An Client Selection Mechanism for FL with Electre

Rafael Veiga[1], **Marcos Seruffo**[1], **Eduardo Cerqueira**[1], **Denis Rosário**[1]

[1]Federal University of Pará (UFPA) - Belém, PA, Brazil

rafael.teixeira.silva@icen.ufpa.br, renan.morais@itec.ufpa.br,

{cerqueira, denis}@ufpa.br

Abstract. *There is a exponential growth of data usage, specially due to the proliferation of connected applications with personalized models for different applications. In this context, Federated Learning (FL) emerges as a promising solution to enable collaborative model training while preserving the privacy and autonomy of participating clients. In a typical FL scenario, clients exhibit significant heterogeneity in terms of data distribution and hardware configurations. In this way, randomly sampling clients in each training round may not fully exploit the local updates from heterogeneous clients, resulting in lower model accuracy, slower convergence rate, degraded fairness, etc. In addition, malicious users could disseminate incorrect weights, which may decrease the accuracy of aggregated models and increase the time for convergence in FL.*

1. Introduction

The integration of big data and deep learning into various applications has significantly enhanced intelligence and efficiency, establishing Machine Learning (ML) as a crucial tool for key stakeholders [Kusano et al. 2023]. However, ML applications necessitate extensive data sharing, which raises substantial communication and privacy concerns [Smestad and Li 2023]. Centralized ML models, which rely on cloud-centric architectures for data storage and processing, often involve user-generated data that contains sensitive information. This centralized approach can result in communication bottlenecks, leading to high latency and increased communication costs.

Anticipating a shift from centralized cloud computing to a distributed edge computing paradigm, FL (FL) emerges as a compelling solution for future ML applications due to its communication-preserving and privacy-enhancing characteristics [Zhang et al. 2023b]. In FL, each client independently constructs its own model without accessing data from other devices [Lobato et al. 2022]. Local models are then aggregated at the cloud or edge servers by a given aggregation policy to produce an accurate global model. This collaborative approach allows for the construction of a unified model while preserving data privacy, as the data remains distributed across various stakeholders. Moreover, FL facilitates continuous learning, adapting ML models without necessitating raw data sharing.

Client selection is a critical component of the FL training process, strategically choosing a subset of devices, or clients, to contribute to model training in each learning round. This selection must ensure valuable sample inclusion, excluding clients who do not add value, to improve the global model while managing communication costs [Smestad and Li 2023]. The selection process must maintain a diverse and representative data sample from various clients, contributing to the model's robustness without centralizing data, thus preserving privacy. Selecting clients based on data quality, availability, and computational capacity optimizes training, enhancing model performance in FL systems.

FL faces significant challenges due to the heterogeneous nature of participating clients, who often possess diverse data distributions and varying hardware characteristics. This heterogeneity results in non-IID data scenarios, where data samples from different clients are not statistically independent and may exhibit distinct statistical distributions [Xiong et al. 2023]. Such diversity can lead to reduced model accuracy, slower convergence rates, and fairness concerns if not adequately addressed. Additionally, the reliance on client-shared trained weights heightens vulnerability to malicious clients injecting erroneous updates, compromising the integrity of the model training process [Le et al. 2023].

Client selection is a significant challenge in FL, where client data and hardware heterogeneity can lead to inefficient training and suboptimal model performance. Randomly selecting clients often fails to leverage the most informative data, resulting in slower convergence rates and reduced accuracy. Researchers have proposed advanced selection methods using entropy or probability based on client attributes to address these issues. These methods prioritize clients with diverse and high-quality data, enhancing training. By strategically selecting clients with high entropy or favorable probabilistic attributes, FL systems can improve model accuracy and achieve faster convergence, thereby ensuring more robust and effective learning outcomes.

This paper introduces RiCAm (Resilience-aware Client Selection Mechanism multi-criteria) to address the challenges of non-IID data FL environments. RiCAm proposes a novel client selection mechanism that incorporates model performance on the client's data, data size, and the entropy of the client's local updates, as well as train metrics such as accuracy and loss of each client. By using AHP to support our algorithm in defining the calculated weights of the algorithm, RiCAm ensures diverse and informative data selection. Additionally, RiCAm employs the ELECTRE method to search for clients with the best-fit attributes to train in each round using AHP weight combined with a ranked multicriteria method. Our experimental evaluation demonstrates that RiCA performs better than the baseline model in terms of accuracy and loss reduction throughout the training rounds.

