Reliable Federated Learning with Auction-Based Incentives at the Extreme Edge

Mhd Saria Allahham, Salimur Choudhury, and Hossam S. Hassanein
School of Computing, Queen's University, Kingston, ON, Canada
Email: {20msa7, s.choudhury}@queensu.ca, hossam@cs.queensu.ca

GLOBECOM 2024 - 2024 IEEE Global Communications Conference | 979-8-3503-5125-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/GLOBECOM52923.2024.10901582

***Abstract*—Extreme Edge Computing (XEC) is a serverless edge computing paradigm where computational tasks are offloaded to and from extreme edge devices (XEDs). XEDs, a subset of IoT devices that consists of consumer-owned devices capable of offering computational resources. Being data-rich, XEDs can facilitate the training of more accurate Machine Learning (ML) models. However, their unpredictable computational behavior, which follows the consumers' usage, and transient availability pose challenges that traditional Federated Learning (FL) approaches may struggle to address. To this end, we propose a new framework for decentralized FL in XEC systems designed to address the computational reliability of XEDs and optimize the computational resource allocation. Moreover, to encourage XEDs' participation in the FL training process, we introduce an Auction-based incentive mechanism. This mechanism models the interactions between XEDs, considering both the computational characteristics and the data quality of XEDs. Furthermore, we present two solution approaches: an optimization approach and a heuristic approach, each introducing a complexity-performance trade-off. Finally, we evaluate and demonstrate the effectiveness of our proposed framework in improving the performance and reliability of XEDs in decentralized learning environments.**

***Index Terms*—Edge computing, extreme Edge Computing, federated learning, incentive mechanism, auction**

I. Introduction

Edge Computing (EC) is a distributed computing paradigm which brings computation and data storage closer to the location where it is needed, aiming to improve response times and save bandwidth [1]. EC often employs edge servers that are closer to IoT devices than cloud servers and handle a substantial amount of processing, which reduces the need to send all data to the cloud for processing [2]. Extreme Edge Computing (XEC) is an extension and a more granular form of EC. It pushes the computing capabilities even further to the edge of the network directly onto the devices generating or capturing the data, and any devices that are within their proximity [3]. This category includes IoT devices, sensors, smartphones, wearables, and other smart devices, henceforth called as Extreme Edge Devices (XEDs). This architectural approach has evolved to address the latency and bandwidth challenges posed by cloud-centric and traditional EC architectures. The XEDs, while individually less powerful than traditional data centers and edge servers, collectively offer substantial computational power in a distributed manner [4]. In the context of XEC, computational reliability (CR) refers to the ability of XEDs to consistently execute their allocated task for a specific timeline under varying conditions, mainly encompassing their computational consistency [5]. CR depends on the computational resources of the XEDs and the computational demand of the requested tasks. The fact that XEDs are typically consumer-owned devices, where user behavior is usually unpredictable, introduces an element of unpredictability in usage patterns, possibly compromising their computational consistency [6].

The concept of reliability in edge computing, particularly in XEC systems, has been explored through various lenses in recent research, such as: connectivity [7], security [8], and most relevantly, computation [9]. For instance, in [9], reliability is associated with the with offloading computationally intensive components from mobile devices to edge servers, addressing challenges such as limited power and server capacity constraints. Whereas the work done in [10] brings reliability into the field of the Internet of Vehicles, stressing the importance of ensuring high-reliability levels for latency-sensitive applications. Here, reliability is considered in terms of both processing nodes and communication links, introducing mechanisms like task allocation and reprocessing to maintain service availability.

Federated learning (FL) is a paradigm that enables decentralized training of ML models on edge devices [11]. The goal of FL is to train a global ML utilizing the devices' data while keeping data privacy, as the data remains on the individual devices and is not shared with a centralized server. A plethora of works have proposed to push FL even further to the extreme edge as a fully decentralized scheme by eliminating the use of servers dependency [12], [13], where any XED can act as a parameter server when needed, and the other available XEDs can be considered as clients or learners. While multiple works [14], [15] have discussed the reliability in FL from different perspectives, none of them addressed the computational reliability, that is, the uncertainty in computational resource allocation due to the randomness of the users usage of their devices.

