha}_{mn}$.
        Return updates $\boldsymbol{w}_{mn}$.
        Broadcast $\boldsymbol{w}_{mn}$ and collect trained SVM models from other HABs, save as $\boldsymbol{W}_m$.
    end for
    Update $\boldsymbol{\Omega}_m$ based on $\boldsymbol{W}_m$ for latest $\boldsymbol{\alpha}_{mn}$.
end for
Output: $\boldsymbol{W}_m := [\boldsymbol{w}_{m1}, \boldsymbol{w}_{m2}, \ldots, \boldsymbol{w}_{mN}]$.

$$\begin{aligned}\operatorname{tr}(\boldsymbol{W}_m(\boldsymbol{\Omega})^{-1}(\boldsymbol{W}_m)^{\mathrm{T}}) &= \operatorname{tr}(\boldsymbol{W}_m(\boldsymbol{\Omega}_m)^{-1}(\boldsymbol{W}_m)^{\mathrm{T}})\operatorname{tr}(\boldsymbol{\Omega}_m) \\ &\geqslant (\operatorname{tr}(\boldsymbol{\Omega}_m)^{-\frac{1}{2}}((\boldsymbol{W}_m)^{\mathrm{T}}\boldsymbol{W}_m)^{\frac{1}{2}}(\boldsymbol{\Omega}_m)^{\frac{1}{2}})^2 \\ &= (\operatorname{tr}((\boldsymbol{W}_m)^{\mathrm{T}}\boldsymbol{W}_m)^{\frac{1}{2}})^2,\end{aligned} \tag{21}$$

where the inequality holds due to the Cauchy-Schwarz inequality for the Frobenius norm. Moreover, $\operatorname{tr}(\boldsymbol{W}_m(\boldsymbol{\Omega}_m)^{-1}(\boldsymbol{W}_m)^{\mathrm{T}})$ achieves its minimum value $(\operatorname{tr}((\boldsymbol{W}_m)^{\mathrm{T}}\boldsymbol{W}_m)^{\frac{1}{2}})^2$ if and only if $(\boldsymbol{\Omega}_n)^{-\frac{1}{2}}((\boldsymbol{W}_m)^{\mathrm{T}}\boldsymbol{W}_m)^{\frac{1}{2}} = a\boldsymbol{\Omega}_m$ for some constant a and $\operatorname{tr}(\boldsymbol{\Omega}_n) = 1$. Given (21), we have

$$\boldsymbol{\Omega}_m = \frac{((\boldsymbol{W}_m)^{\mathrm{T}}\boldsymbol{W}_m)^{\frac{1}{2}}}{\operatorname{tr}(((\boldsymbol{W}_m)^{\mathrm{T}}\boldsymbol{W}_m)^{\frac{1}{2}})}, \tag{22}$$

At each learning step, HAB n first updates $\boldsymbol{w}_{mn}$ based on $\boldsymbol{X}_m$ and $\boldsymbol{\Omega}_m$, then transmits $\boldsymbol{w}_{mn}$ to other HABs and calculates $\boldsymbol{\Omega}_m$. Here, we will ignore the cost for exchanging $\boldsymbol{w}_{mn}$ due to the following reasons. First, the data size of $\boldsymbol{w}_{mn}$ can be neglected compared to the data size of each computational task. Second, once each HAB completes the training process, the trained local model $\boldsymbol{w}_{mn}$ can be used to predict the optimal user association in a sustainable period. As the proposed algorithm converges, the optimal $\boldsymbol{W}_m$ and $\boldsymbol{\Omega}_m$ can be found to solve problem (15). The entire process of training the proposed SVM-based FL algorithm is shown in Algorithm 1.

IV. Optimization of Service Sequence and Task Allocation

Once the user association is determined, the HABs can optimize the service sequence and task allocation for each user so as to solve (14). Note that the optimization of the service sequence and task allocation is independent for each HAB due to the following reasons. First, given the predicted user association, the set of users that is served by each HAB is determined and, hence, each HAB only needs to optimize the service sequence and task allocation scheme for its associated users. Second, each HAB is equipped with a directional antenna to communicate with the served users and, thus, the interference among HABs is negligible. Due to the independence between the optimization of the service sequence and task allocation of each HAB, problem (14) can be decoupled into multiple subproblems. Given the user association, problem (14) for HAB n at time slot t can be rewritten as

$$\min_{\boldsymbol{\beta}_{n,k},\boldsymbol{q}_{n,k}} \sum_{m=1}^{M} \left(\gamma_{\mathrm{E}} e_{m,k}\left(\beta_{mn,k}\right) + \gamma_{\mathrm{T}} t_{m,k}(\beta_{mn,k}, q_{mn,k})\right) \tag{23}$$

$$\text{s. t. } 1 \leqslant q_{mn,k} \leqslant \left|\boldsymbol{a}_{n,k}\right|, q_{mn,k} \in \mathbb{Z}^+, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{23a}$$

$$q_{mn,k} \neq q_{m'n,k}, \forall m \neq m', m, m' \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{23b}$$

$$0 \leqslant \beta_{mn,k} \leqslant 1, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{23c}$$

$$\sum_{m=1}^{M} e_{mn,k}\left(\beta_{mn,k}\right) \leqslant E_k, \forall n \in \mathcal{N}. \tag{23d}$$

Problem (23) is a mixed integer programming problem due to the discrete variable $q_{mn,k}$ and continuous variable $\beta_{mn,k}$. To solve (23), we present an iterative algorithm that first optimizes service sequence variable $\boldsymbol{q}_{n,k}$ with fixed task allocation parameter $\boldsymbol{\beta}_{n,k}$ and then finds the optimal task allocation parameter $\boldsymbol{\beta}_{n,k}$ with optimized service sequence variable $\boldsymbol{q}_{n,k}$.

A. Optimization of Service Sequence with Fixed Task Allocation

Since the energy consumption $e_{m,k}\left(\beta_{mn,k}\right)$ in (23) as well as constraints (23c) and (23d) are only determined by task allocation parameter $\boldsymbol{\beta}_{n,k}$, the optimization of service sequence with fixed task allocation $\boldsymbol{\beta}_{n,k}$ is expressed as

$$\min_{\boldsymbol{q}_{n,k}} \sum_{m=1}^{M} \gamma_{\mathrm{T}} t_{m,k}(\beta_{mn,k}, q_{mn,k}) \tag{24}$$

$$\text{s. t. } 1 \leqslant q_{mn,k} \leqslant \left|\boldsymbol{a}_{n,k}\right|, q_{mn,k} \in \mathbb{Z}^+, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{24a}$$

$$q_{mn,k} \neq q_{m'n,k}, \forall m \neq m', m, m' \in \mathcal{M}, \forall n \in \mathcal{N}. \tag{24b}$$

The optimal problem in (24) is combinatorial due to the integer variable $q_{mn,k}$. Due to the complexity of solving combinatorial problems, the computation is essentially impossible even for a modest-sized wireless network. To overcome this, we first capture the relationship between the total time consumption of all users and the service sequence variable $\boldsymbol{q}_{n,k}$, as shown in Proposition 1. Then, based on Proposition 1, the closed-form expression for the optimal service sequence can be given in Theorem 1.

