Journal of the Brazilian Computer Society, 2024, 30:1, doi: 10.5753/jbcs.2024.4325

A Robust Client Selection Mechanism for Federated Learning Environments

Rafael Veiga [**Federal University of Pará (UFPA)** | ***rafael.teixeira.silva@icen.ufpa.br***]
John Sousa [**Federal University of Pará (UFPA)** | ***john.sousa@itec.ufpa.br***]
Renan Morais [**Federal University of Pará (UFPA)** | ***renan.morais@itec.ufpa.br***]
Lucas Bastos [**Federal University of Pará (UFPA)** | ***lucas.bastos@itec.ufpa.br***]
Wellington Lobato [**University of Campinas (UNICAMP)** | ***wellington.lobato@ic.unicamp.br***]
Denis Rosário [**Federal University of Pará (UFPA)** | ***denis@ufpa.br***]
Eduardo Cerqueira [**Federal University of Pará (UFPA)** | ***cerqueira@ufpa.br***]

Institute of Technology, Federal University of Pará, Av. Perimetral, s/n, Guamá, Belém, PA, 66075-110, Brazil.

Received:29 March 2024 • **Accepted:**16 June 2024 • **Published:** 08 October 2024

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. In this article, we introduce Resilience-aware Client Selection Mechanism for non-IID data and malicious clients in FL environment, called RICA. The proposed mechanism employs data size and entropy as criteria for client selection. In addition, RICA relies Centroid-Based Kernel Alignment (CKA) to identify and exclude potentially malicious clients. Our evaluation shows an improvement of 125% in Accuracy values in a scenario of malicious clients, which means the RICA+CKA demonstrates a more stable and resilient approach, reaching 90% accuracy in a few rounds compared to the default average approach, reached only around 30%. Therefore, results of the behavior of RICA+CKA in different datasets show the evaluation of different numbers of clients reaching around 90% while the other approach does not pass the 50% Accuracy.

Keywords: Federated Learning, Client Selection, Entropy.

1 Introduction

The use of Big data and deep learning in different applications makes our lives more intelligent and efficient, since Machine Learning (ML) has become ubiquitous and essential among key stakeholders [Kusano *et al.*, 2023]. However, ML applications require extensive data sharing, which raises significant communication and privacy concerns [Smestad and Li, 2023]. For instance, data privacy concerns arise due to the inherent nature of centralized ML models, where user-generated data often contains sensitive information. Additionally, existing ML approaches primarily rely on a cloud-centric architecture for data storage and processing. This centralized approach can lead to communication bottlenecks, resulting in unacceptably high latency and communication costs.

The convergence of ML and cloud computing is anticipated to shift towards a distributed edge computing paradigm [Zhang *et al.*, 2023b]. In this context, Federated Learning (FL) emerges as a compelling solution for future ML applications due to its inherent communication-preserving and privacy-enhancing characteristics [McMahan *et al.*, 2017]. Specifically, each client independently constructs its own model without leveraging the data and insights from other devices [Lobato *et al.*, 2022]. The shared local models are aggregated at the cloud or edge servers by a given aggregation policy to produce an accurate global model. In this way, FL is a key collaborative approach to model development. It enables the construction of a unified model while preserving data privacy by keeping the data distributed across various stakeholders. Additionally, FL facilitates continuous learning by allowing the adaptation of the ML model without the need to share raw data.

Client selection is a critical component of the FL training process. This mechanism strategically chooses a subset of participating devices, often referred to as clients, to contribute to the model training in each learning round. However, it is important to select the set of clients with valuable samples, *i.e.*, excluding clients who do not add value to the training and to improve the global model while increasing the cost of communication [Smestad and Li, 2023]. The FL client selection process is crucial to ensure a diverse and representative data sample from various clients. Contributing to the model's robustness without centralizing data, thus maintaining privacy. Selecting clients based on data quality, availability, and computational capacity optimizes training, improving the model performance in FL systems.

FL faces challenges due to the heterogeneous nature of participating clients. Client devices often possess diverse data distributions and varying hardware characteristics. This heterogeneity leads to Non-Independent and Non-Identically Distributed (non-IID) data scenarios. In such scenarios, data samples from different clients are not statistically independent and may exhibit distinct underlying statistical distributions [Xiong *et al.*, 2023]. The diversity in data distribution frequently results in diminished model accuracy, slower convergence rates during training process, and the potential emergence of fairness concerns if not adequately mitigated.

Furthermore, the reliance of FL on clients sharing trained weights heightens the vulnerability to malicious clients injecting erroneous updates, thereby compromising the integrity of the entire model training process [Le *et al.*, 2023]. Model poisoning attacks pose a significant threat in FL, where adversaries can exploit the distributed nature of model training [Ghodsi *et al.*, 2023]. This attack occurs when malicious clients intentionally manipulate their model updates to share them with the aggregation server, potentially including low-quality weights that decreases the prediction accuracy of the aggregated model [Yan *et al.*, 2023]. Due to this manipulation, the model parameters can severely compromise its integrity and performance once integrated into the global model. Distinguishing between malicious updates and the inherent variations observed in honest client updates presents a significant challenge in FL. This is because local model updates from legitimate clients naturally exhibit some degree of variation. Consequently, ensuring model resilience against poisoned updates is crucial for maintaining the accuracy and integrity of the FL system.

One approach to mitigating the impact of non-IID data and malicious users is by clustering clients based on the statistical features of their datasets, optimizing the accuracy of FL models. In addition, it is important to assess the similarity between trained model representations to mitigate the effects of non-IID data and malicious users. In this context, clustering algorithms are an important tool for grouping participants with comparable data distributions or learning objectives, creating clusters with similar models. However, to the best of our knowledge, the issue of designing an efficient client selection mechanism in a scenario with poisoning attacks and non-IID data remains a challenge.

This article introduces RiCA (Resilience-aware Client Selection Mechanism). RiCA addresses the challenges of non-IID data and malicious clients in FL environments. It achieves this by proposing a novel client selection mechanism that incorporates three key factors, namely, model performance on the client's data, data size on the client, and the entropy of the client's local updates. At the client selection phase, RiCA filters clients with larger data sizes and selects the remaining clients after using their Entropy as weights. In addition, RiCA uses the Centroid-Based Kernel Alignment (CKA) method for cluster creation to protect the global model against poisoning attacks, where CKA creates the clusters and gives them a score of similarity to detect malicious clients. Our experimental evaluation demonstrates that the RiCA client selection mechanism achieves superior performance compared to the baseline model in terms of both accuracy and loss reduction throughout the training rounds.

