GWO-Boosted Multi-Attribute Client Selection for Over-The-Air Federated Learning

Maryam Ben Driss•°, Essaid Sabir*, Halima Elbiaze•, Abdoulaye Baniré Diallo•, Mohamed Sadik°
•Department of Computer Science, University of Quebec at Montreal, Montreal, H2L 2C4, Quebec, Canada
*Department of Science and Technology, TÉLUQ, University of Quebec, Montreal, H2S 3L4, Canada
°NEST Research Group, LRI Lab, ENSEM, Hassan II University of Casablanca, Morocco
Emails: {bendriss.maryam, sabir.essaid, elbiaze.halima, diallo.abdoulaye}@uqam.ca, m.sadik@ensem.ac.ma

***Abstract*—Federated Learning (FL) has gained popularity across various industries due to its ability to train machine learning models without explicit sharing of sensitive data. While this paradigm offers significant advantages such as privacy preservation and reduced communication overhead, it also comes with several challenges such as deployment complexity and interoperability issues, particularly in heterogeneous scenarios or resource-constrained environments. Over-the-air (OTA) FL was introduced to address those challenges by sharing model updates without the need for direct device-to-device connections or centralized servers. However, OTA-FL induces some issues related to increased energy consumption, wireless channel variability, and network latency. In this paper, we propose a multi-attribute client selection framework using the Grey Wolf optimizer to limit the number of participants in each round and optimize the OTA-FL process while considering the energy, delay, reliability, and fairness constraints of participating devices. We analyze the performance of our client selection approach in terms of model loss, convergence time, and overall accuracy. Our experimental results show that the proposed multi-attribute client selection can lower energy consumption by up to 43% compared to the random client selection method.**

***Index Terms*—Over-The-Air Federated Learning; Client Selection; Grey Wolf Optimizer; Convergence Speed; Energy Efficiency; Reliability; Fairness.**

I. INTRODUCTION

Artificial intelligence (AI) has the potential to transform many aspects of human society. From healthcare and education to finance, transportation, and beyond, AI's ability to analyze vast amounts of data, make predictions, and automate tasks holds the promise of improving efficiency, accuracy, and overall quality of life. However, traditional machine learning (ML) in massive and sensitive environments faces several challenges due to the nature of large-scale datasets, distributed data sources, and their constraints such as data privacy, limited resources, and network heterogeneity. To address these issues, federated learning (FL) is a promising approach to train ML algorithms where devices collaborate to improve a shared model while preserving users' privacy and reducing communication overhead [1]. Instead of sending raw data to a central server for aggregation, each device maintains its dataset, trains a local model, and sends model updates or gradients to the server that aggregates these updates, and then sends back the refined model to the individual devices. This process iterates until the global model reaches the desired accuracy. Over-the-air federated learning (OTA-FL) [2] is a promising concept that allows clients to share the same spectral resources by simultaneously transmitting their local model updates and aggregating these models over the air in a "one-time" manner, as illustrated in Fig. 1. Thus, OTA-FL can greatly reduce the cost of communicating model updates from the edge devices. Implementing OTA-FL in heterogeneous scenarios, where clients have different data distribution, limited bandwidth, and less reliable network conditions, faces several challenges including limited computing capabilities, data quality, and fairness between FL agents. Thus, the set of participants in each training round is a key factor in addressing these challenges and enhancing the learning process [3]. Careful selection of clients profoundly impacts the overall performance, convergence speed, and robustness of the global model. By strategically choosing clients based on factors such as data quality and computational capabilities, the FL system can effectively navigate through communication constraints, privacy concerns, and other challenges.

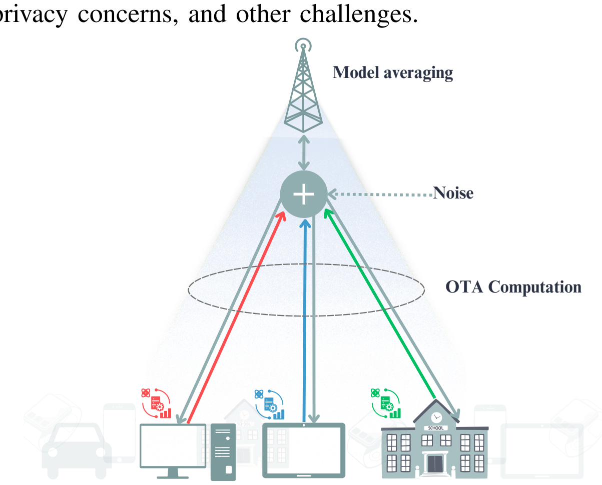

Fig. 1. Over-The-Air (OTA) federated learning process.

In this paper, we present an optimization problem that aims to develop a multi-attribute client selection framework using the grey wolf optimizer (GWO), taking into account several criteria such as model accuracy, communication cost, resource capacity and reliability, and fairness between FL clients. The remainder of this paper is organized as follows: Section II summarizes past research utilizing different techniques and criteria for client selection and presents our contribution.