Proposition 1. Given the data size of each computational task $z_{m,k}$, user association index $a_{mn,k}$, and service sequence index $q_{mn,k}$, the total time consumption of the users that are associated with HAB n will be

$$\sum_{m=1}^{|\boldsymbol{a}_n|} \gamma_{\mathrm{T}} t_{mn,k}(q_{mn,k}, \beta_{mn,k}) = \sum_{m=1}^{|\boldsymbol{a}_n|} \gamma_{\mathrm{T}}(|\boldsymbol{a}_n| - q_{mn,k} + 1) l_{mn,k}(\beta_{mn,k}). \tag{25}$$

Proof: See Appendix A. ■

From Proposition 1, we can see that the time consumption of each user linearly increases with the service sequence index $q_{mn,k}$. Given Proposition 1, we drive the closed-form

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

expression for the optimal service sequence, as shown in the following theorem:

Theorem 1. Given the data size of each computational task $z_{m,k}$, the user association index $a_{mn,k}$, and the task allocation parameter $\beta_{mn,k}$, if the time that HAB n used for processing the computational tasks of its associated users satisfies $l_{1n,k}(\beta_{1n,k}) \leqslant \ldots \leqslant l_{mn,k}(\beta_{mn,k}) \leqslant \ldots \leqslant l_{|\boldsymbol{a}_n|n,k}(\beta_{|\boldsymbol{a}_n|n,k})$, the optimal service sequence of HAB n is $q^*_{1n,k} < \ldots < q^*_{mn,k} < \ldots < q^*_{|\boldsymbol{a}_n|n,k}$.

Proof: See Appendix B. ∎

From Theorem 1, we can see that, the service sequence variable $\boldsymbol{q}_{n,k}$ depends on the time used for processing the computational task. Meanwhile, the optimal service sequence of each HAB is an ascending order of the processing delay among its associated users. Hence, a sorting algorithm, such as bubble sort can be used to find the optimal service sequence.

B. Optimization of Task Allocation with Fixed Service Sequence

Once service sequence $\boldsymbol{q}_{n,k}$ is determined, the weighted sum of the energy and time consumption of all users only depends on the task allocation parameter $\boldsymbol{\beta}_{n,k}$. Hence, given the optimized service sequence index $q^*_{mn,k}$, the optimization problem in (23) can be formulated as

$$\min_{\boldsymbol{\beta}_{n,k}} \sum_{m=1}^{M} \left(\gamma_{\rm E} e_{m,k}(\beta_{mn,k}) + \gamma_{\rm T}(|\boldsymbol{a}_n| - q^*_{mn,k} + 1) l_{m,k}(\beta_{mn,k})\right) \tag{26}$$

$$\text{s. t. } 0 \leqslant \beta_{mn,k} \leqslant 1, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{26a}$$

$$\sum_{m=1}^{M} e_{mn,k}(\beta_{mn,k}) \leqslant E_k, \forall n \in \mathcal{N}, \tag{26b}$$

where $|\boldsymbol{a}_n| - q^*_{mn,k} + 1$ is the coefficient depended on the optimal service sequence index $q^*_{mn,k}$ in Theorem 1. To simplify the objective function in (26), we have

$$\begin{aligned}
&\gamma_{\rm E} e_{m,k}(\beta_{mn,k}) + \gamma_{\rm T}(|\boldsymbol{a}_n| - q^*_{mn,k} + 1) l_{m,k}(\beta_{mn,k}) \\
&= \begin{cases} \gamma_{\rm E}(O_m + \psi_{m,k}(1-\beta_{mn,k}) + \kappa_{m,k}\beta_{mn,k}) + \gamma_{\rm T}\upsilon_{m,k}\pi_{m,k}\beta_{mn,k}, \\ \qquad \text{if } \pi_{m,k}\beta_{mn,k} > \mu_{m,k}(1-\beta_{mn,k}), \\ \gamma_{\rm E}(O_m + \psi_{m,k}(1-\beta_{mn,k}) + \kappa_{m,k}\beta_{mn,k}) + \gamma_{T}\upsilon_{m,k}\mu_{m,k}(1-\beta_{mn,k}), \\ \qquad \text{otherwise}, \end{cases}
\end{aligned}$$

where $\psi_{m,k} = \varsigma_m \left(f^{\rm U}_m\right)^2 z_{m,k}$, $\kappa_{m,k} = \frac{P_{\rm U} z_{m,k}}{u_{mn,k}}$, $\pi_{m,k} = \frac{z_{m,k}}{u_{mn,k}} + \frac{\omega^{\rm B} z_{m,k}}{f^{\rm B}} + \frac{z_{m,k}}{d_{mn,k}}$, $\mu_{m,k} = \frac{\omega^{\rm U}_m z_{m,k}}{f^{\rm U}_m}$, and $\upsilon_{m,k} = |\boldsymbol{a}_n| - q^*_{mn,k} + 1$ can be treated as positive constants at each time slot t. Obviously, the objective function in (26) is a piecewise polynomial function in which the piecewise point is $\hat{\beta}_{mn,k} = \frac{\mu_{m,k}}{\pi_{m,k} + \mu_{m,k}}$ obtained from $\pi_{m,k}\beta_{mn,k} = \mu_{m,k}(1-\beta_{mn,k})$. Given the piecewise point $\hat{\beta}_{mn,k}$, the first-order derivative of the objective function with respect to $\beta_{mn,k}$ in each segment can be given by

$$\begin{aligned}
&\frac{\partial\left(\gamma_{\rm E} e_{m,k}(\beta_{mn,k}) + \gamma_{\rm T}\upsilon_{m,k} l_{m,k}(\beta_{mn,k})\right)}{\partial \beta_{mn,k}} \\
&= \begin{cases} -\psi_m + \kappa_m - \gamma_T \upsilon_{m,k}\mu_m, & 0 < \beta_{mn,k} < \hat{\beta}_{mn,k}, \\ -\psi_m + \kappa_m + \gamma_T \upsilon_{m,k}\pi_m, & \hat{\beta}_{mn,k} < \beta_{mn,k} < 1. \end{cases}
\end{aligned} \tag{27}$$