The rest of this paper is organized as follows. Section 2 presents an overview of works exploring resilience and protection approaches in FL. Section 3 describes our methodology for client selection and methods for improving the weights on the multicriteria. Section 4 explores the simulation model and results related to our method. Finally, Section 5 concludes the paper and outlines directions for future work.

2. Related Works

This section presents key state-of-the-art approaches that enhance the resilience of FL against the uses of client selection and multicriteria methods in FL. In centralized FL scenarios, client selection plays a vital role in the locale of the main clients to improve global accuracy. Thus, multicriteria are used based on information theory and emergent approaches to client selection. Orlandi *et al.* [Orlandi et al. 2023] used an approach to solve this data problem by using an entropy-based approach to the client's distribution data. However, they did not consider other approaches, such as the client's quantity of data and the data's relevance in the training.

FL has emerged as a decentralized approach to address privacy concerns and the challenges of centralized machine learning systems. However, the existing FL protocols, particularly in IoT environments, face limitations due to the heterogeneous nature of client devices and their constrained resources. While previous works like FedMCCS [AbdulRahman et al. 2020] have explored client selection based on CPU, memory, energy, and time, our research takes a different direction. We propose a novel client selection mechanism that considers alternative metrics such as entropy of data, amount of data, and customer convergence accuracy. By focusing on these distinct criteria, we aim to optimize client selection in FL for improved model performance and resource utilization.

3. A Robust Client Selection Mechanism by multi-criteria demand for FL

This section introduces RiCAm (Resilience-aware Client Selection Mechanism), a novel approach designed to enhance the robustness of client selection in FL scenarios. RiCAm leverages a two-stage process: (i) Information-Theoretic Client Selection: The first stage employs information theory principles to select clients. This selection considers both the entropy of client updates, which serves as a measure of data diversity and the data size of each client. Prioritizing clients with smaller data sizes aims to mitigate the negative influence of potentially biased or overly influential large datasets. (ii) ÉLECTRE-Based Client Evaluation: This step strengthens system resilience by leveraging the histoRiCAml behavior of trusted clients to identify potential anomalies in new client updates. ÉLECTRE (Elimination Et Choice Translating Reality) is a multi-criteria decision analysis method that evaluates clients based on various criteria, helping to identify and prioritize the most reliable and consistent clients for participation in the training process.

3.1. Scenario overview

We consider a scenario composed of N devices, denoted as $u_i \in \{u_1, \ldots, u_N\}$. In a typical FL framework, every communication round begins with choosing a group of K client devices to receive the global model and perform training based on their respective datasets D_i. The client selection mechanism aims to choose a set of K clients with valuable samples, thereby reducing the waste of computation resources by excluding data no longer critical for model training. Each dataset D_i consists of a collection of features $x_{k,i}$ (for $k \in \{1, \ldots, \|D_i\|\}$), paired with corresponding labels $y_{k,i}$. The selected clients improve the model *Mn* by training with their datasets D_i. After the training phase, client updates, encompassing either learned model parameters or calculated gradients, are transmitted back to the central aggregation server [Song et al. 2022]. This process enables each device to contribute unique data to the FL process, aiding in developing a comprehensive and robust global model [Barros et al. 2021]. The central server employs a specific aggregation strategy to integrate these updates into a cohesive model. A common strategy is the Federated Averaging (FedAVG) [Liu et al. 2020] algorithm, which calculates the mean of all local models shared by client devices, as computed based on Eq. 1. This aggregation process at the edge servers consolidates client updates, generating a refined global model subsequently distributed back to the selected clients for the next training round [Lobato et al. 2022].

Figure 1 illustrates the proposed FL scenario, emphasizing client selection within a context of poisoning attacks and non-IID data. This setup divides the dataset into "n" users for each training participant. The RiCAm mechanism uses dataset size and Entropy Ei as input to select the set of K clients with relevant data to improve the global model. In addition, RiCAm employs the multicriteria method ELECTRE, utilizing Entropy and data size as parameters to maintain the system's robustness and adaptability to attacks on the Global Model Gm. Specifically, ELECTRE evaluates these criteria to ensure the inclusion of only clients with valuable and diverse data in the aggregation process. As depicted in Figure 1, clients transmit their parameters to the central server after the selection process, encompassing both their datasets and the models refined during each iteration. With the support of the FedAVG algorithm, the central server aggregates the received model updates from participating clients. This aggregation process combines the collective learning progress and generates an updated global model. The central server subsequently distributes this refined model to all participating clients for the next training round.

