IEEE ComSoc | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 14 February 2024; revised 23 June 2024; accepted 16 August 2024.
Date of publication 28 August 2024; date of current version 11 September 2024.

The associate editor coordinating the review of this article and approving it for publication was N. H. Tran.

Digital Object Identifier 10.1109/TMLCN.2024.3450829

Agent Selection Framework for Federated Learning in Resource-Constrained Wireless Networks

MARIA RAFTOPOULOU[1], JOSÉ MAIRTON B. DA SILVA JR.[2] (Member, IEEE), REMCO LITJENS[1,3], H. VINCENT POOR[4] (Life Fellow, IEEE), AND PIET VAN MIEGHEM[1] (Fellow, IEEE)

[1]Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands
[2]Department of Information Technology, Uppsala University, 751 05 Uppsala, Sweden
[3]Department of Networks, Netherlands Organisation for Applied Scientific Research (TNO), 2595 DA The Hague, The Netherlands
[4]Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA

CORRESPONDING AUTHOR: M. RAFTOPOULOU (M.Raftopoulou@tudelft.nl)

This work was supported by the NExTWORKx, a collaboration between TU Delft and Royal KPN N.V. (KPN) on future communication networks. The work of José Mairton B. da Silva Jr. was supported in part by the European Union's Horizon Europe Research and Innovation Program under the Marie Skłodowska-Curie Project Federated Learning Supporting Efficient and Reliable Inference over Vehicular Networks (FLASH), under Grant 101067652; in part the Ericsson Research Foundation; and in part by the Hans Werthén Foundation. The work of H. Vincent Poor was supported by the U.S. National Science Foundation under Grant CNS-2128448 and Grant ECCS-2335876. The work of Piet Van Mieghem was supported by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Program under Grant 101019718.

ABSTRACT Federated learning is an effective method to train a machine learning model without requiring to aggregate the potentially sensitive data of agents in a central server. However, the limited communication bandwidth, the hardware of the agents and a potential application-specific latency requirement impact how many and which agents can participate in the learning process at each communication round. In this paper, we propose a selection metric characterizing each agent's importance with respect to both the learning process and the resource efficiency of its wireless communication channel. Leveraging this importance metric, we formulate a general agent selection optimization problem, which can be adapted to different environments with latency or resource-oriented constraints. Considering an example wireless environment with latency constraints, the agent selection problem reduces to the 0/1 Knapsack problem, which we solve with a fully polynomial approximation. We then evaluate the agent selection policy in different scenarios, using extensive simulations for an example task of object classification of European traffic signs. The results indicate that agent selection policies which consider both learning and channel aspects provide benefits in terms of the attainable global model accuracy and/or the time needed to achieve a targeted accuracy level. However, in scenarios where agents have a limited number of data samples or where the latency requirement is very stringent, a pure learning-based agent selection policy is shown to be more beneficial during the early or late stages of the learning process.

INDEX TERMS Agent selection, federated learning, machine learning, wireless networks.

I. INTRODUCTION

TRADITIONAL machine learning (ML) algorithms are performed at a central location, where large amounts of data are aggregated and used to train the ML model. However, the input data originate at different agents who may be unwilling to share them due to privacy concerns. Additionally, agents can generate a large amount of data in a short period of time, which can saturate the communication channel if all data from all agents need to be centrally aggregated.

Addressing the difficulties of centralized ML algorithms, McMahan et al. [1] introduced *federated learning* (FL), a decentralized ML technique to train a centralized *global model* using decentralized data from multiple agents and without sharing the raw data at the agents. Specifically, the agents train their own *local model*, which has the same neural

© 2024 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

network architecture as the global model, with their own data. After local training, the agents only transmit the tuned parameters of their local model to the FL server, which is then responsible for generating a new global model by combining the received model parameters from the contributing agents. The process repeats for a number of *communication rounds* until the global model converges to a satisfactory accuracy level. `FederatedAveraging` (or `FedAvg`) [1] is the most popular method for FL. Agents apply the stochastic gradient descent (SGD) optimizer, for a number of local iterations, and the global model is generated by averaging the submitted local models.

A challenge of FL is that the exchange of model parameters between the agents and the FL server can come at a high communication cost, especially for models with a large number of parameters [2]. This is of particular relevance in wireless network scenarios, e.g. when considering applications such as autonomous driving and the internet of things, which rely on resource-constrained wireless networks [3]. Also, wireless channels impose further challenges as they are susceptible to interference, have limited resources and their quality varies over location, time and frequency. To address these challenges, a subset of agents is selected to participate in a given communication round. Furthermore, although the `FedAvg` method can perform very well [1], [4], its performance can degrade significantly when data at the agents are not independent and identically distributed (non-IID), i.e., heterogeneous, across all agents [5]. Therefore, the selection of agents influences the convergence time and accuracy of the global model.

A. RELATED WORK

In the literature, the agent selection problem has been addressed from both a pure FL perspective and for the specific setting of a wireless network. From the FL perspective, Charles et al. [6] address the effects of randomly selecting a large number of agents is addressed. Rather than randomly selecting agents, Cho et al. [7] show that selection of agents based on their local loss improves the convergence of the global model, even for scenarios with heterogeneous data. The local loss is also considered by Lai et al. [8], who perform agent selection with a statistical utility function. Nguyen et al. [9] perform agent selection considering the gradient information of each agent, while Chen et al. [10] use the norms of the updates of each agent. Ribero and Vikalo [11] suggest the selection of agents based on the progression of the agents' local weights with respect to time. However, none of the above works consider a wireless network nor provide a clear indication of which metric is the most appropriate to characterize the importance of agents in the learning process.

Considering a wireless network, Hellström et al. [12] provide an overview of importance-aware agent selection with learning and wireless policies. Nishio and Yonetani [13] propose a greedy method to maximize the number of selected agents during a time interval. Yang et al. [14] compare the performance of the random, round robin and proportional fair schedulers, in terms of the FL convergence rate, for scenarios with limited bandwidth and interference. Amiri et al. [15] show that selecting agents based on both their wireless channels and the l_2-norm of their local model update provides better performance than only considering one of the two metrics individually. Zeng et al. [16] concentrate on minimizing the energy consumption, whereas Yu et al. [17] optimize the trade-off between minimizing the energy consumption and maximizing the number of selected agents. Shi et al. [18] consider latency-constrained systems and aim to maximize the model accuracy within a given total latency budget. Fan et al. [19] address latency-constrained systems by minimizing the time duration of each communication round, assuming mobile agents. Moreover, Albaseer et al. [20] address agent selection in scenarios with data and device heterogeneity and limited wireless resources. However, none of the works perform agent selection by characterizing the agents based on their importance in the learning process and their transmission, processing and energy resource consumption. Furthermore, the works in the literature focus on specific objectives of a given problem rather than providing a general agent selection framework, which can be easily adapted to the problem and objective at hand.

The global model convergence, with the `FedAvg` method, under non-IID training data is addressed in the literature in many ways, including with data sharing [21], [22] and with regularization [23], [24], [25]. Convergence guarantees have also been derived for scenarios with non-IID data [26] and it is suggested that for many real-world applications, the `FedAvg` method can provide identical performance for IID and non-IID data [27]. Therefore, motivated by the research so far, we employ the `FedAvg` method in this work.

B. CONTRIBUTIONS AND PAPER ORGANIZATION

Our main contributions are the following:

- We propose a configurable metric to characterize the agents based on their importance in the learning process as well as their resource consumption, which depends on the FL model and the agents' wireless channels and hardware. Such a metric describes the agents better and hence allows for a more appropriate selection of agents based on the scenario considered.
- We propose a general agent selection framework, which considers both agent-specific and system constraints, in the form of an optimization problem. The optimization problem can be adjusted to accommodate different needs and constraints from the application, network and agent perspectives. Thus, the proposed framework can address a number of different scenarios rather than provide a solution to a single specific problem. Moreover, the optimization problem can be easily extended to address the joint agent selection and resource allocation problem in FL scenarios over wireless networks.

- We show that for scenarios with non-IID data, the local loss is a better metric than the deviation between the local and the global model to characterize the importance of an agent in the learning process.
- We demonstrate that the learning accuracy is improved when both learning and channel aspects are considered to characterize the agents. However, when the agents have few samples or when stringent latency requirements apply, a higher global model accuracy is achieved with pure learning-based agent selection policies.

The remainder of this paper is organized as follows. Section II provides the learning and communication models. In Section III, the agent characterization is presented, while the problem formulation is derived in Section IV. Section V presents the considered use case for evaluating the agent selection framework and Section VI provides the evaluation of the agent selection policies, as derived from the framework. Finally, the conclusions and recommendations for future work are presented in Section VII.

II. SYSTEM MODEL

Consider a cellular network with one base station, which also acts as an FL server and a set $\mathcal{V}$ of agents, where $V = |\mathcal{V}|$. The FL server and the agents collaboratively train a global model, without requiring the transmission of the data sets gathered by the agents. Therefore, each agent $v \in \mathcal{V}$ holds its own training data set $\mathcal{K}_v$ and testing data set $\mathcal{K}_{T,v}$, where $K_v = |\mathcal{K}_v|$ and $K_{T,v} = |\mathcal{K}_{T,v}|$ denote the number of training and testing data samples available at agent v, respectively.

At a given communication round i, the FL server selects which agents will participate in the learning and each selected agent $v \in \mathcal{V}_G[i]$ trains its local model, where $\mathcal{V}_G[i]$ is the set containing the selected agents at communication round i. Once each selected agent $v \in \mathcal{V}_G[i]$ finishes its local training, it transmits its local model to the FL server. After all required local model uploads are completed, the FL server is responsible to update the global model for the next communication round $i+1$ and transmit the new global model to each agent $v \in \mathcal{V}$. The process repeats until sufficient accuracy is achieved at the global model, based on an FL server- or agent-specific testing data set, or an application-specific deadline is reached.

An application-specific deadline $T_{\text{APP,MAX}}$ can also be set on the time duration of each communication round i to prevent the selection of agents with limited training power and/or with poor wireless channel quality. Additionally, the FL process can be bound to the available transmission resources $C_{\text{R,MAX}}$ allocated to the FL task, e.g. in a slice in 5G networks, which can restrict the number of selected agents per communication round. Table 1 provides a short description of the most commonly used symbols in this paper. The rest of this section describes in more detail the learning model and the model for the communication between the agents and the FL server.

TABLE 1. List of most commonly used symbols.

Symbol	Description
$\rho_{\{\text{E,L,R,T}\}}$	Constant tuning the relative importance of energy consumption/ learning importance/ transmission resource consumption/ processing resource consumption
$\mathbf{W}_G$	The weights of the global model
$\mathbf{W}_v$	The weights of the local model at agent v
$\mathbf{X}_v$	The input data at agent v
$\mathbf{Y}_v$	The output data at agent v
B	System bandwidth in [MHz]
$C_{\text{R,MAX}}$	Available transmission resources
$C_{E,v}$	Energy consumption of agent v in [J]
$C_{R,v}$	Consumption of transmission resources of agent v
$C_{T,v}$	Consumption of processing resources of agent v in [s]
E_v	Energy level at agent v in [J]
$F(\cdot)$	The loss function of the model
$K = \|\mathcal{K}\|$	The total number of samples
$K_v = \|\mathcal{K}_v\|$	Number of training samples at agent v
$K_{T,v} = \|\mathcal{K}_{T,v}\|$	Number of testing samples at agent v
$Q_{\text{L},v}$	Importance of agent v in the learning process
Q_v	Importance of agent v
R_v	Bit rate at agent v in [Mbps]
S_v	Binary optimization variable for the selection of agent v
$T_{\text{APP,MAX}}$	Application-specific latency budget in [s]
T_v	Transmission time of agent v in [s]
$V = \|\mathcal{V}\|$	Number of agents in the network
$V_G = \|\mathcal{V}_G\|$	Number of selected agents for training
Z	Size of the FL model in [Mbits]
g_{MIN}	Minimum required processing capabilities in [FLOPs]
g_v	Processing capabilities of agent v in [FLOPs]
q_v	Deviation of agent v

A. LEARNING MODEL

Consider an agent v, with training data set $\mathcal{K}_v$. We denote its input data $\mathbf{X}_v = [\mathbf{x}_{v1}, \cdots, \mathbf{x}_{vK_v}]$, where $\mathbf{x}_{vk} \in \mathbb{R}^{n_X}$ denotes the k^{th} input vector to the model of agent v, with n_X as the size of the input vector. Additionally, the output data $\mathbf{Y}_v = [\mathbf{y}_{v1}, \cdots, \mathbf{y}_{vK_v}]$, where $\mathbf{y}_{vk} \in \{0,1\}^{n_C}$ denotes the real output vector associated with the k^{th} input vector $\mathbf{x}_{vk}$, where n_C is the size of the output vector and hence, the number of model outputs. For example, for an object classification learning task with n_C classes, the real output $\mathbf{y}_{vk}$ indicates with value 1 the class that sample k belongs to, while for all other classes, it holds a 0 value.