This article extends the previous work [Sousa *et al.*, 2023], where its main research contributions can be summarized as follows: i) the development of novel approach for client selection based on their data size, while also incorporating entropy to enhance the diversity of data types presented into the training phase; ii) the use of CKA similarity score as a clustering criterion to identify malicious clients based on their models prior to the aggregation step; iii) new evaluation results to demonstrate the performance of the proposed scheme.

The rest of this article is organized as follows. Section 2 presents an overview of works that explore similar proposals related to FL methods of resilience and protection approaches. Section 3 describes our methodology for protecting our aggregation, selection, and resilience cluster approaches. Section 4 explores the simulation model and the results obtained related to the use of our method. Finally, Section 5 concludes the article and directions for future work.

2 Related Work

This section presents key state-of-the-art approaches that enhance the resilience of FL against malicious client attacks or compromised models. In decentralized FL scenarios, client selection plays a crucial role, particularly when dealing with non-IID data distributions across participating devices. Thus, entropy-based selection methods have emerged as a promising approach [Orlandi *et al.*, 2023]. The authors used an approach to solve the problem of this type of data by using an Entropy-based approach to the client's distributions and mode balancing during their training step. However, they did not consider an additional filter of the client's data size.

Ghodsi *et al.* [2023] incorporated statistical bounds in zero-knowledge proofs to identify and discard malicious updates without disclosing private user data, called zPROBE. It achieves Byzantine-resilient and secure FL. However, it lacks in terms of detecting malicious clients. RiCA improves the detection of malicious clients in FL by using CKA. This method leverages the sum of similarity scores computed using CKA to identify clusters with clients exhibiting anomalous behavior, potentially indicative of model poisoning attacks.

Yan *et al.* [2023] introduced a novel defense mechanism called DeFL. This defense utilizes a federated gradient norm vector (FGNV) to detect minor but impactful discrepancies in DNN model updates. DeFL identifies malicious clients with this fine-grained approach and pinpoints CLPs, adapting this information to exclude these clients from the aggregation phase. However, this approach does not consider the model parameters of the clients. In our model, using CKA improves the detection of the malicious client by comparing the similarity with the other clients' parameters.

Sousa *et al.* [2023] investigated the efficacy of an entropy-based client selection mechanism within a FL framework modified for vehicular networks. Through comprehensive simulations involving a network of 20 vehicles, the study demonstrates a significant enhancement in learning performance when utilizing the entropy-based selection method, compared to traditional random selection techniques. However, this approach does not consider the fairness of FL. Choosing only the clients with the most Entropy values may bias the results, which in our proposal will not occur due to the probability applied and defining a minimum number of data size values to choose clients.

Albaseer *et al.* [2021] introduced a client selection mechanism to enhance Clustered Federated Learning (CFL) in wireless edge networks, focusing on reducing training time and improving model convergence without specifically addressing data diversity or richness, unlike entropy-based CKA clusterization methods. This approach ensures equitable client participation but might not fully leverage the potential of selecting clients based on data informativeness. That is crucial for optimizing learning outcomes in environments with heterogeneous data distributions. However, this approach needs to consider the malicious clients on their approach, which means an unsafe aggregation, possibly causing the strategy of clustering to be frustrated. In that way, a similar approach to the author's is using CKA as a resilience control tool for clients by comparing their parameters.

Souza *et al.* [2023] proposed DEEV to address communication challenges and scalability issues by dynamically adapting the number of participating devices and training rounds through a client selection strategy that selects the clients whose accuracy falls below the average. Using a containerized environment, DEEV showcases significant reductions of up to 60% in communication overhead and an impressive 90% in computation overhead compared to existing approaches. Its robust performance in scenarios with non-independent and non-identically distributed data underscores its potential for enhancing FL model efficiency. However, this approach does not account for malicious clients, implying that the system might aggregate their clients during some filtering rounds. Our method deals with choosing fewer clients in a round than the DEEV approach does in the initial rounds.

Regarding malicious clients in FL, Zhang *et al.* [2022] introduces ”FLDetector,” a technique aimed at identifying and mitigating model poisoning attacks by detecting malicious clients. FLDetector leverages the inconsistency in model updates across training iterations to indicate malevolent activity, proposing a mechanism to predict and compare clients' updates for anomalies. While showing promise in enhancing FL security across various datasets and attack models, this method may only partially address challenges posed by advanced attackers capable of mimicking benign behavior, scalability in large networks, privacy implications of the detection process, and adaptability across diverse FL environments. However, this approach does not improve FLDetector's client selection, which is different from our approach; they can choose a not valuable client to train.

Jee Cho *et al.* [2022] presented a novel framework called Power-Of-Choice, a communication and computation-efficient client selection framework that flexibly spans the trade-off between convergence speed and solution bias. The work achieves three times faster convergence and 10% higher accuracy by the author's tests. However, the work does not focus on the data quality or the impact that the data from each client can have on the FL training. However, this approach does not use this approach on malicious clients, which means their approach does not have a detection improvement of this client who can worst their results.

Liu *et al.* [2020] proposed a novel Federated Learning (FL) model called FedGRU that uses Federated Averaging (FedAvg) as the core of the secure parameter aggregation mechanism to collect gradient information from different organizations, the filtering of clients participating in the training is not considered for this algorithm. Therefore, client selection still remains an open challenge in the vehicular field. The proposed model only discussed the reduction of communication overhead without considering the quality of the data.

Sattler *et al.* [2020] explores how the Clustered Federated Learning (CFL) framework functions in environments with Byzantine clients who behave unpredictably or try to sabotage the training process. They conducted experiments with deep neural networks on standard Federated Learning datasets. They found that CFL can accurately identify and remove these Byzantine clients without modifying the framework. However, this approach only improves the client selection in protecting the global model, which means the clients with no values can be chosen more often. Our approach introduces a client selection improvement to reduce the number of rounds and achieve a faster, more stable accuracy.

Ghosh *et al.* [2022] proposed the Iterative Federated Clustering Algorithm (IFCA), which alternately estimates the identities of user clusters and optimizes the model parameters for user clusters through gradient descent. The author shows the advantages of using the clustering method in an FL approach, which is guaranteed to converge, and discusses the optimal statistical error rate. However, this approach uses a different cluster to detect than his cluster but chooses the cluster most relevant to training. Our approach uses the cluster to be more robust and resilient in malicious scenarios.