Section III provides the system model and the objective function. Section IV details the mechanism of the grey wolf optimizer for selecting clients to participate in the learning process. Section V presents experimental results and insights, and Section VI concludes the paper.

II. Related Work and Our Contribution

Effective client selection is crucial for fast convergence, accurate models, fairness, and efficient communication. This section presents a literature review focused on optimizing client selection through various methods and our contribution.

A. Random Selection

This client selection method is achieved by randomly selecting a subset of clients to participate in the FL process. The work in [4] mitigates this problem and performs FL while actively managing clients based on their resource conditions by asking the randomly selected clients to send their resource information and participate in determining which of them go to complete the FL process. However, this approach presents several challenges such as building and maintaining client trust and ensuring high data quality. The random selection strategy is simple to implement but may lead to uneven data distribution and performance.

B. Learning-based Selection

Some papers implement client selection using ML techniques, where a central model predicts which clients provide high-quality updates. For instance, reinforcement learning is deployed to improve client selection performance by involving a reinforcement learning agent that learns a client selection policy [5]. The authors of [6] designed a framework that intelligently chooses the client devices to participate in each round of FL to counterbalance the bias introduced by non-IID data and to speed up convergence. Although this method allows for adaptive client selection strategies, it is computationally intensive, requires additional training, and may be sensitive to the quality of the initial model.

C. Mathematical optimization-based selection

Some methods formulate the client selection strategy as a mathematical optimization problem. Then, clients are selected using mathematical methods such as the Knapsack model in [7], where the authors proposed a framework to balance the trade-off between the energy consumption of the edge clients and the learning accuracy of FL. The authors in [8] proposed a predictive quality of service paradigm that allows devices to self-adjust their power allocation to maintain reliability and latency within the tolerated range of the URLLC application. In [9], the authors proposed a delay-constrained client selection framework for heterogeneous FL in intelligent transportation systems to improve the model performance such as accuracy, training, and transmission time. The multi-armed bandit (MAB) model is used in [10] to work for the hierarchical FL by estimating the participation probability for each client using the following information wireless channel state, local computing resources, and previous performance. The authors of [11] also formulated the client selection problem as an MAB problem to design a selection framework where the network operator learns the number of successful participating clients to improve the training performance as well as under the limited budget on each edge server. Contextual combinatorial MAB is used in [12] to formulate a client selection problem to boost volatile FL by speeding up model convergence, promoting model accuracy, and reducing energy consumption. The authors in [13] leveraged the MAB framework and the virtual queue technique in Lyapunov optimization to conduct client selection with a fairness guarantee in the asynchronous FL framework. Authors of [14] proposed a client selection method using a Genetic algorithm, which enables faster central model training at a lower cost based on the client's cost and the result of its local update. A dynamic and multicriteria scheme for client selection is developed in [15] to offer more volume and heterogeneity of data in the FL process using a genetic algorithm.

D. Our contributions

Based on the related works (See Table I), certain client selection methods choose the clients with the best performance or high resources. This approach results in clients with low-level resource capacity being unable to participate in the training process, and their datasets being ignored. This leads to biased and unfair selection, which ultimately results in an underfitting of the learned global model for those low-level clients. Moreover, some proposed methods suffer from some futility of the clients which train their local models and then the server does not aggregate them. This leads to a waste of client energy. The majority of existing works have concentrated on accuracy and cost criteria for selecting clients to participate in the FL process. Additionally, although efforts have been made to employ GWO to facilitate FL in several ways, GWO has never been used to optimize the client selection problem by improving the global model's accuracy, energy, reliability, and fairness.

Our contributions are summarized below:

- Offering a multi-attribute client selection framework that allows the balancing of accuracy with energy, delay, reliability, and fairness criteria to tackle the OTA-FL challenges such as security risks, limited computational capability, and unstable networks.
- Adopting the grey wolf optimizer to choose the set of eligible clients to join the learning process.
- Evaluating the proposed approach and analyzing the FL model performance in terms of accuracy, convergence time, and energy efficiency.

III. Multi-Attribute Client Selection

We consider an FL framework consisting of a single base station and n clients $N = \{1, 2, \cdots, n\}$. Each client i possesses local data, denoted as D_i. In each round, the server aims to learn a global model with the data D_i distributed across the selected clients.

