






Please cite the Published Version

El-Niss, Ayoub , Alzu'Bi, Ahmad , Abuarqoub, Abdelrahman , Hammoudeh, Mohammad  and Muthanna, Ammar  (2024) SimProx: A Similarity-Based Aggregation in Federated Learning With Client Weight Optimization. IEEE Open Journal of the Communications Society, 5. pp. 7806-7817. ISSN 2644-125X

DOI: <https://doi.org/10.1109/ojcoms.2024.3513816>

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Version: Published Version

Downloaded from: <https://e-space.mmu.ac.uk/637728/>

Usage rights:  [Creative Commons: Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)

Additional Information: This is an open access article which first appeared in IEEE Open Journal of the Communications Society

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

SimProx: A Similarity-Based Aggregation in Federated Learning With Client Weight Optimization

AYOUB EL-NISS¹, AHMAD ALZU'BI¹ (Senior Member, IEEE),
ABDELRAHMAN ABUARQOUB² (Member, IEEE), MOHAMMAD HAMMOUDEH³ (Senior Member, IEEE),
AND AMMAR MUTHANNA⁴ (Senior Member, IEEE)

¹Department of Computer Science, Jordan University of Science and Technology, Irbid 22110, Jordan

²Department of Applied Computing, Cardiff School of Technologies, CF5 2YB Cardiff, U.K.

³Department of Computing and Mathematics, Manchester Metropolitan University, M1 5GD Manchester, U.K.

⁴Department of Telecommunication Networks and Data Transmission, St. Petersburg State University of Telecommunication, 190013 Saint Petersburg, Russia

CORRESPONDING AUTHOR: A. ALZU'BI (e-mail: agalzubi@just.edu.jo)

This work was supported by the St. Petersburg State University of Telecommunications through the Ministry of Science and High Education of the Russian Federation under Grant 075-15-2022-1137.

ABSTRACT Federated Learning (FL) enables decentralized training of machine learning models across multiple clients, preserving data privacy by aggregating locally trained models without sharing raw data. Traditional aggregation methods, such as Federated Averaging (FedAvg), often assume uniform client contributions, leading to suboptimal global models in heterogeneous data environments. This article introduces SimProx, a novel FL approach for aggregation that addresses heterogeneity in data through three key improvements. First, SimProx employs a composite similarity-based weighting mechanism, integrating cosine and Gaussian similarity measures to dynamically optimize client contributions. Then, it incorporates a proximal term in the client weighting scheme, using gradient norms to prioritize updates closer to the global optimum, thereby enhancing model convergence and robustness. Finally, a dynamic parameter learning technique is introduced, which adapts the balance between similarity measures based on data heterogeneity, refining the aggregation process. Extensive experiments on standard benchmarking datasets and real-world multimodal data demonstrate that SimProx significantly outperforms traditional methods like FedAvg in terms of accuracy. SimProx offers a scalable and effective solution for decentralized deep learning in diverse and heterogeneous environments.

INDEX TERMS Federated learning, decentralized network, weighted aggregation, data heterogeneity, deep learning, multimodal classification.

I. INTRODUCTION

FEDERATED Learning (FL) [1] has recently gained significant attention since it attempts to train Machine Learning (ML) models with a decentralized approach, addressing major concerns with data security and privacy. By enabling multiple clients to collaboratively contribute to training ML models without sharing their raw data, FL maintains data confidentiality while utilizing the collective power of distributed datasets. Therefore, federated learning plays an increasingly important role in maintaining privacy and attaining good model performance results as the amount of data generated globally keeps growing [2], [3].

While FL offers many advantages, it faces considerable challenges in building a generalized and robust global model from locally trained models. The essence of FL lies in aggregating local updates from distributed clients to create a unified global model without sharing raw data. The more realistic challenging scenario of heterogeneous FL remains a key challenge despite the tremendous progress made in homogeneous FL, where clients share similar network architectures and analyze similar data distributions [4], [5]. In most large-scale, real-world applications, clients exhibit considerable differences in data distributions, communication networks, and model structures, leading to four primary types of heterogeneity.

Firstly, statistical heterogeneity [6] arises from varying data distributions across clients participating in the FL process. Secondly, model heterogeneity [7] occurs when different clients employ distinct models, which may differ in architecture, size, customizations, learning rates, and hyperparameters. Researchers have attempted to address this issue using various approaches, including model mapping [8], knowledge distillation [9], ensemble learning [10], and meta learning [11]. Thirdly, communication heterogeneity [12] emerges when clients operate under diverse network environments. Lastly, device heterogeneity [13] arises from the varying storage and computation capabilities of devices among participants, which can lead to faults and inactivation of some nodes, known as stragglers [14], [15]. To mitigate these challenges, several methods have been developed, including asynchronous aggregation [16], [17], which enables clients to upload their local updates in a staggered manner.

Many studies have concentrated on FL aggregating techniques that mitigate the impact of statistical heterogeneity on building a global model on the server side [18], [19], [20], starting with FedAvg [21] where a group of clients is randomly selected at each round of training for aggregation. During the aggregation process, the parameters of each client are weighted and averaged to produce a global model. Despite that FedAvg can handle Non-Independent and Identically Distributed (Non-IID) [22] data to some extent, many research investigations [23], [24] have demonstrated that a degradation in the accuracy of FL is practically certain when dealing with Non-IID or heterogeneous data.

The non-IID issue arises in FL because the data distribution across clients often does not adhere to the Independent and Identically Distributed (IID) assumption, which is a cornerstone of many traditional machine learning techniques. This statistical heterogeneity can manifest in various forms, such as differences in data quantity, class distribution, or feature space across clients. Such disparities lead to challenges in achieving a globally optimal model, as the gradients contributed by clients may be biased or conflicting. This imbalance hampers the convergence of the aggregation process, resulting in suboptimal performance and accuracy degradation in the global model. Addressing these issues is critical to ensure that FL can deliver reliable and robust outcomes in real-world applications where data heterogeneity is the norm.

This paper addresses the non-IID issues from a novel perspective based on the following observation: *if two models training on different (skewed) datasets have learned the same information, then this particular information is crucial for producing a generalized model.* Consequently, we propose a new aggregation technique based on the similarity information of client models. Unlike existing approaches that utilize similarity for clustering, the proposed technique, Similarity-Proximal (SimProx), employs a similarity matrix to calculate weights for clients and performs weighted sum aggregation. It incorporates a proximal term that considers

the gradient norms as a proxy for the client's contribution to the global model update. This term indicates how much the client model has changed during the current round of FL. The specific contributions of this work are as follows:

- 1) Propose a method to calculate the similarity matrix using both Gaussian and cosine similarity, which helps the global model to learn from the most relevant and complementary client updates.
- 2) Include a proximal term into the deep FL aggregation to complement the client similarity information captured by the similarity matrix, stimulating the global model to learn more from the client updates that are closer to the global optimum. Additionally, we apply a dynamic calculation of the lambda learning parameter based on the heterogeneity of learnt data.
- 3) Evaluate the performance of the proposed method on real-world multimodal data, textual and visual representation, and two versions of CIFAR testbed.