From (27), we can see that, the optimization problem in (26) is a piecewise linear programming problem that can be rewritten as [32]

$$\min_{\boldsymbol{\beta}_{n,k}} \sum_{m=1}^{M} \left(\gamma_{\rm E} e_{m,k}(\beta_{mn,k}) + \gamma_{\rm T}(|\boldsymbol{a}_n| - q^*_{mn,k} + 1) l_{mn,k}(\beta_{mn,k})\right) \tag{28}$$

$$\text{s. t. } 0 \leqslant \beta_{mn,k} \leqslant 1, \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{28a}$$

$$\left(\frac{z_{m,k}}{u_{mn,k}} + \frac{\omega^{\rm B} z_{m,k}}{f^{\rm B}} + \frac{z_{m,k}}{d_{mn,k}}\right)\beta_{mn,k} \leqslant l_{mn,k}(\beta_{mn,k}), \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{28b}$$

$$\frac{\omega^{\rm U}_m z_{m,k}}{f^{\rm U}_m}(1-\beta_{mn,k}) \leqslant l_{mn,k}(\beta_{mn,k}), \forall m \in \mathcal{M}, \forall n \in \mathcal{N}, \tag{28c}$$

$$\sum_{m=1}^{M} \left(O_n + \varsigma\left(f^{\rm B}\right)^2 z_{m,k}\beta_{mn,k} + \frac{P_{\rm B} z_{m,k}}{d_{mn,k}}\beta_{mn,k}\right) \leqslant E_k, \forall n \in \mathcal{N}. \tag{28d}$$

Problem (28) is linear and convex since the objective functions and constraints are both convex and linear. Hence, problem (28) can be solved by linear programming [33].

Algorithm 2 Iterative Service Sequence and Task Allocation

1: **Input:** The historical computational task X_{mn}, height H of HABs, transmit power $P_{\rm B}$ and $P_{\rm U}$ of HABs and users.
2: **Initialize:** Set the initial task allocation $\boldsymbol{\beta}^{(0)}_{n,k}$, the initial service sequence $\boldsymbol{q}^{(0)}_{n,k}$, the tolerance ϵ, and the iteration number $e = 0$.
3: Input X_{mn} into **Algorithm 1** to predict the user association $\boldsymbol{a}_{n,k}$.
4: Given $\boldsymbol{a}_{n,k}$, each HAB collects the data size of each computational task $z_{m,k}$ from its associated user.
5: Compute objective function $F\left(\boldsymbol{\beta}^{0}_{n,k}, \boldsymbol{q}^{(0)}_{n,k}\right)$.
6: **repeat**
7: $e = e + 1$.
8: With fixed $\boldsymbol{\beta}^{(e-1)}_{n,k}$, obtain the optimal $\boldsymbol{q}^{(e)}_{n,k}$ of problem (24).
9: With fixed $\boldsymbol{q}^{(e)}_{n,k}$, obtain the optimal $\boldsymbol{\beta}^{(e)}_{n,k}$ of problem (26).
10: Compute $F\left(\boldsymbol{\beta}^{(e)}_{n,k}, \boldsymbol{q}^{(e)}_{n,k}\right)$.
11: **Until** $\left|F\left(\boldsymbol{\beta}^{(e-1)}_{n,k}, \boldsymbol{q}^{(e-1)}_{n,k}\right) - F\left(\boldsymbol{\beta}^{(e)}_{n,k}, \boldsymbol{q}^{(e)}_{n,k}\right)\right| \Big/ F\left(\boldsymbol{\beta}^{(e-1)}_{n,k}, \boldsymbol{q}^{(e-1)}_{n,k}\right) < \epsilon$.
12: **Output:** $\boldsymbol{\beta}_{n,k}$ and $\boldsymbol{q}_{n,k}$.

C. Convergence and Complexity of the Proposed Algorithms

Since the convergence of the proposed SVM-based FL algorithm has been analyzed in Lemma 2, we only need to analyze the convergence of the iterative algorithm used to solve problem (23) in this section. We first show that the objective function in (23) is nonincreasing when sequence $(\boldsymbol{\beta}_{n,k}, \boldsymbol{q}_{n,k})$ is updated, as follows.

Lemma 3. The objective function $F(\boldsymbol{\beta}_{n,k}, \boldsymbol{q}_{n,k}) = \sum_{m=1}^{M} \left(\gamma_{\rm E} e_{m,k}(\beta_{mn,k}) + \gamma_{\rm T} t_{m,k}(\beta_{mn,k}, q_{mn,k})\right)$ in (23) is nonincreasing when $(\boldsymbol{\beta}_{n,k}, \boldsymbol{q}_{n,k})$ is updated.

Proof: At each iteration e, the objective function $F(\boldsymbol{\beta}_{n,k}, \boldsymbol{q}_{n,k})$ satisfies the following inequalities

$$F\left(\boldsymbol{\beta}^{(e-1)}_{n,k}, \boldsymbol{q}^{(e-1)}_{n,k}\right) \overset{(a)}{\geqslant} F\left(\boldsymbol{\beta}^{(e-1)}_{n,k}, \boldsymbol{q}^{(e)}_{n,k}\right) \overset{(b)}{\geqslant} F\left(\boldsymbol{\beta}^{(e)}_{n,k}, \boldsymbol{q}^{(e)}_{n,k}\right). \tag{29}$$

Inequality (a) follows from the fact that $\boldsymbol{q}^{(e)}_{n,k}$ is the optimal service sequence with fixed task allocation $\boldsymbol{\beta}^{(e-1)}_{n,k}$ and inequality

TABLE II
SIMULATION PARAMETERS [25] [26]

Parameter	Value	Parameter	Value
B	10 MHz	f_m^{U}	0.5 GHz
P_{B}	20 W	f^{B}	5 GHz
P_{U}	0.5 W	f_c	28 GHz
γ_{E}	0.5	$\varphi_{n,k}$	-20 dB
γ_{T}	0.5	$\varphi_{m,k}$	-20 dB
ς_m	3.44×10^{-23}	χ	1.45 dB/km
ς	3.44×10^{-23}	ρ	65
ω^{U}	1500	N_0	-174 dBm/Hz
ω^{B}	1500	H	17 km

(b) is due to the fact that $\boldsymbol{\beta}_{n,k}^{(e)}$ is the optimal task allocation of problem (26) with fixed service sequence $\boldsymbol{q}_{n,k}^{(e)}$. ■

From Lemma 3, we can see that, $F(\boldsymbol{\beta}_{n,k}, \boldsymbol{q}_{n,k})$ is non-increasing after the update of task allocation and service sequence at each iteration. Furthermore, since $F(\boldsymbol{\beta}_{n,k}, \boldsymbol{q}_{n,k})$ is finitely lower-bounded by 0, the proposed iterative algorithm always converges to a suboptimal solution. The gap between the optimal solution and the suboptimal solution found by the proposed algorithm is determined by the value of initial variables $\boldsymbol{\beta}_{n,k}$ and $\boldsymbol{q}_{n,k}$ [34].