TABLE I
RELATED EXISTING WORKS ON THE MATHEMATICAL OPTIMIZATION-BASED CLIENT SELECTION

Ref	Year	Accuracy	Energy	Delay	Reliability	Fairness	Model
[7]	2021	✓	✓	✓			Knapsack
[9]	2023	✓	✓	✓			Knapsack
[10]	2022	✓	✓				Multi armed bandit
[11]	2022	✓	✓	✓			Multi armed bandit
[12]	2023	✓	✓	✓			Multi armed bandit
[13]	2023	✓	✓	✓		✓	Multi armed bandit
[14]	2023	✓	✓				Genetic algorithm
[15]	2023	✓	✓			✓	Genetic algorithm
Our approach	2023	✓	✓	✓	✓	✓	Grey wolf optimizer

To model the FL problem, we define the weight vector w to capture the parameters related to the global model. We introduce the loss function $l(w, x_j, y_j)$, which captures the FL performance over input vector x_j and output y_j for each D_i. The categorical cross-entropy is used as a loss function in performing the classification problem in our paper. The total loss function of client i writes [16]:

$$F_i(w) = \frac{1}{D_i} \sum_{j=1}^{D_i} l(w, x_j, y_j) \tag{1}$$

The FL training problem can be formulated as follows:

$$\min F(w) = \sum_{i=1}^{n} \frac{D_i}{D} F_i(w), \tag{2}$$

where $D = \sum_{i=1}^{n} D_i$ is the total data samples of all clients.

A. Delay

To implement FL over wireless networks, wireless devices must train the model locally and transmit their results over wireless links. However, this computation and transmission introduce a delay that impacts the overall FL performance. Therefore, it is crucial to optimize the delay for efficient FL implementation.

1) Computation Delay: The computation delay is determined by the type of learning models and the desired learning accuracy ϵ_i, the computation time at user i needed for processing is [16]:

$$\tau_i^c = \frac{C_i D_i}{f_i} \upsilon_i \log_2\left(\frac{1}{\epsilon_i}\right), \tag{3}$$

where $\upsilon_i \log_2(1/\epsilon_i)$ is the number of local iterations required for client i to reach the desired accuracy ϵ_i, C_i (cycles/bit) is the number of CPU cycles required for computing one sample data at user i, and f_i is the computation capacity of user i, which is measured by the number of CPU cycles per second.
2) Transmission Delay: After local computation, all users upload their local FL parameters to the server, the quality of the wireless channel is the primary factor that determines the transmission rate in each round that is given by:

$$r_i = b_i \log_2\left(1 + \frac{g_i p_i}{N_0 b_i}\right), \tag{4}$$

TABLE II
MAIN NOTATIONS USED IN THIS PAPER.

Notation	Meaning
n	Number of clients
i	A single client
D_i	Data samples collected by client i
D	Total data samples of all users
w	Global model parameter vector
x_j	Input vector for each data sample j
y_j	Output vector for each data sample j
$l(w, x_j, y_j)$	Total loss function for client i
$F_i(w)$	Local objective function
$F(w)$	Global objective function
τ_i^c	Computation delay
τ_i^t	Transmission delay
τ_i	Delay requirement
e_i^c	Computation energy
e_i^t	Transmission energy
e_i	Energy consumption requirement
ϵ_i	The desired learning accuracy
C_i	Computation capacity required (CPU cycles per bit)
f_i	Computation capacity of i (CPU cycles per second)
r_i	Transmission rate
b_i	Bandwidth allocated to user i
g_i	Channel gain between user i and the BS
p_i	Transmit power of user i
N_0	Power spectral density of the Gaussian noise
$M(w)$	FL model size
ζ_i	Energy consumption factor of client i
$MTBF_i$	Mean time between failures
$\rho_i(t)$	Reliability of client i
ρ	Reliability requirement
m_i	Number of failures
c_i	Required minimum selection fraction for client i
$\mathbf{X}_p$	Position of the prey
$\mathbf{X}$	Position of the wolf
$\mathbf{A}$ and $\mathbf{C}$	GWO coefficient vectors
$\mathbf{d}$	Distance between the wolf and the prey

where b_i represents the bandwidth allocated to user i, p_i is the transmit power of user i, g_i is the channel gain between user i and the BS, and N_0 is the power spectral density of the Gaussian noise.

The model size determines the transmission time between the client and server, expressed as $M(w)$. The model transmission time is calculated using the following formula:

$$\tau_i^t = \frac{M(w)}{r_i}. \tag{5}$$

B. Energy

Energy is a critical factor to consider when deploying FL, to implement energy-efficient ML algorithms, optimize communications, use low-power hardware accelerators, and develop energy-aware scheduling strategies. Balancing the benefits of FL with the energy constraints of participating devices is crucial for its widespread adoption and long-term sustainability. The energy consumption of each client i is the sum of the energy used to train the model on each client's device and the energy used to transmit the local model from the device to the server.

1) Computation Energy: The computing resources consumed by model training depend on the size of local data D_i, which is expressed as [17]:

$$e_i^c = \zeta_i f_i^2 \cdot \tau_i^c f_i = \zeta_i f_i^2 C_i D_i \upsilon_i \log_2\left(\frac{1}{\epsilon_i}\right). \tag{6}$$

where ζ_i is the energy consumption coefficient depending on the chip of each client i's device. Note that, since the server has a continuous power supply, we do not consider the energy consumption of the server in our problem.