During local training with training data set $\mathcal{K}_v$, the model output (or predictions) $\hat{\mathbf{Y}}_v = [\hat{\mathbf{y}}_{v1}, \cdots, \hat{\mathbf{y}}_{vK_v}]$ is generated, where $\hat{\mathbf{y}}_{vk} \in \mathbb{R}^{n_C}$ denotes the predicted output vector related to the k^{th} input vector $\mathbf{x}_{vk}$. The model output $\hat{\mathbf{Y}}_v$ depends on the considered model architecture, e.g. the number of hidden layers in the case of a deep neural network. The weights $\mathbf{W}_v$ parameterize the considered local model and the goal of the local model is to tune its weights $\mathbf{W}_v$ such that the

predictions $\hat{\mathbf{Y}}_v$ will represent the real output $\mathbf{Y}_v$, given the input data $\mathbf{X}_v$. The relation between the predictions $\hat{\mathbf{y}}_v$ and the real output $\mathbf{Y}_v$ is typically measured with the loss function $F(\mathbf{W}_v; \mathbf{X}_v, \mathbf{Y}_v)$, which also depends on $\mathbf{X}_v$ and $\mathbf{Y}_v$. From here onwards, we omit this dependency for the sake of simplifying the notation.

The objective of tuning the local model is:

$$\min_{\mathbf{W}_v} F(\mathbf{W}_v) = \frac{1}{K_v} \sum_{k \in \mathcal{K}_v} f_k(\mathbf{W}_v), \tag{1}$$

where $f_k(\mathbf{W}_v)$ is the loss function of sample k, which is commonly set to the cross-entropy loss for classification problems [28]. To find the weights $\mathbf{W}_v$ which minimize the loss function $F(\mathbf{W}_v)$, a number of iterations n_{LE} are performed, known as local epochs. Assuming the SGD optimizer [29], the weights $\mathbf{W}_v$ are adapted at every local epoch based on the learning rate η, which controls the learning speed of the model.

In an FL setting, a data set $\mathcal{K}$, where $K = |\mathcal{K}|$, is the collection of the data sets $\mathcal{K}_v$ from all agents in set $\mathcal{V}$ and hence $\mathcal{K} = \cup_{v \in \mathcal{V}} \mathcal{K}_v$. With the `FedAvg` method [1] and assuming that the global model is generated only based on the models of the selected agents, the loss of the FL server, at communication round i, is upper bounded by the weighted average of the local losses

$$\min_{\mathbf{W}_G} F(\mathbf{W}_G[i]) \leq \sum_{v \in \mathcal{V}_G[i]} \frac{K_v}{K} F(\mathbf{W}_v[i]), \tag{2}$$

where $\mathbf{W}_v[i]$ denotes the weights of the local model of agent v, after local training during communication round i and $\mathbf{W}_G[i]$ are the weights of the global model. Using this upper bound, FL approximates the global objective function $F(\mathbf{W}_G[i])$ by the weighted average of the local losses. Then, assuming the SGD optimizer, the weights $\mathbf{W}_G[i]$ of the global model at the end of communication round i are updated as follows:

$$\mathbf{W}_G[i] \leftarrow \sum_{v \in \mathcal{V}_G[i]} \frac{K_v}{K} \mathbf{W}_v[i], \tag{3}$$

and then transmitted to all of the agents for the next communication round $i + 1$.

B. COMMUNICATION MODEL

For the transmission of the local model, assuming a wireless link, we measure the bit rate R_v of agent v in Mbps with the Shannon–Hartley equation [30] as

$$R_v = B_v \log_2 \left(1 + \frac{P_v G_v}{P_N}\right), \tag{4}$$

where B_v is the transmission bandwidth of agent v in MHz, P_v is the transmit power in Watt, G_v is the transmission gain and P_N is the thermal noise power in Watt. The transmission gain G_v is given, in dB, by [30]

$$G_v = 20 \log \left(\frac{c}{4\pi f_C}\right) - 10\gamma \log(d_v) + \psi, \tag{5}$$

where c is the speed of light, f_C is the carrier frequency, d_v is the three-dimensional distance between agent v and the serving base station, γ is the path loss exponent and ψ is a Normally-distributed random variable with zero mean and variance σ^2, capturing the effects of shadow fading. Finally, we ignore the transmission of the global model from the FL server to the agents, which can be assumed to be a broadcast and hence consume relatively few resources.

III. AGENT CHARACTERIZATION

In real-world applications, agents are diverse in terms of their training data as well as processing capabilities (e.g. central processing unit (CPU)) to train their local model, wireless channel quality and energy availability. In this section we define two metrics for measuring the importance of agents in the learning process and we describe the potential resource consumption of the agents.

A. LEARNING PROCESS IMPORTANCE

Non-random agent selection can improve the FL model convergence [7], [8], [9], [10], [11]. Hence, we characterize an agent v at communication round i based on its importance $Q_{L,v}[i]$ in the learning process. Specifically, we consider two metrics which can express the importance $Q_{L,v}[i]$ of agent v: *(a)* the deviation $q_v[i]$ and *(b)* the loss $F(\mathbf{W}_{G,v}[i])$, which are both defined below.

Inspired by regularization to address the challenges of non-IID data [23], we propose as a metric for the importance $Q_{L,v}[i]$ in the learning process, the *deviation*

$$q_v[i] = ||\mathbf{W}_v[i_v] - \mathbf{W_G}[i-1]||_2^2, \tag{6}$$

which represents the deviation between the local model $\mathbf{W}_v[i_v]$ of agent v and the global model $\mathbf{W_G}[i-1]$, where $||\cdot||_2$ denotes the Euclidean norm and i_v denotes the most recent communication round that agent v was selected for learning. The deviations can be calculated at the FL server and consequently used in the agent selection process without any additional signaling from the agents. The reason is due to the assumption that the FL server always stores the weights of agents from their last participation in the learning process until their next participation. Finally, although multiple agents may have the same deviation, the proposed framework ensures that those agents are differentiated during the agent selection process, which is explained with more details in Section IV.

Alternatively, the importance $Q_{L,v}[i]$ in the learning process can also be expressed [7], [22] in terms of the *loss* function $F(\mathbf{W}_{G,v}[i])$ of agent v, which is locally computed at agent v with the testing data set $\mathcal{K}_{T,v}$ and the newly generated global weights $\mathbf{W}_G[i]$ at the end of communication round i. Then, the loss $F(\mathbf{W}_{G,v}[i])$ is transmitted to the FL server as an input for the agent selection process of communication round $i + 1$. The time needed to calculate the loss $F(\mathbf{W}_{G,v}[i])$ is addressed in Section III-B.2 and we consider the corresponding transmission time to be negligible due to the loss being a scalar value. Since all agents need to transmit their loss, the

communication cost for this task is linearly proportional to the number of agents in the network. However, the loss is a scalar value and therefore its impact on the total communication cost is considered non-significant.

B. RESOURCE CONSUMPTION

When an agent participates in the learning process during communication round i, it consumes resources. We characterize the total resource consumption of an agent v based on the resource consumption for the transmission $C_{R,v}[i]$, processing $C_{T,v}[i]$ and energy $C_{E,v}[i]$, respectively. The transmission $C_{R,v}[i]$ and processing $C_{T,v}[i]$ consumption relate to system-specific resource, e.g. bandwidth and time, while the energy $C_{E,v}[i]$ consumption is agent-specific. In the following, we detail the consumption for these three types of resources.

1) TRANSMISSION RESOURCES

The consumption of the transmission, i.e. time-frequency, resources $C_{R,v}[i]$ of agent v is related to the upload of the local model $\mathbf{W}_v[i]$ at communication round i and depends on the communication system. We consider an orthogonal frequency division multiple access (OFDMA) system and assume that the channel is static during the communication round i and over the applied frequency carrier. Considering that the transmission resources are defined in both the time and frequency domains, we define the consumed transmission resources $C_{R,v}[i]$ as:

$$C_{R,v}[i] = T_v[i]B_v[i], \tag{7}$$

where $T_v[i]$ is the transmission time in seconds and $B_v[i]$ is the transmission bandwidth in MHz. Given that the transmission bandwidth $B_v[i]$ is fixed during communication round i, the transmission time $T_v[i] = \frac{Z}{R_v[i]}$, where $R_v[i]$ is the bit rate as given in (4) and Z is the size of the model in Mbits. Then, (7) is re-written as

$$C_{R,v}[i] = \frac{Z}{R_v[i]}B_v[i], \tag{8}$$

which captures the dynamic nature of the channels over time through the bit rate $R_v[i]$. The resource consumption $C_{R,v}[i]$ is important for systems with limited resources because it can be exploited for efficient agent selection. Also, the calculation of the resource consumption $C_{R,v}[i]$ does not require additional communication between the agents and the base station, because the bit rates can be estimated by the base station via the periodic channel quality indicator feedback that all agents report to the network.

2) PROCESSING RESOURCES

The consumption $C_{T,v}$ related to the local model training by agent v is measured in terms of time and depends on the agent's processing capability g_v, as well as on its data set size K_v and other training-related parameters. Assuming a fixed training data set size K_v, the consumption $C_{T,v}$ is the same for any communication round. Specifically, the processing capability g_v of agent v is measured in floating point operations (FLOPs) per second as [31]

$$g_v = n_{\text{CORES},v}\ \nu_v\ \omega_v, \tag{9}$$

where $n_{\text{CORES},v}$ is the number of CPU cores at agent v, ν_v is the CPU clock frequency at agent v in cycles per second and ω_v is the number of FLOPs per cycle at agent v. Then, the time consumption $C_{T,v}$ for training at agent v is

$$C_{T,v} = \left\lceil \frac{K_v}{s_B} \right\rceil \frac{n_{\text{FLOP},G}\ n_{\text{LE}}}{g_v}, \tag{10}$$

where $n_{\text{FLOP},G}$ denotes the number of FLOPs to train the model for a batch of size s_B and $\lceil\cdot\rceil$ represents the ceiling operation.

When the loss $F(\mathbf{W}_{G,v}[i])$ of agent v is considered for the importance $Q_{L,v}[i]$ in the learning process, an additional term can be added to the training time consumption $C_{T,v}$ to represent the loss calculation:

$$C_{T,v} = \left\lceil \frac{K_v}{s_B} \right\rceil \frac{n_{\text{FLOP},G}\ n_{\text{LE}}}{g_v} + \left\lceil \frac{K_{T,v}}{s_B} \right\rceil \frac{n_{\text{FLOP},G}}{g_v}, \tag{11}$$

assuming a fixed testing data set size $K_{T,v}$.

The time consumption $C_{T,v}$ can only be measured locally at the agent and hence, it should be communicated to the FL server when the agent first enters the network. Since this communication occurs only once, the related communication cost is considered negligible, regardless of the number of agents in the network. The knowledge of the time consumption $C_{T,v}$ at the FL server is significant because it allows the FL server to perform resource-efficient agent selection within a given latency bound.

