Computer Science and Information Systems 22(4):1777–1796 https://doi.org/10.2298/CSIS241205070C

Elastic-Trust Hybrid Federated Learning

Yi-Cheng Chen[1], Lin Hui[2,⋆], and Yung-Lin Chu[3]

[1] Dept. of Information Management, National Central University, Taiwan
ycchen@mgt.ncu.edu.tw
[2] Dept. of Computer Science and Information Engineering, Tamkang University, Taiwan
121678@mail.tku.edu.tw
[3] Dept. of Information Management, National Central University, Taiwan
lexlie.yunglinchu@gmail.com

Abstract. Owing to the widespread application of machine learning, increasing attention has been focused on extensive data collection for learning model construction. Recently, with growing concerns about data privacy, private information protection has significantly increased the operation cost and difficulty of boosting model performance. The Federated Learning (FL) technique has been introduced to address this issue by keeping data on client devices and reducing the need to handle sensitive data directly. However, several challenging issues may arise when applying FL, such as data heterogeneity, efficient feature transmission, and additional computational demands. In this study, a novel FL model, Elastic-Trust Hybrid Federated Learning (ET-FL), is introduced with a dual federated learning framework. ET-FL incorporates the trust mechanism and differential aggregation strategy for model optimization and computation reduction. In addition, the proposed model is applied on real-world datasets to show the performance and practicability of promising results.

Keywords: machine learning, federated learning, decentralization, hybrid federated integration

1. Introduction

Over recent decades, machine learning has emerged as a prominent field characterized by rapid advancements and widespread adoption across various industries. Several machine learning techniques have transformed how businesses progress, empowering them to harness large volumes of data to gain insights and inform decision-making. Breakthroughs in algorithms and computational power have resulted in significant enhancements in areas such as predictive analytics, natural language processing, and computer vision, to name a few. The expansion of machine learning has also catalyzed the development of new applications, ranging from personalized recommendations in e-commerce to advanced diagnostics in healthcare.

Recently, with the growing concerns of data privacy regulations, safeguarding personal information and empowering individuals with more authority over private data have become important issues. These regulations, enforced by governments, necessitate businesses to be open about their privacy practices, and to adopt stringent security measures

⋆ Corresponding author

to protect their clients' data. Consequently, users are increasingly mindful of how organizations handle their sensitive information, leading to a heightened focus on data privacy and security. The shift in power dynamics, where individuals have more control over their data, has instilled a sense of security in the digital realm.

However, in traditional machine learning, data are centralized on a single server for training. Without any doubt, larger datasets typically improve model performance. In general, organizations aim to gather extensive data, which requires substantial storage and a high-performance server, making the process resource-intensive and time-consuming. Securing these centralized data, especially when sensitive user information is involved, adds further cost due to the necessary security measures. For example, in healthcare, patient data must be anonymized and encrypted, adding complexity and computation cost. Likewise, the financial sector must implement strict protocols to protect transaction data. Obviously, these measures are essential for preventing breaches and ensuring privacy, but also increase the cost and complexity of traditional machine learning operations.

Federated Learning (FL) is a promising solution to these problems. FL keeps the private data on each device, also called a client, thus removing the burden of implementing security measurements for organizations. Furthermore, while moving the data to each client, the training process could be done in parallel with each client, with less computing power and time. In this situation, the server orchestrates the training process across clients and maintains the consensus model. We use an application to show the significance of FL. Gboard [12] is a keyboard application installed on Android, one of the major operating systems used on mobile devices. It provides a wide range of input languages and has exceeded 1 billion installations. One of the main features of Gboard is that it suggests the next word according to the context that the user has typed in. To improve the recall of suggestions while protecting user privacy, Google has adopted an FL methodology, FedAvg [34], to complete the task successfully.

Nevertheless, transitioning from a single-server setup to a system with multiple instances may suffer several challenging issues when applying FL. These challenges mainly include data heterogeneity, feature transmission efficiency, and extra computing resource consumption. Data heterogeneity manifests as statistical imbalances, with individual clients possessing varying data distributions. In the context of FL, the model on the server acts as a collective representation of the entire system. While the system generally performs adequately in the presence of statistical imbalances, the performance of specific clients may suffer. Furthermore, feature transmission efficiency in decentralized FL encounters tradeoffs between communication efficiency and cost, with various structures such as line, ring, and mesh necessitating considerations about the optimal balance between communication efficiency and cost when spreading features to all clients. Finally, undoubtedly, the customized approach for adapting a shared model or weights requires additional computing resources to effectively complete the task at hand.