2) Transmission Energy: The energy consumption of client i in model transmission is expressed as [17]:

$$e_i^t = p_i \tau_i^t = p_i \frac{M(w)}{r_i}. \tag{7}$$

C. Reliability

Choosing clients capable of completing local training is a crucial maintenance metric to measure performance, safety, and equipment design, especially for critical or complex assets. The reliability of the client's device ensures the trustworthiness, stability, and efficiency of the FL process. It allows FL systems to make informed decisions regarding the participation of clients, data quality, and security, which results in better model performance and a more dependable and robust learning process [18]. The reliability computation of a client i is performed by considering the time between failures i.e. MTBF (mean time between failures), which refers to the average time between two failures and is defined as follows [19]:

$$MTBF_i = \frac{\tau_i^c}{m_i}, \tag{8}$$

where m_i is the number of failures. The client reliability, or the probability of operating without fail for a time t, is denoted by $\rho_i(t)$:

$$\rho_i(t) = e^{-t/MTBF_i}. \tag{9}$$

A higher reliable client device is less likely to fail shortly. This, in turn, reduces the risk of losing the training data or the local model due to unintentional shutdown and network instability.

D. Fairness

During the FL process, the client selection method often prioritizes devices with low latency. However, this bias towards speed may not be fair to clients with high data quality, the local dataset which has a larger size and whose distribution is more similar to the global distribution plays a more important role, and the corresponding clients should participate in more communication rounds. Therefore, it is important to consider the fairness constraint to avoid an overabundance of relevant clients [20]. The fairness constraint is considered to ”tell” each client how many communication rounds they should participate in [21]. We introduce the following constraint on a minimum selection fraction for each client i [22]:

$$\frac{1}{T}\sum_{t=1}^{T} E[a_i(t)] \geq c_i, \tag{10}$$

where $E[.]$ is the expectation operator and $c_i \in (0,1)$ is the minimum fraction of communication rounds required to choose client i. T is the total number of rounds and $a_i(t)$ is a binary variable defined as an indicator with $a_i(t) = 1$ indicating that client i is selected in round t, and $a_i(t) = 0$ otherwise.

E. Problem Formulation

Our approach involves a ”select and train” client selection method where the server invites clients who meet the constraints of accuracy, energy, delay, reliability, and fairness to participate in the FL algorithm. We formulate our problem whose goal is to minimize the loss function of an FL algorithm by optimizing the various wireless parameters, as follows:

$$\min F(w) = \frac{1}{D}\sum_{i=1}^{n}\sum_{j=1}^{D_i} l(w, x_{ji}, y_{ji}) \tag{11}$$

$$s.t. \quad \tau_i^c + \tau_i^t \leq \tau_i, \qquad \forall i \in N \tag{11a}$$

$$0 < e_i^c + e_i^t \leq e_i, \qquad \forall i \in N \tag{11b}$$

$$\rho_i(t) \geq \rho, \qquad \forall i \in N \tag{11c}$$

$$\frac{1}{T}\sum_{t=1}^{T} E[a_i(t)] \geq c_i \quad \forall i \in N \tag{11d}$$

$$\epsilon_{min} \leq \epsilon_i \leq 1 \quad \forall i \in N \tag{11e}$$

$$0 \leq f_i \leq f_i^{max} \quad \forall i \in N \tag{11f}$$

$$0 \leq p_i \leq p_i^{max} \quad \forall i \in N \tag{11g}$$

$$\sum_{i=1}^{n} b_i \leq B \quad \forall i \in N \tag{11h}$$

$$0 \leq c_i \leq 1 \quad \forall i \in N \tag{11i}$$

where γ_T is the maximum delay to join the FL system, γ_E is the energy consumption of the FL algorithm, γ_R is the minimum reliability needed to participate to the FL process.

Constraint (11a) indicates that the execution time of the local tasks and transmission time for all clients should not exceed the maximum completion time for the whole FL algorithm. (11b) is the energy consumption constraint to perform the learning task. Constraint (11c) is the client's device reliability condition for joining the FL algorithm. Constraint (11d) is the fairness constraint to participate in the FL algorithm. The local accuracy constraint is given by (11e). Constraints (11f) and (11g) respectively represent the maximum local computation capacity and average transmit power limits of all clients. Due to the limited bandwidth of the system, we have (11h), where B is the total bandwidth. Constraint (11i) is the fraction of communication rounds required to ensure a fair selection.

IV. Grey Wolf Optimizer-Based client Selection

A. Federated Learning Algorithm

Our FL is depicted in the pseudo-algorithm 1. It is divided into two pieces, one executed by the server and the other by the clients. In the beginning, the server initializes the global model parameters with random values. The server coordinates different rounds of execution. At each round, the server selects the set of clients using Algorithm 2 and, in parallel, sends a copy of the training model. To fine-tune the copy of the training model, each client performs a series of gradient descent steps using its data. After training, each client sends back the weights and biases of the local model to the server. The server aggregates the updates from all clients and starts a new round.