To this end, we propose a novel XEC framework for incentivized FL, which accounts for the unpredictable computational behavior of XEDs. This framework aims to facilitate the FL training process and grant participation of the XEDs until the FL objective is achieved. We summarized the contributions of this work as follows:

- We introduce an XEC model that captures interactions between XEDs, their stochastic behavior in terms of

computational resources and energy consumption, and their reputation within the environment.
- We propose an auction model that motivates XEDs to participate as workers by minimizing their energy consumption. Subsequently, we optimize worker selection based on their reputations.
- We present two distinct solution approaches: an optimization approach and a heuristic approach, introducing a tradeoff between complexity and performance.

II. System Model

In our work, we consider a dynamic environment characterized by the interactions between a single extreme edge device (XED), designated as the service requester, and other XEDs serving as *service providers*, hereafter referred to as *workers*. These workers possess datasets of identical types and nature, such as images or audio files. The service requester has a model that needs to be trained or can make use of the workers' local datasets. Hence, it enables the training of its model via Federated Learning (FL) with other workers. The workers are consumer-owned devices that can enter or exit the environment at any time. To facilitate interaction in the environment, we discretize time into global communication rounds indexed by g. Each communication round g lasts for a short time, henceforth referred to as the coherent time T, during which the environment remains stationary. At each communication round g, an XED, namely, a service requester, initiates an auction for Federated Learning (FL) with workers within its proximity, denoted as $\mathcal{W}_g$, where $|\mathcal{W}_g| = W_g$ represents the number of workers participating in the auction procedure. It is important to note that the set of workers $\mathcal{W}_g$ at time g may differ from those in the previous round, denoted as $\mathcal{W}_{g-1}$, i.e., $\mathcal{W}_g \neq \mathcal{W}_{g-1}$.

A. Federated Learning Model

The FL process is initiated by the requester sending a machine learning model to the workers along with an accuracy requirement θ. To determinate the number of local iterations τ required to achieve this accuracy, we adopt the local accuracy model from [16], which is defined as follows: $\tau = v_1 \log\left(\frac{v_2}{\theta}\right)$, where v_1 and v_2 are hyperparameters influenced by factors including but not limited to the number of data samples that the worker have, the learning rate, and the quality of the data. To assess the quality of data for each worker, we consider the uniform distribution is the ideal case for the worker's data, and we employ the Kullback-Leibler (KL) divergence metric to measure the distance between the worker's data distribution $P(x)$ and the uniform distribution $U(x)$ as follows:

$$D_{KL}(P(x) \mid\mid U(x)) = \sum_x P(x) \log\left(\frac{P(x)}{U(x)}\right) \tag{1}$$

From this, the data quality Q of each worker is quantified as:

$$Q = \exp\left(-D_{KL}(P(x) \mid\mid U(x))\right) \tag{2}$$

This metric implies that the closer the worker's data distribution is to the uniform distribution, the higher the quality of the data, with Q ranging between 0 and 1. As the set of participating workers changes with time, it is intractable to model the number of global communications rounds needed to reach a desired global model accuracy θ_g. Hence, we consider the desired global accuracy as the requester objective, and it will keep initiating auctions for the FL training until it achieves this objective.

B. Worker Reliability

In the modeling of worker reliability, we define the time taken by the worker to complete its task, i.e., the training of the machine learning model as a random variable. The distribution of it, denoted by f, reflects the probability density function (PDF) that characterizes the time dynamics of task completion. As such, the reliability of a worker is quantified as the probability that it completes a task within a given deadline T, represented by the cumulative distribution function (CDF) $F(T)$, defined as: $\mathbf{R}(T) := F(T) = P(t \leq T) = \int_0^T f(t)dt$. We employ the Generalized Pareto Distribution (GPD) from [5], which allows for a broader modeling scope by incorporating features of both the exponential and Pareto distributions. The GPD is defined by the following:

$$F(T; \alpha, \xi) = \begin{cases} 1 - (1 + \frac{T\alpha}{\xi})^{-\xi}, & \text{if } \xi^{-1} > 0, \\ 1 - e^{-\alpha T}, & \text{if } \xi^{-1} = 0, \end{cases} \tag{3}$$

where α and ξ are non-negative parameters that shape the distribution. The parameter ξ primarily represents the asymptotic tail behaviour of the distribution, indicating the long-term reliability potential of a worker, while α determines the rate at which reliability approach that asymptotic tail with time. In fact, the parameter α can be interpreted as the mean task execution time (tasks/sec), that is, how fast can a worker can finish a computational task. The mean execution rate depends mainly on two factors: 1) The computational resources of the worker, 2) The computational demand of the task. Herein, the mean task execution rate is defined as: $\alpha = \mathbb{E}\left[\frac{C}{D}\right]$, where C is the allocated computational resources, i.e., computational capacity (cycles/sec) of the worker, and D is the task demand (cycles/task). We consider the computational capacity that each worker can decide to allocate to be random. Given the inherent uncertainty of the user behavior, a worker cannot commit to a fixed amount of computational resources. Instead, they offer a range or window within which the actual allocated resources can vary. This range is bounded by C_{LB} and C_{UB}, which represent the lower and upper bounds of the computational capacity that a worker can pledge, respectively.

We model the computational capacity C that a worker can allocate as a uniformly distributed random variable $C \sim \text{Uniform}(C_{\text{LB}}, C_{\text{UB}})$, where C_{LB} is strictly positive ($C_{\text{LB}} > 0$) and C_{UB} is constrained by the maximum possible allocation C_{max}, ensuring $C_{\text{UB}} \leq C_{\text{max}}$, and the mean is given by $\mathbb{E}[C] = (C_{\text{LB}} + C_{\text{UB}})/2$. Such distribution assumes that any value within the specified range is equally likely, encapsulating the uncertainty in the exact amount of resources a worker will eventually provide at any given moment.

As for the task computational demand D, it is constant since the the computation needed for training a ML model is known beforehand. The computational demand can be given by $D = \tau\Gamma_c\Gamma_d$, where τ is the number of iterations that was defined in Eq. (1), Γ_c is model computational complexity, i.e., the number computation cycles needed to perform the computation on one data sample (cycles/sample), and Γ_d is the number of data samples available. Consequently, we can re-write the mean task execution rate as the following: $\alpha = \frac{C_{\text{LB}}+C_{\text{UB}}}{2\tau\Gamma_c\Gamma_d}$, and hence, we can define the worker reliability for $\xi^{-1} > 0$, and a coherent time T (i.e., the time deadline before the system changes) as follows:

$$\mathbf{R}(T) = 1 - \left(1 + \frac{T\left(C_{\text{LB}} + C_{\text{UB}}\right)}{2\xi\tau\Gamma_c\Gamma_d}\right)^{-\xi} \quad (4)$$

In our work, it is important to note that we focus solely on computational time and reliability, assuming that the communication channel between XEDs remains constant throughout the coherent time period. We do not address variations in communication time or reliability. Specifically, we assume XEDs operate with constant transmission power and stable channel characteristics, resulting in constant communication times

C. System of Workers Reliability

System reliability is the likelihood that a system will function without failure over a specified period of time. Federated Learning (FL) is analogous to a parallel system where the overall system reliability depends on the completion of any single sub-task among all distributed tasks. Specifically, for a single FL communication round to be successful, only one worker needs to return the trained model; thus, the system of workers meets the minimum functional requirements. From [5], the system reliability can be expressed as:

$$\mathbf{R}_{\text{s}}(T) = 1 - \prod_{w=1}^{W_g}\left(1 - \mathbf{R}_w(T)\right) \quad (5)$$

where $\mathbf{R}_w(T)$ is the the worker w reliability from Eq. (6). This equation shows that the system reliability, $\mathbf{R}_{\text{s}}(T)$, is the complement of the probability that all workers fail to complete their tasks within time T. It emphasizes the advantages of the parallel structure in federated learning (FL), where having more workers significantly improves system reliability, though the completion of the task by just one worker is enough to produce a trained global model