The complexity of the proposed algorithm lies in training the SVM-based FL model and iteratively updating service sequence variable $\boldsymbol{q}_{n,k}$ and task allocation parameter $\boldsymbol{\beta}_{n,k}$. For training the SVM-based FL model, at each iteration, the major complexity that lies in finding a suboptimal solution in each HAB is $\mathcal{O}(\frac{N}{\sqrt{\Theta}}\log_2(1/\Theta))$ [35, Theorem 4.4]. For optimizing service sequence variable $\boldsymbol{q}_{n,k}$ in (24), according to Theorem 1, the complexity of calculating $\boldsymbol{q}_{n,k}$ depends on the sorting algorithm, such as $\mathcal{O}(M^2)$ if bubble sort is used. For optimizing task allocation parameter $\boldsymbol{\beta}_{n,k}$ in (28), the complexity is $\mathcal{O}((2M)^3)$ [36, Theorem 1]. As a result, the proposed algorithm can run independently on the HABs.

V. SIMULATION RESULTS AND ANALYSIS

In our simulations, a MEC-enabled HAB network area having a radius $r = 2.5$ km is considered with $M = 12$ uniformly distributed users and $N = 4$ uniformly distributed HABs. According to ITU guidelines [37], the angular variation in the location of the HABs at 17 km is less than 10 for the worst-case user terminal and thus, the coverage radius of each HAB is less than 1.7 km. The values of other parameters are defined in Table II. All statistical results are averaged over 5,000 independent runs. We use the real city cellular traffic data [38] to train the proposed algorithm. The original traffic data consists of four components: a) the timestamp in UNIX epoch time k, b) the identity of each BS n, c) the number of users associated with a specific BS $M_{n,k}$, and d) the number of bytes associated with a specific BS $Z_{n,k}$. Given this original traffic data, we first calculate the average number of bytes requested by each user using the following equation $z_{m,k} = Z_{n,k}/M_{n,k}$. Then, we use $z_{m,k}$ to represent the data size of each user's computational task. The optimal user associations used for training the SVM model to minimize the utility function of all users are obtained by exhaustive search. The data size of each data sample for each user in a time instant is 100 bits and each HAB collects the information from

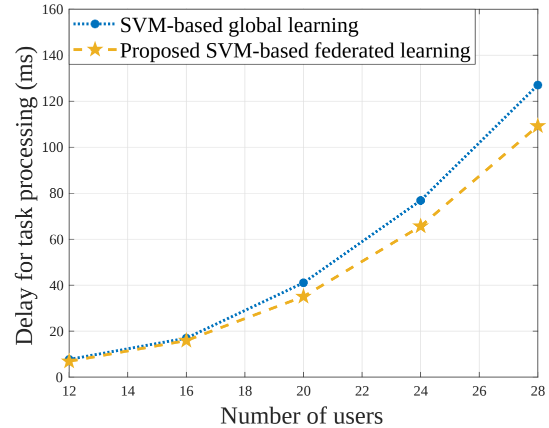


Fig. 2. Delay for task processing changes as the total number of users varies.

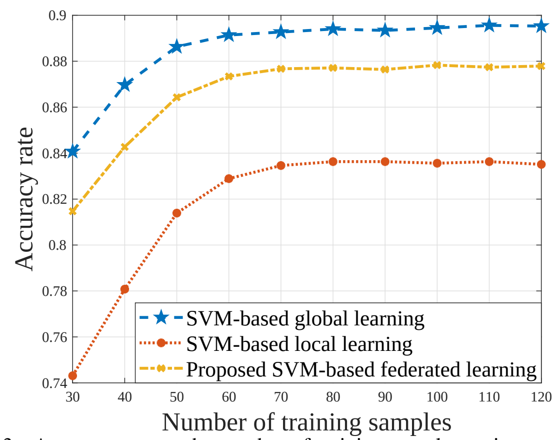


Fig. 3. Accuracy rate as the number of training samples varies.

its associated users in the last 120 time instants. In simulations, we consider two baseline algorithms named SVM-based local learning and SVM-based global learning, respectively. The SVM-based local learning enables each HAB to train its local SVM model individually while the SVM-based global learning requires each HAB to transmit its local dataset to other HABs for training purpose. Compared to the SVM-based local learning algorithm, the proposed algorithm requires each HAB to capture the relationship among SVM models thus improving the performance of the accuracy rate for user association prediction. Compared to the SVM-based global learning algorithm, the proposed SVM-based algorithm only requires the HABs to exchange the trained models. Since the data size of each trained SVM model is much smaller than the data size of the whole local dataset in each HAB, using the proposed algorithm can save time and energy consumption for data sample transmission.

Fig. 2 shows the time consumption of the proposed algorithm and the SVM-based global learning algorithm. From Fig. 2, we can see that, the proposed algorithm reduces the time consumption by up to 15.6% compared to the SVM-based global learning. This is because the SVM-based global learning algorithm requires each HAB to exchange the local dataset with other HABs hence increasing its training time consumption.

In Fig. 3, we show how the accuracy rate changes as the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

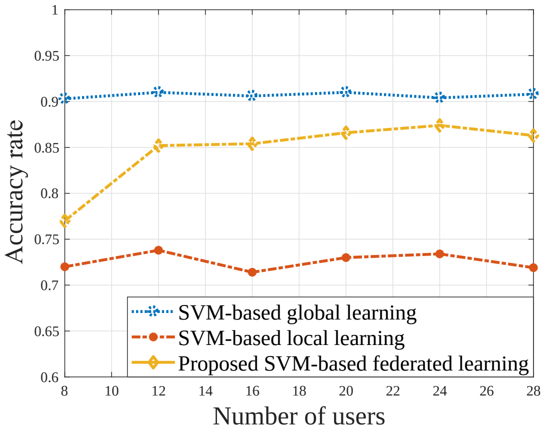


Fig. 4. Accuracy rate as the number of users varies.