3) ENERGY RESOURCES

The energy consumption $C_{E,v}[i]$ of an agent v, during communication round i covers both the training and the wireless transmission. Applying the model in [31], the energy consumption $C_{E,v}[i]$ is given, in Joules, by:

$$C_{E,v}[i] = \frac{e_v}{\omega_v^3} \left\lceil \frac{K_v}{s_B} \right\rceil g_v^2\ n_{\text{FLOP},G}\ n_{\text{LE}} + P_v[i]T_v[i], \tag{12}$$

where e_v is the energy consumption coefficient based on the CPU, measured in Watt(cycles/s)$^{-3}$. When considering agents with energy limitations, the energy consumption $C_{E,v}[i]$ is an important metric because the available energy level $E_v[i]$ of the agent should exceed the energy consumption $C_{E,v}[i]$ required to participate in the learning process.

When energy aspects are taken into account during agent selection, the energy consumption $C_{E,v}[i]$, as well as the total available energy $E_v[i]$, need to be reported to the FL server. Specifically, the training-related energy consumption needs to be reported to the FL server once, when first entering the network, while the transmission-related energy consumption can be estimated at the FL server for the same reasons given for the transmission resource consumption $C_{R,v}[i]$. Furthermore, the energy level $E_v[i]$ of an agent v is dependent on the

communication round i, because it decreases by the agent's energy consumption, every time agent v is selected and hence the energy level $E_v[i]$ should be periodically reported to the FL server.

IV. PROBLEM FORMULATION

We define the *agent importance* $Q_v[i]$ as the metric governing the agent selection process to improve the performance of the FL model by exploiting the different characteristics of the agents. The agent importance $Q_v[i]$ is defined to capture the trade-off between the importance $Q_{L,v}[i]$ of the agent v in the learning process against the total resource consumption of the agent v as follows:

$$Q_v[i] = \frac{Q_{L,v}^{\rho_L}[i]}{C_{R,v}^{\rho_R}[i] C_{T,v}^{\rho_T} C_{E,v}^{\rho_E}[i]}, \tag{13}$$

with $\rho_L + \rho_R + \rho_T + \rho_E = 1$, where $\{\rho_L, \rho_R, \rho_T, \rho_E\} \in [0, 1]$ are constants to tune the relative significance of the learning importance $Q_{L,v}[i]$ and the consumed transmission $C_{R,v}[i]$, as given in (8), processing $C_{T,v}$, as given in (10) or (11), and energy $C_{E,v}[i]$ resources, as given in (12), respectively. Recall that the importance $Q_{L,v}[i]$ of the agent v in the learning process can be expressed in terms of the deviation $q_v[i]$, as given in (6), or the local loss $F(\mathbf{W}_{G,v}[i])$. Moreover, the agent importance $Q_v[i]$ can be fine-tuned to the requirements and constraints of the agents and the system. For example, when the network is very congested, higher emphasis can be given to the transmission resource consumption of agents by increasing the value of ρ_R. Therefore, by appropriately configuring the constants ρ_L, ρ_R, ρ_T and ρ_E in (13), both learning and resource consumption aspects are simultaneously addressed.

For a given communication round i, we formulate the following general agent selection optimization problem to maximize the total agent importance, which can capture both learning and resource consumption aspects:

$$\max_{S_1[i],\cdots S_V[i]} \sum_{v\in\mathcal{V}} Q_v[i] S_v[i] \tag{14a}$$

$$\text{subject to} \quad \sum_{v\in\mathcal{V}} C_{R,v}[i] S_v[i] \le C_{\text{R,MAX}}[i], \tag{14b}$$

$$(C_{T,v} + T_v[i]) S_v[i] \le T_{\text{APP,MAX}}, \quad \forall v \in \mathcal{V}, \tag{14c}$$

$$C_{E,v}[i] S_v[i] \le E_v[i], \quad \forall v \in \mathcal{V}, \tag{14d}$$

$$g_v S_v[i] \ge g_{\text{MIN}}, \quad \forall v \in \mathcal{V}, \tag{14e}$$

$$S_v[i] \in \{0, 1\}, \quad \forall v \in \mathcal{V}, \tag{14f}$$

where the binary optimization variable $S_v[i]$ indicates whether agent v is selected at communication round i or not. Constraint (14b) indicates that the transmission resources allocated to the agents should not exceed the total system resources allocated to the FL task at communication round i. Constraint (14c) shows that the selected agents should train and transmit their models within an application-specific latency budget $T_{\text{APP,MAX}}$. Constraint (14d) ensures that the selected agents have sufficient energy levels to participate in the learning process. Finally, constraint (14e) ensures that the processing capabilities of the selected agents exceed a minimum requirement g_{MIN}, to avoid selecting agents with potentially long training times.

The objective in (14a) depends on the importance $Q_v[i]$ for each agent v, which provides the differentiation between the different agents. In case two or more agents have the same agent importance $Q_v[i]$, the agents are differentiated via the constraints. For example, constraint (14b) differentiates the agents based on the quality of their wireless channels. We remind that the agent importance in (13) serves as a selection metric and thus, the selected agents transmit their local models to the FL server after performing local training. Moreover, the optimization problem in (14) is general and it can be adjusted to the communication system, the application-specific requirements and the energy constraints of the agents. Therefore, some of the constraints in (14) might be irrelevant to specific problems. Furthermore, the optimization problem can be extended to jointly consider agent selection and radio resource allocation, for example when multiple agents can be simultaneously served with beamforming antennas.

Additionally, we consider that the convergence analysis with the proposed framework can be provided based on the convergence analysis for partial agent participation in non-IID scenarios from Zhao et al. [5]. Therefore, we consider that the general convergence of the FL model using this framework is ensured and we leave for future work the specific convergence analysis. Then, the goal of this work is to analyze the performance of the framework in (14) and investigate the trade-offs between learning and wireless communication performance measures.

A. PROBLEM OF INTEREST

To investigate the trade-offs between learning and wireless aspects, we simplify the problem in (14) such that only the necessary constraints are considered. Specifically, we consider the communication system described in Section III and wideband radio resource scheduling. Then, the available transmission resources $C_{\text{R,MAX}}[i]$, in constraint (14b), can be expressed in terms of the application-specific latency budget $T_{\text{APP,MAX}}$. For example, the transmissions of the local models to the FL server will start once the agent with the shortest training time that participates in the given communication round i finishes its training. Considering that the duration of each communication round is given by the latency budget $T_{\text{APP,MAX}}$, the available transmission resources $C_{\text{R,MAX}}[i]$ are given by:

$$C_{\text{R,MAX}}[i] = B\left(T_{\text{APP,MAX}} - \min_{v\in\mathcal{V}_G[i]} C_{T,v} S_v[i]\right), \tag{15}$$

where B is the system bandwidth. Therefore, when applying (15) in constraint (14b), the problem in (14) investigates the scenario where agents have different training times and hence, different hardware and/or data sizes.

When agents with the same training time $C_{T,1} = \cdots = C_{T,V} = C_T$ are considered, (15) is simplified to

$$C_{\mathrm{R,MAX}} = B(T_{\mathrm{APP,MAX}} - C_T), \tag{16}$$

and the available transmission resources $C_{\mathrm{R,MAX}}$ are independent of the communication round i. For the remainder of this work, for simplicity, we consider that agents have the same training time C_T. Although this assumption makes the problem simpler, this problem remains applicable to real-world scenarios. Specifically, it can be assumed that for synchronization purposes, the FL server dictates to all agents the computing capability that should be applied by the agents, which is set to the minimum capability among all agents. Furthermore, when using (16) in constraint (14b), constraint (14c) becomes redundant.

It is also important to consider scenarios with powerful agents, e.g. vehicles or agents that have powerful hardware and access to charging points. For such scenarios, constraints (14d) and (14e) can be ignored. Then, we can further simplify the optimization problem in (14) to:

$$\max_{S_1[i],\cdots S_V[i]} \quad \sum_{v\in\mathcal{V}} Q_v[i]S_v[i] \tag{17a}$$

$$\text{subject to} \quad \sum_{v\in\mathcal{V}} C_{R,v}S_v[i] \leq B(T_{\mathrm{APP,MAX}} - C_T), \tag{17b}$$

$$S_v[i] \in \{0, 1\}, \quad \forall v \in \mathcal{V}. \tag{17c}$$

The problem formulation in (17) emphasizes the latency constraints that might be imposed by the system. Moreover, when problem (17) has all parameters discretized, it becomes the classic 0/1 Knapsack problem, which is a known NP-hard problem [32]. For problems with a small number of variables and constraints a pseudo-polynomial algorithm using dynamic programming solves the integer 0/1 Knapsack problem optimally in $O(VQ)$ time [32], where $Q = \sum_{v\in\mathcal{V}} Q_v$. The complexity can be reduced by a fully polynomial approximation. In this work, we apply the algorithm presented in [32], with $\epsilon = 0.001$, to find the approximate solution to (17). To apply the algorithm solving the integer 0/1 Knapsack problem, all parameters in (17) need to be integers. For this reason, we discretize all parameters by multiplying them with a large number and rounding them to the closest integer.

Another important aspect to discuss is the incurred communication cost in both the uplink and downlink channels due to the FL task. The uplink channel is used for the transmission of the local models to the base station and the problem in (17) considers that the available transmission resources are bounded by $C_{\mathrm{R,MAX}}$. The downlink channel is used for transmitting the global model to the agents and for notifying the agents about their participation in the FL task. In both cases, we assume that broadcast transmissions are used. Typically, the resources needed for the broadcast are network-dependent, such that a minimum bit rate is ensured at the cell edge. Therefore, for both the uplink and downlink channels, there are resources allocated specifically to the FL task, which are fixed during each communication round and do not depend on the number of agents in the network.

B. AGENT SELECTION POLICIES

Depending on the configuration of the agent importance $Q_v[i]$ in (13), different agent selection policies are derived as solutions to problem (17). Considering that all agents have the same hardware and hence the same training time C_T, as given in (10) and (11), we set $\rho_T = 0$ for the agent importance calculation in (13). The energy consumption in relation to training is also the same for all agents. Thus, the total energy consumption $C_{E,v}[i]$, as given in (12), depends only on the model transmission and consequently on the bit rate, which is covered by the consumption of the transmission resources $C_{R,v}[i]$, as given in (8). Thus, we set $\rho_E = 0$ in (13).

Considering that the agent importance $Q_v[i]$ in (13) is now tuned with the constants ρ_L and ρ_R, we consider two extreme cases. For the extreme case of $\rho_L = 1$ and $\rho_R = 0$, two solutions of the problem in (17) are derived. The first considers the deviation $q_v[i]$, whereas the second considers the loss $F(\mathbf{W}_{G,v}[i])$ as a metric for the learning importance $Q_{L,v}[i]$ of agent v. We refer to the two solutions as `max-sum-dev` and `max-sum-loss`, respectively, because they aim to maximize the sum of the deviations/losses over all selected agents. Therefore, these two policies simultaneously address the trade-off between selecting agents that are important to the learning and the limited transmission resources, which is captured in constraint (17b). For the other extreme case, i.e. where $\rho_L = 0$ and $\rho_R = 1$, only one solution exists, which is denoted as `max-sum-rate` because it aims to maximize the sum of the bit rates of the selected agents. Simulations showed that the performance of the policies with $\rho_L \in (0, 1)$ and $\rho_R = 1 - \rho_L$ is bounded by the `max-sum-dev` (or `max-sum-loss`) and the `max-sum-rate` policies. Therefore we do not show these policies in our evaluation.

V. EVALUATION SCENARIO

This section presents the considered scenario for evaluating the agent selection framework and investigating the trade-offs between learning and wireless performance measures, as captured in problem (17). First, we present the considered learning task. Then, we introduce the baseline agent selection policies that will be compared to the policies derived in Section IV-B. Finally, we present the configured learning and wireless parameters.