The remaining part of this paper is structured as follows: Section II introduces the general task of FL and then reviews the related work regarding similarity-based aggregation in heterogeneous settings. Section III presents the methodology adopted in this study. Section IV presents the experimental setups and discusses the model's results in the context of previous approaches using image, textual, and multimodal data. Section V discusses the computational complexity of the proposed method. Finally, Section VI concludes the paper and highlights the main finding.

II. RELATED WORK

This section reviews various existing aggregation approaches related to the proposed work in federated deep learning. To facilitate a thorough understanding of these approaches, we first establish a common task notation with necessary preliminaries.

A. FEDERATED LEARNING PRELIMINARIES

Federated learning is a distributed machine learning approach that enables model training across multiple decentralized devices or servers holding local data samples, without exchanging their data. Unlike traditional centralized machine learning methods that require pooling data to a central server, FL maintains data privacy and security by keeping data on local devices.

Given the list of preliminaries shown in Table 1, consider a FL system with C clients, each holding a local dataset \mathcal{D}_i where $i \in \{1, 2, \dots, C\}$. The objective is to train a global model w by aggregating local models w_i trained on the local datasets \mathcal{D}_i . In order to achieve the primary objectives of FL, the following factors need to be maintained:

- 1) *Data Privacy*: Preserve the privacy of local data by ensuring that raw data remains on the client's device.
- 2) *Communication Efficiency*: Minimize the communication overhead between clients and the central server to make the system scalable and efficient.

TABLE 1. FL task preliminaries.

Notation	Description
C	number of participating clients
C_i	i -th client
D_i	private dataset of client C_i
N_i	number of samples in dataset D_i
N	total number of samples across all clients
w_i	model weights of client i
$f(w_i)$	learned local network model of client C_i
$w_{central}$	centralized model parameters
t	epoch number in the FL process
w_t	global model parameters at epoch t
d_{ij}	Euclidean distance between w_i and w_j
σ	average distance between all clients
s_{ij}	cosine similarity between w_i and w_j
s_g	Gaussian similarity between w_i and w_j
λ	hyperparameter controlling trade-off between s_{ij} and s_g
S	client similarity matrix
α_i	weight of client i

- 3) *Robustness to Heterogeneity*: Handle non-IID data distributions across clients, ensuring that the global model generalizes well despite variations in local data.
- 4) *Decentralized Model Training*: Enable collaborative model training across multiple clients without the need for centralized data aggregation.

Improving aggregation methods is crucial for FL, especially when dealing with non-IID data. In FL, data is distributed across multiple clients, often with significant differences in data distribution and quality. Traditional aggregation methods, such as simple averaging, may not adequately address these disparities, leading to suboptimal model performance [23]. Non-IID data poses additional challenges because the variations in data can cause local models to diverge significantly, complicating the aggregation process [7]. Advanced aggregation methods that consider the underlying data distribution and client similarities can help mitigate these issues, ensuring more robust and accurate global models. By developing and implementing more sophisticated aggregation techniques, FL systems can achieve better generalization, resilience to client variability, and ultimately, more effective and reliable AI solutions in decentralized environments [13], [22].

Several research studies have attempted to improve data aggregation from the client's side, particularly for non-IID data, which typically focus on either local training or the aggregation process. However, Our study's objective is to improve the aggregation process, offering a more adaptive and robust solution to the challenges of FL in diverse real-world settings.

B. RELATED AGGREGATION ALGORITHMS

FedAvg [21] is a fundamental approach in FL, where model updates are aggregated across clients through a weighted average. FedAvg framework involves four key steps in each iteration: the server distributes the global model w_t to all participating clients, each client i updates their local model w_i by performing Stochastic Gradient Descent (SGD) on their local data \mathcal{D}_i , the clients send their updated local models $w_i^{(t+1)}$ back to the server, and the server aggregates the local models by averaging them to produce the new global model, which is defined as follows:

$$w^{t+1} = \frac{1}{C} \sum_{i=1}^C w_i^{(t+1)} \quad (1)$$

where C is the number of clients.

However, FedAvg assumes that the data distributions across clients are identical and independent, which is often not the case in real-world scenarios [25]. To address this limitation, FedProx [26] extends FedAvg by introducing a proximal term to handle heterogeneous data. The objective function for each client includes a proximal term to maintain proximity to the global model:

$$\mathcal{L}_i^{\text{FedProx}}(w) = \mathcal{L}_i(w) + \frac{\mu}{2} |w - w_t|^2 \quad (2)$$

where μ is a regularization parameter. This approach has been shown to improve the robustness of FL systems in the presence of non-IID data.

FedAtt [27] introduces an attention mechanism applied to model parameters to improve aggregation. The global model is optimized by minimizing the weighted distance between it and client models, using attention scores to weigh each client's contribution. FedAtt treats server model parameters as queries and client parameters as keys, calculating layer-wise attention scores as:

$$\alpha_k^l = \text{softmax}(s_k^l) = \text{softmax}(\|w^l - w_k^l\|_p) \quad (3)$$

where s_k^l is the similarity between the global model parameters w^l and the client model parameters w_k^l in the l -th layer, calculated using the norm p .

Other sophisticated similarity-based aggregation approaches were introduced. SimAgg was introduced by Khan et al. [28] to improve model aggregation by weighting client model contributions based on how similar they are to the global model. This method demonstrates significant improvements in settings with highly heterogeneous data. Moreover, Wu and Wang [29] proposed an adaptive weighting approach aimed at speeding up convergence in FL by dynamically adjusting the contribution of each client's update based on its relevance to the global model.

Modeling Overlapping Neighborhoods (MOON) [30] also focuses on better client aggregation strategies by considering overlapping neighborhoods. The loss function includes a contrastive loss term to improve model consistency across overlapping data distributions:

$$\mathcal{L}_i^{\text{MOON}}(w) = \mathcal{L}_i(w) + \gamma \sum_{j \in \mathcal{N}(i)} |w_i - w_j|^2 \quad (4)$$

where $\mathcal{N}(i)$ represents the neighborhood of client i and γ is a weighting parameter. This approach has been shown to improve the robustness of FL systems in scenarios with overlapping data distributions.

Clustered-based FL aggregation [31], [32] is another approach that involves grouping clients into clusters based on the similarity of their data distributions. Each cluster independently trains a local model, and the global model is formed by aggregating the cluster models:

$$w_{t+1}^{\text{cluster}} = \frac{1}{|C_k|} \sum_{i \in C_k} w_i^{(t+1)} \quad (5)$$

where C_k is the k -th cluster and $|C_k|$ is the number of clients in cluster k . Hierarchical FL [33] introduces an additional layer of aggregation, where local models are first aggregated at the cluster level and then at the global level:

$$w_{t+1} = \frac{1}{M} \sum_{k=1}^M w_{t+1}^{\text{cluster}_k} \quad (6)$$

where M is the number of clusters.