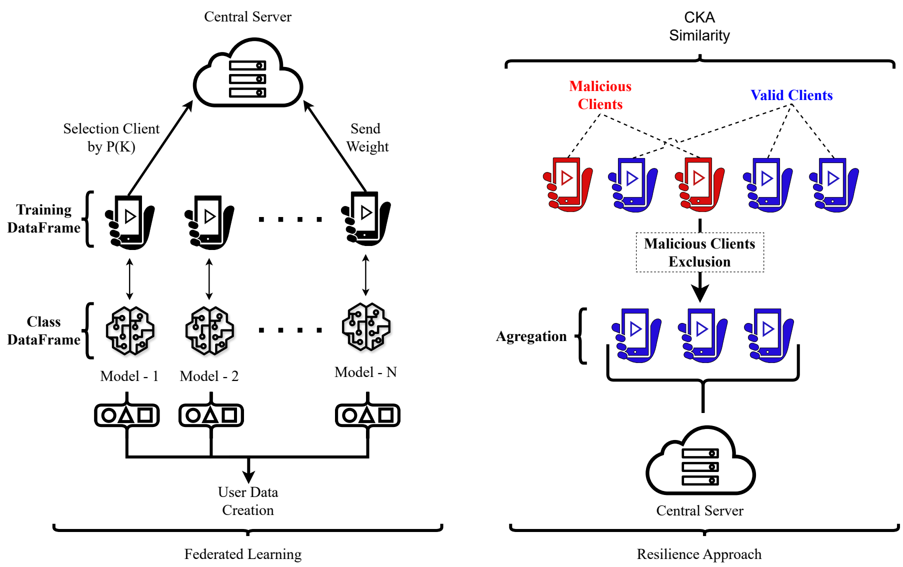

Figure 1. Placeholder do periodic

In FL, we describe data distribution across participating clients using probability distributions. The data distribution of each client, represented by a local dataset D_i, contributes to the overall data distribution across all clients, expressed as the union of these local datasets:

$$D = \bigcup_{i=1}^{K} D_i \tag{1}$$

Let p_i denote the probability distribution associated with the data on client C_i, defined as:

$$p_i(x) = \frac{\text{Number of occurrences of } x \text{ in } D_i}{\text{Total size of } D_i} \tag{2}$$

This probability distribution $p_i(x)$ indicates the likelihood of encountering a specific data point x within the local dataset D_i. The system aggregates information from all clients to construct a global model in the FL process. We iteratively update the global model parameters θ by incorporating the contributions from individual clients, mathematically represented as:

$$\theta_{global} = \frac{1}{K} \sum_{i=1}^{K} \theta_i \tag{3}$$

Here, θ_{global} represents the global model parameters, and θ_i denotes the model parameters updated by client C_i. The probability distributions $p_i(x)$ for each client C_i provide a foundation for evaluating the data diversity within the federated network. In this context, entropy is crucial in assessing and leveraging this diversity.

3.2. Entropy calculation

The performance of the FL training process closely depends on the quality and diversity of client data. Clients show significant data distribution and hardware configuration heterogeneity in a typical FL scenario. Randomly sampling clients in each training round can fail to fully utilize the local updates from these diverse clients, leading to lower model accuracy, slower convergence rates, and reduced fairness. Therefore, employing a client selection scheme based on specific metrics becomes crucial for optimizing the effectiveness of the FL framework [Fu et al. 2023]. The overall success of FL hinges on the quality and diversity of data across participating clients.

The aggregation server employs the ELECTRE method during client selection to identify the most suitable clients. This method considers multiple criteria, including the entropy of data, amount of data, and customer convergence accuracy, to assess each client's potential contribution to the FL process. By analyzing these metrics, the server can decide which clients to select for training, ensuring a balance between data diversity, quantity, and model performance.