A. CLASSIFICATION OF EUROPEAN TRAFFIC SIGNS

As an example application, we perform the learning task of object classification on the European traffic sign data set (ETSD), which consists of 164 classes of signs aggregated over data sets from six European countries [33]. Due to the limited number of training samples per class, we only select the $n_C = 10$ classes with the highest number of samples.

For the classification task, we use a convolutional neural network (CNN) architecture similar to that in Serna and Yuichek [33] and Chiamkurthy [34], which are both

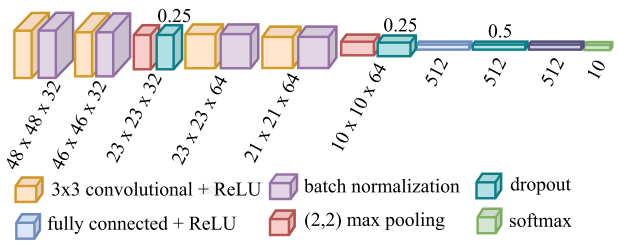


FIGURE 1. The considered CNN architecture to perform object classification task for the ETSD.

inspired by the typical Visual Geometry Group (VGG) architecture [35]. Fig. 1 shows the considered architecture. Specifically, four convolutional layers are activated with a rectified linear unit (ReLU) function and followed by batch normalization. Further, max pooling and dropout regularization with a range of 0.25 are performed. For the fully connected layer, the output of the convolutional layer is flattened and then activated with ReLU. Then, another dropout regularization is performed with a range of 0.5, followed by a batch normalization. Finally, the last layer is activated by a softmax with 10 outputs with each output indicating the probability for each class. Batch normalization makes the network learn robustly [29], while the dropout layers prevent overfitting [29]. In total, the network consists of 3349418 trainable parameters and thus, the size of the model $Z \approx 107$ Mbits, assuming 32-bit precision per parameter.

B. BASELINE AGENT SELECTION POLICIES

Apart from comparing the agent selection policies from Section IV-B to each other, we also compare them to baseline policies from the literature. One widely used agent selection policy is the `FedCS` policy, as introduced by Nishio and Yonetani [13], which is based on a greedy method to maximize the number of selected agents. In the considered evaluation scenario, the `FedCS` policy is identical to the `max-sum-rate` policy, which is derived as a solution to the problem in (17) when setting $\rho_L = 0$ and $\rho_R = 1$. The fact that the `FedCS` and `max-sum-rate` policies are identical shows the effectiveness of our proposed framework in generating diverse policies. This further shows that our agent selection framework is adaptable and allows fine-tuning of the agent selection policy to the scenario under investigation.

Additionally, we also consider the `pow-d` policy as a baseline policy, which was introduced by Cho et al. [7]. The `pow-d` policy only considers learning aspects, where d defines the size of the subset of agents that can be selected for training. Then, from the d randomly selected agents, the m agents with the highest local loss are selected. Considering that in our evaluation scenario the transmission resources are limited, it may occur that not all m agents can participate in the learning. Finally, based on the results shown in [7] and the dimensioning of our scenario, we consider $d = 15$ and $m = 4$.

Moreover, we also consider the `random`, `max-dev` and `max-loss` policies. The latter two baseline policies are derived as an approximation to the solution of the problem in (17). Specifically, they sort the agents in descending order based on their importance $Q_v[i]$ (based on deviation or loss, respectively) and select as many agents as possible until constraint (17b) is violated. In line with this approach, sorting and selecting agents based on their bit rates by setting $\rho_R = 1$, yields a selection policy that is identical to the `max-sum-rate` policy. Hence, it does not offer a further selection policy to be assessed.

The main difference between the policies derived as a solution to problem (17), i.e. `max-sum-dev`, `max-sum-loss` and `max-sum-rate`, and the baseline policies `max-dev`, `max-loss` and `random`, is that the former class of policies *explicitly* takes the bit rates into account, due to constraint (17b). We will refer to these policies as channel-aware policies. The latter class of policies, i.e. `max-dev`, `max-loss` and `random`, considers the bit rates *implicitly* as these policies sort and select as many agents as possible until constraint (17b) is violated. Finally, the baseline policy `pow-d` is *neither* explicitly nor implicitly considering wireless channel aspects, as it aims to select a fixed number of agents, based on their loss, per communication round.

C. LEARNING AND WIRELESS PARAMETERS

In our analysis we consider scenarios with $V = 50$ agents and both IID and non-IID data. For the IID scenario, all agents have the same number of samples K_v, which are evenly distributed over the ten classes. For the non-IID scenario, all agents have K_v samples, which are unevenly split over two classes such that on average all classes are equally represented in the training data set $\mathcal{K}$. Therefore, the non-IID scenario has a highly skewed data distribution that aims to represent a more realistic scenario than the IID scenario. For the calculation of the loss $F(\mathbf{W}_{G,v}[i])$ of agent v at communication round i, the categorical cross-entropy loss function is applied on the testing data set $\mathcal{K}_{T,v}$, which is unique for every agent and three times smaller than the training data set $\mathcal{K}_v$.

For the training, the agents invoke the SGD optimizer with learning rate $\eta = 0.05$, batch size $s_B = 64$ and with each agent performing $n_{\text{LE}} = 2$ local epochs. The number of FLOPs required from the agents to train the CNN for a batch size $s_B = 64$ is measured by the Keras library, in Python, which is $n_{\text{FLOP},G} = 6.55$ GFLOPs. Regarding the hardware of the agents, we consider the processing capabilities $g_v = 64$ GFLOPs per second. Such processing power will be given, for example, by $n_{\text{CORE}} = 1$ CPU core, $\nu = 2$ GHz CPU frequency and $\omega = 32$ FLOPs per cycle. Since the agents are assumed to have the same hardware, their energy coefficient is also identical $e_v = 10^{-27}$ W(cycles/s)$^{-3}$ [36].

For the wireless communication scenario, we consider an urban macro environment at $f_C = 3.5$ GHz and a bandwidth of $B = 50$ MHz [37]. For the wireless propagation, we assume a path loss exponent $\gamma = 3.7$ and shadowing with

$\sigma = 8$ dB, which are typical values for outdoor dense urban environments [30]. Additionally, the agents are uniformly distributed in a cell of radius 150 m. We assume that all agents are at a height of 1.5 m whereas the base station antenna is at a height of 25 m [38]. The transmission of the local model is performed assuming the agents' maximum transmit power $P_v = P_{\text{V,max}} = 24$ dBm [39]. Finally, the thermal noise power is $P_N = -97$ dBm.

VI. EVALUATION

This section presents the evaluation of the considered agent selection policies in terms of the achieved accuracy of the global model, which is measured at the FL server based on its specific testing data set. First, we consider Scenario 0, where all agents have the same bit rates. For this scenario, our aim is to study the policies from a pure learning perspective and hence provide insights into the learning behavior when the deviation $q_v[i]$ and the loss $F(\mathbf{W}_{G,v}[i])$ are applied as metrics for the agent importance $Q_{L,v}[i]$ to the learning. Besides Scenario 0, we compare the policies in Scenarios 1, 2 and 3, in which the agents have distinct bit rates that vary over time. Specifically, in Scenario 1, we show the impact of the wireless channel. Then, in Scenario 2, we investigate the performance of the policies when agents have a lower number of samples K_v. Finally, in Scenario 3, we evaluate the policies under a reduced application-specific latency budget $T_{\text{APP,MAX}}$, thus limiting the permitted communication latency.

Sections VI-A to VI-D present the accuracy of the global model for Scenarios 0-3, respectively. Then, in Section VI-E, we provide a comparison of Scenarios 1, 2 and 3, in terms of what accuracy level is reached within a given deadline and how long it takes to reach a certain accuracy level. Finally, Section VI-F provides the total energy consumed by the agents within a given deadline and to reach a certain accuracy level. We present all results as an average of 70 independent simulations and the source code generating all results is available in [40]. For the sake of presentation, we also include a short summary of the key result observed for each analyzed scenario.

A. SCENARIO 0 - PURE LEARNING PERSPECTIVE

To study the behavior of the different agent selection policies from a pure learning perspective, we consider the scenario where all agents have the same bit rate, which is equal to the average bit rate that can be experienced in the considered wireless environment. Hence, the `max-sum-loss`, `max-sum-dev` and `max-sum-rate/FedCS` policies behave like the `max-loss`, `max-dev` and `random` policies, respectively. Because of the identical bit rates, the number of selected agents per communication round is constant and the same for all policies, even for the `max-loss` policy which requires extra processing time for the loss calculations. We set $K_v = 300$ samples at each agent and hence the training time is equal to $C_T = 1.02$s, excluding the time for the loss calculation for the `max-loss` policy. Finally, setting $T_{\text{APP,MAX}} = 5$s allows approximately 4s of uploading time.

1) IID DATA

Considering the case with IID data, Fig. 2 shows the increase of the accuracy over time and illustrates that the `max-loss` policy exhibits a slower convergence than the `max-dev`, `pow-d` and `random` policies. The slower convergence of the `max-loss` policy is explained by its persistency to select the same agents over time while the `max-dev`, `pow-d` and `random` policies tend to more evenly cover the entire agent population over time. For IID data, all agents have samples from all ten classes and hence, regardless of the selected agents in a given communication round, the loss change of all agents in that communication round will be similar. Consequently, the agent sorting by the `max-loss` policy at the beginning of each communication round, does not change significantly over time and results in frequently selecting the same agents. Therefore, the global model is mostly trained on a subset of the total available samples which leads to a slower convergence.

When an agent is selected for training, its deviation, as calculated in (6), will be relatively small and for every round that the agent is not selected, its deviation will be relatively large. Hence, the `max-dev` policy behaves in a round robin fashion, with some initial agent sorting. With this, an even agent selection is achieved, which allows to consistently train on all available samples. Furthermore, Fig. 2 shows that the `max-dev`, `pow-d` and the `random` policies perform similarly because the `pow-d` and `random` policies also tend to cover the agent population well and hence train the model on all samples. The similarity of the `pow-d` and `random` policies can be attributed to the random component of the `pow-d` policy. Therefore, we have the following important result for the case of the IID data:

Result 1. *From a pure learning perspective, for scenarios with IID data, the learning process benefits from evenly selecting the agents over time and hence the* `max-dev`, `pow-d` *and* `random` *policies tend to outperform the* `max-loss` *policy.*

2) NON-IID DATA

When non-IID data are considered, the agents have samples from only two classes and therefore, the selection of agents in a given communication round is more crucial than for IID data. Also, a larger number of communication rounds is needed to reach a given accuracy level compared to the scenario with IID data. Fig. 3 illustrates that for the non-IID scenario, the `pow-d` policy outperforms all other policies. Moreover, the `max-loss` policy provides better convergence than the `max-dev` policy, in the sense that accuracy fluctuates with the `max-dev` policy. In contrast to IID data, for non-IID data, the losses of the agents after a given communication round will differ depending on the selected agents. Consequently, the `max-loss` policy does not persistently select the same agents. Hence, both the `pow-d` and `max-loss` policies select some agents more often than others, which allows more training on samples that contribute

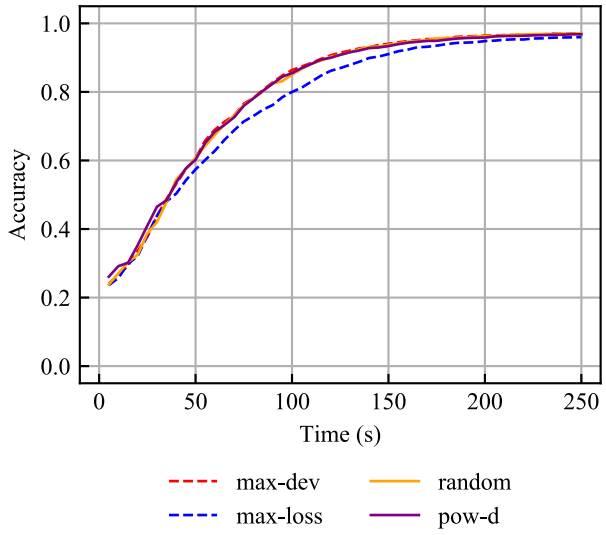


FIGURE 2. Accuracy over time for IID data, when agents have identical bit rates, $K_v = 300$ samples and $T_{\text{app,max}} = 5$s.