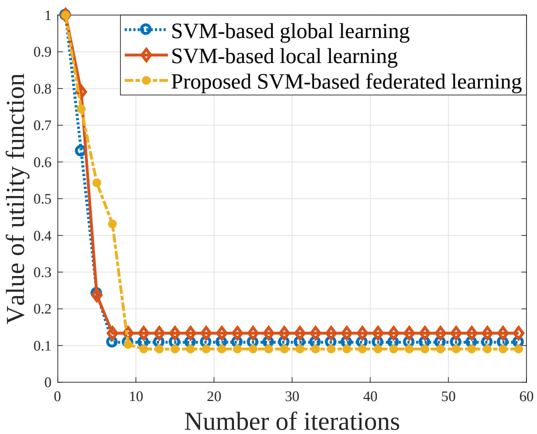
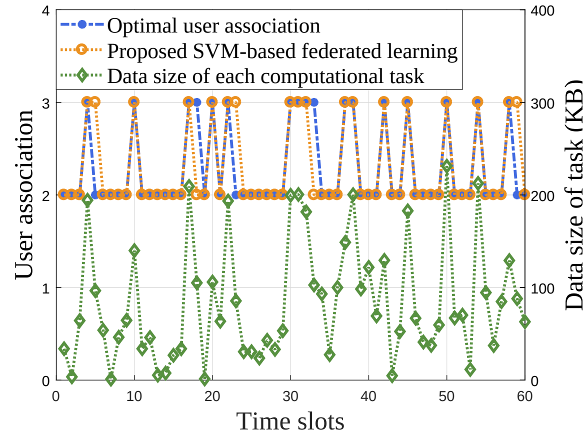
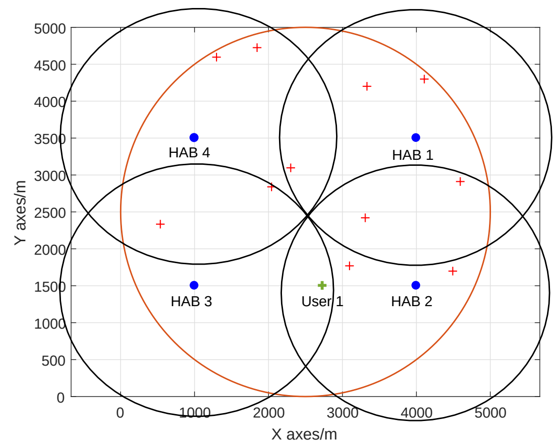


Fig. 5. Value of utility function as the total number of iterations varies.



(a) The prediction of the user association as the data size of the computational task varies.



(b) The map of the distribution of the HABs and users.

Fig. 6. An example of the prediction of the user association performed by the proposed algorithm.

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. From this figure, we see that, 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 2.6% accuracy gap compared to the SVM-based global learning. However, the SVM-based global learning algorithm requires each HAB to transmit all datasets to other HABs for training, which results in high overhead as well as significant energy and time consumption for data transmission.

Fig. 4 shows how the accuracy rate changes as the number of users varies. This figure shows, as the number of users increases, the accuracy rate of the proposed algorithm increases. This is due to the fact that, as the number of users increases, the average energy that is used to process the computational tasks of each user decreases and hence, the probability that each user changes its association increases. In consequence, the information of computational task from each user can be collected by different HABs, thus increasing the correlation between the datasets at the HABs, and the accuracy rate of the proposed algorithm. Fig. 4 also shows that the proposed algorithm yields up to 18.2% gain in terms of the accuracy rate compared to SVM-based local learning. This stems from the fact that the proposed algorithm enables all HABs to cooperatively build a global SVM model for the prediction of each user's association.

Fig. 5 shows the number of iterations needed till convergence for all considered algorithms. From this figure, we can see that, as time elapses, the value of the utility function for the considered algorithms decreases until convergence. Fig. 5 also shows that the proposed algorithm needs 16.7% more iterations needed to converge compared to the SVM-based global learning and SVM-based local learning. This is because the proposed algorithm requires the HABs to exchange their trained SVM parameters. Although exchanging the trained parameters increases the number of iterations needed to converge, the proposed algorithm can achieve a performance gain of up to 19.4% in terms of prediction performance compared to SVM-based local learning.

Fig. 6 shows an example of the prediction of the user association performed by the proposed algorithm for a network with 4 HABs and 12 users. In this figure, we can see that, as the data size of the computational task that is requested by user 1 varies, the prediction of the user association changes, as shown in Fig. 6(a). This implies that the proposed algorithm enables each HAB to predict the optimal user association

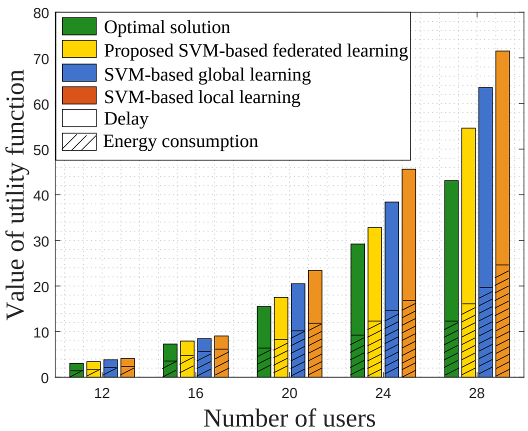


Fig. 7. Value of utility function as the total number of users varies.

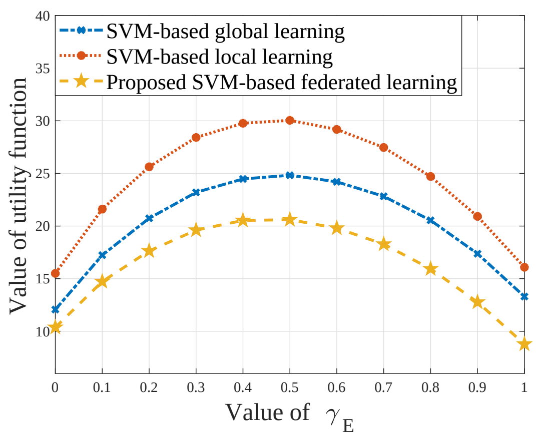


Fig. 8. Value of utility function as γ_E varies.

based on the data size of the computational task that user 1 needs to process currently. Specifically, the proposed algorithm can achieve up to 90% accuracy rate to predict the optimal user association. Fig. 6(a) also shows that user 1 connects to HAB 3 as long as the data size of the computational task is larger than 100 KB, and HAB 2, otherwise. This is due to the fact that as the data size of the computational task is smaller than 100 KB, user 1 associates with HAB 2 for task processing since HAB 2 is nearest to user 1 and have enough energy to process the computational task. Moveover, as the data size of the computational task increases, each HAB needs more energy and time to process the computational task that is offloaded from each user. However, from Fig. 6(b), we can see that, the number of users that are associated with HAB 3 is smaller than the number of users that are associated with HAB 2. Thus, as the data size of the computational task that is offloaded from user 1 increases, HAB 2 does not have enough energy to process the computational tasks from its associated users and hence, user 1 connects to HAB 3 for task processing.

Fig. 7 shows how the value of the utility function changes as the number of users varies. From Fig. 7, we can see that the value of utility function 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 and time consumption for task processing. Fig. 7 also shows that, as the number of users increases, the sum energy consumption increases linearly while the sum time consumption increases exponentially. This is because the sum energy consumption is linearly related to the number of users in the considered TDMA system while the sum of the access delay is exponentially related to the number of users. From Fig. 7, we can also see that the proposed algorithm reduces the value of the utility function by up to 16.1% and 28.3% compared to the SVM-based global learning and SVM-based local learning. These gains stem from the fact that the proposed algorithm enables each HAB to build the SVM model cooperatively without transmitting the local training data samples to the HAB hence reducing energy consumption for local data transmission while guaranteeing a better performance for optimal user association prediction.