In this study, a novel hybrid framework, Elastic-Trust Hybrid Federated Learning (abbreviated as ET-FL), is proposed to tackle the aforementioned obstacles when applying FL in practical domains. We introduce a two-layer hierarchy including local and global tiers. The node in the local tier includes one server and multiple clients. The clients who have similar statistical distributions will be grouped into one node. With the hierarchy of each node (i.e., one server and multiple clients) in the local tier, we could directly apply the state-of-the-art centralized FL methodologies to learn the model and store in each

server. In the global tier, all servers in local nodes are extracted for further processing, using a decentralized approach. We introduce a novel elastic trust mechanism within the global tier to facilitate peer selection, and a merging weight concept to aggregate consensus models from other servers. The weights can be adjusted iteratively, allowing for precise calibration to extract specific features from various nodes at different iterations. Furthermore, we adopt a differential aggregation strategy on global iterations and local rounds to leverage global feature aggregation and resource consumption.

The contributions of this study are as follows:

- To the best of our knowledge, prior studies excluded emphasis on the integration of different FL methodologies. In this study, we developed a novel framework, ET-FL, a sophisticated two-layer architecture which comprises local and global tiers. The local tier learns models using the centralized FL approach, while the global tier utilizes the decentralized FL approach to integrate the learned models.
- Generally, the problem of performance downgrade in clients is mainly attributed to the presence of diverse and heterogeneous data. To address this challenging issue, we propose a strategy to organize clients with similar characteristics into groups. This contribution ensures that the system can maintain a high degree of personalized FL and also efficiently reduce the requirement of computation resources.
- We introduce a trust mechanism for client selection and aggregation weight control. The client selection process could strike a delicate balance between received models and transmission costs. The aggregation weight control ensures the necessary desired attributes throughout the process. With the proposed trust mechanism, ET-FL could optimize network performance and resource allocation.
- ET-FL equips a differential aggregation strategy bridging the global and local tiers. The proposed strategy allows the local consensus models to have ample time to aggregate features within the nodes before exchanging features in the global tier. The strategy effectively optimizes feature exchange efficiency and resource consumption.
- Finally, the proposed ET-FL framework is applied on several real-world datasets to show its performance and practicability.

The organization of the rest of this paper is as follows. Section 2 discusses the Related Work and Section 3 presents the proposed ET-FL framework in detail. We provide the experimental results in a performance study in Section 4, and conclude the paper in Section 5.

2. Related Work

2.1. Federated Learning

Concerning data privacy, the FL architecture was designed with two components: the server and the clients. The clients' private data are not transferred to the server for training; instead, they remain inside each client. To collect the features across clients, the server maintains a consensus model. In the training process, the server distributes the consensus model to the clients for client training and then collects the updated model weight from the clients, aggregating it into the new consensus model. McMahan et al. [34] were the first to propose architecture with the algorithm called FedAvg.

FedAvg suffers multiple challenges in practical use. One is the data heterogeneous problem, which indicates that FedAvg performs poorly when handling non-IID datasets. Researchers have utilized the data heterogeneous problem to improve performance. For example, Li et al. [27] proposed the Federated Proximal (FedProx) methodology which introduces an additional hyperparameter to limit the convergence direction from deviating too far from the consensus model of the client model optimization. Karimireddy et al. [21] proposed the SCAFFOLD methodology which adds two variates to the server and the client. The variates control the gradient direction to prevent the client model gradient from being directed to an optimal point far from the server consensus model. In[45], Wang et al. proposed FedNova, which normalizes the returned gradients by the numbers of the local updates to prevent the aggregated gradient from being pulled away by a larger dataset. Acar et al. [1] indicated that the client's minimal loss will not equal the global minimum loss. Therefore, they proposed the FedDyn methodology, which normalizes the calculated loss on the client to fit the global one. Duan et al. [8] proposed the Astraea framework which adds the role of mediator to manage a subset of training clients to have balanced data in the group view. Both [50] and [15] provided a concept of sharing few data on the clients over the system to resolve the data heterogeneous problem. In [18], Jeong et al. proposed a federated argumentation method using a generator to generate non-balanced data on the client side. However, the training of the generator requires clients to upload a few data to the server for the FL design.