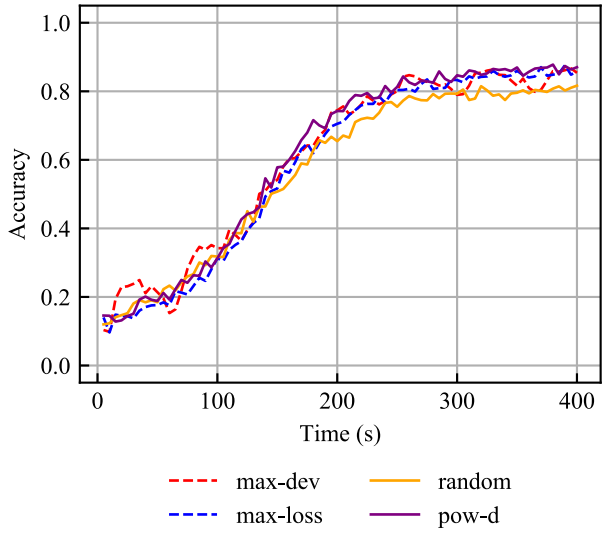


FIGURE 3. Accuracy over time for non-IID data, when agents have identical bit rates, $K_v = 300$ samples and $T_{\text{app,max}} = 5$s.

more to the learning process. Overall, the `pow-d` policy provides better performance compared to the `max-loss` policy because the random aspect of the `pow-d` policy ensures that different agents are considered for selection at each communication round and hence, the `pow-d` policy is less persistent on selecting the same agents. This result highlights that not all agents are equally important to the learning process when the data are non-IID but the bias due to selecting only the agents with the highest losses may limit the achieved global model accuracy.

Fig. 3 also shows that the accuracy achieved by the `max-dev` policy fluctuates over time, where the period of the fluctuation is equal to the time needed to select all agents once, i.e. the round robin period. Since agents do

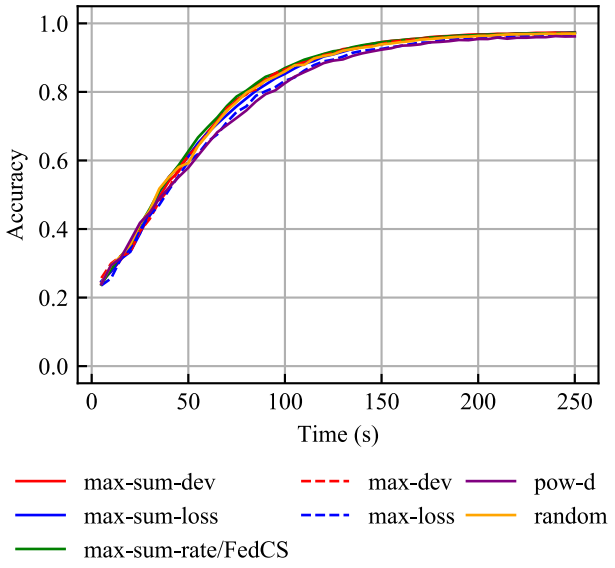


FIGURE 4. Accuracy over time for IID data, considering time-varying wireless channels, $K_v = 300$ samples and $T_{\text{app,max}} = 5$s.

not have equally important data, the initial sorting of the deviations $q_v[i]$ is essentially based on the data importance of the agents. Subsequently, the agents are selected the same number of times and in sequence, which can harm the accuracy and indeed lead to fluctuations. The accuracy fluctuation is amplified by selecting the same number of agents per communication round. Finally, Fig. 3 shows that the `random` policy has the worst performance because it does not consider any learning metrics. Therefore, the key takeaway result for the case of the non-IID data is the following:

Result 2. *From a pure learning perspective, for scenarios with non-IID data, not all agents have equally important data. Hence, the* `pow-d` *policy provides the highest accuracy level and stable gains by selecting the most appropriate agents per communication round while also ensuring that different agents can be selected over time.*

B. SCENARIO 1 - LEARNING AND COMMUNICATION PERSPECTIVE

In this scenario, the agents have distinct bit rates, based on the communication model in (4), with time-varying wireless channels that vary at each communication round. Similarly to Scenario 0, we assume $K_v = 300$ samples at each agent and set $T_{\text{APP,MAX}} = 5$s.

1) IID DATA

Fig. 4 shows the accuracy of the considered policies over time and illustrates that all considered policies, apart from the `max-loss` and `pow-d` policies, perform similarly. The slower convergence of the `max-loss` policy is due to the uneven agent selection, as explained for Scenario 0 in Section VI-A.1. Even though the `max-sum-loss` policy also relies on the loss of the agents, it explicitly

takes into account the bit rates of the agents, which eventually leads to selecting different agents per round and consequently achieving a higher accuracy level than the `max-loss` policy. Moreover, and in contrast to the results in Section VI-A.1, the `pow-d` policy now slightly under-performs because it neither explicitly nor implicitly considers the wireless channels. Fig. 4 also shows that the channel-aware policies, i.e. `max-sum-loss`, `max-sum-dev` and `max-sum-rate/FedCS`, perform similarly to the `max-dev` and `random` policies, which implicitly take the wireless channels into account. This similarity exists despite the fact that the former policies can select more agents than the latter. Hence, we conclude that there are no significant gains from exploiting the wireless channels. For the remaining Scenarios 2 and 3, we will not consider IID data. Therefore, the takeaway result is:

Result 3. *For IID data, the exact agent selection policy is not crucial as long as different agents are selected over time. Additionally, the gains of channel-aware agent selection are minimal.*

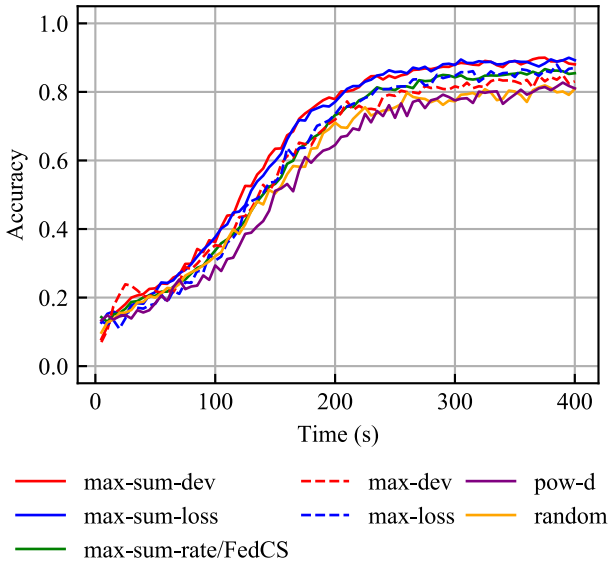


FIGURE 5. **Accuracy over time for non-IID data, considering time-varying wireless channels, $K_v = 300$ samples and $T_{\text{app,max}} = 5$s.**

2) NON-IID DATA

In Scenario 0, the `max-sum-loss`, `max-sum-dev` and `max-sum-rate/FedCS` policies are identical to the `max-loss`, `max-dev` and `random` policies, respectively. In Scenario 1, they behave differently as a result of the variable wireless channels. Fig. 5 shows the accuracy of the policies over time and illustrates that the `max-sum-loss`, `max-sum-dev` and `max-sum-rate/FedCS` policies provide higher accuracy levels than in Scenario 0. The reason is that they can exploit the gains from the wireless channels, which lead to selecting more agents, and hence training on more samples per communication round. The `max-sum-loss` and `max-sum-dev` policies behave similarly and achieve a higher accuracy than the other policies throughout the considered time period. Thus, we can conclude that agent selection based on both channel and learning aspects is beneficial to the learning process, regardless of the learning metric considered, i.e. the deviations $q_v[i]$ or the loss $F(\mathbf{W}_{G,v}[i])$.

Additionally, Fig. 5 shows that the `max-sum-rate/FedCS` and `max-loss` policies perform similarly, even though the `max-sum-rate/FedCS` policy selects on average approximately double the number of agents per round compared to the `max-loss` policy. This result highlights the effectiveness of the loss $F(\mathbf{W}_{G,v}[i])$ as a metric to indicate the importance of an agent in the learning process. Similar to our observation from Fig. 4, Fig. 5 shows that the `pow-d` policy under-performs, which contrasts with its best performance in Fig. 3 in Scenario 0. Once more, the reason is that the `pow-d` policy neither explicitly nor implicitly considers the wireless channels. Moreover, when comparing Figs. 3 and 5, the policies that implicitly take the wireless channels into account, i.e. `max-loss`, `max-dev` and `random`, behave similarly. However, the accuracy with the `max-dev` policy in Fig. 5 does not fluctuate as it did in Fig. 3, which is due to the fact that the wireless channel variation impacts the number of selected agents per communication round. This leads to the averaging of the peaks that were observed in Fig. 3. Therefore, the important message is:

Result 4. *Choosing agents based on both channel and learning aspects is advantageous for the learning process in non-IID data scenarios. The learning aspect ensures the selection of agents with suitable data, while the channel aspect benefits from the wireless channels, thus enabling the selection of as many as possible agents per communication round.*

3) COMPARISON WITH THE CIFAR-10 DATASET

To show the effectiveness and adaptability of our proposed framework, we also study the performance of the policies with the CIFAR-10 dataset [41], which is a complex dataset commonly used in the literature. For the training, we consider the CNN as used in [13], whereas the agents invoke the SGD optimizer with learning rate $\eta = 0.1$, batch size $s_B = 64$ and each agent performs $n_{\text{LE}} = 5$ local epochs. Additionally, we consider a non-IID scenario, where each agent holds $K_v = 600$ training samples and $K_{T,v} = 100$ testing samples. For the sake of a fair comparison with the ETSD, we adjust the hardware of the agents such that the time interval for uploading the FL models is the same when using both datasets.

Fig. 6 shows the accuracy over time when training with the CIFAR-10 dataset. Although higher accuracy levels have been reported in the literature when training with the CIFAR-10 dataset, the model we use is sufficient for our evaluations. We highlight that our goal is to evaluate our proposed framework and to investigate the trade-off between learning and wireless communication performance measures, rather than

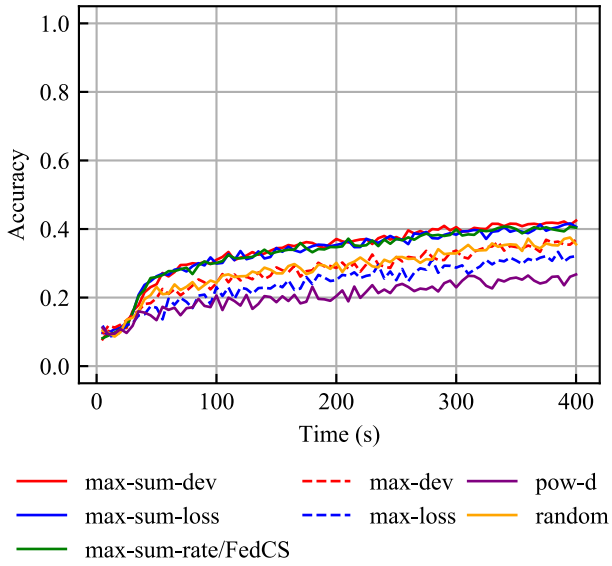


FIGURE 6. Accuracy over time for non-IID data with the CIFAR-10 dataset, considering time-varying wireless channels and $T_{\text{app,max}} = 5$s.

