

Research Article

Enhancing Communication Efficiency in Federated Learning Through Client Contribution Selection Based Normalized L2 Norm with Cosine Similarity and L2 Norm Filtering

Goran Saman Nariman^{1,2,*}, Hozan Khalid Hamarashid³

¹ Department of Computer Science, College of Science, Charmo University, Chamchamal, 46023, Iraq

² Department of Information Technology, College of Science and Technology, University of Human Development, Sulaimani, 46001, Iraq

³ Information Technology Department, Computer Science Institute, Sulaimani Polytechnic University, Sulaimani, 46001, Iraq

*Corresponding Author: Goran Saman Nariman, E-mail: goran.saman@chu.edu.iq

Article Info	Abstract
Article History	Federated Learning (FL) is a decentralized machine learning framework in which clients train models locally and share only their updates with a central server for aggregation. However, FL faces critical communication efficiency and scalability challenges, particularly in large-scale networks where high communication overhead and slow convergence impede deployment. This paper proposes a novel client contribution selection strategy that combines Cosine Similarity and L2 Norm Filtering (CLF-C) with a Normalized L2-Based Client Selection (NLCS) mechanism to enhance communication efficiency and model convergence. The proposed approach prioritizes client updates that are both significant in magnitude and directionally aligned with the global model, thereby reducing unnecessary participation and improving aggregation quality. Experimental evaluations on the MNIST dataset demonstrate that the proposed method achieves 96.8% accuracy in 68 communication rounds using 468 clients, outperforming baseline models such as FedAvg (96.5%, 80 rounds, 800 clients) and FedProx (96.3%, 90 rounds, 900 clients). In addition to reducing communication rounds, the method enhances model robustness by filtering out misaligned or potentially adversarial clients. Importantly, it introduces no additional computational overhead for clients, making it well-suited for resource-constrained environments. This work contributes a scalable and secure solution for improving the efficiency of FL without compromising model performance.
Received Mar 01, 2025	
Revised Apr 27, 2025	
Accepted May 05, 2025	
Keywords	
Federated Learning	
Communication Efficiency	
Client Selection	
L2 Norm	
Cosine Similarity	
Decentralized ML	



Copyright: © 2025 Goran Saman Nariman and Hozan Khalid Hamarashid. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY 4.0) license.

1. Introduction

Federated Learning (FL) has emerged as a promising paradigm for decentralized machine learning, enabling multiple clients to train a global model collaboratively without sharing raw data. This approach is particularly beneficial in scenarios where data privacy and security are paramount, such as healthcare [1, 2], finance [3], and edge computing [4, 5]. However, FL faces significant challenges, particularly regarding

communication efficiency and scalability. In large-scale decentralized networks, the high communication overhead and slow convergence rates hinder the practical deployment of FL systems [6-8].

Traditional FL methods, such as Federated Averaging (FedAvg), require all clients to participate in every communication round, leading to inefficiencies. This approach often results in excessive communication overhead and resource consumption, which is especially problematic in environments where clients have limited computational and communication resources, such as mobile devices or IoT nodes. To address these issues, recent research has focused on client selection mechanisms prioritizing high-contribution clients, thereby reducing communication rounds and accelerating convergence [9-11].

According to Nariman et al. [12], communication round reduction and compression techniques for enhancing communication efficiency in FL can be broadly divided into two main classes. Compression techniques, such as quantization and sparsification, aim to reduce the size of the data transmitted between clients and the server. However, these methods often impose additional computational overhead on the client side, requiring extra processing to compress and decompress model updates. This can be particularly burdensome for resource-constrained devices, which may lack the necessary computational power to handle such tasks efficiently. In contrast, communication round reduction techniques focus on minimizing the number of communication rounds by selectively involving only the most relevant clients in each round. This approach reduces communication overhead and avoids additional computational burdens on clients, making it more suitable for resource-constrained environments where scalability remains critical in real-world FL deployments [13]. Given these considerations, this paper focuses on communication round reduction as a more practical and scalable solution for improving communication efficiency in FL.

Before delving into the proposed method, it is essential to understand the roles of cosine similarity and the L2 norm in evaluating client contributions. Cosine similarity measures the directional alignment between a client's model updates and the global model, ensuring that updates are consistent with the overall learning objective. On the other hand, the L2 norm quantifies the magnitude of the updates, indicating how significantly a client's local training has altered the model parameters. Combining these two metrics enables the identification of clients whose updates are both substantial in magnitude and well-aligned with the global model, thereby ensuring that only high-quality contributions are aggregated.

This paper proposes a novel client contribution selection mechanism based on Normalized L2 Norm and Cosine Similarity Filtering to enhance communication efficiency in FL. The proposed method selectively includes clients whose updates are substantial (measured by L2 norm) and well-aligned with the global model (measured by cosine similarity). By employing a two-stage filtering process, Cosine Similarity and L2 Norm Filtering with Clustering (CLF-C) is followed by Normalized L2-Based Client Selection (NLCS), ensuring robust and efficient aggregation of client updates. This method reduces the number of communication rounds and eliminates unnecessary client participation, thereby conserving client and server

resources. The key contributions of this work are as follows:

- **Communication Efficiency:** The proposed method significantly reduces communication overhead by selectively including high-contribution clients.
- **Faster Convergence:** The exclusion of low-contribution and misaligned updates accelerates the convergence speed of the global model.
- **Enhanced Security:** The combined magnitude and alignment filtering improve robustness against adversarial or faulty clients, ensuring that only high-quality updates are aggregated.
- **Scalability:** The proposed solution is lightweight and scalable, requiring no additional computational burden on clients, making it ideal for resource-constrained environments.

The remainder of this paper is organized as follows: Section 2 reviews related work on communication-efficient FL and client selection mechanisms. Section 3 details the proposed methodology, including the NLCS and CLF-C algorithms. Section 4 presents the experimental setup and evaluation results. Finally, Section 5 outlines limitations and future research directions.

2. Literature Review

FL has been widely explored to enhance communication efficiency, improve client selection, and optimize model aggregation strategies. Several methods leverage gradient-based filtering, clustering, and adaptive selection to enhance convergence and robustness in FL systems.

Wu and Wang [14] introduced FedPNS, a probabilistic node selection strategy that adjusts client participation probabilities based on the alignment of the gradient. By analyzing the inner product between local and global gradients, their method prioritizes clients that contribute positively to model convergence, reducing communication overhead while accelerating training. Cho et al. [15] developed POWER-OF-CHOICE, a biased client selection strategy that prioritizes clients with higher local loss values to enhance convergence speed. By dynamically selecting the most informative clients, their method achieves up to 3× faster convergence and 10% higher test accuracy than random selection.

Duan et al. [16] introduced FedGroup, a clustered FL framework that groups clients based on the similarity directions. The method uses a decomposed cosine similarity measure to enhance the efficiency of client clustering and to reduce the impact of non-IID data distributions. By splitting the optimization task into sub-models trained within these clusters, FedGroup improves learning stability while preserving scalability through an auxiliary global model. Additionally, they explored a Euclidean distance variant of cosine similarity for high-dimensional clustering, demonstrating notable improvements in accuracy on the MNIST and FEMNIST datasets. Similarly, Yu et al. [17] explored using cosine similarity to cluster clients based on their local updates, which helps refine client selection by grouping those with similar data characteristics.