Table 1 presents a comprehensive overview of the key attributes of reviewed studies pertaining to the challenge of client selection, encompassing clustering techniques, client selection methods, detection of malicious clients, and robust methodologies for FL approaches. Based on the state-of-the-art analysis, it is imperative to adopt a robust FL approach to addresses the challenges of non-IID data and malicious clients in FL environments. Therefore, efficient client selection is pivotal for training the global model effectively. To achieve this objective, insights gleaned from other FL studies must be leveraged to expedite the identification of the most suitable approaches. Consequently, by integrating considerations of data distributions and size, the significance of clients within the client set can be enhanced, thereby establishing a robust methodology for detecting and eliminating malicious clients.

3 A Robust Client Selection Mechanism for Federated Learning

This section introduces RiCA (Resilience-aware Client Selection Mechanism), a novel approach designed to enhance the robustness and resilience of client selection in FL scenarios. RiCA 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 on each

Table 1. Related Works

Work	Malicious Clients	Client Selection	Clusterization	Resilience/Robustness
Ghodsi *et al.* [2023]	Yes	No	Yes	Yes
Yan *et al.* [2023]	Yes	No	No	Yes
Sousa *et al.* [2023]	No	Yes	No	No
Albaseer *et al.* [2021]	No	Yes	Yes	No
Souza *et al.* [2023]	No	Yes	No	Yes
Zhang *et al.* [2022]	Yes	No	Yes	Yes
Jee Cho *et al.* [2022]	No	Yes	Yes	Yes
Liu *et al.* [2020]	No	No	Yes	Yes
Sattler *et al.* [2020]	Yes	No	Yes	Yes
Ghosh *et al.* [2022]	No	No	Yes	Yes
RiCA [2024]	Yes	Yes	Yes	Yes

client. Prioritizing clients with smaller data sizes aims to mitigate the negative influence of potentially biased or overly influential large datasets. (ii) CKA-based Clustering: Following client selection, RiCA utilizes Kernel Conditional Covariance Analysis (CKA) for clustering. This clustering groups newly selected clients with those that previously participated in the training process. This step strengthens system resilience by leveraging the historical behavior of trusted clients to identify potential anomalies in new client updates.

3.1 Scenario overview

We consider a scenario composed of N devices, denoted as $u_i \in \{u_1, \dots, u_N\}$. We adopt a typical FL framework, initiating every communication round by choosing a group of K client devices, referred to as Client, to receive the global model, perform the training based on its Dataset D_i. The client selection mechanism must select a set of K clients with valuable samples to reduce the waste of computation resources, where it must remove the learning whose data are no longer critical for the model training. The dataset D_i consist of a collection of features $x_{k,i}$, for $k \in \{1, \dots, \|D_i\|\}$, with each feature paired with a corresponding label $y_{k,i}$. The selected clients improve the Model *Mn* by training with its dataset D_i.

After the training phase, client updates are transmitted back to the central aggregation server [Song *et al.*, 2022]. These updates can encompass either the learned model parameters or the calculated gradients. This approach allows each device to contribute its unique data to the FL process, aiding in developing a comprehensive and robust global model [Barros *et al.*, 2021]. Upon receiving these updates, the central server employs a specific aggregation strategy to integrate the updates into a cohesive model. A common strategy used is the Federated Averaging (FedAVG) [Liu *et al.*, 2020] algorithm, which calculates the mean of all local models shared by client devices, which is computed based on Eq. 1. This aggregation process at the edge servers plays a critical role in consolidating the client updates and generating a refined global model. The aggregated model, incorporating the collective learning progress from participating devices, is subsequently distributed back to the selected clients for the next training round [Lobato *et al.*, 2022].

$$W_t = \Sigma_{i=1}^{m} \frac{n_i}{n} W_i \tag{1}$$

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 RiCA mechanism uses datasets size and Entropy Ei as input to select the set of K clients with relevant data to improve the global model. In addition, RiCA uses CKA to maintain the system robustly and adaptable to attacks on the Global Model Gm. Specifically, CKA evaluates the average similarity among clients within each cluster, thus safeguarding against the inclusion of malicious users in the aggregation process. As depicted in Figure 1, after the selection process, clients transmit their parameters to the central server, 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 back to all participating clients for the next training round. Table 2 summarizes the list of main symbols used to introduce RiCA mechanism.

In FL, the distribution of data among participating clients can be mathematically expressed using probability distributions. The distribution of data on each client C_i is represented by a local dataset D_i, and the overall distribution of data across all clients can be expressed as the union of these local datasets:

$$D = \bigcup_{i=1}^{K} D_i \tag{2}$$

Now, let p_i denote the probability distribution associated with the data on client C_i, which can be expressed as:

$$p_i(x) = \frac{\text{Number of occurrences of } x \text{ in } D_i}{\text{Total size of } D_i} \tag{3}$$

This probability distribution $p_i(x)$ captures the likelihood of encountering a specific data point x within the local dataset D_i. The FL process involves aggregating information from all clients to build a global model. The aggre-

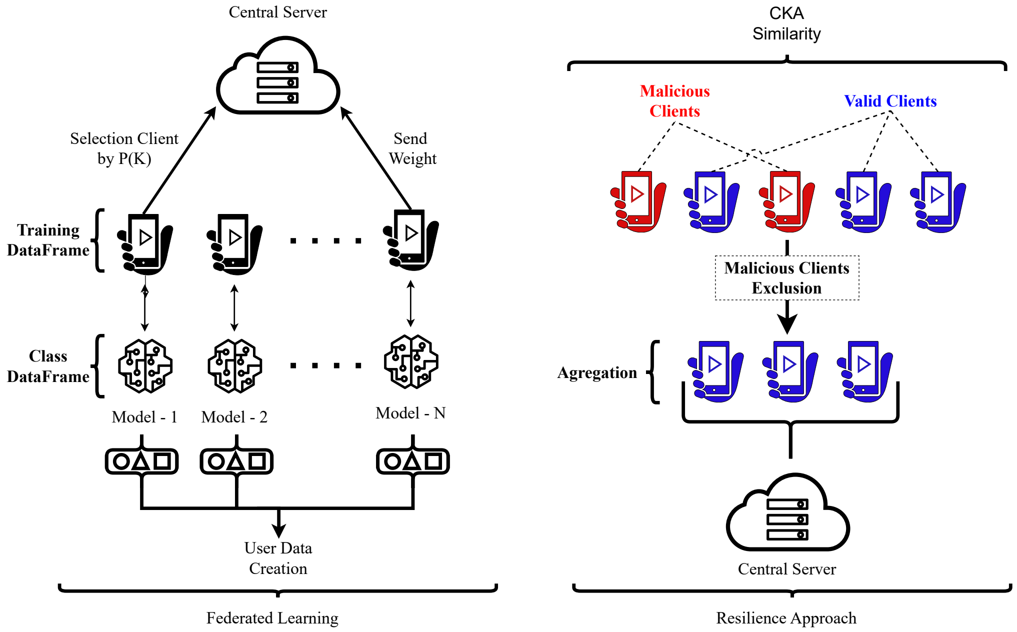


Figure 1. Scenario overview for robust global model attacks

Table 2. List of Symbols

Acronyms	Description
Dn	Dataset
Gm	Global Model
Mn	Model
Wk	Client weight within the cluster
Wi	Model Weight
K	Participating clients
P(k)	Probability of K
En	The sum result in the total data from all Entropy
Ek	Number of data from that client Entropy
H	Entropy
θ_{global}	Global aggregated parameters