Our proposed method bridges the gap in existing FL approaches by introducing a novel similarity-based weighting mechanism designed to improve model aggregation. Unlike prior methods that predominantly rely on either cosine similarity, which focuses on the alignment of model directions, or Gaussian similarity, which depends on the Euclidean distance between client models, our approach leverages a combination of both. This dual-similarity calculation enhances the accuracy and robustness of the aggregation process. Furthermore, while many existing similarity-based methods modify the loss function or compute separate similarity scores for each layer of the model—introducing significant computational overhead on resource-constrained edge devices—our method shifts the computational load to the cloud. By assigning a single aggregated weight for each client, our approach achieves greater scalability and efficiency for real-world applications. The process involves calculating a unified similarity score using both cosine and Gaussian similarity, normalizing client weights, and performing similarity-weighted aggregation of client models. To further enhance the system’s robustness, particularly in scenarios with highly heterogeneous data, we incorporate a proximal term with dynamic parameter learning, ensuring improved adaptability and performance.

III. METHODOLOGY

In this work, we propose a novel FL approach that incorporates client similarity information to improve the overall model performance. Figure 1 shows the generic pipeline of the proposed similarity-based aggregation of FL results from the client side to the server side. A set of clients contribute to the learning procedure performed on the global model broadcasted by a Central server. The results and weights of each single client trained on its local data are aggregated

into a similarity matrix that indicates the similarity-weighted scores to understand the diversity of clients’ models.

A proximal term with dynamic parameter learning is also involved in the learning procedure, providing the final weighted sum of results aggregated from the clients in each training round. This process is performed over many rounds to improve the convergence and robustness of the FL paradigm. We first establish the baseline distance calculation between clients, followed by detailing each step of the proposed approach in the subsequent subsections.

Let $\mathcal{C} = 1, 2, \dots, m$ be the set of clients, and let $\mathbf{w}_i \in \mathbb{R}^d$ be the model weights of client i , where d is the dimensionality of the model. To quantify the overall similarity between clients, we define the average distance between clients as:

$$\sigma = \frac{1}{\binom{m}{2}} \sum_{i=1}^m \sum_{j=i+1}^m \|\mathbf{w}_i - \mathbf{w}_j\|_2 \quad (7)$$

where $\|\cdot\|_2$ denotes the Euclidean norm. This average distance metric captures the diversity of the client models, with a smaller value indicating that clients are more similar, and a larger value indicating that clients are more diverse.

A. CLIENT SIMILARITY MATRIX

The core of the proposed aggregation process relies on calculating the similarity between the clients’ results. This section explains how Gaussian and Cosine functions are used to determine the similarity matrix, incorporating a dynamically learned hyperparameter to manage the trade-off between similarity scores.

The rationale for employing similarity-based aggregation lies in its ability to handle data heterogeneity more effectively than simple averaging. In FL systems, clients operate on datasets that may differ significantly in terms of content, size, and distribution. By focusing on clients whose models exhibit greater similarity, SimProx optimizes the aggregation process by giving more weight to clients whose updates are most likely to contribute to a well-generalized global model. This reduces the noise introduced by dissimilar client updates and improves overall model robustness.

1) GAUSSIAN SIMILARITY

The Gaussian similarity term in the client similarity matrix \mathbf{S} is defined as:

$$s_{ij}^{\text{Gaussian}} = \exp\left(-\frac{\|\mathbf{w}_i - \mathbf{w}_j\|_2^2}{2\sigma^2}\right) \quad (8)$$

This term captures the Euclidean distance between the client model weights \mathbf{w}_i and \mathbf{w}_j , normalized by the average distance σ between all client models. The key intuition behind the Gaussian similarity is that clients with more similar model weights, as measured by the Euclidean distance, are likely to provide more relevant and complementary updates to the global model. The Gaussian function is used to map the Euclidean distances to similarity scores, with smaller distances resulting in higher similarity scores.

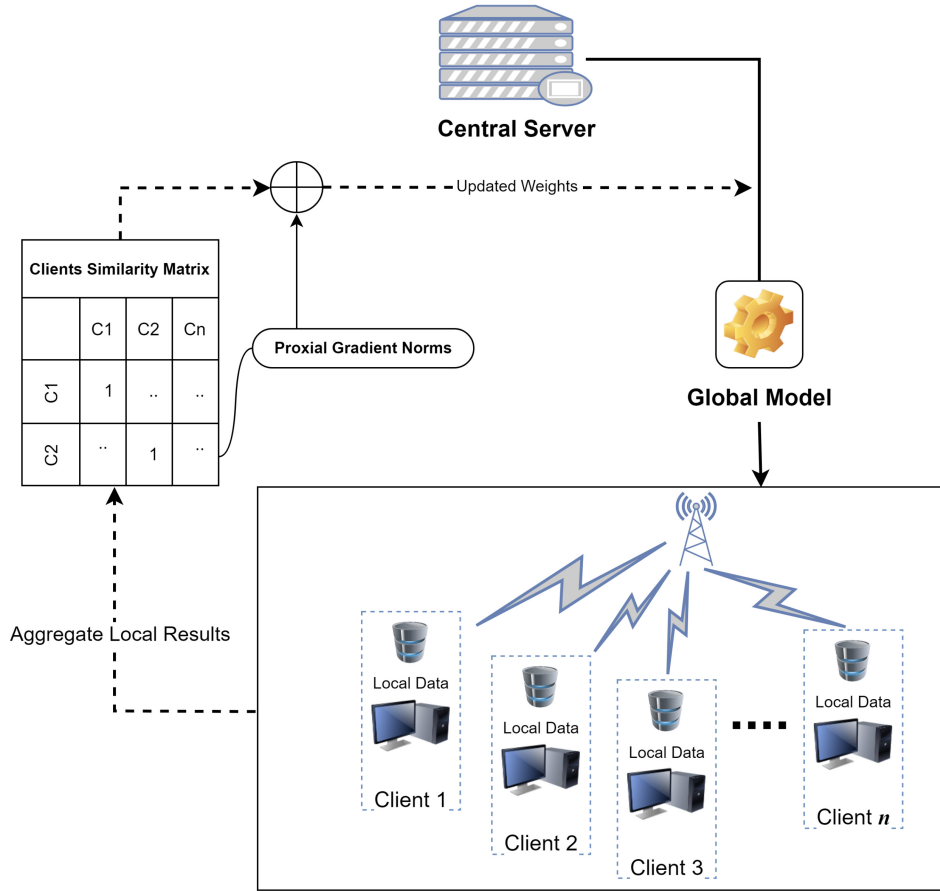


FIGURE 1. The general pipeline of the proposed similarity-based FL aggregation approach.

2) COSINE SIMILARITY

The cosine similarity term in the client similarity matrix \mathbf{S} is defined as:

$$S_{ij}^{\text{Cosine}} = \frac{\mathbf{w}_i^\top \mathbf{w}_j}{\|\mathbf{w}_i\|_2 \|\mathbf{w}_j\|_2} \quad (9)$$

This term captures the angular difference between the client model weights \mathbf{w}_i and \mathbf{w}_j , as measured by the cosine of the angle between them. The cosine similarity provides a complementary perspective to the Gaussian similarity. While the Gaussian similarity focuses on the Euclidean distance between the model weights, the cosine similarity captures the alignment or “direction” of the model weights.

Clients with more similar model directions, as indicated by a higher cosine similarity score, are likely to provide updates that are more aligned with the global model’s learning objective. These updates can help the global model converge faster and achieve better generalization performance. Also The cosine similarity is invariant to the scaling of the model weights, as it only depends on the relative direction of the vectors.