Additionally, Chen et al. [18] proposed an optimal client sampling scheme that reduces communication

bottlenecks by selecting clients based on the L2 norm of their updates. This method ensures informative client participation, minimizing communication overhead while maintaining convergence guarantees for Distributed SGD and Federated Averaging. Their findings show that leveraging update norms achieves efficiency comparable to full participation. Cheng et al. [19] introduced a differentially private FL framework that utilizes Bounded Local Update Regularization (BLUR) and Local Update Sparsification (LUS) to improve model utility while maintaining user-level differential privacy. Their approach regularizes the L2 norm of local updates before clipping, ensuring updates remain within a controlled range to mitigate the negative impact of noise perturbation. By reducing the norm of local updates, their method enhances convergence stability while preserving privacy guarantees.

Artificial neural networks have been applied to model Quality of Experience (QoE) based on QoS metrics for multimedia services on cellular networks, helping capture user satisfaction beyond network-level performance [20]. Uthansakul et al. [21] proposed an ANN-based QoE model combined with a particle genetic algorithm to enable the self-tuning of service priority factors for efficient LTE resource allocation. A federated learning approach has been proposed to support decentralized smartphone recommendation, preserving user privacy while maintaining accuracy [22]. Kawoosa et al. [23] proposed an XGBoost-based ensemble method for detecting smart-meter energy theft by analyzing consumption patterns and injecting synthetic attack scenarios.

While prior research has explored client selection in FL using either the L2 norm (to measure update magnitude) or cosine similarity (to assess alignment with the global model), to the best of our knowledge, no previous work has combined both within a unified framework. This integration is crucial, as it enhances communication efficiency and security by ensuring that only significant and well-aligned updates are included in the aggregation. Additionally, introducing the Normalized L2 Norm standardizes client contributions, preventing dominance by outliers and promoting fair participation among all clients. By addressing these gaps, our approach enhances model robustness, accelerates convergence, and mitigates risks associated with adversarial or low-quality updates in decentralized learning environments.

3. Methodology

The proposed approach aims to reduce the number of clients participating in each round of the FL process and the number of models involved in the global model aggregation. Clients whose contributions are less significant, determined by their alignment with the global model, are excluded from the aggregation process and next communication round. This methodology consists of two main phases. First, the Normalized L2-Based Client Selection (NLCS) algorithm, which forms the core of the approach, filters out clients based on their contribution magnitude. Second, the Cosine and L2-Based Filtering with Clustering (CLF-C) algorithm acts as a pre-filter to identify which clients meet the criteria to proceed to the NLCS stage.

Each algorithm is presented along with its rationale and demonstrated effectiveness in enhancing communication efficiency, strengthening security, and accelerating global model convergence. Figure 1 illustrates the whole process of the proposed approach.

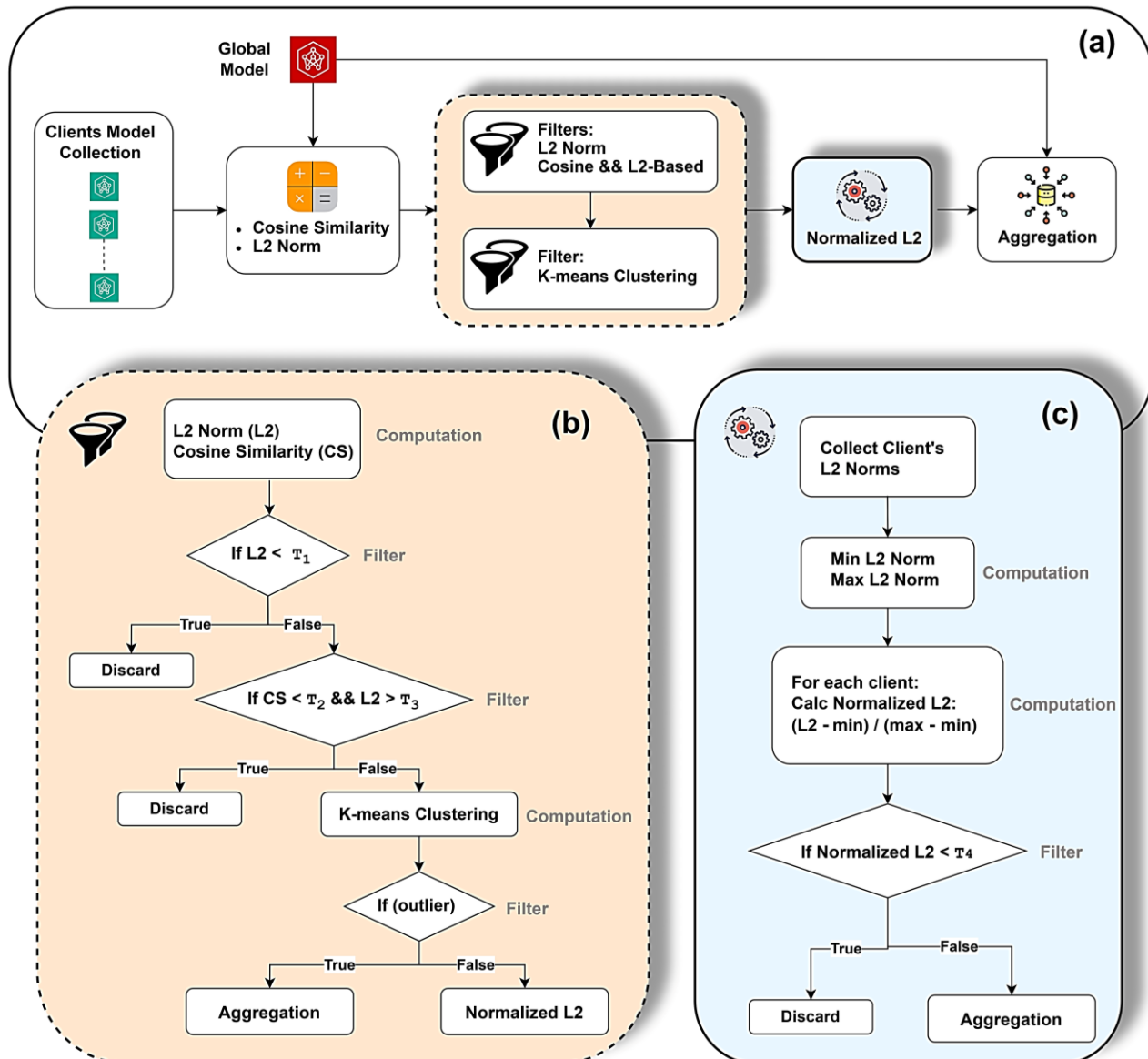


Figure 1. Overview of the proposed client selection approach. (a) Abstract representation of the overall methodology. (b) Flowchart of the CLF-C filtering process. (c) Flowchart of the Normalized L2 norm process, where T_1 , T_2 , T_3 , and T_4 represent the defined thresholds for L2 norm, combined cosine similarity and L2 norm, and normalized L2 norm, respectively

3.1. Normalized L2-Based Client Selection (NLCS)

The FL process begins with sharing the initial global model with clients during the first communication round, where parameter weights are randomly initialized. The initial stage emphasizes discarding less-contributive clients to reduce communication overhead while promoting fair participation by normalizing client contributions using L2 norms. This approach reduces communication costs and accelerates convergence by focusing aggregation on impactful updates, thereby improving the global model's accuracy and

achieving convergence more quickly. The process of the proposed approach, as shown in Algorithm 1, consists of the following steps in each round of the FL process until convergence is reached.

A. Collection and Storage of Client Parameters

At the end of each round, active clients send their updated model parameters to the central server, which stores them in a Vector Database (VD) for efficient operations.

B. Cosine and L2-Based Filtering with Clustering (CLF-C) (Alg. 2)

All clients' L2 norms and cosine similarity are prepared for further filtering. Without this intermediate filtering step, the subsequent Normalized L2 phase may be negatively affected, potentially hindering the convergence of the global model. This filtering process is detailed in Algorithm 2, illustrated in Figure 1(b), and further explained in the following subsection.