In this context, RiCAm correlates directly with the characteristics of diverse and high-quality client data. Specifically, diversity ensures that the model can generalize well across different data distributions, while high-quality data contributes to the accuracy and reliability of the learned model. Striking a balance between these two aspects is essential for robust and adaptable model training. Traditional client selection methods may fall short in addressing the nuanced interplay between data diversity and quality. Biased or suboptimal model updates may result from overlooking client data's distributional characteristics, hindering the FL system's overall performance. RiCAm leverages entropy as a key metric during client selection. Entropy serves as a measure of data diversity, allowing RiCAm to prioritize a set of top clients. This approach fosters a more comprehensive and balanced representation of the underlying data landscape within the training process.

RiCAm determines entropy by assessing the probability distribution of symbols or messages that a source can produce [Fu et al. 2023]. By calculating the entropy of individual clients' data, RiCAm gains insights into the degree of randomness or uncertainty within the data. This entropy-based client selection enables FL algorithms to identify the most relevant and diverse data, thereby facilitating learning models that can effectively handle heterogeneity. By selecting clients with high entropy, RiCAm ensures that the learned models accurately represent the entire network, capturing variations in driving behavior, traffic patterns, and network connectivity.

The Shannon entropy formula is applied to measure data diversity formally. For example, consider two datasets, A and B. Dataset A, with an even distribution of labels $0, 1$ each appearing with a probability of 50%, has an entropy of $-0.5 \log_2(0.5) - 0.5 \log_2(0.5) = 1$ bit. Conversely, Dataset B, with a skewed distribution where label 0 occurs 90% and label 1 10% of the time, has an entropy of $-0.9 \log_2(0.9) - 0.1 \log_2(0.1) \approx 0.47$ bit. This example illustrates that Dataset A, with higher entropy, exhibits a more balanced and diverse distribution of classes compared to Dataset B. High entropy in data labels signifies diverse data, mitigating overfitting and bias by ensuring a balanced representation of classes, thus enhancing model performance across various scenarios.

RiCAm uses the Shannon Entropy formula, as described in Eq. 4, where $H(X)$ represents the dataset entropy and $P(x)$ the probability of observing a particular value x in the dataset. We select clients with high-entropy datasets for their diverse and informative data, which can enhance FL model performance. Applying Eq. 4 in our scenario yields the entropy calculation for client selection ranking in FL, described in Eq. 5. Here, k_m refers to the class of the data point d_{ni}, representing an individual data point in d_n. Prioritizing clients with higher entropy values introduces greater data diversity into the training process, fostering the development of more generalizable and adaptable models. Consequently, the resulting models exhibit improved accuracy and applicability across diverse FL scenarios, making them well-suited for non-IID environments.

$$H(X) = -\sum_{x} P(x) \log P(x) \tag{4}$$

$$H(d_n) = -\sum_{j=1}^{m} P(k_m) \log P(k_m) \tag{5}$$

The RiCAm approach employs a filtering strategy that prioritizes clients based on data size and subsequently utilizes entropy to determine their weight. This methodology ensures equitable selection and distribution of clients, with the selection process being initiated after the client is established, considering information relayed to the server, such as clients' data values and label diversity, among other factors. The use of weight enhances the efficacy of training the Convolutional Neural Network (CNN), guaranteeing that each iteration or round of the global model training integrates the most suitable set of clients.

3.3. Intelligent Client Selection Based on Data Convergence with Electre

The Analytic Hierarchy Process (AHP) determines the weights assigned to different criteria considered in the client selection process. These criteria may include data size, entropy, model performance, and other relevant metrics. Using AHP, we can establish a hierarchical structure of criteria and pairwise comparisons to derive the relative importance of each criterion. That ensures a more informed and balanced approach to client selection, as the weights reflect the specific requirements and priorities of the FL task. Calculate the weight of each criterion by finding the eigenvector corresponding to the maximum eigenvalue λ_{max} of the pairwise comparison matrix and normalizing it such that the sum of the weights is 1, as shown in 6.

$$w_i = \frac{\lambda_{max} - n}{n - 1} \tag{6}$$

In our approach, ELECTRE ranks the available clients based on their weighted scores obtained from the AHP analysis. The methodology then evaluates the clients using ELECTRE, involving pairwise comparisons and outranking relations to determine the most suitable clients for training in the next round. By considering multiple criteria and their weights, ELECTRE enables a comprehensive assessment of each client's potential contribution to the FL process. That ensures that the selected clients are not only diverse in terms of data distribution but also possess the necessary qualities to enhance the overall model performance. The concordance index, which measures the degree to which one alternative outranks another, is calculated as the weighted sum of the criteria for which the performance of the first alternative is at least as good as the performance of the second alternative, as shown in 7.