3) HYBRID SIMILARITY MEASURE

By combining the Gaussian similarity and the cosine similarity, the algorithm can capture both the Euclidean distance and

the angular difference between the client models, providing a more comprehensive assessment of client similarity. This hybrid similarity measure helps the global model learn from the most relevant and complementary client updates, leading to improved performance and faster convergence. We merge the Gaussian similarity and cosine similarity into a unified similarity matrix \mathbf{S} using a hyperparameter $\lambda \in [0, 1]$ to control the trade-off between the two similarity measures:

The use of both cosine and Gaussian similarity is grounded in the principle that alignment (captured by cosine similarity) and proximity (captured by Gaussian similarity) together provide a richer understanding of model behavior. Cosine similarity alone might fail to distinguish between models that are directionally aligned but distant in magnitude, while Gaussian similarity may overlook models that share a similar direction but differ slightly in magnitude. By combining the two, SimProx ensures that client models are weighted based on both their direction and magnitude, which leads to more accurate aggregation and faster convergence. This dual-similarity approach addresses the limitations of using a single metric in isolation, thus ensuring that SimProx learns from updates that are both highly relevant and balanced in their influence on the global model.

$$S_{ij} = \lambda S_{ij}^{\text{Cosine}} + (1 - \lambda) S_{ij}^{\text{Gaussian}} \quad (10)$$

We incorporate a dynamic lambda adjustment mechanism to adaptively control the balance between cosine and Gaussian similarity measures. The average cosine similarity between the client models and the global model is calculated at every round. If this average falls below a predefined heterogeneity threshold, indicating high client diversity, The lambda value is proportionally reduced to the ratio of average similarity to the threshold. Otherwise, the base lambda value is maintained. Formally, let s_i be the cosine similarity between the i -th client model and the global model, and $\bar{s} = \frac{1}{N} \sum_{i=1}^N s_i$ be the average similarity. Given a base lambda λ_0 and heterogeneity threshold τ , we adjust lambda as:

$$\lambda = \begin{cases} \lambda_0 \cdot \left(\frac{\bar{s}}{\tau}\right) & \text{if } \bar{s} < \tau \\ \lambda_0 & \text{otherwise} \end{cases}$$

The λ parameter plays a critical role in balancing the trade-off between cosine and Gaussian similarity measures, directly influencing how client contributions are weighted during aggregation. In our experiments, we observed that extreme values of λ –either too small (close to 0) or too large (close to 1)–result in suboptimal performance, particularly in heterogeneous, non-IID environments. When λ is set too low, the model over-relies on Gaussian similarity, which prioritizes clients with numerically close updates, potentially disregarding important directional trends. Conversely, a high λ value overemphasizes cosine similarity, focusing excessively on directional alignment without adequately accounting for the magnitude of updates, slowing down convergence. Our empirical evidence indicates that intermediate values between 0.6 and 0.7 offer the optimal balance, with an average accuracy decrease of 3% when values fall above or below this range, allowing the model to incorporate both similarity measures effectively, thereby improving convergence speed and model accuracy.

Given the sensitivity of λ , our method includes a dynamic adjustment mechanism to further optimize its impact during the training process. This mechanism adapts λ based on the current level of heterogeneity across clients, utilizing a predefined threshold to dynamically balance between the cosine and Gaussian similarity measures. When the average cosine similarity between the client models and the global model falls below this heterogeneity threshold, is proportionally reduced, ensuring that client updates that are numerically closer are given more weight. This dynamic adaptation has proven effective in our experiments, improving the robustness of the SimProx method in environments with varying levels of data heterogeneity. Although our initial findings indicate that the optimal λ typically falls between 0.6 and 0.7, future work could explore the dynamic adjustment strategy across a broader range of datasets and FL settings to further solidify these insights.

B. PROXIMAL GRADIENT NORMS

In addition to the client similarity, we also consider the gradient norms $\mathbf{g} \in \mathbb{R}^m$ as a proxy for the client’s contribution to the global model update. The gradient norm

g_i is defined as the Euclidean norm of the difference between the client’s current model weights \mathbf{w}_i and its previous model weights $\mathbf{w}_i^{\text{prev}}$:

$$g_i = \|\mathbf{w}_i - \mathbf{w}_i^{\text{prev}}\|_2 \tag{11}$$

The gradient norm g_i serves as an indicator of how much the client model has changed during the current round of FL. Clients with smaller gradient norms are likely to have model updates that are closer to the global optimum, as their current model is already well-aligned with the global model.

The inclusion of proximal gradient norms is motivated by the observation that not all model updates are equally beneficial to the global model. In FL systems, certain clients may undergo substantial model updates due to local data characteristics, but these large updates could destabilize the global model if incorporated uncritically. By prioritizing clients with smaller gradient norms, SimProx stabilizes the learning process, reducing the risk of overshooting and ensuring smoother convergence. This approach is aligned with the theoretical understanding of gradient-based optimization, where smaller updates closer to the optimum are generally more reliable indicators of model convergence. The proximal term thus enhances the resilience of SimProx in handling the variability in client contributions, leading to more stable and accurate global model updates.

This proximal term complements the client similarity information captured by the similarity matrix \mathbf{S} . Together, these two components allow the algorithm to identify the most relevant client updates and incorporate them more effectively into the global model update, leading to improved performance and faster convergence of the FL process.

C. CLIENT WEIGHTS

The client weights $\alpha \in \mathbb{R}^m$ are calculated as follows:

$$\alpha_i = \exp(-g_i) \left(1 + \frac{\sum_{j \neq i} S_{ij}}{m - 1} \right) \tag{12}$$

This formulation combines two key components: the gradient norms \mathbf{g} and the client similarity matrix \mathbf{S} . By combining the gradient norm term and the client similarity term, the algorithm assigns higher weights to clients that have both smaller gradient norms (indicating updates closer to the global optimum) and higher average similarity to the rest of the client population (indicating more relevant and complementary updates). This dual consideration of the client’s contribution and similarity allows the global model to learn more effectively from the most valuable client updates, leading to improved performance and faster convergence.

The $m - 1$ term in the equation arises from the normalization of the similarity scores, where $\sum_{j \neq i} S_{ij}$ calculates the total similarity of client i with all other $m - 1$ clients, and dividing by $m - 1$ gives the average similarity. This normalization ensures that the weight α_i is independent of the total number of clients, making the approach scalable. In graph-theoretic terms, the similarity matrix S can be

seen as the adjacency matrix of a fully connected graph, where nodes represent clients, and edges represent pairwise similarities. The term $\frac{\sum_{j \neq i} S_{ij}}{m-1}$ corresponds to the normalized degree (or average similarity) of a node in this graph, reflecting how representative or connected client i is to the rest of the population. Combining this with the gradient norm g_i , which prioritizes clients closer to the global optimum, allows the algorithm to weight client updates based on both individual reliability and collective relevance, leading to a more effective and balanced aggregation process.

The mathematical properties of the individual components, such as boundedness, monotonicity, and normalization, ensure that the client weight calculation is well-behaved and provides a principled way to prioritize the most relevant client updates during the FL process.

After calculating the client weights α using the formulation in Equation (12), we perform an additional normalization step:

$$\alpha \leftarrow \alpha / \sum_{i=1}^m \alpha_i \quad (13)$$

Then a softmax function is applied to the weights to provide more evenly weight distribution.