D. Worker Reputation

In the considered environment, it is important to characterize each worker by a metric that distinguishes them from others. For simplicity, we assume that this metric, termed as the worker's reputation, remains constant over the coherent time period T. We propose the following formula for the worker's reputation:

$$\phi(T) = \mathbf{R}(T) \cdot Q \cdot \log(\Gamma_d) \quad (6)$$

where $\mathbf{R}(T)$ represents the reliability, Q denotes the data quality, and Γ_d is the number of data samples. The reliability is a critical factor contributing to the worker's reputation as it reflects the likelihood of finishing the model training within the given deadline, ensuring dependable participation in the federated learning process. On the other hand, data quality Q is essential for enhancing the overall accuracy of the federated model; higher quality data leads to more accurate and generalizable model training outcomes. Furthermore, the number of data samples Γ_d directly influences the model's learning capacity, with more samples typically providing a richer basis for learning. To ensure fairness between workers, the logarithm of Γ_d is used in the reputation formula to moderate the impact of very large data sets.

E. Worker Energy Consumption

Given the computational capacity C, the energy consumption is defined as E and given by the following model:

$$E = \frac{\kappa}{2}\tau\Gamma_c\Gamma_d C^2 \quad (7)$$

where κ is a coefficient that depends on the chip architecture [16].

When C is modeled as a random variable, E subsequently becomes a function of this random variable, implying that it possesses a distribution and specific statistical moments. This dependency on C introduces variability in E, making its analysis essential for understanding the overall energy consumption dynamics for each worker.

Lemma 1 Given $k = \frac{\kappa}{2}\tau\Gamma_c\Gamma_d$, the energy consumption E follows a distribution $f_E(e)$ characterized by the following PDF:

$$f_E(e) = \frac{1}{2\sqrt{ke}\left(C_{UB} - C_{LB}\right)} \quad (8)$$

with mean $\mu_E = \frac{k(C_{UB}^3 - C_{LB}^3)}{3(C_{UB}-C_{LB})}$ and variance $\sigma_E^2 = \frac{k^2(C_{UB}^5 - C_{LB}^5)}{5(C_{UB}-C_{LB})} - \mu_E^2$, valid for $0 < kC_{LB}^2 \leq e \leq kC_{UB}^2$.

Proof. The proof is omitted and will be included in a future publication.

Lemma 1 characterizes the distribution of energy consumption E in terms of the computational capacity C, which varies randomly between a lower bound C_{LB} and an upper bound C_{UB}. The denominator $(C_{\text{UB}} - C_{\text{LB}})$ in the expression for $f_E(e)$ indicates that the spread between C_{UB} and C_{LB} moderates the PDF's scale, affecting how energy usage is distributed across the possible range of C.

III. Auction Model for Reliable Federated Learning

Recall that for each global communication round g, the service requester initiates an auction. The auction procedure at each round consists of the following sequential steps: (1) The requester initiates an auction by broadcasting, requesting bids and specifying FL task requirements. (2) The Workers then compute their bids and decide whether to participate based on their calculations. (3) The collection of bids by the requester and determination of winning workers (4) Distribution of the machine learning model for the FL training. (5) Collection of trained models and results. (6) Disbursement of payments to the workers. The service requester keeps initiating auction rounds until a condition is met, e.g., the model reached the desired accuracy. The auction procedure elements that require decision-making include the calculation of bids by the workers and the determination of winning bids by the service requester. Upon receiving the details of the FL task during an auction broadcast, each worker evaluates the potential cost of participating. This evaluation includes assessing the computational resources needed and the energy costs incurred. Based on this assessment, each worker decides whether to participate. Those who choose to participate then submit their bids, reflecting the cost they expect to be compensated for. Once all bids are submitted, the service requester aims to select the most suitable workers while adhering to budget constraints and ensuring high system reliability.