$$\sum_{j=1}^{m} w_j g_j(a_i) \geq \sum_{j=1}^{m} w_j g_j(a_k) \tag{7}$$

Integrating AHP and ELECTRE in our intelligent client selection mechanism offers several advantages. AHP structures and systematizes weight determination, ensuring alignment of the selection process with the specific goals and constraints of the FL task. ELECTRE, on the other hand, enables a comprehensive evaluation of clients based on multiple criteria, leading to a more informed and practical selection of clients for training. This approach enhances the efficiency of the client selection process and contributes to the overall performance and robustness of the FL model. The discordance index, which measures the degree to which one alternative is worse than another on the most important criteria, is calculated as shown in 8.

$$C(a_i, a_k) = \frac{1}{1 + d(a_i, a_k)} \tag{8}$$

In our other approach, MACBETH evaluates the available clients based on their weighted scores obtained from the AHP analysis. The methodology then uses MACBETH to assess the clients, involving pairwise comparisons to establish a scale of preferences and determine the most suitable clients for training in the next round. By considering multiple criteria and their weights, MACBETH facilitates a nuanced assessment of each client's potential contribution to the FL process. That ensures that the selected clients are not only diverse in terms of data distribution but also possess the necessary qualities to enhance the overall model performance. The pairwise comparison matrix, which measures the relative importance of criteria, is calculated using the MACBETH scale of preferences, as shown in 9.

$$M_{ij} = \begin{cases} 0 & \text{if } a_i \text{ is equally preferred to } a_j \\ x & \text{if } a_i \text{ is preferred to } a_j \text{ by } x \text{ units} \end{cases} \tag{9}$$

Integrating AHP and MACBETH in our intelligent client selection mechanism offers several advantages. AHP structures and systematizes weight determination, ensuring alignment of the selection process with the specific goals and constraints of the FL task. Conversely, MACBETH enables a detailed evaluation of clients based on multiple criteria, leading to a more informed and practical selection of clients for training. This approach enhances the efficiency of the client selection process and contributes to the overall performance and robustness of the FL model. Check the consistency of the pairwise comparison matrix to ensure the validity of the derived scale, as shown in 10.

$$\text{Consistency Index} = \frac{\lambda_{\max} - n}{n - 1} \tag{10}$$

Algorithm 1 illustrates the client selection phase using the AHP, ELECTRE, and MACBETH methodologies to address the client selection challenges in federated learning. Starting with the calculation of the total data size, denoted as D_{total}, the algorithm first evaluates the volumetric contribution of each client. This step is crucial for identifying clients with sufficient data samples to facilitate practical training and enhance overall performance.

Next, the algorithm calculates the entropy E_i for each client using Shannon entropy, where $p(x)$ represents the probability distribution of classes in the client's data. This metric helps quantify the diversity or uncertainty within the client's dataset, which is vital for the client selection. Additionally, the local model performance loss L_i is computed for each client, adding another layer of assessment based on data quality.

The algorithm then defines weights for each criterion (data size, entropy, and loss) using the AHP method, ensuring a structured and systematic approach to weight determination. Each client's data size D_i, entropy $E_{i_{\text{norm}}}$, and loss $L_{i_{\text{norm}}}$ are normalized to facilitate a fair comparison. These normalized values are then combined using the weighted sum formula $S_i = w_D \cdot D_i + w_E \cdot E_{i_{\text{norm}}} + w_L \cdot L_{i_{\text{norm}}}$, where w_D, w_E, and w_L are the weights derived from the AHP analysis.

The clients are ranked based on their combined scores S_i, and the top 20% of clients are selected using ELECTRE or MACBETH. These multi-criteria decision-making methods further refine the client selection by considering various data quality and relevance aspects. Finally, the selected subset of clients is used for federated learning training, ensuring a diverse and representative dataset that enhances the overall model performance.

The algorithm achieves a robust and efficient client selection mechanism by integrating data size, entropy, and loss into the client selection process and employing sophisticated decision-making techniques like ELECTRE and MACBETH. This approach not only improves the quality of the training data but also ensures the resilience and robustness of the federated learning model.