The pre-softmax normalization step ensures that the weights sum to 1 before applying the softmax function. This normalization is crucial for emphasizing the relative shape of the weights vector, α , rather than its absolute magnitude. Since the softmax function inherently sharpens input differences due to its exponential component (e^{α_i}), pre-normalization mitigates the risk of excessively large values disproportionately dominating the output. By dampening extreme variations in the input, this approach controls the spread of the softmax outputs and ensures a more balanced response. Furthermore, empirical evidence suggests that pre-normalization improves convergence, as it provides a consistent and well-defined range of inputs to the softmax function, thereby enhancing the stability of the optimization process.

$$\text{softmax}(\alpha)_i = \frac{e^{\alpha_i}}{\sum_{j=1}^n e^{\alpha_j}} \quad (14)$$

This normalization ensures that the client weights sum up to 1, i.e., $\sum_{i=1}^m \alpha_i = 1$.

The softmax function is a widely used normalization technique that converts unbounded weights into a probabilistic distribution. By applying softmax, SimProx ensures that no single client's update dominates the aggregation process, thereby preventing overfitting to specific clients. This probabilistic scaling aligns with the need for smooth and consistent updates in FL, where extreme variations in client contributions can lead to instability and oscillations in model performance. The use of softmax normalization is scientifically justified by its ability to smooth out the aggregation process, ensuring that all client contributions are balanced and scaled appropriately, contributing to a more reliable and stable global model update process.

Algorithm 1 SimProx

```

1: procedure FEDERATEDLEARNING( $\mathbf{w}^{\text{global}}$ ,  $\mathbf{w}^{\text{client}}$ ,  $\mathbf{w}^{\text{prev}}$ )
2:    $\mathbf{S} \leftarrow \text{ComputeClientSimilarityMatrix}(\mathbf{w}^{\text{client}})$   $\triangleright$ 
   Compute the client similarity matrix
3:    $\alpha \leftarrow \text{ComputeClientWeights}(\mathbf{w}^{\text{client}}, \mathbf{S}, \mathbf{w}^{\text{prev}})$   $\triangleright$ 
   Compute client weights
4:    $\alpha \leftarrow \alpha / \sum_{i=1}^m \alpha_i$   $\triangleright$  Normalize the weights
5:    $\alpha \leftarrow \text{Softmax}(\alpha)$   $\triangleright$  Apply softmax
6:   for  $k \in \mathbf{w}^{\text{global}}$  do
7:      $\mathbf{u} \leftarrow \mathbf{0}$ 
8:     for  $i \leftarrow 1$  to  $m$  do
9:        $\mathbf{u} \leftarrow \mathbf{u} + \alpha_i \mathbf{w}_i[k]$   $\triangleright$  Weighted aggregation of
   client updates
10:    end for
11:     $\mathbf{w}^{\text{global}}[k] \leftarrow \mathbf{u}$   $\triangleright$  Update global model
12:  end for
13:   $\mathbf{w}^{\text{client}} \leftarrow \mathbf{w}^{\text{global}}$   $\triangleright$  Update client models
14: end procedure

```

The normalization step serves two important purposes, which are proper weighting and interpretability.

On one hand, normalizing the client weights ensures that the weighted sum of the client updates \mathbf{w}_i is a valid model update, as the weights are properly scaled to sum up to 1. This allows the global model $\mathbf{w}^{\text{global}}$ to be updated as a convex combination of the client updates, which is a necessary condition for the FL algorithm to converge. On the other hand, the normalized client weights α can be interpreted as the relative importance or contribution of each client to the global model update. This provides a clear and intuitive way to understand the role of different clients in the FL process.

The overall FL algorithm incorporating the client similarity information is presented in Algorithm 1.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

The proposed method is implemented using Pytorch [34] and executed on a GPU-enabled cloud environment. The SGD optimizer is used with the learning rate set to 0.01 and weight decay to 0.001. The batch size is set to 64. The number of local epochs is set to 20 epochs for all FL approaches. For our approach, the optimal value of the lambda parameter was 0.7. For CIFAR-10 and CIFAR-100, the experiments were conducted 10 times, starting with 10 rounds, then repeated with 20 rounds and so on, up to 100 rounds. We compare the proposed SimProx approach with the state-of-the-art method FedAvg, along with the related methods SimAgg [28] and FedProx [26]. For collaborator selection, we use a subset of the available collaborators (30%) in each round. To accommodate system heterogeneity, where collaborator contributions may vary unpredictably, we simulate a random selection process in each round. To ensure that all collaborators are engaged uniformly over

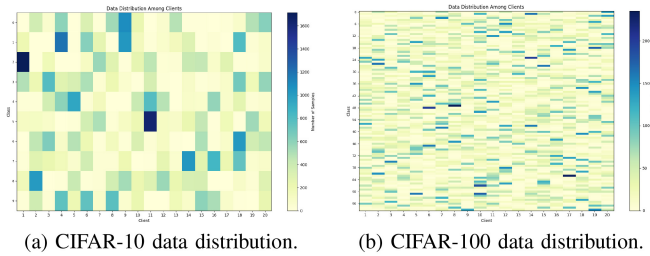
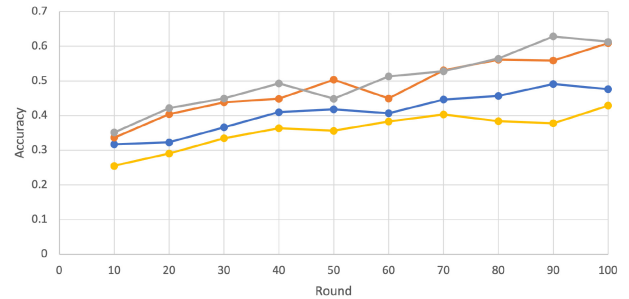


FIGURE 2. Data distribution across 20 clients for CIFAR-10 and CIFAR-100.

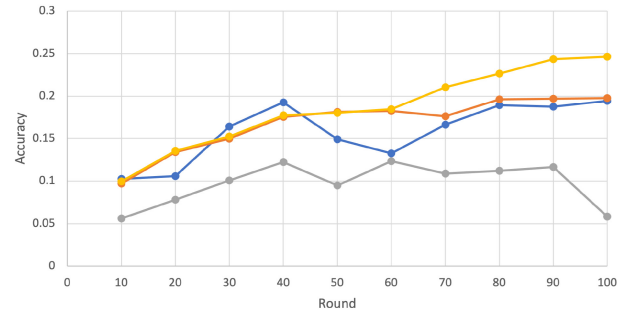
time, we utilize a sliding window mechanism over the randomized collaborator index. This method ensures that, after all collaborators have participated, a new randomized order is generated to enhance learning efficiency. The sliding window approach is favored over random selection to guarantee consistent participation from all collaborators. The sliding-window size is set to 30% of the total number of collaborators within each partition. This approach guarantees that each client contributes to the model’s updates over time, while also introducing an element of randomness to prevent overfitting to any particular client order.