Fig. 8 shows how the value of utility function changes as $\gamma_{\rm E}$ varies. From this figure, we can see that, as $\gamma_{\rm E}$ increases, the value of the utility function increases at first and then decreases. This is due to the fact that, as $\gamma_{\rm E}$ decreases to 0, each HAB only focuses on the minimization of time consumption. To this end, each HAB fully utilizes its limited energy to process the computational task requested by each user and thus, reducing the time consumption used for local computing. Meanwhile, as $\gamma_{\rm E}$ increases to 1, each HAB emphasizes on the minimization of energy consumption, which means that each HAB encourages the users to process its computational task locally and hence, reducing the energy consumption for computational task transmission.

Fig. 9 shows how the value of $\gamma_{\rm E}$ affects the energy consumption. From Fig. 9, we can see that, as $\gamma_{\rm E}$ increases, the energy consumption decreases. This is due to the fact that, as the value of $\gamma_{\rm E}$ increases, the proposed algorithm will focus on the minimization of energy consumption. Thus, the HABs encourage each user to process its own computational task locally so as to reduce the energy consumption for computational task transmission.

Fig. 10 shows how the time consumption changes as the value of $\gamma_{\rm T}$ varies. From Fig. 10, we can see that, as $\gamma_{\rm T}$ increases, the time consumption decreases. This is because as the value of $\gamma_{\rm T}$ increases, each HAB will focus on the minimization of time consumption. To this end, each HAB seeks to fully utilize its energy to compute the computational tasks thus, reducing the time used for task processing. Fig. 10 also shows that the proposed algorithm reduces the time used for computational task transmission by up to 25.7% and 34.3% compared to the SVM-based global learning and SVM-based local learning as $\gamma_{\rm T} = 1$. This implies that the proposed algorithm enables the HABs to exchange the trained SVM models so as to improve the prediction performance, thus finding the optimal user association and reducing the time used for computational task transmission.

VI. CONCLUSION

In this paper, we have studied the problem of minimizing energy and time consumption for task computation and transmission in a MEC-enabled balloon network. We have formulated this problem as an optimization problem that seeks to minimize the weighted sum of the energy and time consumption of all users. To solve this problem, we have developed an SVM-based FL algorithm which enables each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

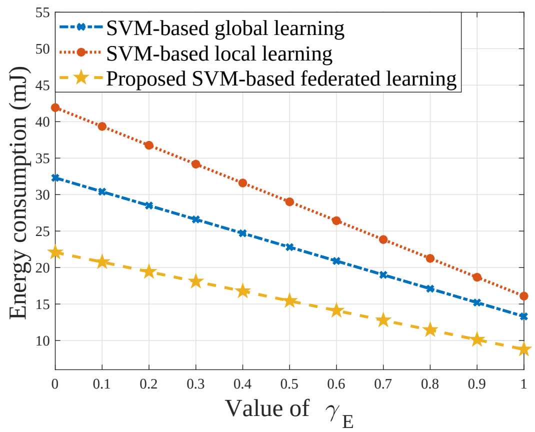


Fig. 9. Energy consumption as γ_{E} varies.

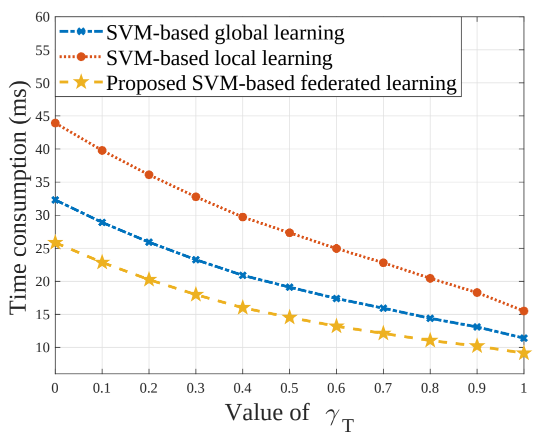


Fig. 10. Time consumption as γ_{T} varies.

HAB to cooperatively train an optimal SVM model using its own data. The 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. Based on the optimal prediction, the optimization of service sequence and task allocation are determined so as to minimize the energy and time consumption for task computing and transmission. Simulation results have demonstrated that the proposed approach yields significant gains in terms of sum energy and time consumption compared to conventional approaches.

Appendix

A. Proof of Proposition 1

The enumeration method is used to prove Proposition 1.

- If the number of users associated with HAB n is 1, i.e., $|\boldsymbol{a}_{n,k}| = 1$, then the sum delay for processing the computational task that is requested by user m is

$$\begin{aligned}
&\sum_{m=1}^{|\boldsymbol{a}_{n,k}|} \gamma_{\mathrm{T}} t_{mn,k}(q_{mn,k}, \beta_{mn,k}) \\
&= \gamma_{\mathrm{T}} l^{\mathrm{S}}_{mn,k}(q_{mn,k}) + \gamma_{\mathrm{T}} l_m(\beta_{mn,k}) \\
&= \gamma_{\mathrm{T}} l_m(\beta_{mn,k}) \\
&= \sum_{m=1}^{|\boldsymbol{a}_{n,k}|} \gamma_{\mathrm{T}}(|\boldsymbol{a}_{n,k}| - q_{mn,k}+1)\, l_{mn,k}(\beta_{mn,k}),
\end{aligned} \tag{30}$$

where the second equality follows from the fact that user m is served firstly and hence, it does not have access delay, i.e., $l^{\mathrm{S}}_{mn,k} = 0$ and the third equality is due to the fact that $|\boldsymbol{a}_{n,k}| = q_{mn,k} = 1$.

- If the number of users associated with HAB n is 2, i.e., $|\boldsymbol{a}_{n,k}| = 2$, then the sum delay for processing the computational tasks that are requested by the two users is given by

$$\begin{aligned}
\sum_{m=1}^{|\boldsymbol{a}_{n,k}|} \gamma_{\mathrm{T}} t_{mn,k}(q_{mn,k}, \beta_{mn,k}) = \gamma_{\mathrm{T}}\big(l^{\mathrm{S}}_{m'n,k}(q_{m'n,k}) + l_{m'n,k}(\beta_{m'n,k}) \\
+ l^{\mathrm{S}}_{mn,k}(q_{mn,k}) + l_{mn,k}(\beta_{mn,k})\big).
\end{aligned} \tag{31}$$