A. Worker Bid Calculation

For each worker, the cost is assumed to be the incurred energy consumption, which is a random variable with a known distribution and statistical moments. For each worker, the aim is to choose the computational capacity window, namely, C_{LB} and C_{UB}, with the aim to minimize the expected energy consumption, denoted as μ_E. However, to mitigate potential energy spikes due to a broad range of resource allocation $(C_{UB} - C_{LB})$, we also consider minimizing the variance σ_E^2. This leads to the following stochastic optimization problem for the worker's bid calculation, referred to as **P1**:

$$\begin{aligned} \mathbf{P1}: \underset{C_{LB},C_{UB}}{\text{Minimize}} \quad & \mu_E + \lambda\sigma_E^2 && \text{(9a)} \\ \text{subject to:} \quad & \mathbf{R}(T) \geq \epsilon_{\mathrm{n}}, && \text{(9b)} \\ & 0 \leq C_{LB} < C_{UB}, && \text{(9c)} \\ & 0 \leq C_{UB} \leq C_{\max}. && \text{(9d)} \end{aligned}$$

where λ is a weighting parameter for the energy consumption variance, and ϵ_{n} represents the minimum reliability requirement given the coherent time T. The parameter λ acts like a knob for more robust control over the potential risk of high energy costs. The minimum reliability ϵ_{n} is a requirements from the service requester, denoting the reliability required to achieve the model training within the given time deadline.

Lemma 2: Given the worker maximum computational capacity $C_{\max}$, a time deadline T, and worker minimum reliability ϵ_n, the problem **P1** is feasible if and only if:

$$\frac{\xi\tau\Gamma_c\Gamma_d}{T_{\text{deadline}}}\left(\exp\left(\frac{-\log(1-\epsilon_{\mathrm{n}})}{\xi}\right)-1\right) < C_{\max} \tag{10}$$

Proof. The proof is omitted and will be included in a future publication.

This lemma establishes that a worker must possess sufficient computational resources to participate in the FL training, subject to the requirements set by the service requester. Essentially, it serves as a criterion for the participation decision: if a worker can meet the requirements while adhering to the minimum reliability ϵ_n, it will participate. Conversely, if no feasible solution exists that respects ϵ_n, the worker will opt out of participation.

Lemma 3: Given a positive constant k, the problem $\mathbf{P1}$ is convex within the convex hull defined by C_{LB} and C_{UB}, that is, there exists an optimal solution that achieves the minimum energy consumption and ensures the completion of the task before the time deadline T with at least ϵ_n reliability.

Proof. The proof is omitted and will be included in a future publication.

After each workers decides its optimal bid in terms of C_{LB} and C_{UB}, with an expected energy consumption μ_E, the service requester collect these bids, and proceeds to determine the winners.

B. Winner Determination

For each worker, we assume the cost is the incurred energy consumption, such that each worker receives compensation based on its consumed energy. We define the cost ς for a worker as $\varsigma = \rho\ .\ \mu_E$ where ρ represents the monetary cost per joule (e.g., \$/joule). Workers with greater computational resources typically pose higher monetary costs.

From the requester perspective, the goal of the auction process is to select as many workers as possible, prioritizing those with the highest reliability, best data quality and sufficient amount of data, while also achieving a system reliability of ϵ_s. Given the set of participating workers is denoted as $\mathcal{W}_g^p$, where $\mathcal{W}_g^p \subseteq \mathcal{W}_g$, we formulate the winner determination problem, denoted as $\mathbf{P2}$, as follows:

$$\begin{aligned} \mathbf{P2}: \max_{x_w} \sum_{w\in\mathcal{W}_g^p} & \phi_w x_w && \text{(11a)} \\ \text{s.t.} \quad & \mathbf{R}_{\mathrm{s}}(T) \geq \epsilon_{\mathrm{s}}, && \text{(11b)} \\ & \sum_{w\in\mathcal{W}_g^p} \varsigma_w x_w \leq B, && \text{(11c)} \\ & \sum_{win\mathcal{W}_g^p} x_w \leq K_{\max}, && \text{(11d)} \\ & x_w \in \{0,1\}, \forall x_w \in \mathcal{W}_g^p && \text{(11e)} \end{aligned}$$

where x_w is a binary variable indicating whether worker w is selected, ϕ_w represents the reputation of worker w, B is the budget allocated for a single round, $\mathbf{R}_{\mathrm{s}}(T)$ is the system reliability given time T from Eq. (7), and $K_{\max}$ is an optional constraint on the maximum number of workers to prevent overwhelming the service requester.