gated model parameters θ are iteratively updated by incorporating contributions from individual clients, which can be represented as:

$$\theta_{global} = \frac{1}{K}\sum_{i=1}^{K}\theta_i \quad (4)$$

Here, θ_{global} represents the global model parameters, and θ_i represents the model parameters updated by client C_i. This distribution, captured by the probability distributions $p_i(x)$ for each client C_i, lays the groundwork for assessing the diversity of data in the federated network. It is within this context that entropy emerges as a pivotal tool.

3.2 Clients Selection

The performance of FL training process is heavily linked to the quality and diversity of the client data. 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 this context, the client selection scheme based on specific metrics for participation in the FL process becomes a critical factor in determining the effectiveness of the learning framework [Fu *et al.*, 2023]. The effectiveness of FL critically depends on the quality and diversity of data distributed across participating clients

During the client selection phase, the aggregation server uses the data size as a criterion to identify the top clients. The server evaluates the performance of the model trained on each client's data. Based on this evaluation, a weight is assigned to each client, reflecting its potential contribution to the training process. After that, the server calculates the probability of choosing a client for training based on its relative importance compared to others within the same cluster. The Eq. 5 ensures that clients with superior relevance in terms of the diversity of classes in each client and more extensive data to train, which have a higher probability of participating in the training process, preventing the occurrence of an overly dominant "superfit" scenario. En denotes the aggregate count of all appearances of each K in this collective sum, added to the *Ek*. We employ these dataset strategies to evaluate the *P(K)* for a client's submission to our server, showcasing the ability to identify the most pertinent clients. This method ensures fairness in FL, unlike other selection strategies that often prioritize a fixed group based on predefined selection criteria.

$$P(K=k) = \frac{E_k}{\sum_{n=0}^{N} E_n} \quad (5)$$

In this context, RiCA 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. RiCA leverages entropy as a key metric during client selection. Entropy serves as a measure of data diversity, allowing RiCA 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.

The RiCA entropy is computed by evaluating the probability distribution of symbols or messages that a source can produce [Fu *et al.*, 2023]. By calculating the entropy of individual clients' data, RiCA gains an understanding of the degree of randomness or uncertainty inherent in the data. Leveraging entropy-based client selection allows FL algorithms to identify the most pertinent and diverse data, thus facilitating the learning of models that can effectively accommodate heterogeneity. Through the selection of clients with high entropy, RiCA can ensure that learned models accurately represent the entire network, capturing variations in driving behavior, traffic patterns, and network connectivity.

We can formally measure the concept of data diversity by applying the Shannon entropy formula. For instance, consider two datasets, denoted by A and B. Dataset A possesses an even distribution of labels, where each label $0, 1$ appears with a probability of 50% . Following this, we can calculate the Entropy of this specific dataset, which is $-0.5\log_2(0.5) - 0.5\log_2(0.5) = 1bit$. In contrast, Dataset B has a skewed distribution with labels $0, 1$, where label 0 occurs in 90% and label 1 occurs in 10% of the time. The entropy of this scenario is $-0.9\log_2(0.9) - 0.1\log_2(0.1) \approx 0.47bit$. This scenario demonstrates that dataset A, with its higher entropy, exhibits a more balanced and diverse distribution of classes compared to dataset B.

This example illustrates that a high entropy in data labels signifies diverse data, mitigating overfitting and bias by ensuring a balanced representation of classes. Such diversity empowers models to capture general patterns and subtle nuances, thereby enhancing their performance across various scenarios. By selecting clients with high-entropy datasets, FL environments gain the capability to enhance model generalization and fairness, highlighting entropy as a strategic factor in the development of effective machine learning models.

RiCA uses the Shannon Entropy formula to calculate the entropy, as described in Eq. 6. $H(X)$ is the entropy of the dataset, $P(x)$ is the probability of observing a particular value x in the dataset, and log is the natural logarithm. Clients with datasets exhibiting a high entropy level are chosen due to their possession of diverse and informative data, which has the potential to enhance the performance of the FL model.

$$H(X) = -\sum_{x} P(x)\log P(x) \tag{6}$$

Applying the Eq. 6 in our scenario gives the entropy calculation for client selection ranking in FL, and the resulting equation can be described in Eq. 7. In this way, k_m refers to the class of the data point d_{ni}, which represents an individual data point in d_n. Prioritizing clients with higher entropy values during selection injects greater data diversity into the training process. This fosters the development of models that are more generalizable and adaptable to real-world variations. Consequently, the resulting model exhibits improved performance not only in terms of accuracy but also in its applicability across diverse FL scenarios, making it well-suited for non-IID environments.

$$H(d_n) = -\sum_{j=1}^{m} P(k_m)\log P(k_m) \tag{7}$$

Although the entropy-based approach to client selection in FL presents notable benefits, particularly in bolstering model resilience and diversity, it is not exempt from limitations. A prominent drawback is its perceived suitability primarily for classification tasks, attributed to the conventional application of entropy in assessing uncertainty or variability within categorical datasets. This perception emerges from the classical information theory context, where entropy quantifies the unpredictability of a system's state, a concept directly applicable to the distribution of categories or classes in classification tasks.

The RiCA approach employs a filtering strategy that prioritizes clients based on data size and subsequently utilizes entropy to determine their probability. 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 probability 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.

Algorithm 1 illustrates the operational procedure of RICA on the server for addressing client selection challenges. Beginning with a focus on the fundamental objective of clients, the algorithm prioritizes filtering clients with greater relevance. In addressing non-IID clients, RiCA aims to identify optimal fits for training in each round of FL. The initial filter targets clients with appropriate data sizes, ensuring a sufficient quantity of samples for effective training. Each client is required to possess numerous samples to facilitate accurate evaluation during training and enhance the overall round performance.