Algorithm 1 OTA-FL with Multi-Attribute Client Selection

Base Station Side:
Initialize the global model W_0
for $t \leftarrow 0$ to T **do**
  Select client set $\mathcal{C}$ using **Algorithm 2**
  Broadcast W_t to selected clients (i.e., $\mathcal{C}$).
  Receive the over-the-air aggregated global model W_{t+1}.
end for

Selected Client Side:
At each round t:
Receive current global model W_t.
Train local model and produce model update W_{t+1}^c.
Send W_{t+1}^c to the server.

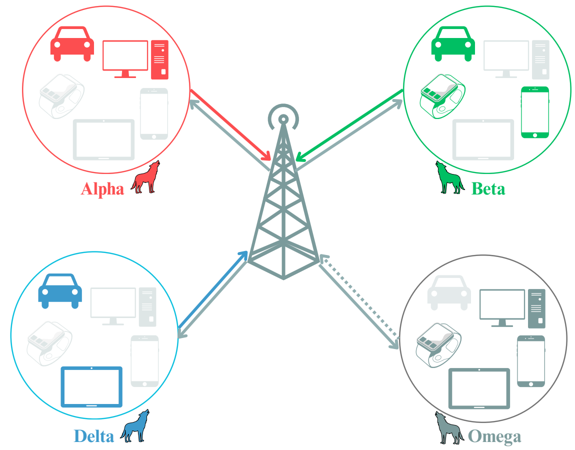

Fig. 2. The wolf in the GWO is the set of clients in the FL process: The selected clients are shown in bold pictures, while transparent pictures represent clients that have not been selected

B. Client Selection Algorithm

The GWO is a metaheuristic algorithm that mimics the social hierarchy and hunting behaviors of the grey wolves to catch the prey in nature. It's used to solve different problems such as global optimization problems due to the advantages of fewer parameters, simple principles, and implementation [23]. In this work, we employ the grey wolf model for Optimizing the client selection problem (Eq. 11), wherein the wolf is represented as the set of clients that are eligible to join the learning process (See Fig.2).

Let's assume that there are S solutions (sets of clients) in the search space, GWO classifies these solutions based on the objective function (Eq.11) for four categories as follows: the best solution is alpha (α), the second-best is beta (β), the third-best delta (δ) and the rest solutions are omega (ω). The best three solutions (α, β, δ) are used to guide the other solutions (ω) for improving the search space. During the optimization, there are three main phases of hunting behavior: Encircling, hunting, and attacking which will be detailed later.

1) Encyrcling: When the grey wolves are on the hunt, they start by creating a circle around the prey. The mathematical model of the encircling phase is developed using the following equations:

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A} \times \mathbf{d}. \tag{12}$$

The distance $\mathbf{d}$ between the wolf and the prey is calculated by the following equation:

$$\mathbf{d} = |\mathbf{C} \times \mathbf{X}_p(t) - \mathbf{X}(t)|, \tag{13}$$

where t is the current iteration, $\mathbf{X}_p$ is the position of the prey and $\mathbf{X}$ is the position of the wolf. $\mathbf{A}$ and $\mathbf{C}$ are coefficient vectors defined as follows:

$$\mathbf{A} = 2\mathbf{a} \times \mathbf{r}_1 - \mathbf{a}, \tag{14}$$

$$\mathbf{C} = 2\mathbf{r}_2. \tag{15}$$

The components of $\mathbf{a}$ are linearly decreased from 2 to 0 over iterations and can be calculated by:

$$a = 2 - t \times 2/max_{itr}, \tag{16}$$

where max_{itr} is the maximum number of iterations. $\mathbf{r}_1$ and $\mathbf{r}_2$ are random vectors in $[0, 1]$.

2) Hunting: During this phase, the three most promising solutions denoted by (α, β, δ) are obtained. As for the other research agents (ω), they need to update their positions by moving towards the average of the three best-known positions since they have better knowledge about the optimal location of the prey. In this regard, the following equations have been presented with $i \in \{\alpha, \beta, \delta\}$:

$$\mathbf{X}_i(t+1) = \mathbf{X}_i(t) - \mathbf{a}_i \times \mathbf{d}_i, \tag{17}$$

where $\mathbf{d}_i$ is estimated using the following:

$$\mathbf{d}_i = \left| \mathbf{C}_i \times \mathbf{X}_i(t) - \mathbf{X}(t) \right|. \tag{18}$$

Let p_i be the positive weight associated with wolf $i \in \{\alpha, \beta, \delta\}$ such that $\sum_i p_i = 1$. Given the positions of wolves α, β, and δ, a good estimation of the average position of the optimal solution at round t is given by:

$$\mathbf{X}(t+1) = \sum_{i \in \{\alpha,\beta,\delta\}} p_i \cdot \mathbf{X}_i(t+1). \tag{19}$$

3) Attacking: GWO finishes hunting by attacking the prey when it stops moving, to model approaching the prey we use Eq. (16) as the parameter a is responsible for making the balance between exploration and exploitation, the value of a linearly decreased from 2 to 0 over iterations, consequently, the parameter A takes a random value in the interval $[-2a, 2a]$ given by Eq. (14). The wolves take a random position when $A > 1$ or $A < -1$ and they are forced to move towards the prey when $-1 \leq A \leq 1$.