Lemma 4: Given a system reliability ϵ_s, the minimum number of participating workers $N_{\min}$ needed for problem $P2$ to be feasible is given by:

$$N_{\min} \geq \left\lceil \frac{\log(1-\epsilon_s)}{\log(1-\mathbf{R}_{\min}(T))} \right\rceil \tag{12}$$

where $\mathbf{R}_{\min}(T)$ is the reliability of the worker with the lowest reliability among the participating worker group $\mathcal{W}_g^p$.

Proof. The proof is omitted and will be included in a future publication.

Lemma 4 provides a lower bound on the number of participating worker so the FL training process can succeed with at least one worker with probability ϵ_s. If the number of participating workers is less than $N_{\min}$, the requester will halt the auction and wait for the next global round with new set of workers.

IV. Solution Approaches

Recall that problem $\mathbf{P1}$ is convex for workers who satisfy the feasibility condition in Eq. (12). Consequently, a worker can utilize an optimizer, such as an interior-point method, to compute its optimal resources (C^*_{LB} and C^*_{UB}) to allocate to the FL task. This type of optimizer is both fast and efficient, capable of finding the optimal solution when the problem is convex. In contrast, $\mathbf{P2}$ is a Binary Integer Linear Program (BILP), which is inherently NP-hard. We propose two approaches to solve $\mathbf{P2}$: an optimization approach and a heuristic approach.

A. Optimization Approach

Based on Lemma 4, we can reformulate $\mathbf{P2}$ by eliminating the constraint in (13b), and re-write constraint (13d) as follows:

$$N_{\min} \leq \sum_{w \in \mathcal{W}_g^p} x_w \leq K_{\max} \tag{13}$$

The original reliability constraint is replaced by a lower bound on the number of workers $N_{\min}$, since having this minimum number achieves the target system reliability requirement. Subsequently, the BILP can be passed to a solver equipped with a branch-and-bound algorithm, which is designed to find an optimal solution. However, since there is no guarantee that the optimization algorithm can run in polynomial time, this violates the auction requirement of computational efficiency. Therefore, we retain this approach as a benchmark and propose a simpler, yet effective, heuristic algorithm as a practical alternative.

B. Heuristic Approach

In the proposed heuristic approach, we begin by sorting the workers based on their reliability, denoted as $(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_K)$, in descending order such that $\mathbf{R}_i \geq \mathbf{R}_{i+1}$. The initial step involves selecting the top $N_{\min}$ workers to evaluate whether their combined costs are within the allocated budget.

- If the budget accommodates the costs of these top $N_{\min}$ workers, a solution is obtained, and the process terminates successfully.
- If the budget does not suffice, we proceed to consider the next set of $N_{\min}$ most reliable workers, excluding those already considered in the previous set.
- This selection and evaluation process is repeated iteratively, moving down the list of sorted workers.

If after checking all possible combinations up to K, no subset of workers meets the budget constraints, then the problem itself is infeasible under the current budget and worker costs, and no solution can be obtained.

V. Simulation Results

In this section, we first demonstrate how the node reliability ϵ_n and system reliability ϵ_s metrics affect the system. Afterwards, we evaluate the FL training process in the system. We ran the simulations considering 50 workers, each with random values for $C_{\max}$, ξ, and Γ_d. For FL, we adopt a SqueezeNet deep learning model with 750k parameters, a target local accuracy θ of 0.01, and a desired global model accuracy of θ_g of 0.001.