C. Normalizing L2 Norms Across Clients

To ensure fair client comparisons, the central server normalizes each client's L2 norm, scaling the values relative to the observed range across all clients. This provides a consistent and unbiased basis for evaluation, as illustrated in Figure 1(c).

$$\text{Normalized } L2(\text{client}_i) = \frac{L2_{\text{client}_i} - \min(L2)}{\max(L2) - \min(L2)} \quad (1)$$

By standardizing the L2 norms, the process highlights clients with significant variability in updates, identifying those with stronger contributions to the learning process.

D. Client Selection Based on Contribution Threshold

Using the normalized L2 values, the server identifies and discards clients with normalized L2 values below a defined threshold T . These clients are considered to have non-significant contributions for the current and upcoming rounds.

$$\text{Discard } \text{client}_i (D) \leftarrow \text{if } \text{Normalized } L2(\text{client}_i) < T \quad (2)$$

Clients with contributions below this threshold are excluded from aggregation and the next rounds, reducing communication load without compromising model quality.

E. Aggregating Parameters from Selected Clients

Clients whose Normalized L2 norms fall below the threshold T are discarded and excluded from aggregation. The remaining clients are aggregated to form the new global model.

$$x_{k+1} \leftarrow \frac{1}{|S|} \sum_{i \in S} \text{Parameter}_{\text{client}_i}, \text{ where } S = [N]/D \quad (3)$$

3.2. Cosine and L2-Based Filtering with Clustering (CLF-C)

Before diving into the details of the filtering mechanisms, it is essential to understand the rationale behind using these filters in the proposed approach. As outlined in Algorithm 1, the normalized L2 values

are computed by considering the minimum and maximum L2 values across all clients. However, clients with extreme L2 values can heavily influence this normalization process, resulting in outliers. If one or a few clients have significantly higher L2 values, the normalized L2 threshold becomes skewed, potentially preventing most clients in the round from passing the threshold, even if their parameter updates carry valuable information.

Algorithm 1. Normalized L2-based Client Selection for Communication Efficiency (NLCS)

1:	Input: Initial global model $\mathbf{x}_1$, client parameter $\mathbf{P} = \{parameter_{client_1}, \dots, parameter_{client_N}\}$, threshold T , discarded clients D
2:	for each round $k = 1, \dots, K$ do
3:	Broadcasts $\mathbf{x}_k$ to all clients $i \in [N] \setminus D$
4:	for each active client $i \in [N] \setminus D$ (in parallel) do
5:	Compute updated local model parameters $parameter_{client_i}$
6:	Send $parameter_{client_i}$ to the master
7:	Master store $parameter_{client_i}$ in vector database
8:	End for
Alg. X2: Filter (clients) based on Cosine and L2 value to participate in Normalized L2.	
9:	Compute min(L2) and max(L2) across active clients
10:	for each active client $i \in [N] \setminus D$ do
11:	Compute $Normalized\ L2\ (client_i) = \frac{L2_{client_i} - \min(L2)}{\max(L2) - \min(L2)}$
12:	if $Normalized\ L2_i < T$ then $D \leftarrow client_i$
13:	else
14:	Include $client_i$ in aggregation.
15:	end if
16:	end for
17:	$\mathbf{x}_{k+1} \leftarrow \frac{1}{ S } \sum_{i \in S} Parameter_{client_i}$, where $S = [N] \setminus D$
18:	end for

Another scenario arises when a client has a high L2 norm, but its contribution is not aligned to improve the global model. This misalignment can occur due to several factors, such as adversarial behavior, where an attacker intentionally disrupts the FL process, or client malfunction, where a client trains on irrelevant data, leading to undesirable parameter updates. In these cases, high L2 values may mislead the aggregation process, resulting in model degradation rather than improvement.

Two filters have been introduced to address these issues, as shown in Algorithm 2 and Figure 1(b). The first filter identifies and discards clients whose contributions are misaligned with the global model's

enhancement. This filter ensures that only clients making meaningful contributions are considered for further processing. After passing this first filter, clients undergo a second filter that clusters clients based on their L2 norm values. The clustering step identifies and removes outlier clients whose extremely high L2 values could distort the normalized L2 filtering process. These outlier clients bypass the normalization stage and proceed directly to the aggregation process, ensuring their extreme contributions do not adversely affect the model's convergence.

3.2.1. Cosine Similarity and L2 Norm Filtering

This filter evaluates the alignment and magnitude of client updates using Cosine Similarity and L2 norm. Cosine Similarity evaluates the directional alignment of a client update with the global model, while the L2 norm measures the magnitude of the parameter updates. L2 norm and cosine similarity are calculated in Alg. 2, lines 5 and 6.

1. Low Alignment or Magnitude: Clients are excluded from the aggregation if their updates exhibit either low directional alignment with the global model or insufficient magnitude, indicating limited contribution to learning.

$$\text{Discard } client_i (D) \leftarrow \text{if } \cos_sim_{client_i} < T_{cos} \text{ or } L2_{client_i} < T_{L2} \quad (4)$$

2. High Magnitude with Misalignment: This condition is designed to identify clients whose updates are extensive and poorly aligned with the global model, as such updates can negatively impact convergence and model stability.

$$\text{Discard } client_i (D) \leftarrow \text{if } L2_{client_i} > T_{L2_high} \text{ AND } \cos_sim_{client_i} < T_{cos_high} \quad (5)$$

The distinction between standard (T_{cos}, T_{L2}) and high thresholds ($T_{cos_high}, T_{L2_high}$) reflects the need to address diverse client behaviors:

- Cosine Similarity Thresholds: The standard threshold identifies poorly aligned updates, while the high threshold captures misaligned clients with significant magnitudes.
- L2 Norm Thresholds: The standard threshold removes minor updates, while the high threshold identifies extreme outliers with potentially destabilizing effects.

High thresholds are set more stringently to prevent disruptions from large but misaligned updates. Clients passing this stage proceed to the second filter for further evaluation.

3.2.2. Clustering-Based Outlier Detection

The second filter utilizes a clustering approach to manage clients with extreme L2 norm values, which can disrupt the Normalized L2 process by skewing the normalization and preventing other clients from meeting the threshold. These high L2 norms often result from non-IID data distributions, where clients' local data differs significantly, leading to larger update magnitudes.

The clustering mechanism works by grouping clients based on their L2 norm values using a clustering algorithm, such as k-means. The clients are partitioned into distinct clusters, and the cluster with the highest average L2 norm is identified as the outlier cluster. The clients in this outlier cluster bypass the Normalized L2 filtering process and proceed directly to the aggregation stage, as shown in Algorithm 2.