Algorithm 2 Grey Wolf Optimizer-Based client Selection

```
Initialize the grey wolf population X
Initialize a, A, and C
Calculate the fitness of each search agent
X_α = the best search agent
X_β = the second best agent
X_δ = the third best search agent
while t < max_itr do
    for each search agent do
        Randomly initialize r_1 and r_2
        Update the position of the current search agent using Eq.(19)
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    t = t + 1
end while
return X_α        ▷ Best solution: Set of clients to join the FL
```

The multi-attribute client selection is provided in Algorithm.2. First, the GWO parameters are initialized by the base station. Second, the GWO calculates the score of the set of best clients based on the lowest loss value, lowest computation and transmission delay, lowest energy consumption, highest reliability, and fairness. The best score value is sent to the BS from each set of clients. Finally, the local models are trained by the best clients with the best score (i.e., alpha solution) and sent to the base station for aggregation via OTA communication.

V. Experimental Investigation

To evaluate the performance of our approach, we conducted experiments to analyze the global model performance as well as study the impact of delay, energy, reliability, and fairness. In this section, we provide a comparison between the random client selection of 3 clients and then 5 clients, loss-aware client selection using GWO, and our multi-attribute client selection approach based on the local model accuracy, energy, delay, reliability, and fairness using the MNIST dataset.

A. Experimental setup

We implemented an FL model using the MNIST dataset, which consists of 60,000 28 x 28 images of handwritten digits ranging from 0 to 9, these images were distributed among 10 clients to train the FL model where each client has its hardware parameters which allowed us to calculate the total delay using Eq.(3) and Eq.(5), the total energy using Eq. (6) and Eq.(7), reliability using Eq.(9) and fairness using Eq.(10). Our study was conducted on a cloud-based platform, Google Colab T4 GPU. We used Python version 3, TensorFlow version 2.3.0, and Keras version 2.4.3 to create our experimental code. Our classification problem was solved using the convolutional neural network (CNN) algorithm to train local models with the stochastic gradient descent (SGD) technique for training acceleration. Our study aims to improve the FL performance using GWO by selecting clients that can achieve the best score regarding prediction, delay, energy, reliability, and fairness.

B. Experimental Results

To show the performance of our client selection approach in OTA-FL, we analyze the FL model on the MNIST classification problem. The analysis focused on measuring the global model loss, convergence time, accuracy, and energy consumption. We also compare the results of our solution with various other methods including random client selection and GWO loss-aware client selection.

Loss probability: Fig. 3 shows the loss of the FL model across different communication rounds using random client selection of 3 clients and 5 clients, loss-based client selection, and our client selection method. It is clear that the model loss probability obtained using our approach is lower and converges faster compared to the other selection methods.

Convergence Time: Fig. 4 presents a comparison of the FL convergence time for 10 communication rounds, where our approach generally has lower values compared to GWO loss-based client selection and random selection, suggesting better performance in terms of convergence speed.

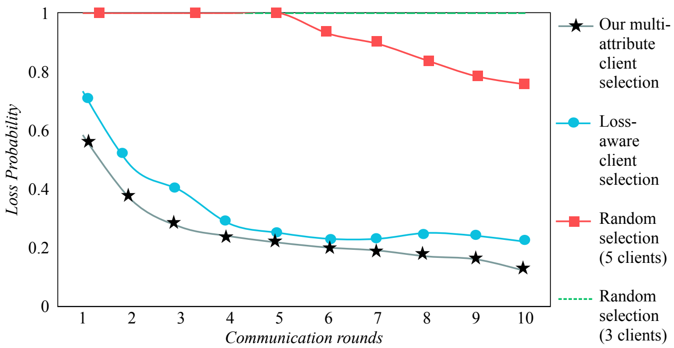

Fig. 3. FL model loss under different client selection schemes.

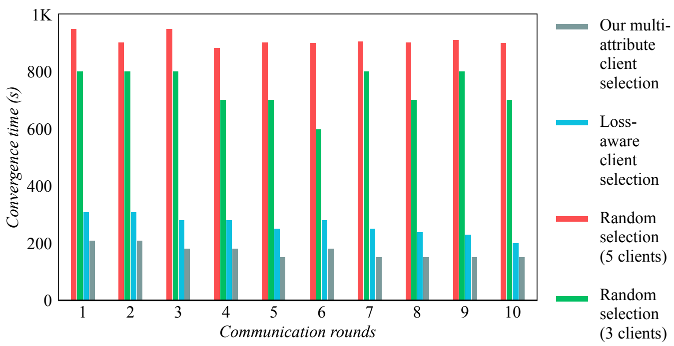

Fig. 4. Convergence time under different client selection schemes.

Accuracy: The Global model accuracy is increased across iterations as seen in Fig. 5 While Table III summarizes the global model accuracy achieved using different client selection methods, where the highest value is obtained using our multi-attribute client selection scheme.