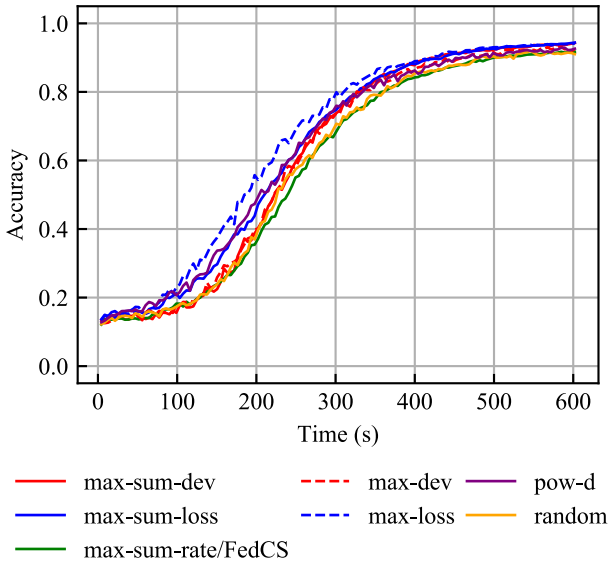


FIGURE 7. Accuracy over time for non-IID data, considering time-varying wireless channels, $K_v = 100$ samples and $T_{\text{app,max}} = 4.3$s.

provide the highest accuracy level. Moreover, more works in the literature report similar accuracy levels when considering non-IID data [5], [13]. Additionally, Fig. 6 shows an increasing trend in the achieved accuracy, which implies that with more time and communication rounds, a higher accuracy level may be achieved.

Fig. 6 also shows that the `max-sum-dev`, `max-sum-loss` and `max-sum-rate/FedCS` policies behave similarly and provide the highest accuracy level. Therefore, we can conclude that the policies that explicitly take the wireless channels into account perform the best because they can select more agents than the policies that implicitly consider the channels or not consider them at all. Thus, we conclude that with the complex CIFAR-10 dataset, it is crucial that many agents are selected for training such that more training samples can contribute to the training.

When comparing the results with the ETSD and CIFAR-10 datasets, we conclude that our proposed framework is effective, as the policies generated from our framework, i.e., `max-sum-dev`, `max-sum-loss` and `max-sum-rate/FedCS`, perform the best in both datasets. Moreover, the `max-sum-dev` and `max-sum-loss` policies, that take both learning and wireless channel aspects into account, achieve the highest accuracy with both datasets. The only difference between the results with the two datasets is that the pure wireless based `max-sum-rate/FedCS` policy matches the performance of the `max-sum-dev` and `max-sum-loss` policies when the complex CIFAR-10 dataset is used.

C. SCENARIO 2 - DIFFERENT NUMBER OF SAMPLES

In this scenario, we continue to have varying bit rates, while only considering non-IID data and reducing the number of samples per agent from $K_v = 300$ to $K_v = 100$. Due to the reduction of the training time C_T given a lower number of samples K_v, for comparison reasons, we adjust the application-specific latency budget to $T_{\text{APP,MAX}} = 4.3$s. With this, we ensure that the time interval for uploading the FL models is the same as in Scenarios 0 and 1.

Fig. 7 shows the accuracy of the policies over time and in comparison to Scenario 1, it takes a longer time for the accuracy to reach a more stable level because the agents now hold less data. Moreover, Fig. 7 shows that during the initial learning phase (until 400s), the `max-loss` policy learns more quickly than the other policies. The good performance of the `max-loss` policy is a consequence of selecting the most appropriate agents for the learning, which is more crucial in this scenario, since given that the agents have fewer samples, the likelihood of an agent possessing non-beneficial data for the learning process is higher. During this initial learning phase, the `max-sum-loss` and `max-sum-dev` policies perform worse than the `max-loss` policy because they sacrifice agents that are important to the learning process for less important agents with high bit rates. Moreover, the `pow-d` policy performs worse than the `max-loss` policy because it is limited in the number of agents that can be selected at each communication round. After 400s, the `max-sum-loss`, `max-sum-dev` and `pow-d` policies perform similarly to the `max-loss` policy because they selected enough agents with important data over time.

Moreover, Fig. 7 shows that the `max-sum-rate/FedCS` policy consistently under-performs during the learning process, despite being the policy that selects the most agents per communication round. This poor performance is attributed to not at all taking into account the learning aspect, which is dominant in this scenario. We can therefore get the following takeaway message:

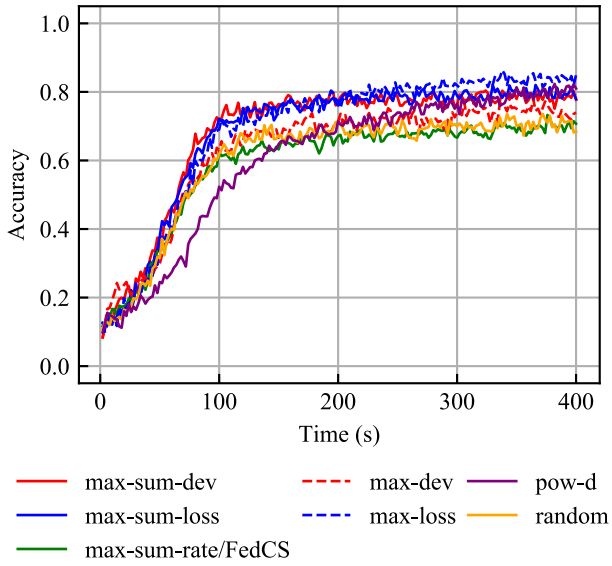


FIGURE 8. **Accuracy over time for non-IID data, considering time-varying wireless channels, $K_v = 300$ samples and $T_{\text{app,max}} = 2$s.**

Result 5. *When agents have a small data set size in a non-IID setting, the agent selection becomes very important, especially during the initial learning phase. For this reason, the* `max-loss` *policy provides higher accuracy during the initial learning phase than other channel- and learning-aware policies.*

D. SCENARIO 3 - DIFFERENT LATENCY BUDGET

To investigate the impact of the application-specific latency budget $T_{\text{APP,MAX}}$ on the accuracy, we consider a scenario with non-IID data, $K_v = 300$ samples and $T_{\text{APP,MAX}} = 2$s, instead of $T_{\text{APP,MAX}} = 5$s that was considered in Scenarios 0 and 1. The substantial reduction of $T_{\text{APP,MAX}}$ limits the number of agents that can be selected in a communication round as well as the set of agents that can be selected. The reason is that the cell edge agents, who suffer from low bit rates, may only sporadically be able to transmit their local model within the latency budget $T_{\text{APP,MAX}}$. Therefore, lower accuracy levels are expected within a given time period, compared to Scenario 1.

Fig. 8 shows the accuracy of the policies over time, which fluctuate more than in Scenario 1 because the global model is updated more frequently. Specifically, within 400s, 80 and 200 communication rounds are executed in Scenarios 1 and 3, respectively. Moreover, Fig. 8 shows that the initial learning phase in this scenario lasts for about 100s while in Scenario 1, it lasts for about 200s, as a result of setting a different latency budget $T_{\text{APP,MAX}}$. However, in both scenarios, the initial learning phase lasts for a comparable number of communication rounds.

Another observation from Fig. 8 is that the performance of the `max-sum-loss` and `max-sum-dev` policies is better than the performance of the `max-loss` policy until 200s, when the `max-loss` policy becomes the best performing policy. The reason is that the reduction of the latency budget $T_{\text{APP,MAX}}$ limits the extra number of agents that the channel-aware policies can select compared to the policies that implicitly take the channels into account. Hence, until 200s, there are some gains from exploiting the wireless channel but after 200s the accuracy with the channel-aware policies does not improve further, because the channel-aware policies avoid selecting agents with poor bit rates. Due to this reason, the `max-loss` policy can converge to a higher accuracy level in the long term. This implies that it selects agents at the cell edge more often than the channel-aware policies. For the same reason, the `max-sum-rate/FedCS` policy under-performs, thus making it slightly worse than the `random` policy. Furthermore, the `pow-d` policy initially under-performs due to being fully unaware of the wireless channels but it eventually reaches a high accuracy level. The reason is that similarly to `max-loss` policy, it selects more persistently important agents at the cell edge. Overall, the key message from the analysis of Scenario 3 is:

TABLE 2. **Accuracy level reached for every policy after 300 seconds for each scenario, where the highest accuracy per scenario is marked in bold.**

Policy	Scenario 1	Scenario 2	Scenario 3
`max-sum-dev`	**0.87**	0.72	0.77
`max-sum-loss`	**0.87**	0.73	0.79
`max-sum-rate/FedCS`	0.84	0.66	0.68
`max-dev`	0.81	0.71	0.71
`max-loss`	0.84	**0.76**	**0.82**
`pow-d`	0.78	0.73	0.76
`random`	0.77	0.68	0.70

Result 6. *A short latency budget $T_{\text{APP,MAX}}$ in a non-IID setting limits the gains of the channel-aware policies. In the long term, the* `max-loss` *policy can provide a higher accuracy because it selects agents with persistently poor bit rates that have beneficial data for the learning process.*

E. SCENARIO COMPARISONS

The policies in Scenarios 1-3 can be compared in terms of what accuracy levels they have reached by a given deadline as well as in terms of how much time is needed to reach a certain targeted accuracy level.

1) DEADLINE

Considering that some applications may require the training to be completed within a given deadline, we compare the policies over the three scenarios by a deadline of 300s, i.e., 5 minutes. Fig. 9 shows the accuracy for every policy and scenario around the 300s deadline, while Table 2 shows the measured accuracy level, which is derived by averaging the accuracy over a 30-second period, therefore from 270s to 300s. The averaging of the accuracy in Table 2 is performed to ensure that the provided results are not dominated by the accuracy fluctuations. From both Fig. 9 and Table 2,

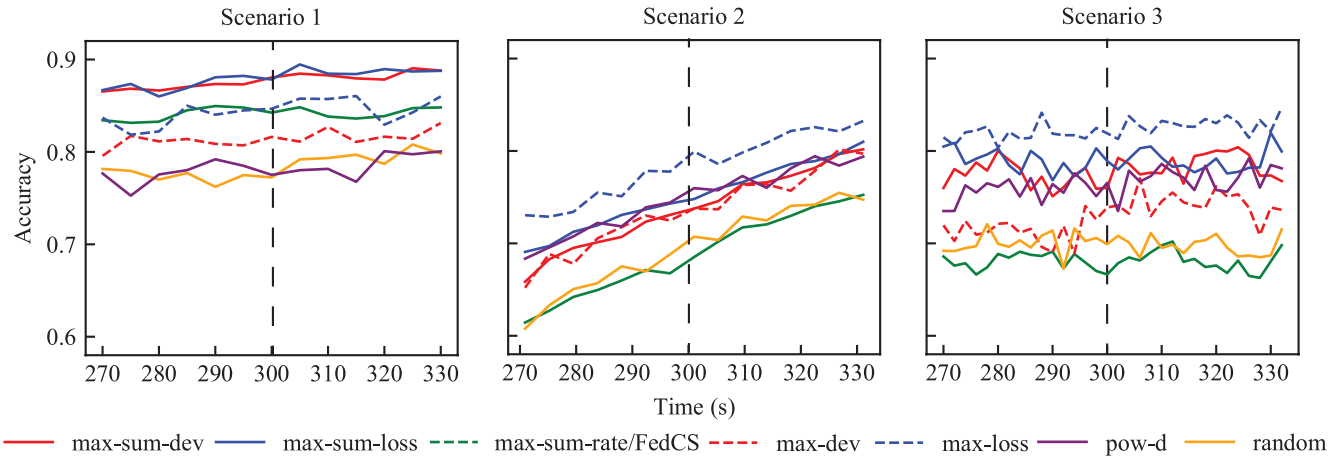


FIGURE 9. **Accuracy of the considered policies for non-IID data around the time intervals of interest, where the dashed line indicates the 300 seconds deadline.**

TABLE 3. **Time, in seconds, needed to reach the 75%, 80% and 85% Accuracy levels for every policy in each scenario, where the shortest time per level and scenario is marked in bold.**

Policy	Scenario 1			Scenario 2			Scenario 3		
	75%	80%	85%	75%	80%	85%	75%	80%	85%
max-sum-dev	**195**	**225**	270	318	344	378	**150**	-	-
max-sum-loss	200	**225**	**255**	314	340	374	156	358	-
max-sum-rate/FedCS	225	250	355	340	370	(421)	-	-	-
max-dev	230	275	-	318	348	391	364	-	-
max-loss	220	250	315	**297**	**323**	**357**	168	**242**	-
pow-d	270	375	-	314	344	400	275	384	-
random	260	380	-	335	370	(409)	-	-	-

we observe that each policy reaches a higher accuracy level in Scenario 1 than in Scenarios 2 and 3. The policies in Scenario 1 perform better than in Scenario 2 because the agents do not suffer from a small data set size. In Scenario 1, the policies perform better than in Scenario 3 because the larger latency budget $T_{\text{APP,MAX}}$ allows the selection of more agents in a given communication round. Additionally, Fig. 9 shows that even though the accuracy of the policies in Scenarios 1 and 3 are roughly stable, the accuracy of the policies in Scenario 2 is still sharply increasing, because more communication rounds are needed to reach convergence when the agents have a small data set.