Algorithm 2. Cosine and L2-based Filtering (CLF-C)

```

1: Input: client parameter  $\mathbf{P} = \{parameter_{client_1}, \dots, parameter_{client_N}\}$ , thresholds  $T_{cos}$ ,  $T_{L2}$ ,  $T_{cos\_high}$ ,  $T_{L2\_high}$ , discarded clients  $\mathbf{D}$ 
2: for each round  $k = 1, \dots, K$  do
4:   for each active client  $i \in [N] \setminus \mathbf{D}$  (in parallel) do
5:      $L2_{client_i} = \sqrt{\sum_{j=1}^N (parameter_{client_i,j} - parameter_{client_i,global})^2}$ 
6:      $cos\_sim_{client_i} = \text{CosineSimilarity}(parameter_{client_i} - parameter_{client\_global})$ 
8:   End for
9:   for each active client  $i \in [N] \setminus \mathbf{D}$  do
10:    if  $cos\_sim_{client_i} < T_{cos}$  or  $L2_{client_i} < T_{L2}$  Then
11:       $\mathbf{D} \leftarrow \mathbf{D} \cup \{i\}$  // Clients discarded for low alignment or insufficient magnitude.
12:    else if  $L2_{client_i} > T_{L2\_high}$  AND  $cos\_sim_{client_i} < T_{cos\_high}$  Then
13:       $\mathbf{D} \leftarrow \mathbf{D} \cup \{i\}$  //Clients discarded for destabilizing updates
14:    end if
15:  End for
16:  Apply k-means clustering to the set of clients based on their L2 norms into  $k$  cluster.
    For each cluster  $C_j$  (where  $j \in \{1, \dots, k\}$ ) compute the mean L2 norm:
18:     $\mu_{C_j} = \frac{1}{|C_j|} \sum_{i \in C_j} L2(client_i)$ 
    Identify the cluster  $C_{outlaier}$  with the highest mean L2 norm:
19:     $C_{outlaier} = \arg \max_{j \in \{1, \dots, k\}} \mu_{C_j}$ 
20:    Clients  $i \in C_{outlaier}$  bypass further filtering and proceed directly to aggregation.
22: End for
23: Output: Updated discarded set  $\mathbf{D}$  and send  $C_{outlaier}$  to global model aggregation.
  
```

3.3. Security-Oriented Filtering

The proposed method enhances security by leveraging the combined filtering mechanism of cosine similarity and L2 norm to detect and exclude adversarial clients or those affected by data poisoning. This stage strengthens the security and reliability of the FL process by targeting two fundamental characteristics of adversarial behavior: directional misalignment and abnormal update magnitude. Below, we provide a

theoretical analysis and additional experimental validation in the next section to strengthen these claims.

For an adversarial detection mechanism, let Δw_i represent the update from client i , and Δw_g denote the global model update direction.

a. Cosine Similarity: Measures alignment:

$$\cos(\theta_i) = \frac{\Delta w_i \cdot \Delta w_g}{\|\Delta w_i\| \|\Delta w_g\|} \quad (6)$$

Low values ($\cos(\theta_i) \ll 1$) indicate misalignment, suggesting adversarial or noisy updates.

b. L2 Norm: Quantifies update magnitude:

Let W_g denote the current global model and W_i be the model parameters of client i after local training. The client update is defined as:

$$\Delta w_i = w_i - w_g \quad (7)$$

The L2 norm of this update quantifies the magnitude of deviation:

$$\|\Delta w_i\|_2 = \sqrt{\sum_{j=1}^d (w_{i,j} - w_{g,j})^2} \quad (8)$$

A high L2 norm indicates a strong deviation from the global model, which may reflect either a meaningful contribution or a potential attack when misaligned. By combining L2 norm and directional alignment, the proposed method filters out large and inconsistent updates. This forms the basis of the CLF-C algorithm described in Algorithm 2, with experimental validation provided in Section 4.2.3.

4. Evaluation

A real-world simulation was implemented to evaluate the proposed approach. The results were then compared with a baseline FL framework to assess improvements in communication efficiency and global model convergence. To do that, this section has been broken down into two main subsections.

4.1. Experimental Setup

The experimental setup was designed to evaluate the proposed approach for enhancing communication efficiency in FL by simulating a decentralized FL environment with multiple clients and implementing selective client contribution mechanisms. Key components included dataset preparation, client distribution, model architecture, client contribution selection mechanisms, and federated training. To ensure consistent evaluation, the same initial random model parameters were used across all experimental runs by fixing the random seed in PyTorch, ensuring variations in results were solely due to threshold changes and not random initialization.

4.1.1. Dataset Preparation

The experiments used the MNIST [24] dataset, a widely used benchmark for evaluating ML models.

According to Nariman et al. [12], who reviewed 76 research papers, 34 utilized the MNIST dataset, highlighting its popularity in FL studies. Additionally, the dataset was chosen because the approaches used for comparison with the proposed method, including FedProx and CMFL, also rely on MNIST, making it a standard for fair and consistent evaluation. Nariman et al. [12] identify these two approaches as the most commonly used for evaluation in FL research. MNIST consists of 60,000 training images and 10,000 test images of handwritten digits, each of size 28x28 pixels. The training dataset was partitioned among 100 clients to simulate an FL environment to mimic decentralized data storage.

To simulate a realistic non-IID data scenario, the pathological non-IID approach, described by McMahan et al. [25] and Briggs et al. [26], was selected, where labels are partitioned so that each client receives data corresponding to only two specific labels. For instance, one client might receive all images labeled as 1 and 9, while another might receive images labeled as 3 and 8. This method introduces substantial heterogeneity in the data distribution, mimicking real-world scenarios where client datasets are inherently unbalanced.

4.1.2. Clients Distribution

The FL setup included 100 clients, with 10 randomly selected to participate in each communication round. Each client was simulated as an independent node with its own local subset of the dataset, and a local model was trained accordingly. Clients sent their model parameter updates to a central server after local training. This selective participation reflects real-world constraints such as limited communication bandwidth and resource availability.

4.1.3. Model Architecture

The selected model architecture is a convolutional neural network (CNN), chosen for its ability to strike a balance between computational efficiency and strong predictive performance. According to Nariman et al. [12], CNNs are the most commonly used models in FL applications.

The architecture consisted of three convolutional layers, each with 3x3 kernels, stride 2, and padding 1: Layer 1 with 32 filters, Layer 2 with 64 filters, and Layer 3 with 128 filters. It also included two fully connected layers: Layer 1 reduced 2048 input features to 128 features, and Layer 2 reduced 128 input features to 10 output classes corresponding to digit labels. ReLU activations were applied after each layer, and the final output was passed through a SoftMax activation to compute class probabilities.

Each client locally trained the model, initialized with random weights, using the Stochastic Gradient Descent (SGD) optimizer with a learning rate 0.1. The training was performed for five local epochs with a mini-batch size 32. These values were chosen based on common practice in FL literature, balancing convergence speed and computational cost effectively.

The learning rate (0.1) was selected empirically after initial testing with lower and higher values (e.g., 0.01 and 0.5), where 0.1 consistently achieved faster and more stable convergence without overshooting.

Similarly, five epochs offered a practical trade-off between local update strength and communication cost per round, while larger values sometimes caused overfitting and divergence. The batch size was set to 32 to align with standard training practices for MNIST and ensure compatibility with most client devices.

To ensure consistent evaluation and fair comparisons across different threshold configurations, the same initial random model parameters were used in all experimental runs. This was achieved by setting a fixed random seed in PyTorch, which guarantees that the initialization of model parameters remains identical across different experiments. This approach ensures that variations in performance can be attributed solely to changes in thresholds rather than random initialization effects, thus enhancing the reliability and reproducibility of the results.

4.1.4. Client Contribution Selection Mechanisms

The core novelty of the proposed approach lies in the selective participation of clients, achieved through the NLCS and the CLF-C mechanism. These mechanisms reduce communication rounds and accelerate global model convergence by excluding less-contributive clients.

4.1.5. Federated Training and Aggregation

Each communication round involved the following steps:

1. Client Training: Selected clients trained their local models on their respective datasets for five epochs.
2. Client Update Transmission: Clients sent their updated model parameters to the central server.
3. Contribution Evaluation and Client Filtering: Clients were filtered using the NLCS and CLF-C algorithms.
4. Global Aggregation: The parameters of participating clients were averaged to update the global model.
5. Broadcast: The updated global model was shared with all clients for the next round.

The process was conducted over 100 communication rounds, with client participation dynamically adjusted based on their evaluated contributions.