Initially, the process starts by aggregating the total data size from all participants, denoted as D_{total}. This step is instrumental in ascertaining the volumetric contribution of each client, which subsequently informs the selection of the top 30% of clients. The inclusion of data size as a selection criterion reflects the inherent value placed on larger datasets within the algorithm's decision-making process. Afterwards, the algorithm employs Shannon entropy to calculate each client's Entropy E_i. To quantify and assess data diversity within the client's dataset, RiCA employs Entropy as a crucial metric for client selection. By integrating data size and entropy values with the probability of weights, denoted as W_i, the algorithm identifies the primary clients for training. The coefficients β in this function balance the weighting assigned to both data size and diversity.

Algorithm 1 also presents our second filtering mechanism, aimed at selecting a subset of clients for training. This filter enhances the initial selection process by further refining client selection criteria based on data quality. During the final selection phase, it is imperative to calculate the client selection probabilities P_i based on their weights. The entropy filter is designed to identify clients exhibiting the highest entropy, signifying increased diversity or uncertainty in their data. Prioritizing clients with higher entropy, and subsequently filtering out 20% of the client set, the probability P_i will determine the top clients for training. This ensures a diverse dataset with ample size for training the CNN, thereby enhancing the model training by increasing the quantity of relevant data. By employing these two strategies to identify relevant data, a robust learning framework is established, minimizing the risks of overfitting. This approach allows us to leverage the benefits of training clients with larger data sizes and higher entropy, ultimately enhancing the overall training process and model quality.

Algorithm 1: Client Selection Phase

```
Calculate the total data size $D_{\text{total}}$ for all clients;
Select top 30% of clients based on data size;
for each client $i$ in the top 30% do
    Calculate the individual entropy $E_i$ of client $i$ using Shannon entropy:;
    $E_i = -\sum p(x) \log p(x)$;
    where $p(x)$ is the probability of occurrence of class $x$ in client $i$'s data;
end
Calculate a combined weight for each client $W_i$ using a polynomial function of data size and entropy;
for each client $i$ in the top 30% do
    $W_i = \beta \cdot E_i$;
    where $\beta$ are coefficients that balance the importance of entropy;
end
Normalize the weights $W_i$ to get selection probabilities $P_i$;
for each client $i$ in the top 30% do
    $P_i = \frac{W_i}{\sum_j W_j}$ where $j$ is in the top 30%;
end
Use $P_i$ to probabilistically select 20% of the clients;
for each client $i$ in the top 30% do
    Select client $i$ with probability $P_i$ until 20% of total clients are chosen;
end
Proceed with FL training using the selected subset of clients;
```

3.3 Resilience by Similarity Relevance

RiCA leverages the CKA metric due to its emergence as a robust tool for evaluating the similarity between representations across diverse layers of neural networks or distinct models. CKA quantifies the alignment between two datasets by assessing the similarity of their respective feature representations, with similarity values ranging from 0 to 1. A value of 0 signifies no similarity, while 1 indicates identical representations [Raghu *et al.*, 2021].

CKA helps to determine how closely the features learned by a model on one task align with those learned on another, guiding the selection of layers for transfer or fine-tuning. CKA employs various techniques to assess the similarity between matrices. In our approach, we use the Dot Product-based Similarity method, which relies on Equation 8 to compute the relative dot products of the samples, where H represents the centering matrix. The Dot Product-based Similarity method performs well in dynamic and non-iid FL scenarios.

$$HSIC(K, L) = \frac{1}{(n-1)^2} tr(KHLH), \tag{8}$$

The alignment is then normalized to provide a similarity score between 0 and 1 based on Eq. 9, where 1 indicates perfect alignment.

$$CKA(K, L) = \frac{HSIC(K, L)}{\sqrt{HSIC(K, K) HSIC(L, L)}} \tag{9}$$

The Algorithm 2 introduces an approach method for model protection global model based on the CKA similarity matrix. RiCA leverages client similarity to identify potential anomalies or malicious actors within the FL framework. RiCA focuses on clients whose data distributions exhibit minimal similarity to the expected patterns for the specific classification task. For instance, in an image classification scenario, clients with data containing irrelevant or significantly different imagery would be flagged for further investigation.

RICA relies on CKA similarity scores between clients' weights. The crux of the algorithm lies in its capacity to compute an average similarity, S_{i_avg}, for each client by averaging their similarity scores with every other client. This average serves as a metric to gauge the extent of alignment between the data distributions of different clients.

The CKA-based clustering stage in RiCA employs parameters to seek similarity by generating a confusion matrix for each client, which then transmits weights to the server, assigning a value to each client. Thus, it is essential to assign values to clients trained using similar data modalities, such as images, sounds, or natural language processing. This value aggregates clients within a cluster based on their similarity, with clusters exhibiting low similarity likely to contain malicious clients for FL aggregation. Consequently, it becomes necessary to exclude selected clients from clusters with low similarity, thereby enhancing the safety of FL aggregation for Refined Clients.

The similarity threshold, denoted as τ, acts as a hyperparameter determining the necessary level of similarity for excluding a client from the current set. Clients exceeding this threshold are deemed to have an adequate level of similarity and are therefore expected to make more meaningful contributions to the federated learning model. This threshold also represents the probability that a client was not trained for the same objectives as the other clients. By employing this thresholding mechanism, the algorithm ensures the retention of clients whose weights align closely with the broader data distribution, thereby potentially reducing variability and improving the robustness of the model. Thus, the algorithm assures the intricate considerations involved in optimizing client selection for FL, highlighting the intricate relationship between similarity, diversity, and model effectiveness.

Algorithm 2: Pulling Out Clients Phase

```
Selected clients with their weights $W_i$, CKA similarity threshold $\tau$;
Initialize an empty list, RefinedClients;
Compute the CKA similarity matrix $S$ for all pairs of selected clients;
for each client $i$ in the selected clients do
    Compute the average similarity $S_{i\_avg}$ by averaging all values in row $i$ of $S$;
    if $S_{i\_avg} > \tau$ then
        Add client $i$ to RefinedClients along with their weight $W_i$;
    end
end
return RefinedClients
```

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 obtained 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 used the MNIST, FMNIST, and CIFAR-10 datasets for our experiments (widely used public datasets for training and testing model validations). The MNIST dataset consists of several images, FMNIST comprises other images, and CIFAR-10 features color images of animals and objects. Therefore, our evaluation methodology will assess the performance of RiCA across these three distinct types 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 of the global model. We also assess the RICA scheme under malicious scenarios to compare its efficacy in selecting clients, despite the potential selection of malicious clients compared to the Accuracy and Loss. The evaluation of client selection based on their entropy-weighted weights will serve as a benchmark comparison to the default FL approach. The default FL means the most common strategy base for FL, using only random selection. Table 3 summarizes the simulation setup and parameters.