Table 2 also shows that among the policies that implicitly take the channels into account and the `pow-d` policy that does not consider the channels, the `max-loss` policy converges to a higher accuracy level. Moreover, the `max-sum-dev` and `max-sum-loss` policies converge to approximately the same accuracy level, regardless of the scenario and they provide the highest accuracy in Scenario 1, as explained in Result 4. However, in Scenario 2, where the agents have limited samples and in Scenario 3, where the latency-budget $T_{\text{APP,MAX}}$ is short, the highest accuracy level is provided by the `max-loss` policy, for the reasons provided in Results 5 and 6, respectively.

2) ACCURACY TARGET

Some applications require to train the global model until a specific accuracy target is met. Therefore, we compare the policies in the three scenarios in terms of how much time is needed to reach the 75%, 80% and 85% accuracy levels. We consider that an accuracy level is reached if the average accuracy over a period of 30s is above the accuracy target. Table 3 shows the time in seconds to reach each accuracy level, where a hyphen indicates that the accuracy level could not be reached within the simulated 400s while the values in parenthesis under Scenario 2 indicate that the accuracy level is measured after 400s.

Table 3 shows that the accuracy levels are reached faster in Scenario 1 than in Scenario 2. The reason is that agents have more samples in Scenario 1 and hence, the FL server can train on more samples in a given time period. Table 3 also shows that when the latency budget $T_{\text{APP,MAX}}$ is set to a small value, i.e. in Scenario 3, the `max-sum-dev`, `max-sum-loss` and `max-loss` policies reach the 75% accuracy level faster than when $T_{\text{APP,MAX}}$ is set to a larger value, i.e. in Scenario 1. This is because in Scenario 3 more communication rounds are performed within a given time interval than in Scenario 1. However, in Scenario 1, higher accuracy levels can be achieved within the 400s time interval

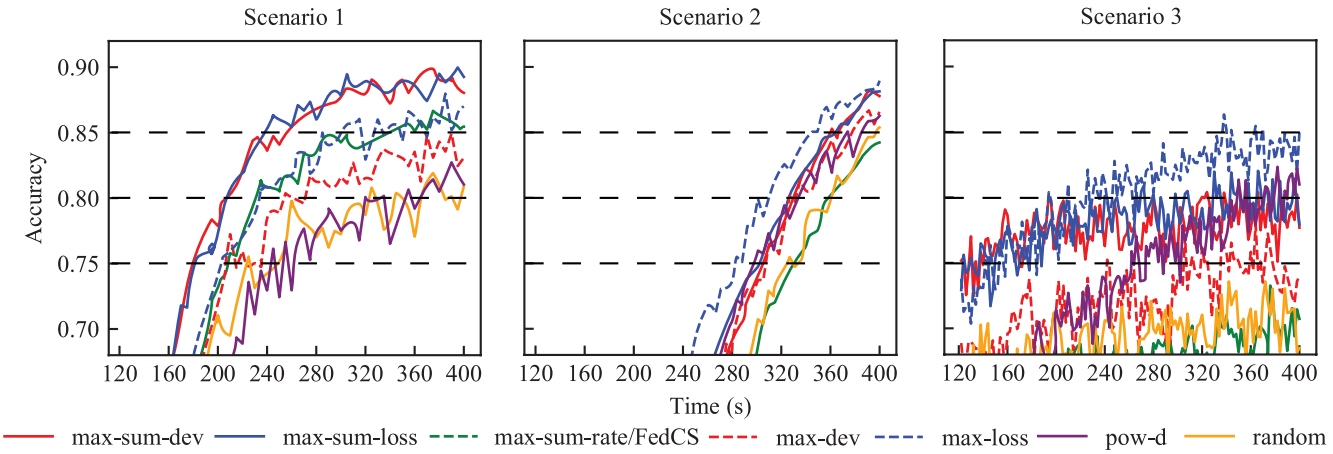


FIGURE 10. **Accuracy of the considered policies for non-IID data around the accuracy levels of interest, where the dashed lines indicate the 75%, 80% and 85% accuracy levels.**

compared to Scenario 3 because more agents can consistently contribute to the learning process. For example, in Scenario 1, the `max-sum-dev` policy can reach the 85% accuracy level within 270s while in Scenario 3, none of the policies can reach the 85% accuracy level within 400s.

Moreover, Table 3 shows that the policies in Scenario 2 generally reach the 75% accuracy target at a later time compared to Scenarios 1 and 3. However, higher accuracy targets can be achieved in Scenario 2 than in Scenario 3, within the 400s time period. This observation is also illustrated in Fig. 10, as the accuracy curves in Scenario 2 are still in an increasing phase while in Scenario 3 they are fairly constant, which shows that the accuracy will not further improve significantly. The only two policies that are in an increasing phase in Scenario 3 are the `max-loss` and `pow-d` policies because they select agents on the cell edge more persistently. Yet, their increase is more modest compared to Scenario 2. This result highlights that the more agents can participate in the learning process, even if those agents have a small data set, the higher accuracy levels can be achieved within a given long-term time period, because more diverse data are used for the training. Therefore, the comparison among scenarios has the following important messages:

Result 7. *When the latency-budget $T_{APP,MAX}$ is set to a small value, the policies initially learn faster than when the latency-budget $T_{APP,MAX}$ is large. However, in the long term, a higher accuracy is achieved with a large latency-budget $T_{APP,MAX}$.*

Result 8. *Regardless of the considered scenario, the more agents with diverse data are selected, the higher the accuracy level that can be achieved.*

F. ENERGY CONSIDERATIONS

In this section, we analyze the total energy consumption of the agents for every policy and scenario. Fig. 11 shows the energy consumption in Joules after a 300-second time interval and it illustrates that the total energy consumption is dominated by

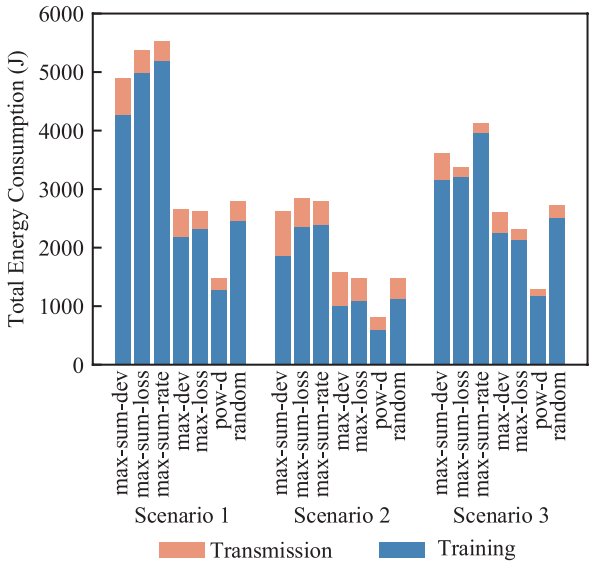


FIGURE 11. **Total energy consumption of agents for every policy and scenario after a time period of 300 seconds.**

the training, rather than the transmissions. Therefore, the total energy consumption depends primarily on the total number of agents selected within the given time interval. Consequently, the energy consumption is higher when channel-aware policies are applied, regardless of the scenario, as such policies generally select more agents. For example, the `pow-d` policy, which does not consider the wireless channels, selects the least amount of agents per communication round, thus leading to the lowest energy consumption, as also shown in Fig. 11.

Fig. 11 also shows that the ratio of the training energy consumption between the policies considering explicitly and implicitly the wireless channels, is smaller in Scenario 3 than in Scenarios 1 and 2. The reason is that the channel-aware policies select fewer agents in Scenario 3 compared to Scenarios 1 and 2 due to the shorter latency budget $T_{\text{APP,MAX}}$.

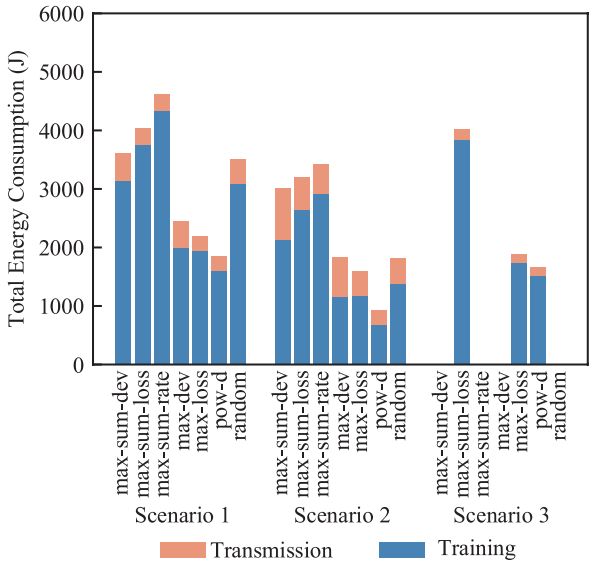


FIGURE 12. **Total energy consumption of agents for every policy and scenario to reach an accuracy level of 80%.**

Despite more rounds occurring within the 5-minute interval, the total number of selected agents is lower than in Scenario 1. Consequently, both transmission and training energy consumption are reduced in Scenario 3 compared to Scenario 1. Moreover, the total energy consumption in Scenario 2 is lower than in Scenarios 1 and 3, which is due to the agents having fewer samples and consequently, shorter training times. However, because of the shorter training times, more communication rounds are performed during the 5-minute time interval compared to Scenario 1. In addition, Scenarios 1 and 2 have a similar number of agents selected per communication round, implying that the transmission related energy consumption is higher in Scenario 2 than in Scenario 1.

Table 2 shows that in Scenario 1, the `max-sum-dev` and `max-sum-loss` policies provide the highest accuracy, whereas Fig. 11 shows that their energy consumption is high. However, the `max-loss` policy achieves a slightly lower accuracy than the two above-mentioned policies while consuming about half the amount of energy. Additionally, the `max-loss` policy achieves the highest accuracy in Scenarios 2 and 3 while also consuming comparatively lower energy due to selecting few agents. Therefore, the `max-loss` policy provides a good trade-off between accuracy and energy consumption.