4.1.6. Evaluation Metrics

The proposed approach was evaluated on two key metrics:

1. Model Performance: The global model's accuracy and loss were measured on a separate test set after each communication round.
2. Communication Efficiency: The reduction in communication overhead was assessed by tracking the number of discarded clients and the number of communication rounds.

Implementing the entire process is straightforward, with the advantage that the approach does not impose additional computation on the clients. The implementation is publicly available [27].

4.2. Results and Discussion

This section presents a detailed analysis of the experimental results obtained by implementing the proposed client contribution selection mechanism, which utilizes normalized L2 norm and cosine similarity filtering. The results are compared against standard FL baselines, such as FedAvg and FedProx, with an emphasis on communication efficiency, convergence behavior, and model performance.

Table 1 summarizes the results of different threshold configurations applied to the proposed approach regarding the number of communication rounds required for convergence, model accuracy, and the number of discarded clients. The best-performing configuration achieved 96.8% accuracy in 68 communication rounds, discarding 212 clients. This demonstrates the effectiveness of the proposed approach in achieving high accuracy with fewer communication rounds and reduced client participation.

Table 1. Performance of the proposed method under varying threshold configurations for normalized L2 norm, L2 norm, and combined cosine-L2 filtering

No. of Rounds	Accuracy	No. of discarded clients	Normalized L2 Threshold	L2 Norm Threshold	Cosine Similarity Threshold	L2 for Cosine Filter
68	96.70%	195	0.2	0.4	0.97	0.6-0.3
68	96.70%	209				
68	96.80%	212		0.4		
92	97.20%	298				
87	97%	283	0.25	0.3	0.98	0.5
92	96.60%	303		0.2		
75	96.10%	284		0.4		
68	96.3%	238		0.3		
92	97.20%	312	0.275	0.1	0.96	0.6
90	96.50%	326		0.2		
90	96.50%	326		0.3		
90	96.50%	326				
68	96.50%	253	0.3		0.97	0.3, 0.4, 0.5, 0.6
68	96.50%	253				
68	96.50%	253				
68	96.50%	253				
92	96.80%	365	0.35	0.4	0.95	0.3, 0.5
68	96.40%	250				
79	96.80%	288				
92	96.80%	324				
95	96.50%	408	0.4	0.5	0.98	0.5
90	97%	367				
10	67.80%	928				

The proposed approach does not introduce additional client computational requirements, making it particularly advantageous for resource-constrained environments. Clients generally favor approaches prioritizing communication efficiency over computational complexity, as minimizing local processing aligns better with the constraints of resource-limited devices while still supporting global model optimization.

4.2.1. Communication Efficiency and Convergence Behavior

The proposed method significantly reduces communication overhead while accelerating convergence. Using normalized L2 norm and cosine similarity-based filtering, clients with low or misaligned contributions are excluded from aggregation. In the best-performing configuration, an average of 3.12 clients per round were discarded, resulting in a total of 212 discarded clients across 68 rounds.

Compared to FedAvg, which requires all clients to participate in every round, and FedProx, which introduces additional computational costs on the client side, the proposed approach demonstrates a more targeted and efficient use of communication resources. This targeted approach avoids unnecessary computational burdens on clients, making it suitable for resource-constrained environments. The rapid convergence of the proposed approach is evident from its ability to achieve 96.8% accuracy in only 68 rounds. In comparison:

- FedAvg: Requires 80 rounds to reach 96.5% accuracy under similar conditions.
- FedProx: Achieves 96.3% accuracy in approximately 90 rounds but incurs additional client-side computation due to the proximal term.

This improvement in convergence speed is attributed to the selective participation mechanism, which ensures that only clients with significant contributions are included in the aggregation. The proposed method accelerates the learning process and stabilizes global model updates by eliminating noise introduced by low-contribution clients.

4.2.2. Accuracy Analysis

The accuracy achieved by the proposed approach is consistent with, or slightly superior to, existing methods while requiring fewer communication rounds. For example, while FedProx achieves a comparable accuracy of 96.3% in 90 rounds, the proposed method reaches 96.8% in only 68 rounds. This performance gain demonstrates the proposed client selection mechanism's ability to enhance model quality by focusing on high-impact updates.

Moreover, the results highlight the robustness of the proposed approach in handling non-IID data distributions. The combination of normalized L2 norm filtering and cosine similarity ensures that updates are significant in magnitude and well-aligned with the global model's objectives, mitigating the adverse effects of data heterogeneity.

4.2.3. Security Analysis Against Adversarial Clients

A key advantage of the combined filtering mechanism, which utilizes both cosine similarity and L2 norm, is its ability to enhance security by identifying and excluding potential intrusions. Specifically, when a client's L2 norm is high, but its cosine similarity with the global model is low, this indicates that the client's updates, despite their magnitude, are misaligned with the global model's objectives. Misalignment could result from adversarial behavior, client malfunctions, or data inconsistencies.

To validate these theoretical claims, we conducted experiments to assess the robustness of the proposed CLF-C mechanism against adversarial clients. We simulated data poisoning attacks by injecting controlled noise into a subset of clients, following established benchmarks for Byzantine attacks in FL [28, 29]. This approach is widely adopted in the literature to evaluate the robustness of FL systems [30, 31]. The attack simulation was conducted by selecting 20% of the clients in each round as adversaries and perturbing their model updates with zero-mean Gaussian noise:

$$\tilde{w}_i = w_i + N(0, \sigma^2), \sigma = 0.5 \quad (9)$$

where:

- w_i : original parameters from client i .
- $\tilde{w}_i$: perturbed (poisoned) parameters.
- $N(0, \sigma^2)$: Gaussian noise with $\sigma = 0.5$, a standard deviation empirically validated in FL robustness studies [28, 30, 31].

This noise magnitude represents a realistic threat scenario that is neither too weak to be ineffective nor too strong to be easily detectable, making it suitable for evaluating the proposed defense mechanism [30].

The detection results showed that the CLF-C mechanism successfully identified and excluded all adversarial clients throughout the FL process. This perfect detection rate (100%) was achieved because:

1. L2 Norm: Adversarial updates exhibited norms 2-3 $\times$ higher than benign clients, consistently exceeding the detection threshold.
2. Cosine Similarity: Adversarial updates showed severe misalignment with the global model's direction, with cosine similarity values dropping below 0.3.

These results demonstrate that the proposed filtering mechanism effectively distinguishes between legitimate updates and adversarial perturbations, thereby maintaining the integrity of the global model while preserving its performance.

4.2.4. Comparative Performance: FedAvg, FedProx, and the Proposed Approach

Figure 2 visually compares the accuracy curves for the proposed approach, FedAvg, and FedProx over 100 communication rounds, with detailed numerical results summarized in Table 2. Notable observations from this comparison include:

The proposed approach achieves a steep increase in accuracy during the early rounds, reaching 96.8%

accuracy by round 68.

- FedAvg requires approximately 80 rounds to achieve comparable accuracy, reflecting a slower convergence rate due to the inclusion of all clients regardless of their contribution.
- FedProx demonstrates marginally faster convergence than FedAvg but incurs additional computational overhead on the client side, which may not be feasible in resource-limited environments.

Additionally, FedProx's additional client-side computation stems from including a proximal term in the local objective function. This proximal term penalizes deviations of local parameters (ω) from the global model (ω_g), ensuring stability in non-IID settings. However, it introduces:

1. Extra Gradient Computation: Clients must compute the gradient of the proximal term and add it to the gradient during backpropagation.
2. Memory Overhead: Clients must store a copy of the global model to calculate this term.
3. Computational Impact: Although relatively lightweight for simple models, the additional steps may strain resource-constrained clients, especially in large-scale settings.