Other approaches to improving centralized FL performance include applying optimizers to FL, resolving physical limitations, inquiring into the safety of transferred content, etc. [23,5,40,49] have tackled the communication challenges. Selecting the training client is another approach to improving FL performance; this approach was adopted by [37,20,36], while [35,29] defined new merging weight mechanisms, and [4,2,46] delved into security for aggregation. Aside from the aforementioned approaches, some researchers have adapted machine learning methodologies to FL. [39,28,31,44] implemented optimizers in FL, whereas [17,25,51,24,38] implemented knowledge distillation methodologies.

Instead of pursuing better performance using one consensus model on a system with heterogeneous data, some researchers have deployed different models on the client side. This approach is called personalized FL. Arvazagan et al. [3] proposed the FedPer algorithm. In addition to the shared consensus model across the system, FedPer adds an extra layer on top of the consensus model. Furthermore, this added layer does not attend the aggregation to enhance the client feature. Fallah et al. [10] proposed the Per-FedAvg model which splits the client training into two steps using two optimizers to generate the global and local gradients. The global gradient is aggregated to update the consensus model. Liang et al. [32] proposed the LG-FedAvg methodology. Each client maintains the global and local models and updates them by both models' loss in succession.

The concept of training one global model and fitting it to the different tasks of the different datasets in meta-learning and multi-task learning can help to solve the data heterogeneous problem by viewing each dataset as a different task. Smith et al. [42] introduced MOCHA, which identifies different clients as performing different tasks. To moderate the weight between clients, the server maintains a matrix that identifies the relationship between each client. The training process optimizes the client model and relationship matrix. In [9], Eicher et al. categorized the dataset by the time span in the day, such as midnight, morning, noon, etc. Different training rounds pick different time spans' data and clients

for training. In[22], Khodak et al. proposed ARUBA, which maintains an extra parameter, the learning rate, on the server side to better adapt the consensus model to different tasks, which is also adjusted in rounds. Li et al. [26] proposed MOON, a personalized methodology using contrastive learning to minimize the distance between the consensus and local models.

According to the aforementioned research, FL methodologies have diverse approaches to improving performance. However, the most frequently addressed issue is the data heterogeneous problem. The effect of data heterogeneity has even opened up a new branch of FL methodologies called personalized FL. Considering these studies, we divided the clients into different nodes according to the statistics. In addition, we adopted the centralized FL architecture inside nodes.

2.2. Decentralized Federated Learning

A decentralized network is structured without a central authority, relying instead on a distributed architecture where each node operates independently. In this network, control is not vested in a single entity, but is distributed among all participating nodes. Each node has the ability to make decisions, process data, and communicate with other nodes autonomously. The lack of a central control point means that there is no single point of failure, enhancing the network's fault tolerance and reliability.

With these decentralized network features, each node can gain control of its own model and dataset. Furthermore, while each client manages a personal dataset and model, it is a perfect situation to dive into the solution of dataset heterogeneity. These reasons encourage researchers to start their research in decentralized FL. Kalra et al. [19] proposed a model called ProxyFL, in which the consensus model is used only as a proxy model for exchanging the network features. Yue et al. [48]proposed the FedDCM method, which shares the same concept with [19], but uses distillation to exchange the feature between the proxy model and the local model. Gholami et al. [11] proposed a merging weight called trust, which was evaluated by the contribution in the previous round.

With the increased connections between clients in a decentralized network, carefully selecting peers is a new aspect of utilizing the framework. Tang et al. [43] provided a method that selects the peer by the connection bandwidth, while Masmoudi et al. [33] introduced OCD-FL, which includes a peer selection method determined by the energy consumption and knowledge gain.

Other than the research mentioned above, resolving communication efficiency and revising the existing centralized FL methodologies to a decentralized network are also available to improve the framework. Hu et al. [14] divided the model into segments and retained different segments from different peers to replace the aggregation and reduce the communication cost, whereas Li et al. [30] introduced a model using knowledge distillation in the decentralized setting.

In the practical usage of decentralized FL, the network designs vary. For example, [46-48] adopted decentralized FL in practical use in the medical category, but used different network architecture. Huang et al. [16] used line architecture, Chang et al. [6] used a ring network, while Xu et al. [47] used a mesh network in their decentralized FL network architecture.