Table 3. Simulation Parameters

Description	Values
Number of Clients	30, 40, 50, 60 Clients
Number of Clusters	2, 3, 5 clusters
Datasets used	MNIST, CIFAR-10, FMNIST
Default number of clients	50 clients
Default number of clusters	5 clusters
Default dataset	MNIST

We compare these mechanism with metrics commonly used for authentication, namely Accuracy and Loss. The Accuracy metric is calculated by dividing the number of hits (positive) by the total number of examples. This calculation applies to data that include examples for each class and accounts for misses. Moreover, hit penalties for each class are the same.

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 (positive) 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, Equation 10 can be expressed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

This equation suggests that Accuracy is inherently straightforward and intuitive, offering a quick assessment of the model's performance. However, the simplicity of the accuracy metric also brings about limitations, particularly in scenarios with imbalanced datasets.

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

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, making it practical for assessing the certainty of predictions. The cross-entropy loss for that sample is the sum of $\log(\hat{y}_{ic})$ across all classes C. The overall loss is obtained by averaging these sums over all N samples. This approach penalizes confident but incorrect predictions, with the penalty increasing 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 shows the accuracy of the RiCA algorithm with the default state-of-the-art method, employing FedAVG with a random client selection. The default approach encounters numerous challenges when handling unexpected attacks within the scenario. Consequently, the default method yields poor results, failing to surpass 40% accuracy in our simulation. RiCA improves the client selection process and the accuracy (approximately 50%) when compared to the default scheme. The RiCA+CKA proposal surpasses other approaches within just a few rounds. RiCA+CKA achieves superior results even when compared to using only RiCA for five rounds, demonstrating an improvement of 125% over the other methods. Based on these accuracy results, the RICA+CKA mechanism shows in the evaluation a robustness and resilience approach to identify malicious clients and faster train evaluation.

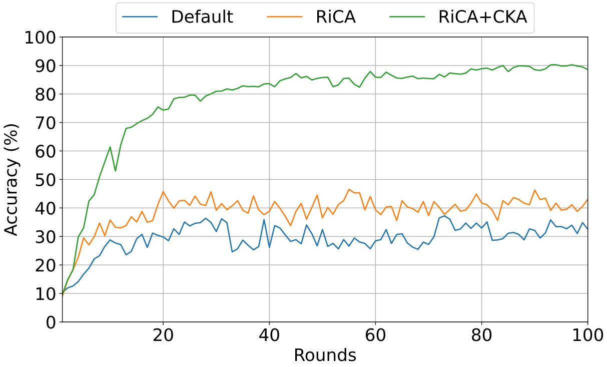


Figure 2. Accuracy measurements

Figure 3 analyzes the loss measurements and provides insight into the performance of our FL algorithm RiCA+CKA. The Default baseline algorithm starts with a significantly higher loss that gradually declines but remains above 1.5 throughout the training rounds. The RiCA algorithm exhibits an improvement in reducing loss, stabilizing just below the 1.0 threshold. The RiCA+CKA solution outperforms the other approaches and demonstrates a swift reduction in loss, dipping below 0.5 within the first 20 rounds and consistently maintaining this low level after that. This pattern signifies that RiCA+CKA rapidly converges to a robust solution and sustains a minimal loss, indicative of a high model resilience and accuracy level. Such performance underscores the effectiveness of configuring RiCA with CKA for FL applications, especially when model stability and reliability are critical. Based on this, the RICA mechanism also shows an improvement related to the loss patterns in the evaluation step. In that way, the RICA+CKA comparison demonstrates the improvement of that approach in a resilient FL method during this evaluation, which can improve the protection of the global model.