A. Reliability Analysis

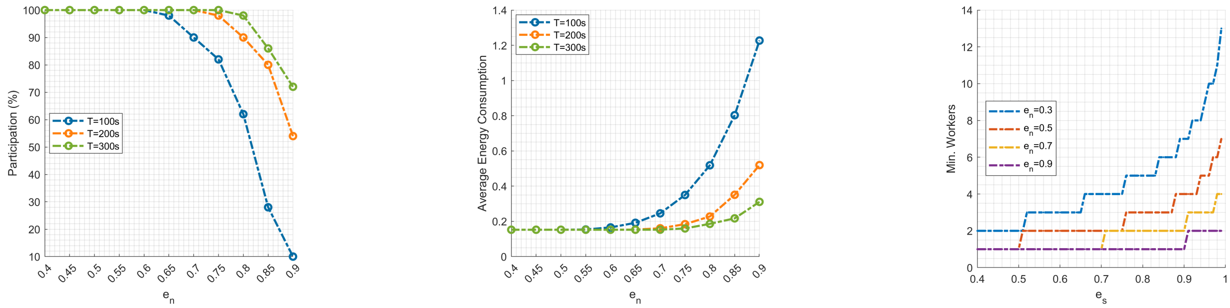
The effects of reliability on the considered system are depicted in Figure 1. In Figure 1 (a), the participation rate against the node reliability requirement ϵ_n is shown, with different time deadlines T. We can notice that, as we increase ϵ_n, the participation rate drops, since less amount of workers can meet this requirement. Moreover, as we decrease the time deadline, less workers will be able to finish their task on time with a given ϵ_n. In Figure 1 (b), we can notice that the average workers energy consumption increases as we increase ϵ_n, and decrease the time deadline. In fact, with higher values for ϵ_n and tight deadline, the requester needs to recruit workers with more computational capacity, and hence, more energy consumption. In Figure 1 (b), we show the minimum number of workers needed to satisfy a system reliability of ϵ_s with varying ϵ_n. It can be seen that as we increase ϵ_s, a higher number of workers is needed to guarantee this system reliability. However, as we increase ϵ_n, we needed less workers overall, since only highly reliable workers will be available, and only recruiting them will suffice the system reliability requirement.

B. Performance Comparison

The comparison between the benchmark, optimization approach and the heuristic approach is depicted in Figure 2. We assume our benchmark is the ideal scenario when all the nodes participate in the FL process, and the goal to be as close as possible to that performance. The FL training loss is shown in Figure 2 (a). We can notice that the optimization approach loss is close to the benchmark, maintaining a good performance by recruiting reliable workers with better data, while the heuristic approach suffers from unstable training performance. In fact, since the heuristic approach recruit exactly the minimum number of workers needed, it suffers from lack of data. Not only that, but as it can be seen from 2 (b), it does not favour workers with good data quality, but it favours highly reliable workers. The latter fact can be noticed in Figure 2 (c), where the percentage of workers that did not drop during the FL training, refered as the retention rate, is shown. The heuristic maintain a higher retention rate during the training, while the optimization retains a less retention rate, since it does not favour the reliability, but it balances it with the data quality.

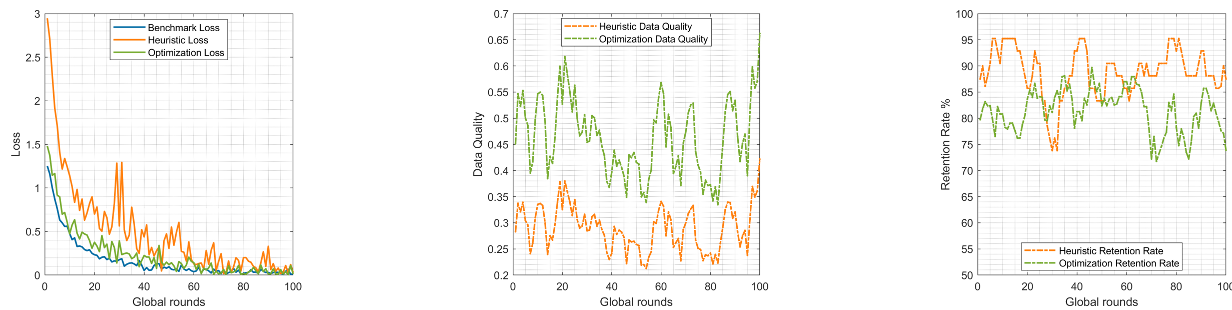
VI. Conclusion

This work introduced a novel framework for incentivized Federated Learning (FL) within Extreme Edge Computing