TABLE III
GLOBAL ACCURACY AND ENERGY EFFICIENCY.

Client Selection Method	Accuracy (%)	EE (%/joule)
Random selection (3 clients)	68	0.46
Random selection (5 clients)	73	0.41
Loss-aware client selection	92	0.80
Our multi-attribute client selection	98	1.08

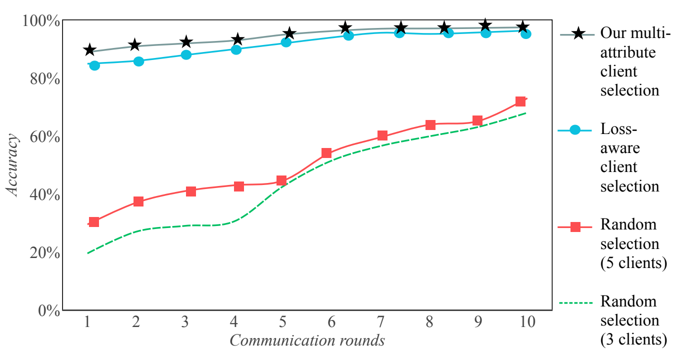

Fig. 5. FL accuracy under different client selection schemes.

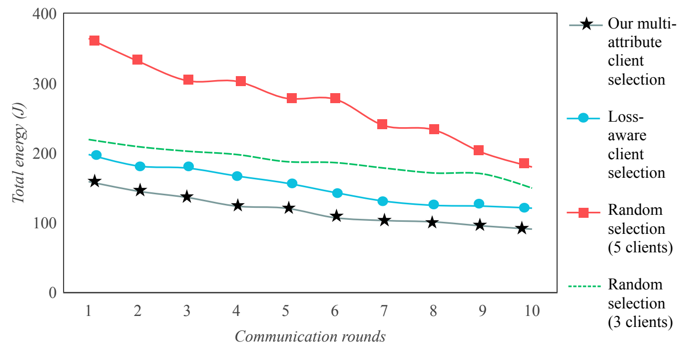

Fig. 6. Total energy consumption under different client selection schemes.

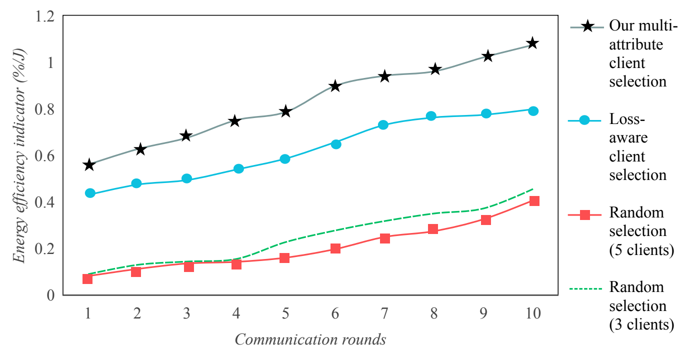

Fig. 7. Energy efficiency indicator under different client selection schemes.

Energy Efficiency: To assess the energy efficiency of our scheme and compare it with existing literature, we define the following energy efficiency function:

$$EE \triangleq \frac{\text{Global Accuracy}}{\text{Total Energy}}. \tag{20}$$

From Fig. 6, we observe a reduction in the total energy consumption with the proposed client selection method compared to other methods. We can relate the energy consumption directly to the model's accuracy. Our solution yields an accuracy of 1.08% per joule consumed (See Table III). Additionally, our approach enhances the energy efficiency indicator where one joule will allow for more precision as seen in Fig. 7.

Our multi-attribute client selection aims to identify clients that contribute to the training efficiently by finding the balance between achieving a satisfactory level of accuracy (98% after 10 global iterations), managing the computational costs, and reducing the time required for model updates. Additionally, this client selection scheme is designed to be fair where all participants have an equal opportunity to contribute to the model training and we make sure that important clients participate in more communication rounds to maintain fairness.

C. Discussion & Insights

By leveraging the GWO, we aim to optimize the process of choosing clients based on multiple attributes crucial to the success of OTA-FL. One key aspect is the ability of our solution to enhance the selection of clients based on their proficiency in providing informative updates. In OTA-FL, the quality of model updates plays a pivotal role in the overall learning process. Clients capable of contributing insightful and relevant updates contribute significantly to the effectiveness and accuracy of the federated learning model. The GWO helps us identify and prioritize clients with a higher potential for delivering informative contributions, thereby enriching the learning experience. Moreover, the GWO assists in striking a balance between the informative updates and the associated communication costs. In FL, communication overhead is a critical consideration, especially in wireless environments where bandwidth may be limited. Our approach aids in optimizing the trade-off between selecting clients with valuable insights and minimizing the overall communication costs. This ensures that the learning process remains efficient and scalable, even in resource-constrained OTA-FL scenarios. Additionally, fairness is a key principle in client selection for OTA-FL. Our approach incorporates fairness considerations into the client selection process. This ensures that all participating clients have a fair opportunity to contribute to the FL model. In conclusion, the integration of the GWO in our multi-attribute client selection helps to achieve a well-balanced and optimized approach. It empowers to prioritize clients based on their ability to provide informative updates, minimize communication costs, and uphold fairness in the FL ecosystem. This holistic approach not only enhances the energy efficiency of OTA-FL but also contributes to the development of more robust and equitable ML models. Moving forward, we acknowledge the need for a comprehensive analysis of the scalability of the solution to understand how the increasing number of devices will affect the efficacy of our solution. We committed to addressing this aspect in future work.