B. PERFORMANCE ON CIFAR IMAGES

We conduct experiments on CIFAR-10 and CIFAR-100[35] both of which are widely recognized as benchmark datasets in the context of heterogeneous FL. To ensure a rigorous evaluation, we utilized the ResNet-18 [36] and DensNet-121 [37] architectures for both datasets. The two models are well-established Convolutional Neural Network (CNN) that have demonstrated robust performance in image classification tasks and serve as a consistent baseline in FL studies. While our primary focus is on introducing and validating a novel aggregation approach, ResNet-18 and DensNet-121 provide a reliable foundation for assessing the performance of our method across diverse experimental settings. The CIFAR-10 dataset contains 60,000 color images of size 32x32, divided into 10 mutually exclusive classes with 6,000 images per class. It includes 50,000 training images and 10,000 test images. The CIFAR-100 dataset consists of 100 classes, each containing 600 images, with 500 training images and 100 testing images per class. Figure 2 shows the samples of CIFAR-10 and CIFAR-100 distributed in the default settings across 20 clients participating in the FL process. Like previous studies [38], [39], we use Dirichlet distribution to generate the non-IID data partition among parties. Specifically, we sample $p_k \sim \text{Dir}_N(\beta)$ and allocate a $p_{k,j}$ proportion of the instances of class k to party j , where $\text{Dir}(\beta)$ is the Dirichlet distribution with a concentration parameter β (0.5 by default). With the above partitioning strategy, each party can have relatively few (even no) data samples in some classes. We set the number of parties to 20 where 6 clients are picked randomly to participate in the learning process in each round to simulate a real world scenario.



(a) Performance on CIFAR-10 dataset using ResNet-18.



(b) Performance on CIFAR-100 dataset using ResNet-18.

FIGURE 3. Average accuracy on CIFAR-10 and CIFAR-100 using ResNet-18.

With ResNet-18, Figure 3(a) shows the average classification accuracy computed every 10 rounds on CIFAR-10 over multiple communication rounds for four different methods: FedAvg, FedProx, SimAgg, and SimProx. SimProx consistently achieves the highest average accuracy, across all communication rounds compared to the other methods. The curve indicates that SimProx is particularly effective in handling the data, which improves convergence speed and model robustness. Both SimProx and FedProx show steady improvement, outperforming FedAvg and SimAgg and maintaining the lead throughout the evaluation process. However, SimProx shows approximately a 2% improvement in accuracy on CIFAR-10 compared to FedProx, reaching a peak accuracy of 0.628 with 90 rounds, whereas FedProx achieves its best accuracy of 0.609 with 100 rounds.

Figure 3(b) also demonstrates how SimProx performs better than the other methods on CIFAR-100 under the same non-IID settings, achieving the highest classification accuracy of 0.247 overall with 100 rounds. The results of the SimProx aggregation approach show superior and consistent performance starting from 60 rounds onward, earlier than the other methods. With a 5–6% accuracy gain over FedAvg and FedProx, SimProx proves its reliability and effectiveness even with the more complex CIFAR-100 dataset.

Figure 4 presents the average accuracy scores obtained from ten test experiments conducted for each aggregation approach on CIFAR-10 and CIFAR-100. SimProx outperforms all the other methods, achieving the highest accuracy.

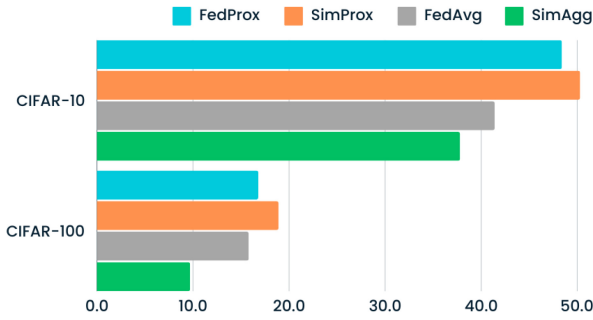


FIGURE 4. Average accuracy of all methods on CIFAR-10 and CIFAR-100.

TABLE 2. SimProx performance on CIFAR-10 and CIFAR-100 with ResNet-18 and DenseNet-121.

	Model	Accuracy	Precision	Recall	F1-Score
CIFAR-10	ResNet-18	0.674	0.673	0.674	0.670
	DenseNet-121	0.817	0.813	0.815	0.814
CIFAR-100	ResNet-18	0.336	0.351	0.336	0.322
	DenseNet-121	0.471	0.494	0.471	0.474

These accuracy results suggest that SimProx offers a robust and efficient aggregation approach to FL.

Further performance enhancements for SimProx could be achieved through the use of alternative deep learning backbones and hyperparameter tuning, although this is beyond the primary scope of this study. Table 2 demonstrates that SimProx’s performance on CIFAR-10 improves by 21% and on CIFAR-100 by 41% on average across all metrics when using DenseNet-121. The highest accuracy achieved using DenseNet-121 over 100 rounds is 0.817 on CIFAR-10 and 0.471 on CIFAR-100, with precision, recall, and F1-score displaying similarly comparable values to the accuracy.

It is worth mentioning that ResNet-18 with its identity shortcuts bypass residual blocks to preserve features, potentially limiting the network’s representational power and learning capacity for SimProx. In contrast, DenseNet-121 uses dense concatenation across all subsequent layers, avoiding direct summation and retaining features from previous layers. While DenseNet-121 has demonstrated more efficient feature utilization for SimProx, outperforming ResNet-18, it demands more GPU memory due to concatenation operations and increases training time by an average of 15%.

C. PERFORMANCE ON TEXTUAL DATA

To further assess the robustness and adaptability of our proposed SimProx aggregation approach in federated learning (FL), we conducted experiments on Banking77 [40], a widely used benchmarking dataset in intent detection for customer service applications. This dataset, comprising over 13,000 customer queries labeled across 77 distinct intents, provides a challenging, real-world setting with inherently diverse and nuanced language, making it well-suited for evaluating FL approaches in non-IID environments. We implemented our model using a pretrained BERT-based

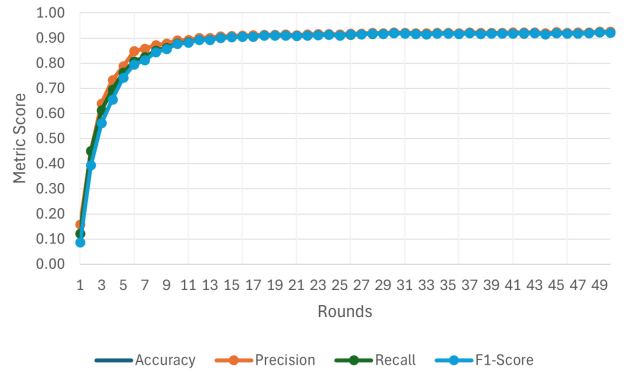


FIGURE 5. Performance of SimProx on Banking77 over 50 rounds.

TABLE 3. Results of SimProx and FedAvg on Banking77.

Method	Accuracy	Precision	Recall	F1-score
FedAvg [42]	0.916	0.920	0.916	0.916
SimProx	0.922	0.925	0.922	0.922

uncased model [41] for text embeddings, enabling us to capture semantic representations of the intents while keeping parameters consistent across clients. By comparing SimProx to the baseline FedAvg aggregation approach, we aimed to demonstrate SimProx’s performance in accuracy and convergence, especially under data heterogeneity. Specifically, we simulated a non-IID setup by assigning each client a subset of intents and varying the number of examples per client.