In contrast, the proposed approach does not impose additional computational requirements on the client side. This makes it more practical for environments where minimizing client-side resource usage is critical. The results highlight the dual advantage of the proposed approach, which achieves high accuracy with fewer communication rounds and reduces client computational requirements.

Table 2. The comparative results between the proposed approaches and other approaches

Approaches	No. of rounds	Accuracy	Total participating clients
Proposed	68	96.8%	468
FedAvg	80	96.5%	800
FedProx	90	96.3%	900

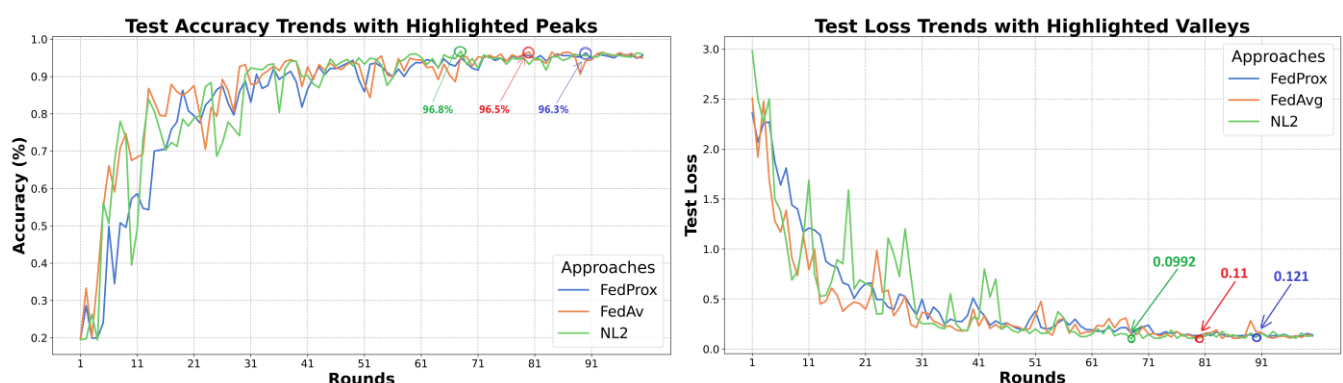


Figure 2. Accuracy comparison over 100 communication rounds for the proposed approach, FedAvg, and FedProx. Highlighted is the proposed method, achieving 96.8% accuracy in 68 rounds while discarding 212 clients

4.2.5. Threshold Analysis

The thresholds used in the proposed approach significantly impact performance, particularly in terms

of accuracy, convergence speed, and communication efficiency. Here, we analyze the effects of the Normalized L2, L2 norm, and Combined thresholds (cosine similarity and L2 norm), focusing on identifying ranges where accuracy begins to degrade. Additionally, we examine how these thresholds interact and influence each other based on the data in Table 1.

Normalized L2 Threshold:

- **Optimal Range:** The best results are achieved when the Normalized L2 threshold is between 0.2 and 0.25. This range enables faster convergence (e.g., 68 rounds) while maintaining high accuracy (96.8%).
- **Degradation Range:** When the threshold exceeds 0.3, accuracy begins to decline, as observed in configurations with thresholds such as 0.35, where accuracy stabilizes, but communication efficiency worsens, resulting in more discarded clients. Similarly, thresholds below 0.2 include too many low-contribution clients, resulting in increased communication overhead without notable accuracy gains.

L2 Norm Threshold:

- **Optimal Range:** The optimal L2 norm threshold is approximately 0.4, which strikes a balance between communication efficiency and model accuracy. For instance, a threshold of 0.4 achieves a high accuracy of 96.8% with relatively few discarded clients (e.g., 212).
- **Degradation Range:** When the L2 norm threshold falls below 0.3, the system becomes overly permissive, leading to slower convergence and degraded accuracy. Conversely, thresholds above 0.4 discard too many clients, reducing update diversity and impairing model quality.

Combined Thresholds (Cosine Similarity and L2 Norm):

- **Optimal Range:** The combination (Cosine = 0.98, L2 = 0.5) yields the most consistent balance, achieving 96.8% accuracy in 68 rounds. This configuration effectively selects high-contribution, well-aligned updates.
- **Degradation Range:** When the cosine similarity threshold drops below 0.95, accuracy declines as poorly aligned updates are included. Conversely, increasing the L2 threshold beyond 0.5 discards too many clients, reducing the number of meaningful updates.

It is important to highlight that the selected threshold values fall within specific effective ranges identified through extensive experimentation. For the Normalized L2 and L2 norm, the tested values span from 0.2 to 0.5, and for cosine similarity, 95% to 98%. Performance noticeably degraded when thresholds were set outside of these ranges. As shown in Table 1, the most effective configurations are located near the center of these ranges, indicating a performance peak and validating the chosen intervals.

However, these threshold values are not expected to be universally applicable. The current evaluation was conducted on the MNIST dataset, which contains only 10 classes and relatively low-dimensional input data. Threshold effectiveness inherently depends on the dataset's characteristics, including data distribution,

class balance, feature dimensionality, and model complexity. As such, these thresholds should be considered dataset-specific. Generalizing them across datasets may lead to suboptimal results, particularly when applied to more complex datasets with higher class counts or non-uniform distributions.

Future work will evaluate the proposed method using diverse datasets and explore adaptive threshold selection strategies. This could involve dataset-aware heuristics or dynamic threshold tuning mechanisms that adjust in real-time based on update statistics, model behavior, or validation feedback.

4.2.6. Theoretical Evaluation of the Combination Strategy

The effectiveness of the proposed method stems from its integrated use of Cosine Similarity and Normalized L2 Norm, which jointly evaluate the direction and magnitude of client updates. This combination addresses the core challenge in FL: selecting updates that meaningfully contribute to global model improvement, particularly under non-IID conditions.

Cosine similarity is defined as $\cos(\theta) = \frac{\Delta w_i \cdot \Delta w_g}{\|\Delta w_i\| \|\Delta w_g\|}$, measures how well a client's update aligns with the global model's direction. However, it is insensitive to the scale of the update. An update with perfect alignment but negligible magnitude ($\|\Delta w_i\| \approx 0$) contributes little to convergence.

Normalized L2 Norm, computed as shown in Eq. 1, quantifies the relative strength of updates and prevents domination by outliers. On its own, however, it cannot distinguish between helpful and harmful large updates. By combining these two metrics, the proposed method effectively:

- Eliminates low-impact updates, either small in magnitude or poorly aligned.
- Blocks harmful updates, those that are large in size but misaligned, which may arise from adversarial or noisy clients.

This combined filtering approach provides clear advantages over existing techniques:

1. FedAvg applies uniform aggregation across all participating clients, treating all updates equally regardless of their quality. This allows weak or noisy updates to dilute the global model, especially under heterogeneous data. As a result, convergence is slower, and communication overhead remains high due to unnecessary client participation.
2. FedProx, in contrast, modifies the local training objective by adding a proximal term that penalizes deviation from the global model. This encourages local updates to remain close to the global parameters, providing implicit alignment, especially in non-IID settings. However, FedProx does not evaluate the updates explicitly based on magnitude or direction. As a result, it may still include updates that contribute little to learning or even degrade performance. For instance, a client may produce a negligible update ($\|\Delta w\| \approx 0$) that is well-aligned but ineffective. Alternatively, a client may generate a large update ($\|\Delta w\| \gg 0$) with poor alignment ($\cos(\theta) \ll 1$), potentially harming convergence. Since FedProx does not filter updates based on these measurable criteria, it remains vulnerable to low-value and disruptive contributions.