Fig. 12 shows the total energy consumption of the agents per policy and scenario, until the time that the 80% accuracy level is reached. The absence of a bar in Fig. 12, as it occurs for policies in Scenario 3, implies that the 80% accuracy level was not reached within 400s. Comparing the energy consumption in Fig. 12 to the accuracy levels in Table 3, a trade-off between time and energy is observed in Scenario 1. Specifically, the `max-sum-dev` and `max-sum-loss` policies reach the 80% accuracy level the fastest. However, both policies have a higher energy consumption than the `max-loss` policy, which reaches the 80% accuracy level 25s later. Another observation is that, despite the policies in Scenario 2 taking longer to reach the 80% accuracy level compared to Scenario 1, they consume less energy. Overall, our takeaway message from the energy impact on the system is:

Result 9. *The* `max-loss` *policy provides a good balance between achieving high accuracy levels fairly quickly and consuming less energy due to selecting fewer agents per communication round.*

VII. CONCLUSION AND FUTURE WORK

This work has investigated the agent selection problem for FL in wireless communication environments. We proposed a general optimization problem which can be adapted to a range of applications, depending on the needs and capabilities of the network and the agents. We focused on the important subproblem of latency-constrained networks, which is a 0/1 Knapsack problem. We obtained its solution with a pseudo-polynomial algorithm with low complexity. Then, we conducted extensive simulations, which lead to the important Results 1-9. Overall, our Results 1-9 indicated that the loss is a very good metric to describe the importance of an agent in the learning process. Additionally, we showed that the policies derived from the optimization problem, performed better than only channel-aware and only learning-aware policies, as indicated in Result 4. Moreover, we also showed in Results 5-6 that learning-based policies performed well when the agents had few samples and when the wireless channel could not be largely exploited due to short latency budgets.

For future works, there are several interesting directions worth investigating. First of all, the specific convergence analysis that highlights both the wireless and learning aspects is left for future work. In this work we calculated the deviation by equally considering all the parameter features of the model. For future works, it is interesting to investigate a weighted calculation of the deviations, in which only the parameter features of the model that are important to the learning are considered. Moreover, as Results 5-8 indicated, adaptive policies based on the scenario and communication round are expected to further improve the accuracy. Additionally, the problem with agents having different hardware and/or data set sizes is worth investigating due to the trade-off between learning accuracy, latency deadline and energy consumption showed in Results 5-9. Finally, we also leave for future work the problem of joint agent selection and resource allocation.

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, vol. 54, Fort Lauderdale, FL, USA, 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf

[2] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[3] Z. Du, C. Wu, T. Yoshinaga, K. A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular Internet of Things: Recent advances and open issues," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45–61, 2020.

[4] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastruct. Deep Learn.*, Dec. 2018, pp. 1–8.

[5] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[6] Z. Charles, Z. Garrett, Z. Huo, S. Shmulyian, and V. Smith, "On large-cohort training for federated learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–11. [Online]. Available: https://openreview.net/pdf?id=Kb26p7chwhf

[7] Y. J. Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *Proc. 25th Int. Conf. Artif. Intell. Stat.*, 2022, pp. 10351–10375. [Online]. Available: https://proceedings.mlr.press/v151/jee-cho22a.html

[8] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in *Proc. 15th USENIX Symp. Oper. Syst.-Design Implement.*, 2021, pp. 19–35. [Online]. Available: https://www.usenix.org/conference/osdi21/presentation/lai

[9] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. Vincent Poor, "Fast-convergent federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 201–218, Jan. 2021.

[10] W. Chen, S. Horváth, and P. Richtárik, "Optimal client sampling for federated learning," *Trans. Mach. Learn. Res.*, Oct. 2022. [Online]. Available: https://openreview.net/forum?id=8GvRCWKHIL

[11] M. Ribero and H. Vikalo, "Reducing communication in federated learning via efficient client sampling," *Pattern Recognit.*, vol. 148, Apr. 2024, Art. no. 110122.

[12] H. Hellström et al., "Wireless for machine learning: A survey," *Found. Trends Signal Process.*, vol. 15, no. 4, pp. 290–399, 2022.

[13] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.

[14] H. H. Yang, Z. Liu, T. Q. S. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 317–333, Jan. 2020.

[15] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3643–3658, Jun. 2021.

[16] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient radio resource allocation for federated edge learning," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[17] L. Yu, R. Albelaihi, X. Sun, N. Ansari, and M. Devetsikiotis, "Jointly optimizing client selection and resource management in wireless federated learning for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4385–4395, Mar. 2022.

[18] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2021.

[19] K. Fan, W. Chen, J. Li, X. Deng, X. Han, and M. Ding, "Mobility-aware joint user scheduling and resource allocation for low latency federated learning," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2023, pp. 1–6.

[20] A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Data-driven participant selection and bandwidth allocation for heterogeneous federated edge learning," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 53, no. 9, pp. 5848–5860, Sep. 2023.

[21] W. Zhang, X. Wang, P. Zhou, W. Wu, and X. Zhang, "Client selection for federated learning with non-IID data in mobile edge computing," *IEEE Access*, vol. 9, pp. 24462–24474, 2021.

[22] L. Huang, Y. Yin, Z. Fu, S. Zhang, H. Deng, and D. Liu, "LoAdaBoost: Loss-based AdaBoost federated machine learning with reduced computational complexity on IID and non-IID intensive care data," *PLoS ONE*, vol. 15, no. 4, Apr. 2020, Art. no. e0230706.

[23] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. 3rd Mach. Learn. Syst. Conf.*, 2020, pp. 429–450. [Online]. Available: https://proceedings.mlsys.org/paperfiles/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf

[24] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5132–5143. [Online]. Available: https://proceedings.mlr.press/v119/karimireddy20a.html

[25] D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *Proc. Int. Conf. Learn. Represent.*, 2021. [Online]. Available: https://openreview.net/pdf?id=B7v4QMR6Z9w

[26] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on Non-IID data," in *Proc. Int. Conf. Learn. Represent.*, 2020. [Online]. Available: https://openreview.net/forum?id=HJxNAnVtDS

[27] J. Wang, R. Das, G. Joshi, S. Kale, Z. Xu, and T. Zhang, "On the unreasonable effectiveness of federated averaging with heterogeneous data," 2022, *arXiv:2206.04723*.

[28] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," 2017, *arXiv:1702.05659*.

[29] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2022.

[30] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[31] Q. Zeng, Y. Du, K. Huang, and K. K. Leung, "Energy-efficient resource management for federated edge learning with CPU-GPU heterogeneous computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, pp. 7947–7962, Dec. 2021.

[32] B. Korte and J. Vygen, *Combinatorial Optimization, Theory and Algorithms*, 5th ed., Berlin, Germany: Springer, 2012, pp. 459–470, doi: 10.1007/978-3-642-24488-9.

[33] C. G. Serna and Y. Ruichek, "Classification of traffic signs: The European dataset," *IEEE Access*, vol. 6, pp. 78136–78148, 2018.

[34] S. Chilamkurthy. (2017). *Keras Tutorial-Traffic Sign Recognition*. [Online]. Available: https://chsasank.com/keras-tutorial.html

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[36] 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 Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Jan. 2021.

[37] *Study on Scenarios and Requirements for Next Generation Access Technologies*, Standard TR 38.913; 5G, 3GPP, 2022. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2996

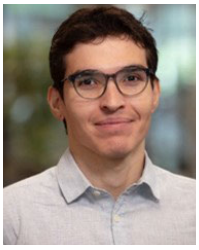
[38] *Study on Channel Model for Frequencies From 0.5 to 100 GHz*, Standard TR 38.901; 5G, 3GPP, 2024. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173

[39] *User Equipment (UE) Radio Transmission and Reception (TDD)*, Standard TS 25.102, 3GPP, 2022. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1152

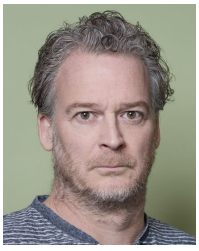
[40] M. Raftopoulou and J. M. B. da Silva Jr., "FLoverWireless: System-level simulator for FL over wireless networks," Zenodo, Jun. 2024, doi: 10.5281/zenodo.12506109. [Online]. Available: https://zenodo.org/records/12506109

[41] A. Krizhevsky. (2009). *Learning Multiple Layers of Features From Tiny Images*. [Online]. Available: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

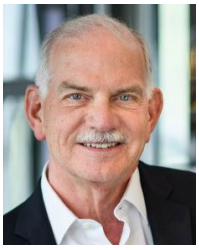
MARIA RAFTOPOULOU received the M.Eng. degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 2016, the M.Sc. degree (cum laude) in telecommunications and sensing systems from Delft University of Technology, The Netherlands, in 2018, and the Ph.D. degree from Delft University of Technology in 2024. She is currently a Scientist Innovator with the Netherlands Organisation for Applied Scientific Research (TNO), The Netherlands. From 2018 to 2019, she was a Technology Young Talent with KPN, The Netherlands.



JOSÉ MAIRTON B. DA SILVA JR. (Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in telecommunications engineering from the Federal University of Ceara, Brazil, in 2012 and 2014, respectively, and the Ph.D. degree from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2019. He is currently an Assistant Professor with the Division of Computer Systems, Uppsala University, Sweden. He was a Post-Doctoral Researcher with the KTH Royal Institute of Technology, from 2019 to 2021. He was a Marie Sklodowska-Curie Post-Doctoral Fellow with Princeton University, USA, and the KTH Royal Institute of Technology, from 2022 to 2023. His research interests include distributed machine learning and optimization over wireless communications. He has served as the Secretary for the IEEE Communications Society Emerging Technology Initiative on Full Duplex Communications, from 2018 to 2021. He has been involved in the organization of many IEEE conferences and workshops, including co-chairing ICMLCN 2024 and SECON 2022-2023. He gave several tutorials at many IEEE flagship conferences, including ICASSP, PIMRC, ICC, and GLOBECOM.



REMCO LITJENS received the M.Sc. degree (Hons.) in econometrics from Tilburg University, The Netherlands, in 1994, the M.Sc. degree in electrical engineering and computer sciences from the University of California at Berkeley, USA, in 1996, and the Ph.D. degree in applied mathematics from the University of Twente, The Netherlands, in 2003. He was a Technical Scientific Researcher with KPN Research, The Netherlands. He was a Senior Scientist with TNO, The Netherlands, as of 2003. Since 2014, he has been holding a part-time associate professorship with Delft University of Technology, The Netherlands. His core expertise is on the development, assessment, planning, and (self-)optimisation of mobile/wireless networking technologies. He has served on the project board of the European projects SOCRATES and SEMAFOUR, on the management committees of COST 279 and COST 290, and on several technical programme committees of international conferences and workshops. He has published over 100 articles in journals, magazines, and conference proceedings (with Best Paper Awards received at WWIC'08, WCNC'14 and EuCNC'15), a co-inventor of over 25 granted or filed patents, and has written/contributed to five books.



H. VINCENT POOR (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana–Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. From 2006 to 2016, he served as the Dean of Princeton's School of Engineering and Applied Science, and he has also held visiting appointments with several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, and energy systems and related fields. Among his publications in these areas is the book *Machine Learning and Wireless Communications* (Cambridge University Press, 2022). He is a member of the National Academy of Engineering and the National Academy of Sciences and a foreign member of the Royal Society and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.

PIET VAN MIEGHEM (Fellow, IEEE) received the master's and Ph.D. degrees in electrical engineering from KU Leuven, Belgium, in 1987 and 1991, respectively. He has been a Professor with Delft University of Technology and the Chairman of the Section Network Architectures and Services (NAS), since 1998. Before joining Delft, he was with the Interuniversity Micro Electronic Center (IMEC), from 1987 to 1991. From 1993 to 1998, he was a member of the Alcatel Corporate Research Center, Antwerp. He was a Visiting Scientist with MIT (1992–1993); and a Visiting Professor with UCLA, in 2005, Cornell University, in 2009, Stanford University, in 2015, and Princeton University, in 2022. He is the author of four books, such as *Performance Analysis of Communications Networks and Systems*, *Data Communications Networking*, *Graph Spectra for Complex Networks*, and *Performance Analysis of Complex Networks and Systems*. He is a Board Member of the Netherlands Platform of Complex Systems, a Steering Committee Member of the Dutch Network Science Society, and an External Faculty Member with the Institute for Advanced Study (IAS) of the University of Amsterdam. He received the Advanced ERC Grant 2020 for ViSiON, Virus Spread in Networks. Currently, he serves on the editorial board of the *OUP Journal of Complex Networks*. He was a member of the editorial board of *Computer Networks* (2005–2006), IEEE/ACM TRANSACTIONS ON NETWORKING (2008–2012), *Journal of Discrete Mathematics* (2012–2014), and *Computer Communications* (2012–2015).