The performance results on the Banking77 dataset, as shown in Figure 5, demonstrate that SimProx achieved a notably accuracy and fast convergence rate. After 15 communication rounds, SimProx reached an accuracy of 0.903, while it achieved an accuracy of 0.922 with 50 rounds. SimProx also shows relatively similar rates of precision, recall, and F1-score, suggesting that SimProx effectively mitigates the challenges posed by non-IID data, particularly in scenarios where intent representations are not uniformly distributed across clients. SimProx’s similarity-based weighting allowed the global model to prioritize updates from clients with higher relevance to the aggregated intent distribution, leading to a more generalized model that accurately captures intent diversity across all clients. Table 3 also shows that SimProx and FedAvg perform very similarly on the Banking77 dataset across all evaluation metrics.

D. PERFORMANCE ON MULTIMODAL DATA

To assess the performance of the proposed aggregation method on different type of data, we conducted a comprehensive evaluation on the MEDIC dataset [43], a publicly available dataset comprising image-tweet pairs related to various disaster events. This dataset serves as a representative real-world multimodal data source, presenting the challenges inherent in social media data during critical events, such as class imbalance and noisy data.

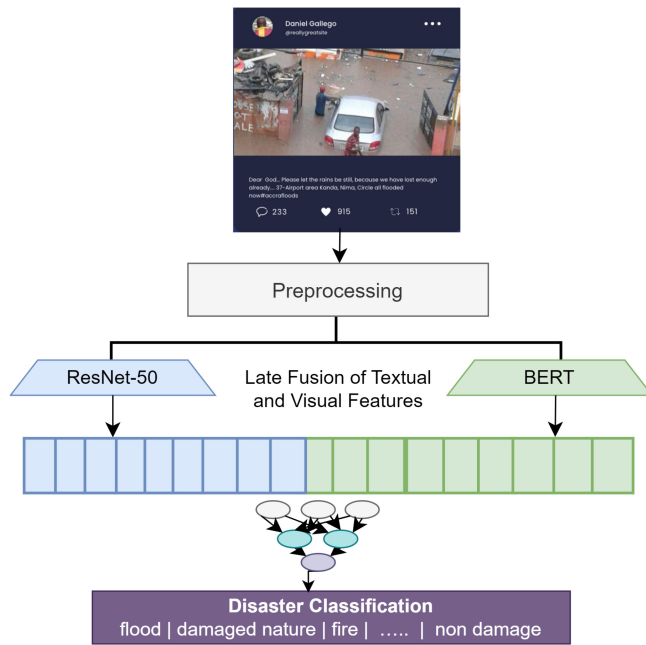


FIGURE 6. The multimodal deep learning model used for disaster event classification.

The MEDIC dataset consists of 5,831 image-tweet pairs, divided into a training set of 5,247 samples and a test set of 584 samples. To mitigate the class imbalance issue, we employed a range of data augmentation techniques, including random horizontal flipping, color jittering, and random rotation, to enhance the diversity of the training data.

We configured the FL setup with 20 client models, similar to the CIFAR experimental setup, randomly selecting clients to participate in each training round. The performance of our proposed aggregation method was evaluated against the traditional FedAvg algorithm implemented in [42] on the MEDIC dataset, with results reported in terms of key metrics, including accuracy, precision, recall, and F1-score, averaged over 31 training rounds.

For feature extraction, we leveraged pre-trained models to capture both textual and visual information, as shown in Figure 6. Specifically, we utilized the BERT model [41] to extract meaningful features from the textual content of the tweets, and the ResNet50 [36] architecture to extract visual features from the corresponding images. The extracted features were then combined using a late fusion approach, where the outputs of the text and image models were concatenated to form a unified representation, enabling our model to capture the complementary information present in both modalities.

The experimental results, presented in Table 4, demonstrate the superiority of SimProx aggregation method compared to FedAvg, emphasizing its effectiveness in leveraging the multimodal nature of the data and the collaborative nature of the FL environment. The improved classification performance, in terms of accuracy, precision, recall, and F1-score, highlights the advantages of our approach in

TABLE 4. Performance comparison on MEDIC dataset.

Method	Accuracy	Precision	Recall	F1-score	Round
FedAvg [42]	0.851	0.858	0.851	0.852	31
SimProx	0.930	0.932	0.930	0.930	11

addressing the challenges inherent in real-world multimodal data.

The findings of this study demonstrate the effectiveness of our proposed aggregation method in the context of real-world multimodal data, highlighting its potential to improve disaster event classification performance while addressing data privacy and scalability concerns.

V. COMPLEXITY ANALYSIS

The SimProx method introduces a hybrid similarity-based aggregation that integrates both cosine and Gaussian similarity measures alongside a proximal term for client update weighting, which increases its computational complexity relative to traditional aggregation methods such as FedAvg, FedProx, and SimAgg. Specifically, SimProx’s use of a similarity matrix, which calculates pairwise similarities between client models, scales quadratically with the number of clients, resulting in an overall complexity of $\mathcal{O}(C^2 \cdot p)$, where C represents the number of clients and p denotes the number of model parameters. The proximal term, which evaluates the gradient norms of client updates, incurs an additional complexity of $\mathcal{O}(C \cdot p)$.

In contrast, simpler methods like FedAvg have a lower complexity of $\mathcal{O}(C \cdot p)$ due to their reliance on a straightforward weighted averaging process, but they suffer from poor performance in handling heterogeneous, non-IID data distributions. FedProx, while introducing a proximal term to improve robustness in heterogeneous settings, adds additional computational load on edge devices by penalizing updates that diverge from the global model, which increases the complexity of local computations. Although the complexity of FedProx remains $\mathcal{O}(C \cdot p)$, the proximal term requires extra gradient calculations, making local updates slower and more computationally intensive. Similarly, SimAgg, while leveraging similarity measures for client weighting, lacks the proximal regularization, exhibiting a complexity of $\mathcal{O}(C^2 \cdot p)$ without the added benefits of SimProx’s enhanced optimization.

The increased computational complexity of SimProx translates into significant performance gains, particularly in non-IID environments. The propose method accelerates convergence and enhances model robustness. In comparison, FedProx, while handling heterogeneity better than FedAvg, imposes more computational burden on edge devices as they must perform additional local gradient computations to maintain proximity to the global model, potentially affecting resource-constrained devices. Experimental results demonstrate that SimProx outperforms FedAvg, FedProx, and SimAgg in terms of accuracy and convergence speed,

especially in scenarios with high data heterogeneity, as shown in the CIFAR-10, CIFAR-100, and Banking77 datasets. While the quadratic complexity of SimProx may pose scalability challenges in large-scale federated learning systems, its effectiveness in handling heterogeneous data distributions makes it a valuable method for environments where data variability is a critical concern.

While SimProx offers several advantages, it is not without limitations, particularly in addressing device heterogeneity. A key challenge lies in the requirement for the server to collect all client models to calculate similarity scores. This process may introduce delays in the learning cycle, especially as the number of participating clients increases significantly. Such delays may impede the responsiveness of the system in dynamic or resource-constrained environments [44]. Future research could explore the adaptation of this method to buffered asynchronous federated learning frameworks [45], which may alleviate delays by enabling clients to upload updates at different times. Additionally, efforts to reduce the computational complexity of the similarity calculation process would further enhance the feasibility of the proposed approach in large-scale, real-world deployments.