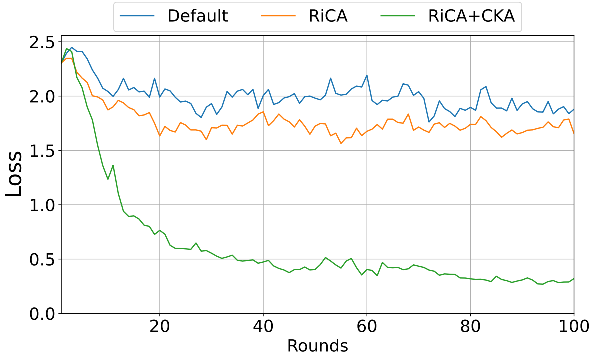


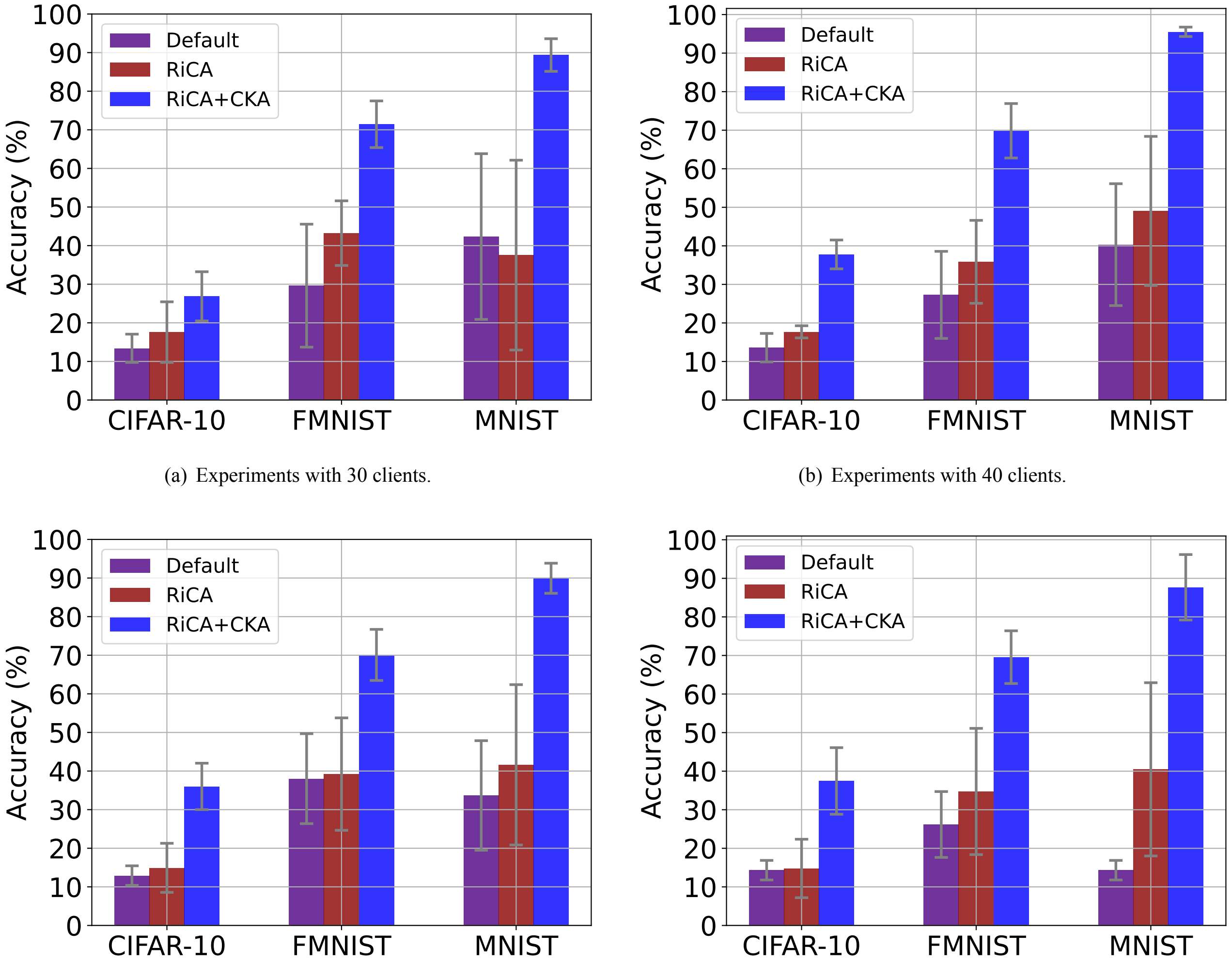
Figure 3. Loss measurements

Figure 4 presents a comparative evaluation of the three client selection approaches (Default, RiCA, and RiCA+CKA) across three datasets (CIFAR-10, FMNIST, and MNIST). The graphs depict the impact of varying client numbers (30 to 60) on model accuracy. The bar graphs allow for a clear visual comparison of the performance across different selection strategies and datasets.

Figure 4(a) presents the evaluation results of the three approaches with 30 clients on each dataset. RiCA+CKA demonstrates the highest evaluation across all datasets. Even in cases where results did not surpass 90%, such as MNIST, RiCA+CKA achieved approximately 90% accuracy with a consistent standard deviation. For the FMNIST dataset, RiCA+CKA achieves around 70% accuracy, while the default approach only reaches approximately 30% accuracy and exhibits a larger standard deviation. Only RiCA exhibits a minimal standard deviation and a potential value exceeding 40%. This observation holds true for the CIFAR-10 dataset as well, where a similar distribution is observed. However, CIFAR-10 presents a more arduous training scenario compared to others, resulting in lower values. The RiCA+CKA mechanism improved the results by approximately 26% with a deviation of around 5%, while the default scheme on CIFAR-10 only reached 12% of accuracy.

Figure 4(b) presents the evaluation results of the three approaches with 40 clients on each dataset. RICA+CKA demonstrates enhanced stability and yields results reaching approximately 95% when it is configured with MNIST. The RICA accuracy is of around 20%, 35%, and 50% for CIFAR-10, FMNIST, and MNIST, respectively. In contrast, the Default approach yields improvements of approximately 15%, 30%, and 40% for the same datasets, respectively.

Figures 4(c) and 4(d) depict the evaluation results of con-



(a) Experiments with 30 clients.

(b) Experiments with 40 clients.

(c) Experiments with 50 clients.

(d) Experiments with 60 clients.

Figure 4. Evaluation results of Client Variation Impact Across Comparative Dataset Analyses.

figuring the system with 50 and 60 clients, respectively. RiCA+CKA still achieves better accuracy results in all scenarios. For instance, the RiCA+CKA accuracy reaches around 90%, 78%, and 38% when the system is configured with CIFAR-10, FMNIST, and MNIST, respectively. In that way, based on the resilience of RICA+CKA, it shows possible to compare and find the malicious clients in each dataset, improving the Accuracy values in all evaluations on these figures.

Figure 5 illustrates the influence of cluster variation on the accuracy of the RiCA+CKA mechanism across 100 rounds. The accuracy steadily increases when utilizing 2 clusters, reaching near the 80% mark. This trend suggests efficient learning with a restricted number of clusters. The results show that configuring the system with 3 clusters results in only a marginal improvement. However, when utilizing 5 clusters, there is a minor decline in the convergence rate, leading to accuracy stabilizing slightly below that of the configurations with 2 and 3 clusters. That suggests that increasing the number of clusters may introduce complexity beyond a certain threshold without producing proportional gains in performance. While the method with 3 and 5 clusters shows evaluations around 90%, which means the method with 3 clusters already satisfies our criteria. The results suggest that while the RiCA+CKA algorithm exhibits resilience across various clustering configurations, achieving an optimal cluster quantity entails balancing the trade-off between accuracy and algorithmic efficiency. Based on this evaluation, different numbers of clusters in the RICA+CKA directly link to the number of malicious clients or the total number of clients. That parameter possibly removes clients from the aggregation step without necessarily being malicious to the global model.

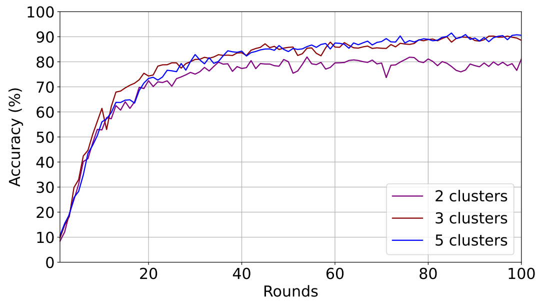


Figure 5. Cluster variation measurements

Examining the accuracy and cluster variation Figures for the RiCA+CKA mechanism uncovers nuanced insights into its behavior across various federated environments. In scenarios featuring fewer clusters, there is a rapid accuracy increase followed by a plateau, signifying that a smaller cluster set enables swift and effective model convergence. However, as the number of clusters increases, the accuracy benefit diminishes, implying the presence of an optimal cluster count for attaining peak performance without unnecessary computational complexity. In varying numbers of clients, the enhancement in accuracy is more noticeable in larger groups, emphasizing the benefit of a diverse dataset intrinsic to a broader client base. Nonetheless, one must consider the incremental gains alongside potential increases in communication and computational overhead. These findings underscore the significance of optimizing the FL environment, striking a balance between cluster quantity and client diversity to fully leverage the capabilities of the RiCA+CKA mechanism.