VI. Conclusion

In this paper, we have introduced a multi-attribute client selection approach for over-the-air federated learning. We have used the grey wolf optimizer to choose the best set of clients to participate in each communication round, which helps to improve the performance of the global model and speed up the convergence time. We consider several factors when selecting clients, such as model loss, energy consumption, delay, data quality, reliability, and fairness. Our proposed approach achieves better results compared to standard client selection methods, as it improves the model accuracy, convergence speed, and energy efficiency in over-the-air federated learning.

References

[1] M. B. Driss, E. Sabir, H. Elbiaze, and W. Saad, "Federated learning for 6g: Paradigms, taxonomy, recent advances and insights," *arXiv preprint arXiv:2312.04688*, 2023.

[2] B. Xiao, X. Yu, W. Ni, X. Wang, and H. V. Poor, "Over-the-air federated learning: Status quo, open challenges, and future directions," *arXiv preprint arXiv:2307.00974*, 2023.

[3] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. Nguyen, and C. S. Hong, "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 88–93, 2020.

[4] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *ICC 2019-2019 IEEE international conference on communications (ICC)*, pp. 1–7, IEEE, 2019.

[5] Z. Cheng, X. Fan, N. Chen, M. Liwang, L. Huang, and X. Wang, "Learning-based client selection for multiple federated learning services with constrained monetary budgets," *ICT Express*, 2023.

[6] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1698–1707, IEEE, 2020.

[7] J. Zheng, K. Li, E. Tovar, and M. Guizani, "Federated learning for energy-balanced client selection in mobile edge computing," in *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1942–1947, IEEE, 2021.

[8] S. Abouzahir, E. Sabir, H. Elbiaze, and M. Sadik, "Federated power control for predictive qos in 5g and beyond: A proof of concept for urllc," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, pp. 1–7, IEEE, 2023.

[9] W. Zhang, Y. Chen, Y. Jiang, and J. Liu, "Delay-constrained client selection for heterogeneous federated learning in intelligent transportation systems," *IEEE Transactions on Network Science and Engineering*, 2023.

[10] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic client selection for federated learning with volatile clients," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20055–20070, 2022.

[11] Z. Qu, R. Duan, L. Chen, J. Xu, Z. Lu, and Y. Liu, "Context-aware online client selection for hierarchical federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4353–4367, 2022.

[12] F. Shi, W. Lin, L. Fan, X. Lai, and X. Wang, "Efficient client selection based on contextual combinatorial multi-arm bandits," *IEEE Transactions on Wireless Communications*, 2023.

[13] H. Zhu, Y. Zhou, H. Qian, Y. Shi, X. Chen, and Y. Yang, "Online client selection for asynchronous federated learning with fairness consideration," *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2493–2506, 2022.

[14] D. Kang and C. W. Ahn, "Ga approach to optimize training client set in federated learning," *IEEE Access*, 2023.

[15] M. Chahoud, H. Sami, A. Mourad, S. Otoum, H. Otrok, J. Bentahar, and M. Guizani, "On-demand-fl: A dynamic and efficient multi-criteria federated learning client deployment scheme," *IEEE Internet of Things Journal*, 2023.

[16] Z. Yang, M. Chen, W. Saad, C. S. Hong, M. Shikh-Bahaei, H. V. Poor, and S. Cui, "Delay minimization for federated learning over wireless communication networks," *arXiv preprint arXiv:2007.03462*, 2020.

[17] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 269–283, 2020.

[18] J. Park, J. Moon, T. Kim, P. Wu, T. Imbiriba, P. Closas, and S. Kim, "Federated learning for indoor localization via model reliability with dropout," *IEEE Communications Letters*, vol. 26, no. 7, pp. 1553–1557, 2022.

[19] M. Sharma and P. Kaur, "Reliable federated learning in a cloud-fog-iot environment," *The Journal of Supercomputing*, pp. 1–24, 2023.

[20] C. Smestad and J. Li, "A systematic literature review on client selection in federated learning," *arXiv preprint arXiv:2306.04862*, 2023.

[21] W. Xia, T. Q. Quek, K. Guo, W. Wen, H. H. Yang, and H. Zhu, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7108–7123, 2020.

[22] F. Li, J. Liu, and B. Ji, "Combinatorial sleeping bandits with fairness constraints," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1799–1813, 2019.

[23] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.