From the above literature, which benefits from the feature of a decentralized network, decentralized FL has a natural advantage in handling heterogeneous data. In addition,

communication efficiency remains an approach to utilize the framework with some new research criteria, such as the peer selection problem. Considering these mentioned aspects, we adopted the decentralized FL setting in the global tier with the trust mechanism for controlling the merging weight and peer selection.

We propose a hybrid framework mixing the two architectures with client grouping and trust mechanisms. We present a detailed explanation of our proposed model, Elastic-Trust Hybrid Federated Learning, in the next section.

3. The Proposed Framework: ET-FL

Our first goal was to address the challenge of dealing with heterogeneous data and to avoid using extra computational resources on clients for a personalized approach. Instead, we proposed a client grouping strategy with a centralized server inside each node. To facilitate the exchange of features between nodes, we introduced a global tier to gather all the servers and build a network. We also aimed to remove control from any specific server and to form a decentralized network known as the global tier network. Additionally, we acknowledge that the optimal model weight derived from all the data may not always be the best for practical usage. To address this, we introduced the elastic Trust mechanism for each node to determine the preferred weight for global aggregation in different iterations. This approach also serves as a peer selection method. Finally, we observed that a synchronous setting for global tier training and local tier training leads to insufficient feature information exchange. As a solution, we proposed the differential aggregation strategy.

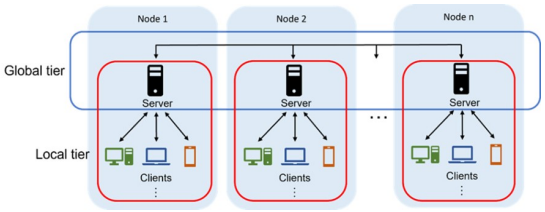

Fig. 1. The system architecture of ET-FL

Figure 1 displays the architecture of the system design. In the following section, we denote our nodes $\mathcal{N} = \{1, 2, 3, \dots\}$, and server $\mathcal{S} = \{s_n \mid n \in \mathcal{N}\}$ to better illustrate our architecture.

3.1. Local Tier

In the local tier, we group clients into clusters based on their statistical characteristics. Each cluster has a server assigned to manage the training process within the cluster. We denote $C_s = \{1, 2, 3, \dots\}$ as the clients that are grouped under a specific server. This organizational structure is illustrated in Figure 2.

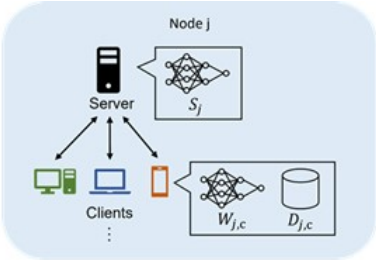

Fig. 2. The node structure

Each client device maintains a private dataset $\mathcal{D} = \{d_{s,c} \mid c \in \mathcal{C}_s,\ s \in \mathcal{S}\}$ as well as a local model $\mathcal{W} = \{w_{s,c} \mid c \in \mathcal{C}_s,\ s \in \mathcal{S}\}$. Meanwhile, the server stores a consensus model S and is responsible for coordinating the training process. In star network architecture, centralized FL methodologies such as the three most mentioned methodologies, FedAvg, FedProx, and SCAFFOLD, can be effectively implemented. Our objective function is given as follows:

$$w^{(t+1)} = \arg\min_{w} \mathcal{F}_L(w) \tag{1}$$

The FedAvg algorithm represents a pioneering approach to employing FL. The training begins with the server initializing the consensus model weight and randomly selecting a subset of clients for training. Subsequently, the server transmits the consensus model weight to the chosen clients to initialize the training process on their end. Upon reception of the model weight, the clients apply it to their respective models and commence the training procedure using their local datasets over a specified number of epochs. Upon completion of training, the client returns the model weight to the server. Post-receiving all the model weights from the selected clients, the server aggregates the model weights using the weight calculated from the training data size of the clients. The server then leverages the accumulated gradient to optimize the consensus model. These steps constitute one round, and a comprehensive training process encompasses several rounds. We denote the server weight $\mathcal{W}_f$ and the selected client $\mathcal{K}$ to express the aggregation as follows:

$$w_s = \frac{1}{M} \sum_{k \in \mathcal{K}} n_k w_k, \quad \text{where } M = \sum_{k \in \mathcal{K}} n_k \tag{2}$$