VI. CONCLUSION

In this article, we introduced SimProx, a novel similarity-based aggregation method for federated deep learning that optimizes client weights using a combination of cosine and Gaussian similarity measures. Our approach addresses the challenges of data heterogeneity by effectively weighting client contributions based on computed similarities. Experimental results demonstrate that SimProx significantly improves the accuracy and robustness of the aggregated global model, outperforming traditional methods like FedAvg, particularly in non-IID settings and real-world multimodal datasets. The proposed method enhances model performance, also reduces training time, showcasing its potential for practical applications in decentralized environments. While SimProx offers substantial advantages, it also introduces increased computational complexity, which may pose challenges in large-scale FL systems to maintain scalability. Exploring additional similarity measures and addressing communication and computational heterogeneity are also promising directions for future research.

REFERENCES

- [1] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.
- [2] M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous federated learning: State-of-the-art and research challenges," *ACM Comput. Surv.*, vol. 56, no. 3, pp. 1–44, 2023.
- [3] S. Baker and W. Xiang, "Artificial intelligence of things for smarter healthcare: A survey of advancements, challenges, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1261–1293, 2nd Quart., 2023. [Online]. Available: <https://doi.org/10.1109/comst.2023.3256323>
- [4] B. Liu, N. Lv, Y. Guo, and Y. Li, "Recent advances on federated learning: A systematic survey," *Neurocomputing*, vol. 597, Sep. 2024, Art. no. 128019.
- [5] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.
- [6] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2022, pp. 1739–1748.
- [7] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8397–8406.
- [8] S. Alam, L. Liu, M. Yan, and M. Zhang, "FedRolex: Model-heterogeneous federated learning with rolling sub-model extraction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 29677–29690.
- [9] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 12878–12889.
- [10] T. Lin, L. Kong, S. U. Stich, and M. Jaggi, "Ensemble distillation for robust model fusion in federated learning," in *Proc. 34th Adv. Neural Inf. Process. Syst.*, 2020, pp. 2351–2363.
- [11] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning: A meta-learning approach," 2020, *arXiv:2002.07948*.
- [12] E. Diao, J. Ding, and V. Tarokh, "HeteroFL: Computation and communication efficient federated learning for heterogeneous clients," 2021, *arXiv:2010.01264*.
- [13] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *Comput. Sci. Rev.*, vol. 50, Nov. 2023, Art. no. 100595.
- [14] J. Hong, H. Wang, Z. Wang, and J. Zhou, "Efficient split-mix federated learning for on-demand and in-situ customization," 2022, *arXiv:2203.09747*.
- [15] J. Yoon, G. Park, W. Jeong, and S. J. Hwang, "Bitwidth heterogeneous federated learning with progressive weight dequantization," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 25552–25565.
- [16] C. Zhou, H. Tian, H. Zhang, J. Zhang, M. Dong, and J. Jia, "TEA-fed: Time-efficient asynchronous federated learning for edge computing," in *Proc. 18th ACM Int. Conf. Comput. Front.*, 2021, pp. 30–37.
- [17] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.
- [18] L. Qu et al., "Rethinking architecture design for tackling data heterogeneity in federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10061–10071.
- [19] Z. Qu, X. Li, R. Duan, Y. Liu, B. Tang, and Z. Lu, "Generalized federated learning via sharpness aware minimization," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 18250–18280.
- [20] Z. Tang, Y. Zhang, S. Shi, X. He, B. Han, and X. Chu, "Virtual homogeneity learning: Defending against data heterogeneity in federated learning," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 21111–21132.
- [21] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastruct. Deep Learn.*, 2018, pp. 1–8.
- [22] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021.
- [23] X. Ma, J. Zhu, Z. Lin, S. Chen, and Y. Qin, "A state-of-the-art survey on solving non-IID data in federated learning," *Future Gener. Comput. Syst.*, vol. 135, pp. 244–258, Oct. 2022.
- [24] Y. Tan, Y. Liu, G. Long, J. Jiang, Q. Lu, and C. Zhang, "Federated learning on non-IID graphs via structural knowledge sharing," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 9953–9961.
- [25] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.
- [26] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020, *arXiv:1812.06127*.

- [27] S. Ji, S. Pan, G. Long, X. Li, J. Jiang, and Z. Huang, "Learning private neural language modeling with attentive aggregation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2019, pp. 1–8.
- [28] M. I. Khan, M. Jafaritadi, E. Alhoniemi, E. Kontio, and S. A. Khan, "Adaptive weight aggregation in federated learning for brain tumor segmentation," in *Proc. 7th Int. Workshop MICCAI Brainlesion Workshop*, 2021, pp. 455–469.
- [29] H. Wu and P. Wang, "Fast-convergent federated learning with adaptive weighting," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 4, pp. 1078–1088, Dec. 2021.
- [30] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10713–10722.
- [31] Z. Chen, S. Yu, F. Chen, F. Wang, X. Liu, and R. H. Deng, "Lightweight privacy-preserving cross-cluster federated learning with heterogeneous data," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 7404–7419, 2024.
- [32] Y. Yan, X. Tong, and S. Wang, "Clustered federated learning in heterogeneous environment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12796–12809, Sep. 2024.
- [33] C. Briggs, Z. Fan, and P. Andras, "Federated learning with hierarchical clustering of local updates to improve training on non-IID data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2020, pp. 1–9.
- [34] A. Paszke et al., "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.
- [35] A. Krizhevsky, (Univ. Toronto, Toronto, ON, Canada). *Learning Multiple Layers of Features From Tiny Images*. (2009). [Online]. Available: <https://www.cs.toronto.edu/kriz/Learn.-features-2009-TR.pdf>
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [37] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
- [38] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 7252–7261.
- [39] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," 2020, *arXiv:2002.06440*.
- [40] I. Casanueva, T. Temčinas, D. Gerz, M. Henderson, and I. Vulić, "Efficient intent detection with dual sentence encoders," 2020, *arXiv:2003.04807*.
- [41] A. Vaswani et al., "Attention is all you need," 2023, *arXiv:1706.03762*.
- [42] A. El-Niss, A. Alzu'Bi, and A. Abuarqoub, "Multimodal fusion for disaster event classification on social media: A deep federated learning approach," in *Proc. 7th Int. Conf. Future Netw. Distrib. Syst.*, 2024, pp. 758–763.
- [43] F. Alam, T. Alam, M. A. Hasan, A. Hasnat, M. Imran, and F. Ofli, "MEDIC: A multi-task learning dataset for disaster image classification," *Neural Comput. Appl.*, vol. 35, no. 3, pp. 2609–2632, 2023.
- [44] K. Pfeiffer, M. Rapp, R. Khalili, and J. Henkel, "Federated learning for computationally constrained heterogeneous devices: A survey," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–27, 2023.
- [45] J. Nguyen et al., "Federated learning with buffered asynchronous aggregation," in *Proc. 25th Int. Conf. Artif. Intell. Statist.*, 2022, pp. 3581–3607.