In contrast, the proposed method explicitly selects clients whose updates are substantial and directionally aligned with the objective of the global model. This leads to faster convergence, improved resistance to noisy or adversarial updates, and a significant reduction in communication overhead, as demonstrated in the evaluation results throughout Section 4.2.

4.2.7. Comparative Analysis of State-of-the-Art FL Approaches

To contextualize the effectiveness of the proposed approach, a comparative analysis is provided against several leading methods in FL that focus on communication efficiency and client selection. Table 3 outlines the performance of each method in terms of communication reduction, impact on model accuracy, client-side computational overhead, a brief functional description, and whether the method incorporates security enhancements.

Table 3. Comparative Analysis of Communication-Efficient FL Methods with Client Selection

Method	Communication Reduction	Accuracy Impact	Client-side Computation	Short Description	Security Enhancement
Adaptive-OU [32]	Up to 50%	Comparable or better than FedAvg	Moderate (norm calculation & filtering)	Selects informative updates and estimates the rest using an Ornstein–Uhlenbeck (OU) process.	No
HACCS [33]	18–38% reduction in convergence time	Maintains accuracy	Yes (histogram, clustering, noise)	It uses privacy-aware clustering based on histogram representations of data.	Partial
FEDGS [34]	~59% fewer training rounds	+3.5% over FedAvg	None (uses server-side GBP-CS)	Gradient-based permutation clusters clients into stable supernodes for edge-side aggregation.	Partial
CFL with Scheduling [35]	Up to 50% faster convergence	Comparable or better	Minimal (client latency reporting)	Schedules clients in cluster-based FL using latency-aware orchestration.	No
AoU-based FL [36]	~35% reduction in rounds and latency	+0.1–0.15 (non-IID MNIST)	None (server-side ANN inference)	Combines Age-of-Update and NOMA uplink with server-side prediction of missing updates.	Partial
E3CS [37]	Up to 2× faster convergence	Maintains baseline accuracy	None (Exp3-based stochastic selection)	Adapts client selection under volatile client conditions using multi-armed bandit logic.	No
Proposed Method (Ours)	~24.4% fewer rounds and 48% fewer clients	+0.5% over FedAvg	None	Uses normalized L2 norm and cosine similarity filtering to select impactful, well-aligned clients.	Yes

Regarding communication efficiency, while methods like FEDGS and E3CS demonstrate significant reductions in communication rounds, they often rely on additional assumptions, such as hierarchical infrastructure (an Additional server) or probabilistic modeling. In contrast, our method achieves meaningful communication savings through a deterministic filtering process that does not require added architectural complexity or randomness, making it suitable for more general deployment scenarios.

The proposed method is lightweight when evaluating client-side computational demands, requiring no additional processing on client devices. This contrasts with strategies like Adaptive-OU and HACCS, which require local computations or privacy-preserving transformations. By offloading all selection logic

to the server, our approach remains efficient for use in constrained or mobile client environments.

The proposed method also contributes to model robustness by identifying and excluding either misaligned or substantial updates, which can indicate adversarial or noisy behavior. Combining magnitude (L2 norm) and directional alignment (cosine similarity) introduces a lightweight safeguard that enhances model integrity without additional cost. This added layer of reliability is particularly valuable in sensitive or data-critical applications.

5. Limitations and Future Work

While the proposed approach demonstrates significant improvements in communication efficiency and model performance, there are areas for potential enhancement:

1. **Adaptive and Multi-Objective Threshold Optimization:** The current implementation relies on predefined thresholds for normalized L2 norm and cosine similarity, which were empirically tuned based on the MNIST dataset. Future work will investigate adaptive thresholding strategies that adjust automatically during training based on client update statistics, such as mean and standard deviation of norm values or alignment trends over rounds. One potential direction is to utilize validation loss feedback to adjust thresholds for improved trade-offs between convergence and participation dynamically.

Additionally, multi-objective optimization methods, such as evolutionary algorithms or Pareto front analysis, can jointly optimize communication overhead, model accuracy, and robustness. These strategies would allow the threshold selection process to become self-tuning, making the approach more practical for deployment in dynamic and heterogeneous environments.

2. **Scalability to Larger Models and Datasets:** While the experiments were conducted on the MNIST dataset, future studies could evaluate the scalability of the proposed method to larger, more complex datasets and models.

In particular, deep learning architectures and high-dimensional inputs (e.g., in datasets like CIFAR-10, ImageNet, or sensor-rich time series) can introduce substantial variance in client updates. This variance may challenge the stability of both L2 norm and cosine similarity calculations, especially as model parameters increase. Future work will explore techniques such as layer-wise contribution analysis and low-rank projections to reduce computational costs while maintaining effectiveness in such settings.

3. **Generalization of Threshold Ranges:** The current threshold ranges were optimized for MNIST, which is relatively simple in terms of input dimensionality and class count. Thresholds may need to be re-tuned for more complex or domain-specific datasets. Future work will include cross-dataset evaluations to determine how sensitive the filtering performance is to data complexity and to explore whether meta-learning techniques or dataset-specific heuristics can support generalizable threshold strategies.
4. **Robustness to Adversarial Clients:** Additional mechanisms could be integrated to detect and mitigate adversarial behaviors, further enhancing the robustness of the proposed approach.
5. **Real-World Deployment:** The method will be extended to real-world domains such as healthcare and

finance, where privacy, data heterogeneity, and communication efficiency are critical. To validate its practical applicability, future evaluations will include domain-specific datasets (e.g., MIMIC-III [38] and financial transaction data).

6. Conclusions

This study addressed the critical challenge of communication efficiency in Federated Learning (FL) by introducing a novel client contribution selection mechanism based on Normalized L2 Norm and Cosine Similarity Filtering. The proposed approach effectively reduces unnecessary client participation by selecting significant and well-aligned updates with the global model, thereby minimizing communication overhead while maintaining high model performance. Experimental results on the MNIST dataset demonstrated that the proposed method achieves 96.8% accuracy in just 68 communication rounds, requiring only 468 participating clients, compared to FedAvg (96.5% in 80 rounds, 800 clients) and FedProx (96.3% in 90 rounds, 900 clients). These results highlight the method's effectiveness in accelerating convergence and reducing communication costs, making it well-suited for large-scale, decentralized FL applications. Beyond communication efficiency, the proposed method enhances model security by detecting and excluding adversarial or misaligned clients, thus preserving the integrity of the global model. Importantly, this is achieved without imposing additional computational overhead on clients, making the approach scalable and practical for real-world FL scenarios with resource constraints.

Future research will focus on adaptive threshold selection to optimize client filtering based on real-time model updates. Additionally, the method will be extended to larger and more complex datasets, including real-world applications in healthcare and finance. Further investigations into multi-objective optimization techniques could refine the balance between communication efficiency and model performance, enhancing the robustness of FL in highly heterogeneous environments.

Acknowledgements: We sincerely thank the University of Human Development, Kurdistan Region, Iraq, for providing access to a high-performance computer server, which facilitated the implementation and testing of the proposed approaches.

Funding: The authors declare that no funding was received for this research.

Declaration of Competing Interest: The authors declare they have no known competing interests.

Research Involving Human/Animal Participants: This article does not involve any studies conducted by authors on animals or human participants.

References

- [1] G. S. Nariman and H. K. Hamarashid, "Hierarchical federated learning for health trend prediction and anomaly detection using pharmacy data: from zone to national scale," *International Journal of Data Science and Analytics*, pp:1-20, 2025, doi: 10.1007/s41060-025-00756-5.