4. Evaluation

This section presents the simulation setup employed to evaluate the performance and efficiency of RiCA. We first describe the simulated FL scenario, including the underlying framework, database characteristics, and simulation parameters. We also present the ob-

Algorithm 1: Client Selection Phase

$D_{\text{total}} \leftarrow \sum_i C_i.\text{dataSize}$;
for *each client* i **do**
  $p(x) \leftarrow$ Probability distribution of classes in C_i's data;
  $E_i \leftarrow -\sum p(x) \log p(x)$;
  $L_i \leftarrow$ Local model performance loss for C_i;
end
$\{w_D, w_E, w_L\} \leftarrow$ AHP();
for *each client* i **do**
  $D_i \leftarrow \frac{C_i.\text{dataSize}}{D_{\text{total}}}$;
  $E_{i_{\text{norm}}} \leftarrow$ NormalizeEntropy(E_i);
  $L_{i_{\text{norm}}} \leftarrow$ NormalizeLoss(L_i);
end
for *each client* i **do**
  $S_i \leftarrow w_D \cdot D_i + w_E \cdot E_{i_{\text{norm}}} + w_L \cdot L_{i_{\text{norm}}}$;
end
Rank clients based on their combined scores;
$\{C_{\text{ranked}}\} \leftarrow$ ELECTRE(S_i) or MACBETH(S_i);
$\{C_{\text{selected}}\} \leftarrow$ top 20% of $\{C_{\text{ranked}}\}$;
Proceed with FL training using $\{C_{\text{selected}}\}$;

tained results, focusing on the metrics of accuracy and loss for the global model. This analysis aims to assess the effectiveness and resource utilization of RiCA compared to baseline approaches.

4.1. Simulation Description

We conducted an extensive simulation study utilizing the PFLib, a versatile framework introduced [Zhang et al. 2023a][1]. The framework was executed on a server with the following specifications: i9-13900K(32), 128 GB RAM, and Dual RTX 4090 GPUs, running on a Ubuntu Server operating system. We also used three different sets of pictures: MNIST (handwritten numbers), FMNIST (pictures of clothes), and CIFAR-10 (pictures of animals and things). By testing RiCAm on these different picture sets, we wanted to see how well it does with different kinds of data.

The CNN model architecture employed in the experiment consisted of two convolutional layers with 5x5 filter sizes, followed by 2x2 max-pooling operations after each convolutional layer. To ensure a realistic representation of data distribution challenges, we employed non-IID data throughout the experiments. This non-IID data was modeled using a Dirichlet distribution. We use label distribution to characterize the local data distribution among clients by a proportion, as described in [Ma et al. 2022]). The Dirichlet distribution, with a concentration parameter of $\beta = 0.2$, determines this proportion of samples, as outlined in [Li et al. 2021].

In addition, we evaluate the RICA+CKA approach across various client scenarios to demonstrate how it can enhance accuracy more rapidly while preserving the security

[1]https://github.com/TsingZ0/PFLlib

of the global model. We also assess the RiCAm scheme in different scenarios to compare its efficacy in selecting clients, despite the potential selection of clients, compared to the Accuracy and Loss. Evaluating client selection based on their entropy-weighted weights will be a benchmark comparison to the default FL approach. The default FL is the most common FL strategy base, using only random selection. Table 1 summarizes the simulation setup and parameters.

Table 1. Simulation Parameters

Description	**Values**
Number of Clients	10, 25, 50 Clients
Multicriteria method default	Electre
Multicriteria methodd default	Electre and Macbeth
Datasets used	MNIST, CIFAR-10, FMNIST
Default dataset	MNIST

We evaluate the performance of the proposed mechanisms using established metrics commonly employed in FL environments, namely, Accuracy and Loss. Accuracy is computed by dividing the number of correct predictions (optimistic) by the total number of examples. In a more formal mathematical representation, where *TP* denotes true positives, *TN* denotes true negatives, and *FP* and *FN* denote false positives and false negatives, respectively, accuracy as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

The loss metric quantifies the discrepancy between the model's predictions and the actual labels. Lower loss values indicate better model performance. In essence, the loss metric reflects the cost associated with prediction errors. For classification tasks, Cross-Entropy Loss (also known as Log Loss) measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy Loss increases as the predicted probability diverges from the actual label. The cross-entropy Loss for a sample is the negative sum of the logarithm of the predicted probability for the correct class. The overall Loss is obtained by averaging these sums across all samples. This approach penalizes confident but incorrect predictions, increasing the penalty as the predicted probability diverges from the actual class. The objective of model training is to minimize this Loss, thereby enhancing the model's accuracy in classifying samples.