5 Conclusion

This article introduces RiCA, a Resilience-aware Client Selection Mechanism aimed at enhancing the performance of Federated Learning (FL) environments, particularly in scenarios involving non-Independently and Identically Distributed (non-IID) data and malicious clients. RiCA improves the FL client selection scheme by introducing a novel mechanism that considers three crucial factors during the selection process: the model performance on the client's data, the data size of the client, and the entropy of the client's local updates. Simulation results demonstrate that both RiCA alone and when configured alongside CKA (RiCA+CKA) outperform the results obtained from the baseline approach. For instance, the accuracy results for RiCA and RiCA+CKA are 49% and 95%, respectively, when the system is configured with MNIST and 40 clients.

For future work, the authors pretend to improve the selection approach by using parameters with other weights besides entropy. That means that Alan inspires a client's network parameters and processing to choose more adaptations and faster clients. Another approach dealing with resilience methods using the FL approach is to determine whether malicious clients are more realistic and different from each other to create a scenario with more variety. In that way, the simulation will improve by simulating sophisticated attacks in the global model.

Declarations

Acknowledgements

This research was partially funded by CNPq grant #405940/2022-0 and CAPES grant #88887.954253/2024-00. This study was also funded, in part, by the São Paulo Research Foundation (FAPESP), Brazil, under Process Number #2023/00673-7, and received additional support from the Amazon Foundation for Studies and Research Support (FAPESPA). This project was supported by the Brazilian Ministry of Science, Technology, and Innovations, with resources from Law nº 8,248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex, and published under Arquitetura Cognitiva (Phase 3), DOU 01245.003479/2024-10.

Authors' Contributions

R.V and R.M conceived and planned this study and experiments. R.V. and R.M. carried out the experiments, the simulations and computed the results. L.B., W.L, D.R., E.C. contributed to the interpretation of the results. R.V took the lead in writing the manuscript with final review of L.B, D.R and E.C. All authors provided critical feedback and helped shape the research, analysis and manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

All data and materials used in this work are based on the PFLib framework, publicly available at https://github.com/TsingZ0/PFLlib. Our modifications and contributions to the framework, including all the work related to this paper, can be accessed at https://github.com/OrenanM/Trabalho-IC/tree/master/PFL-Non-IID-RESILIENCE. Additional datasets used in our experiments and instructions on how to use and extract them are also provided through the original PFLib repository.

References

Albaseer, A., Abdallah, M., Al-Fuqaha, A., and Erbad, A. (2021). Client Selection Approach in Support of Clustered Federated Learning over Wireless Edge Networks. *2021 IEEE Global Communications Conference, GLOBECOM 2021 - Proceedings*. DOI: 10.1109/GLOBECOM46510.2021.9685938.

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. DOI: 10.5753/wgrs.2021.17181.

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.

Ghodsi, Z., Javaheripi, M., Sheybani, N., Zhang, X., Huang, K., and Koushanfar, F. (2023). zprobe: Zero peek robustness checks for federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4860–4870. Available at: `https://openaccess.thecvf.com/content/ICCV2023/html/Ghodsi_zPROBE_Zero_Peek_Robustness_Checks_for_Federated_Learning_ICCV_2023_paper.html`.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. (2022). An efficient framework for clustered federated learning. *IEEE Transactions on Information Theory*, 68(12):8076–8091. Available at: `https://proceedings.neurips.cc/paper_files/paper/2020/hash/e32cc80bf07915058ce90722ee17bb71-Abstract.html`.

Jee Cho, Y., Wang, J., and Joshi, G. (2022). Towards understanding biased client selection in federated learning. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 10351–10375. PMLR. Available at: `https://proceedings.mlr.press/v151/jee-cho22a`.

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*. DOI: 10.1109/TIFS.2023.3295949.

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. Available at: `https://openaccess.thecvf.com/content/CVPR2021/html/Li_Model-Contrastive_Federated_Learning_CVPR_2021_paper.html`.

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. DOI: 10.1109/JIOT.2020.2991401.

Lobato, W., Costa, J. B., Souza, A. M., Rosario, D., Sommer, C., and Villas, L. A. (2022). FLEXE: Investigating Federated Learning in Connected Autonomous Vehicle Simulations. *IEEE Vehicular Technology Conference*, 2022-Septe. DOI: 10.1109/VTC2022-Fall57202.2022.10012905.

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. DOI: 10.1016/j.future.2022.05.003.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Available at: `https://proceedings.mlr.press/v54/mcmahan17a?ref=https://githubhelp.com`.

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*. DOI: 10.1109/ACCESS.2023.3298704.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128. Available at: `https://proceedings.neurips.cc/paper_files/paper/2021/hash/652cf38361a209088302ba2b8b7f51e0-Abstract.html`.

Sattler, F., Müller, K.-R., Wiegand, T., and Samek, W. (2020). On the byzantine robustness of clustered federated learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8861–8865. IEEE. DOI: 10.1109/ICASSP40776.2020.9054676.

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. DOI: 10.1145/3593434.3593438.

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. DOI: 10.1109/ITSC55140.2022.9922064.

Sousa, J. L. R., Lobato, W., Rosário, D., Cerqueira, E., and Villas, L. A. (2023). Entropy-based client selection mechanism for vehicular federated environments. In *Proceedings of the 22nd Workshop on Performance of Computer and Communication Systems (WPERFORMANCE)*, pages 37–48. SBC. DOI: 10.5753/wperformance.2023.230700.

Souza, A., Bittencourt, L., Cerqueira, E., Loureiro, A., and Villas, L. (2023). Dispositivos, eu escolho vocês: Seleção de clientes adaptativa para comunicação eficiente em aprendizado federado. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 1–14, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbrc.2023.499.

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. Available at: `https://openaccess.thecvf.com/content/CVPR2023/html/Xiong_FedDM_Iterative_Distribution_Matching_for_Communication-Efficient_Federated_Learning_CVPR_2023_paper.htmll`.

Yan, G., Wang, H., Yuan, X., and Li, J. (2023). Defl: defending against model poisoning attacks in federated learning via critical learning periods awareness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10711–10719. DOI: 10.1609/aaai.v37i9.26271.

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. DOI: 10.1609/aaai.v37i9.26330.

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. DOI: 10.1109/MVT.2023.3297793.

Zhang, Z., Cao, X., Jia, J., and Gong, N. Z. (2022). FLDetector: Defending Federated Learning Against Model Poisoning Attacks via Detecting Malicious Clients. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2545–2555. DOI: 10.1145/3534678.3539231.