- [2] S. Jiang, Y. Li, F. Firouzi, and K. Chakrabarty, "Federated clustered multi-domain learning for health monitoring," *Scientific Reports*, vol. 14, no. 1, p. 903, 2024, doi: 10.1038/s41598-024-51344-9.
- [3] M. Asad and S. Otoum, "Bppfl: a blockchain-based framework for privacy-preserving federated learning," *Cluster Computing*, vol. 28, no. 2, p. 126, 2024, doi: 10.1007/s10586-024-04834-4.
- [4] C. Ma, X. Li, B. Huang, G. Li, and F. Li, "Personalized client-edge-cloud hierarchical federated learning in mobile edge computing," *Journal of Cloud Computing*, vol. 13, no. 1, p. 161, 2024, doi: 10.1186/s13677-024-00721-w.
- [5] J. Li, M. Lu, J. Zhang, and J. Wu, "Aldp-fl: an adaptive local differential privacy-based federated learning mechanism for IoT," *International Journal of Information Security*, vol. 24, no. 1, p. 23, 2024, doi: 10.1007/s10207-024-00933-w.
- [6] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 2, pp. 513-535, 2023/02/01 2023, doi: 10.1007/s13042-022-01647-y.
- [7] S. Ji *et al.*, "Emerging trends in federated learning: from model fusion to federated X learning," *International Journal of Machine Learning and Cybernetics*, vol. 15, no. 9, pp. 3769-3790, 2024, doi: 10.1007/s13042-024-02119-1.
- [8] J. Xu, X. Li, K. Zhu, L. Zhou, and Y. Zhao, "Two-stage model fusion scheme based on knowledge distillation for stragglers in federated learning," *International Journal of Machine Learning and Cybernetics*, vol. 16, pp. 3067–3083 2024, doi: 10.1007/s13042-024-02436-5.
- [9] S. D N, A. B, and S. Ambesange, "Smart client selection strategies for enhanced federated learning in digital healthcare applications," *Multimedia Tools and Applications*, pp:1-16, 2024, doi: 10.1007/s11042-024-19403-5.
- [10] Y. Liu, H. Xia, W. Li, and T. Niu, "Mitigating bias in heterogeneous federated learning via stratified client selection," *Peer-to-Peer Networking and Applications*, vol. 18, no. 2, p. 11, 2025, doi: 10.1007/s12083-024-01886-6.
- [11] L. Hu, Y. Hu, L. Jiang, and W. Long, "Federated learning client selection algorithm based on gradient similarity," *Cluster Computing*, vol. 28, no. 2, p. 134, 2024, doi: 10.1007/s10586-024-04846-0.
- [12] G. S. Nariman and H. K. Hamarashid, "Communication overhead reduction in federated learning: a review," *International Journal of Data Science and Analytics*, vol. 19, pp: 185–216, 2024, doi: 10.1007/s41060-024-00691-x.
- [13] B. Soudan, S. Abbas, A. Kubba, O. Abu Waraga, M. Abu Talib, and Q. Nasir, "Scalability and performance evaluation of federated learning frameworks: a comparative analysis," *International Journal of Machine Learning and Cybernetics*, vol. 16, pp: 3329–3343 2025, doi: 10.1007/s13042-024-02453-4.
- [14] H. Wu and P. Wang, "Node Selection Toward Faster Convergence for Federated Learning on Non-IID Data," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3099-3111, 2022, doi: 10.1109/TNSE.2022.3146399.
- [15] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.
- [16] M. Duan *et al.*, "FedGroup: Efficient Federated Learning via Decomposed Similarity-Based Clustering," in *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom)*, 30 Sept.-3 Oct. 2021 2021, pp. 228-237, doi: 10.1109/ISPA-BDCLOUD-SocialCom-SustainCom52081.2021.00042.
- [17] L. Yu, W. Nie, L. Xin, and M. Guo, "Clustered federated learning based on data distribution," in *Proceedings of the 3rd International Conference on Advanced Information Science and System*, 2021, pp. 1-5.
- [18] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.
- [19] A. Cheng, P. Wang, X. S. Zhang, and J. Cheng, "Differentially private federated learning with local regularization and sparsification," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10122-10131.
- [20] P. Uthansakul, P. Anchuen, M. Uthansakul, and A. A. Khan, "Estimating and Synthesizing QoE Based on QoS

- Measurement for Improving Multimedia Services on Cellular Networks Using ANN Method," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 389-402, 2020, doi: 10.1109/TNSM.2019.2946091.
- [21] P. Uthansakul, P. Anchuen, M. Uthansakul, and A. A. Khan, "QoE-aware self-tuning of service priority factor for resource allocation optimization in LTE networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 887-900, 2019.
- [22] A. A. Khan, M. Driss, W. Boulila, G. A. Sampedro, S. Abbas, and C. Wechtaisong, "Privacy Preserved and Decentralized Smartphone Recommendation System," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 4617-4624, 2024, doi: 10.1109/TCE.2023.3323406.
- [23] A. I. Kawoosa, D. Prashar, M. Faheem, N. Jha, and A. A. Khan, "Using machine learning ensemble method for detection of energy theft in smart meters," *IET Generation, Transmission & Distribution*, vol. 17, no. 21, pp. 4794-4809, 2023.
- [24] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141-142, 2012.
- [25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017: PMLR, pp. 1273-1282.
- [26] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020: IEEE, pp. 1-9.
- [27] Goran1984. "Enhancing-Communication-Efficiency-in-FL." GitHub. <https://github.com/Goran1984/Enhancing-Communication-Efficiency-in-FL/blob/dc90beb41adb46c78b14e9a18cbca08eec9dacc9/Main-1> (accessed Feb 28, 2025, 2025).
- [28] M. Fang, X. Cao, J. Jia, and N. Gong, "Local model poisoning attacks to {Byzantine-Robust} federated learning," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1605-1622.
- [29] L. Lyu *et al.*, "Privacy and Robustness in Federated Learning: Attacks and Defenses," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 7, pp. 8726-8746, 2024, doi: 10.1109/TNNLS.2022.3216981.
- [30] G. Baruch, M. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," *Advances in Neural Information Processing Systems*, vol. 32, pp.1-11, 2019.
- [31] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*, 2020: PMLR, pp. 2938-2948.
- [32] M. Ribero and H. Vikalo, "Reducing communication in federated learning via efficient client sampling," *Pattern Recognition*, vol. 148, p. 110122, 2024.
- [33] J. Wolfrath, N. Sreekumar, D. Kumar, Y. Wang, and A. Chandra, "Haccs: Heterogeneity-aware clustered client selection for accelerated federated learning," in *2022 IEEE international parallel and distributed processing symposium (IPDPS)*, 2022: IEEE, pp. 985-995.
- [34] Z. Li *et al.*, "Data heterogeneity-robust federated learning via group client selection in industrial IoT," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17844-17857, 2022.
- [35] A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Erbad, "Client selection approach in support of clustered federated learning over wireless edge networks," in *2021 IEEE global communications conference (GLOBECOM)*, 2021: IEEE, pp. 1-6.
- [36] B. Wu, F. Fang, and X. Wang, "Joint Age-Based Client Selection and Resource Allocation for Communication-Efficient Federated Learning Over NOMA Networks," *IEEE Transactions on Communications*, vol. 72, no. 1, pp. 179-192, 2024, doi: 10.1109/TCOMM.2023.3317300.
- [37] T. Huang, W. Lin, L. Shen, K. Li, and A. Y. Zomaya, "Stochastic Client Selection for Federated Learning With Volatile Clients," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20055-20070, 2022, doi: 10.1109/JIOT.2022.3172113.
- [38] A. E. W. Johnson *et al.*, "MIMIC-III, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, p. 160035, 2016/05/24 2016, doi: 10.1038/sdata.2016.35.