4.2. Results

Figure 2 presents the accuracy performance of various client selection methods, including Random Selection, Electre Selection, Macbeth Selection, and RiCA, over 30 rounds. The Random Selection method shows a slow and inconsistent improvement, struggling to exceed an accuracy of 50% throughout the simulation. Electre Selection and Macbeth Selection demonstrate more stable and higher performance, achieving accuracies close to 80% by the end of the rounds. These results underscore the effectiveness of

Electre Selection and Macbeth Selection in enhancing client selection processes and significantly improving model accuracy compared to Random Selection. The evaluation indicates that both Electre Selection and Macbeth Selection offer considerable robustness and efficiency, making them valuable approaches for federated learning scenarios.

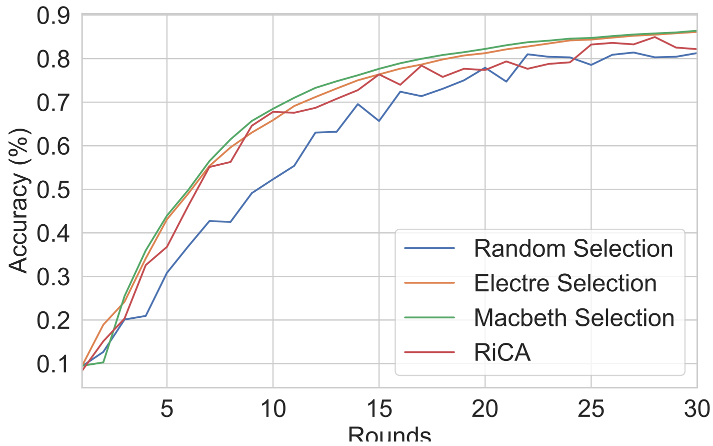

Figure 2. Accuracy measurements

Figure 3 illustrates the loss performance of various client selection methods, including Random Selection, Electre Selection, Macbeth Selection, and RiCA, over 30 rounds. The Random Selection method exhibits a slower reduction in loss, maintaining higher values throughout the simulation. Electre Selection and Macbeth Selection demonstrate more rapid and stable reductions in loss, converging to values around 0.5 by the end of the rounds. These results highlight the effectiveness of Electre Selection and Macbeth Selection in reducing model loss more efficiently than Random Selection. The evaluation indicates that both Electre Selection and Macbeth Selection offer significant robustness and efficiency, making them valuable approaches for improving the performance of federated learning models.

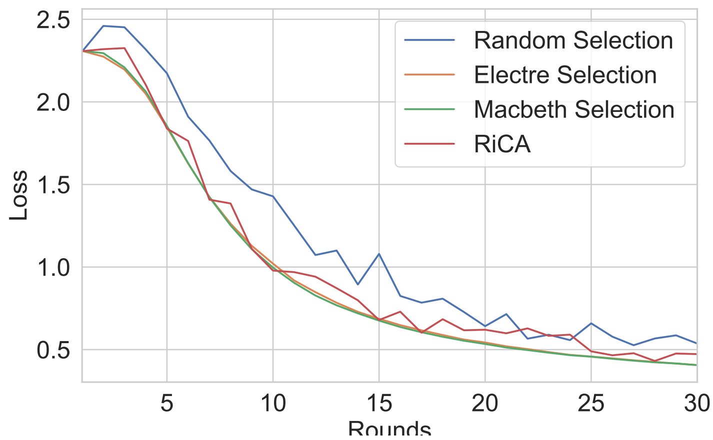

Figure 3. Loss measurements

Figure 4 illustrates the False Rejection Rate (FRR) measurements for different client selection methods, including Random Selection, Electre Selection, Macbeth Selection, and RiCA, over 30 rounds. The Random Selection method shows a higher and more fluctuating FRR, initially peaking close to 0.9 and gradually decreasing to around 0.3 by the end of the rounds. Electre Selection and Macbeth Selection demonstrate a more consistent and rapid reduction in FRR, stabilizing at approximately 0.2 towards the latter rounds. These results underscore the effectiveness of Electre Selection and Macbeth Selection in improving client selection processes by achieving lower FRR compared to Random Selection. The evaluation highlights the robustness and precision of Electre Selection and Macbeth Selection in minimizing FRR, making them highly effective approaches for federated learning scenarios where reducing false rejections is crucial for maintaining model accuracy and reliability.