In (31), for the user m' that is served firstly, i.e., $q_{m'n,k} = 1$, we have $l^{\mathrm{S}}_{m'n,k}(q_{m'n,k}) = 0$. For the user that is served secondly, i.e., $q_{mn,k} = 2$, we have $l^{\mathrm{S}}_{mn,k}(q_{mn,k}) = l_{m'n,k}(\beta_{m'n,k})$. Therefore, (31) can be simplified as

$$\begin{aligned}
&\sum_{m=1}^{|\boldsymbol{a}_{n,k}|} \gamma_{\mathrm{T}} t_{mn,k}(q_{mn,k}, \beta_{mn,k}) \\
&= \gamma_{\mathrm{T}}(l_{m'n,k}(\beta_{m'n,k}) + l_{m'n,k}(\beta_{m'n,k}) + l_{mn,k}(\beta_{mn,k})) \\
&= \gamma_{\mathrm{T}}(2 l_{m'n,k}(\beta_{m'n,k}) + l_{mn,k}(\beta_{mn,k})) \\
&= \sum_{m=1}^{|\boldsymbol{a}_{n,k}|} \gamma_{\mathrm{T}}(|\boldsymbol{a}_{n,k}| - q_{mn,k}+1)\, l_{mn,k}(\beta_{mn,k}),
\end{aligned} \tag{32}$$

where the first equality is due to the fact that $l^{\mathrm{S}}_{m'n,k}(q_{m'n,k}) = 0$ and the third equality is obtained from the fact that $|\boldsymbol{a}_{n,k}| = 2$, $q_{m'n,k} = 1$, and $q_{mn,k} = 2$.

- Using the enumeration method, if the number of users that are associated with HAB n is $|\boldsymbol{a}_{n,k}|$, and the processing delay for each associated user is given, the sum delay for processing the computational tasks that are requested by the associated users is given by

$$\begin{aligned}
&\sum_{m=1}^{|\boldsymbol{a}_{n,k}|} \gamma_{\mathrm{T}} t_{mn,k}(q_{mn,k}, \beta_{mn,k}) \\
&= \sum_{m=1}^{|\boldsymbol{a}_{n,k}|} \gamma_{\mathrm{T}}(|\boldsymbol{a}_{n,k}| - q_{mn,k}+1)\, l_{mn,k}(\beta_{mn,k}),
\end{aligned} \tag{33}$$

where $|\boldsymbol{a}_{n,k}| - q_{mn,k}$ represents the number of users that need to be served when user m has already been served. Hence, $(|\boldsymbol{a}_{n,k}| - q_{mn,k}+1) l_{mn,k}(\beta_{mn,k})$ is user m's sum delay that consists of the processing delay of user m, $l_{mn,k}(\beta_{mn,k})$, and the access delay caused by user m, $(|\boldsymbol{a}_{n,k}| - q_{mn,k}) l_{mn,k}(\beta_{mn,k})$.

This completes the proof.

B. Proof of Theorem 1

We use contradiction method to prove Theorem 1. We assume that the time that HAB n used to process the computational tasks satisfies $l_{1n,k}(\beta_{1n,k}) \leqslant \ldots \leqslant l_{in,k}(\beta_{in,k}) \leqslant \ldots \leqslant l_{mn,k}(\beta_{mn,k}) \leqslant \ldots \leqslant l_{|\boldsymbol{a}_n|n,k}(\beta_{|\boldsymbol{a}_n|n,k})$. Hence, based on Theorem 1, we have $q^*_{1n,k} < \ldots < q^*_{(i-1)n,k} < q^*_{in,k} < q^*_{(i+1)n,k} < \ldots < q^*_{(m-1)n,k} < q^*_{mn,k} < q^*_{(m+1)n,k} < \ldots < q^*_{|\boldsymbol{a}_n|n,k}$. The total time consumption of the users associated with HAB n can be given by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2021.3080078, IEEE Internet of Things Journal

$$\sum_{m=1}^{|\boldsymbol{a}_n|}\gamma_{\mathrm{T}}t_{mn,k}=\gamma_{\mathrm{T}}\left(\sum_{q_{jn,t}=1}^{q^*_{(i-1)n,t}}t_{jn,k}+t_{in,k}\left(q^*_{in,k},\beta_{in,k}\right)\right.$$
$$\left.+\sum_{q_{jn,t}=q^*_{(i+1)n,k}}^{q^*_{(m-1)n,k}}t_{jn,k}+t_{mn,k}\left(q^*_{mn,k},\beta_{mn,k}\right)+\sum_{q_{jn,k}=q^*_{(m+1)n,k}}^{q_{|\boldsymbol{a}_n|n,k}}t_{jn,k}\right). \tag{34}$$

where $t_{jn,k}$ is short for $t_{jn,k}(q_{jn,k},\beta_{jn,k})$. When the optimal service index of user m is exchanged with the service index of user i, the service sequence will be $q^*_{1n,t}<\ldots<q^*_{(i-1)n,t}<q'_{mn,k}<q^*_{(i+1)n,t}<\ldots<q^*_{(m-1)n,t}<q'_{in,t}<q^*_{(m+1)n,t}<\ldots<q^*_{|\boldsymbol{a}_n|n,t}$ where $q'_{mn,k}=q^*_{in,k}$ and $q'_{in,t}=q^*_{mn,k}$. Thus, the total processing delay for all users can be given by

$$\sum_{m=1}^{|\boldsymbol{a}_n|}\gamma_{\mathrm{T}}t_{mn,k}=\gamma_{\mathrm{T}}\left(\sum_{q_{jn,t}=1}^{q^*_{(i-1)n,t}}t_{jn,k}+t_{mn,k}\left(q'_{mn,k},\beta_{mn,k}\right)\right.$$
$$\left.+\sum_{q_{jn,t}=q^*_{(i+1)n,k}}^{q^*_{(m-1)n,k}}t_{jn,k}+t_{in,k}\left(q'_{in,k},\beta_{in,k}\right)+\sum_{q_{jn,k}=q^*_{(m+1)n,k}}^{q_{|\boldsymbol{a}_n|n,k}}t_{jn,k}\right). \tag{35}$$

The gap between (34) and (35) is given by

$$\begin{aligned}
&(34)–(35)\\
&=\gamma_{\mathrm{T}}\left(t_{in,k}\left(q^*_{in,k},\beta_{in,k}\right)+t_{mn,k}\left(q^*_{mn,k},\beta_{mn,k}\right)\right.\\
&\qquad\left.-t_{in,k}\left(q'_{in,k},\beta_{in,k}\right)-t_{mn,k}\left(q'_{mn,k},\beta_{mn,k}\right)\right)\\
&=\gamma_{\mathrm{T}}\left(t_{in,k}\left(q^*_{in,k}-q'_{in,k},\beta_{in,k}\right)+t_{mn,k}\left(q^*_{mn,k}-q'_{mn,k},\beta_{mn,k}\right)\right)\\
&=\gamma_{\mathrm{T}}\left(\left(q^*_{in,k}-q'_{in,k}\right)l_{in,k}\left(\beta_{in,k}\right)+\left(q^*_{mn,k}-q'_{mn,k}\right)l_{mn,k}\left(\beta_{mn,k}\right)\right)\\
&=\gamma_{\mathrm{T}}\left(q^*_{mn,k}-q^*_{in,k}\right)\left(l_{mn,k}\left(\beta_{mn,k}\right)-l_{in,k}\left(\beta_{in,k}\right)\right)
\end{aligned}$$

where the last equation follows from the fact that $q'_{mn,k}=q^*_{in,k}$ and $q'_{in,k}=q^*_{mn,k}$. Since $q^*_{in,k}<q^*_{mn,k}$ and $l_{in,k}\left(\beta_{in,k}\right)\leqslant l_{mn,k}\left(\beta_{mn,k}\right)$, we have $(34)-(35)\geqslant 0$. Therefore, as the optimal service sequence of each HAB n changes, the time $\sum\limits_{m=1}^{|\boldsymbol{a}_n|}\gamma_{\mathrm{T}}t_{mn,k}$ needed for processing the computational task increases. This completes the proof.