(a) (b) (c)

Figure 1: Reliability analysis in terms of (a) participation, (b) energy consumption, and (c) minimum number of workers needed



(a) (b) (c)

Figure 2: Performance comparison in terms of (a) FL training loss, (b) workers data quality, and (c) workers retention rate

(XEC). The proposed framework captures the characteristics of Extreme Edge Devices (XEDs), including their stochastic computational behavior and energy usage. Additionally, we have proposed an auction-based mechanism to encourage XEDs to engage as workers, focusing on minimizing energy consumption while optimizing the selection based on device reliability and reputation. Moreover, we have outlined two solution strategies, an optimization approach and a heuristic approach, each providing a different trade-off between complexity and performance. Finally, we evaluated the proposed solution approaches in terms of the FL training task.

References

[1] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE access*, vol. 8, pp. 85 714–85 728, 2020.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[3] J. Portilla, G. Mujica, J.-S. Lee, and T. Riesgo, "The extreme edge at the bottom of the internet of things: A review," *IEEE Sensors Journal*, vol. 19, no. 9, pp. 3179–3190, 2019.

[4] E. Covi, E. Donati, X. Liang, D. Kappel, H. Heidari, M. Payvand, and W. Wang, "Adaptive extreme edge computing for wearable devices," *Frontiers in Neuroscience*, vol. 15, p. 611300, 2021.

[5] M. S. Allahham, A. Mohamed, A. Erbad, and H. Hassanein, "On the modeling of reliability in extreme edge computing systems," in *2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, 2022, pp. 1–6.

[6] S.-W. Ko, S. J. Kim, H. Jung, and S. W. Choi, "Computation offloading and service caching for mobile edge computing under personalized service preference," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6568–6583, 2022.

[7] A. E. Zonouz, L. Xing, V. M. Vokkarane, and Y. L. Sun, "Reliability-oriented single-path routing protocols in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 4059–4068, 2014.

[8] B. Wang, M. Li, X. Jin, and C. Guo, "A reliable iot edge computing trust management mechanism for smart cities," *IEEE Access*, vol. 8, pp. 46 373–46 399, 2020.

[9] L. Dong, W. Wu, Q. Guo, M. N. Satpute, T. Znati, and D. Z. Du, "Reliability-aware offloading and allocation in multilevel edge computing system," *IEEE Transactions on Reliability*, vol. 70, no. 1, pp. 200–211, 2021.

[10] X. Hou, Z. Ren, J. Wang, W. Cheng, Y. Ren, K.-C. Chen, and H. Zhang, "Reliable computation offloading for edge-computing-enabled software-defined iov," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7097–7111, 2020.

[11] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[12] M. S. Allahham, A. Mohamed, A. Erbad, and M. Guizani, "Motivating learners in multiorchestrator mobile edge learning: A stackelberg game approach," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 46, no. 1, pp. 69–76, 2022.

[13] M. S. Allahham, S. Sorour, A. Mohamed, A. Erbad, and M. Guizani, "Energy-efficient device assignment and task allocation in multi-orchestrator mobile edge learning," in *2021 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2021, pp. 1–6.

[14] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[15] S. Math, P. Tam, and S. Kim, "Reliable federated learning systems based on intelligent resource sharing scheme for big data internet of things," *Ieee Access*, vol. 9, pp. 108 091–108 100, 2021.

[16] N. H. Tran, W. Bao, A. Zomaya, M. N. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *IEEE INFOCOM 2019-IEEE conference on computer communications*. IEEE, 2019, pp. 1387–1395.