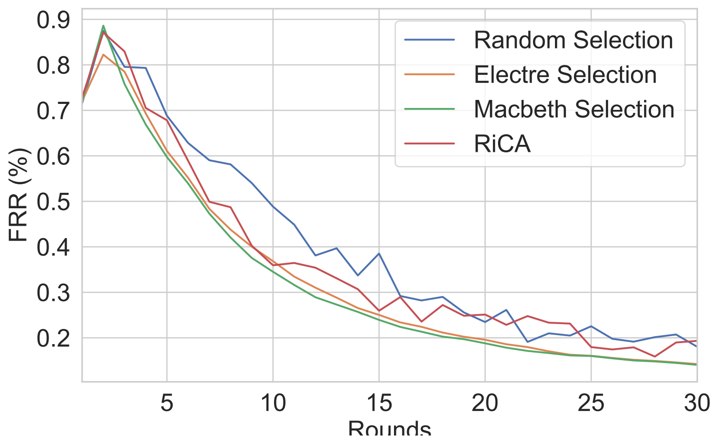

Figure 4. False Rejection Rate measurements

5. Conclusion and Future Works

References

AbdulRahman, S., Tout, H., Mourad, A., and Talhi, C. (2020). Fedmccs: Multicriteria client selection model for optimal iot federated learning. *IEEE Internet of Things Journal*, 8(6):4723–4735.

Barros, A., Rosário, D., Cerqueira, E., and da Fonseca, N. L. (2021). A strategy to the reduction of communication overhead and overfitting in federated learning. In *Anais do XXVI Workshop de Gerência e Operação de Redes e Serviços*, pages 1–13. SBC.

Fu, L., Zhang, H., Gao, G., Zhang, M., and Liu, X. (2023). Client Selection in Federated Learning: Principles, Challenges, and Opportunities. *IEEE Internet of Things Journal*, 10(24):21811–21819.

Kusano, K. D., Scanlon, J. M., Chen, Y.-H., McMurry, T. L., Chen, R., Gode, T., and Victor, T. (2023). Comparison of waymo rider-only crash data to human benchmarks at 7.1 million miles.

Le, J., Zhang, D., Lei, X., Jiao, L., Zeng, K., and Liao, X. (2023). Privacy-preserving federated learning with malicious clients and honest-but-curious servers. *IEEE Transactions on Information Forensics and Security*.

Li, Q., He, B., and Song, D. (2021). Model-contrastive federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10713–10722.

Liu, Y., Yu, J. J., Kang, J., Niyato, D., and Zhang, S. (2020). Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach. *IEEE Internet of Things Journal*, 7(8):7751–7763.

Lobato, W., Costa, J. B. D. D., Souza, A. M. d., Rosário, D., Sommer, C., and Villas, L. A. (2022). Flexe: Investigating federated learning in connected autonomous vehicle simulations. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pages 1–5.

Ma, X., Zhu, J., Lin, Z., Chen, S., and Qin, Y. (2022). A state-of-the-art survey on solving non-iid data in federated learning. *Future Generation Computer Systems*, 135:244–258.

Orlandi, F. C., Dos Anjos, J. C., Santana, J. F. d. P., Leithardt, V. R., and Geyer, C. F. (2023). Entropy to mitigate non-iid data problem on federated learning for the edge intelligence environment. *IEEE Access*.

Smestad, C. and Li, J. (2023). A systematic literature review on client selection in federated learning. In *Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering*, EASE '23, page 2–11, New York, NY, USA. Association for Computing Machinery.

Song, R., Zhou, L., Lakshminarasimhan, V., Festag, A., and Knoll, A. (2022). Federated Learning Framework Coping with Hierarchical Heterogeneity in Cooperative ITS. In *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE.

Xiong, Y., Wang, R., Cheng, M., Yu, F., and Hsieh, C.-J. (2023). Feddm: Iterative distribution matching for communication-efficient federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16323–16332.

Zhang, J., Hua, Y., Wang, H., Song, T., Xue, Z., Ma, R., and Guan, H. (2023a). Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11237–11244.

Zhang, X., Liu, J., Hu, T., Chang, Z., Zhang, Y., and Min, G. (2023b). Federated learning-assisted vehicular edge computing: Architecture and research directions. *IEEE Vehicular Technology Magazine*, pages 2–11.