References

[1] P. G. Sudheesh, M. Mozaffari, M. Magarini, W. Saad, and P. Muthuchidambaranathan, "Sum-rate analysis for high altitude platform (HAP) drones with tethered balloon relay," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1240–1243, Jun. 2018.

[2] X. Cao, P. Yang, M. Alzenad, X. Xi, D. Wu, and H. Yanikomeroglu, "Airborne communication networks: A survey," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1907–1926, Sept. 2018.

[3] Q. Wu, J. Xu, and R. Zhang, "Capacity characterization of UAV-enabled two-user broadcast channel," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1955–1971, Aug. 2018.

[4] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, Thirdquarter. 2019.

[5] A. Alzidaneen, A. Alsharoa, and M. Alouini, "Resource and placement optimization for multiple UAVs using backhaul tethered balloons," *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 543–547, Dec. 2020.

[6] Y. Liu, D. Grace, and P. D. Mitchell, "Exploiting platform diversity for GoS improvement for users with different high altitude platform availability," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 196–203, Jan. 2009.

[7] H. Kimm and H. Kimm, "Android payload system development for high-altitude balloons: Capturing periodic images along with GPS and sensor data," in *Proc. of IEEE International Congress on Electronics, Electrical Engineering and Computing (INTERCON)*, Piura, Peru, Aug. 2016.

[8] N. Zhao, X. Pang, Z. Li, Y. Chen, F. Li, Z. Ding, and M. Alouini, "Joint trajectory and precoding optimization for UAV-assisted NOMA networks," *IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3723–3735, May 2019.

[9] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4576–4589, Sept. 2019.

[10] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, Sept. 2018.

[11] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station for maximum coverage of users with different QoS requirements," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 38–41, Feb. 2018.

[12] A. Ibrahim and A. S. Alfa, "Using Lagrangian relaxation for radio resource allocation in high altitude platforms," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5823–5835, Oct. 2015.

[13] H. Zhang, S. Huang, C. Jiang, K. Long, V. C. M. Leung, and H. V. Poor, "Energy efficient user association and power allocation in millimeter-wave-based ultra dense networks with energy harvesting base stations," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 1936–1947, Sept. 2017.

[14] S. Moon, H. Kim, and Y. Yi, "Brute: Energy-efficient user association in cellular networks from population game perspective," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 663–675, Jan. 2016.

[15] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4692–4707, Oct. 2019.

[16] R. Xie, X. Jia, and K. Wu, "Adaptive online decision method for initial congestion window in 5G mobile edge computing using deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 389–403, Feb. 2020.

[17] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4585–4600, Sept. 2019.

[18] 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 & Tutorials*, vol. 21, no. 4, pp. 3039–3071, Fourthquarter 2019.

[19] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020.

[20] F. Zhou and R. Q. Hu, "Computation efficiency maximization in wireless-powered mobile edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 5, pp. 3170–3184, May 2020.

[21] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. of IEEE International Conference on Communications (ICC)*, Shanghai, China, May 2019.

[22] S. Zhang, W. Quan, J. Li, W. Shi, P. Yang, and X. Shen, "Air-ground integrated vehicular network slicing with content pushing and caching," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2114–2127, Sept. 2018.

[23] O. Semiari, W. Saad, and M. Bennis, "Joint millimeter wave and microwave resources allocation in cellular networks with dual-mode base stations," *IEEE Transactions on Wireless Communications*, vol. 16, no. 7, pp. 4802–4816, Jul. 2017.

[24] L. X. Cai, L. Cai, X. Shen, and J. W. Mark, "Rex: A randomized exclusive region based scheduling scheme for mmwave WPANs with directional antenna," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 113–121, Jan. 2010.

[25] F. Yuan, Y. H. Lee, Y. S. Meng, S. Manandhar, and J. T. Ong, "High-resolution ITU-R cloud attenuation model for satellite communications in tropical region," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 9, pp. 6115–6122, Sept. 2019.

[26] E. Falletti, M. Laddomada, M. Mondin, and F. Sellone, "Integrated services from high-altitude platforms: A flexible communication system," *IEEE Communications Magazine*, vol. 44, no. 2, pp. 85–94, Feb. 2006.

[27] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication efficient federated learning," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 118, no. 17, Apr. 2021.

[28] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, 2001.

[29] V. Smith, C. K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," in *Advance International Conference on Neural Information Processing Systems (NIPS)*, CA, USA, Dec. 2017.

[30] V. Smith, S. Forte, C. Ma, M. Takáč, M. I. Jordan, and M. Jaggi, "CoCoA: A general framework for communication-efficient distributed optimization," *Journal of Machine Learning Research*, vol. 18, no. 230, pp. 1–49, Nov. 2016.

[31] 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," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[32] S.Boyd and L.Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[33] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 3.0," 2015, Available Online: http://cvxr.com/cvx.

[34] X. Pang, G. Gui, N. Zhao, W. Zhang, Y. Chen, Z. Ding, and F. Adachi, "Uplink precoding optimization for NOMA cellular-connected UAV networks," *IEEE Transactions on Communications*, vol. 68, no. 2, pp. 1271–1283, Feb. 2020.

[35] I. Necoara and A. Patrascu, "Iteration complexity analysis of dual first-order methods for conic convex programming," *Optimization Methods & Software*, vol. 31, no. 3, pp. 645–678, 2014.

[36] Z. Yang, W. Xu, and M. Shikh-Bahaei, "Energy efficient UAV communication with energy harvesting," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1913–1927, Feb. 2020.

[37] Recommendation ITU-R M.1456, *Minimum performance characteristics and operational conditions for high altitude platform stations*, International Telecommunications Union, 2000.

[38] J. Long, *City Cellular Traffic Map (C2TM)*, Available Online: https://github.com/caesar0301/city-cellular-traffic-map[accessed on May 2021].