The FedAvg methodology has introduced a new avenue for research, prompting many researchers to explore its applications. FedProx has emerged as a key player in this field, with a particular focus on addressing the challenges posed by data heterogeneity. In order to mitigate the risk of highly diverse client datasets misleading the consensus model, FedProx incorporates a penalty mechanism that accounts for the discrepancy between the local model and the consensus model. This ensures that the client's search for the optimal model weight is guided by a refined equation, thereby enhancing the overall efficacy of the approach. The new equation of the client for searching for the optimal model weight is as follows:

$$w^{(t+1)} = \arg\min_{w} F_k(w) + \frac{\mu}{2} \|w - w^t\|^2 \tag{3}$$

Although we describe the client process as a search for the best model weight, it involves receiving the current round model weight $\mathcal{W}^{\sqcup}$ from the server, thereby controlling the client's distance from the server. Furthermore, the parameter $\Updownarrow\sqcap$ is crucial in determining how closely the client remains near the server. Essentially, a higher $\Updownarrow\sqcap$ level increases the client's difficulty finding an optimal space away from the server.

SCAFFOLD addresses the issue of data heterogeneity by introducing server and client control variates, which help control the client model stepping during training. These variates indicate the stepping direction in the previous round. As the client undergoes training, the gradient is adjusted using the gap between the server and client variates. The specific formula for this correction is provided as follows:

$$w_k^{(t+1)} = w_k^t - \eta_l \nabla \mathcal{L}(w_k^t) - c_k + c_s \tag{4}$$

where c_k and c_s refer to the client and server control variables. At the end of each round, these variables undergo updates based on a predefined formula. This update ensures that the variables accurately reflect the state of the client-server interaction. The formula is listed as follows:

$$c_k^{(t+1)} = c_k^t - c_s + \frac{1}{K\eta_l}\left(w_s - w_k^{(t+1)}\right), \quad \text{where } K \text{ stands for number of epochs} \tag{5}$$

$$c_s^{(t+1)} = c_s^t + \eta_g \left(\frac{1}{S}\sum_{i\in\mathcal{S}} w_i^{(t+1)} - w_s^t\right), \quad \text{where } S \text{ is the number of selected clients} \tag{6}$$

3.2. Global Tier

Within our local network infrastructure, clients are segmented into distinct nodes for organizational purposes. Each node's server is interconnected to facilitate the seamless exchange of features across the various nodes. This interconnected network, known as the global tier, plays a pivotal role in our operations. To ensure that server control is uniformly distributed and feature exchange is conducted with optimal efficiency, we have implemented a mesh network within the global tier. Utilizing this mesh network enables us to manage equal server controllability and to maximize the efficiency of feature exchange across nodes. Figure 3 depicts the intricate interactions between a server and other servers within the global tier, providing a comprehensive illustration of our network architecture.

Figure 3 illustrates how a server retrieves model weights from other servers, known as an aggregation step. We introduce a novel concept called "Trust" for merging weights during aggregation. Trust governs the feature weight gathered from the servers. Higher trust weight indicates the greater significance of the corresponding model's feature. Trust also serves as a method for selecting peers while adjusting the trust weight to zero. Moreover, trust is an independent setting that differs from the servers and is elastic over iterations. Trust (T) is represented as follows:

$$T \subseteq \left\{(x, y, t) \mid (x, y) \in \mathbb{N}^2,\ t \in [0, 1]\right\} \tag{7}$$

We denote the aggregated model in the global tier as S_con. We can express the relation of trust as follows:

$$\mathcal{S}_{\text{con}} = \left\{ \sum_{t \in T_s,\ w \in \mathcal{S}} t \cdot w \,\middle|\, s \in \mathcal{S} \right\} \tag{8}$$

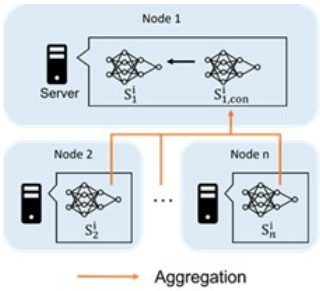

Fig. 3. Global tier aggregation

Algorithm 1 Elastic-Trust Hybrid Federated Learning

initialize the consensus models S
for i in Iterations I **do**
  for s in Servers S **in parallel do**
    $s \leftarrow LocalTierTraining(s)$
  end for
  $S_{con} \leftarrow \{\Sigma_{t \in T_s, w \in S} t * w | s \in S\}$
  $S \leftarrow S_{con}$
end for

LocalTierTraining(s):
for r in Rounds R **do**
  $s \leftarrow FL(s)$
end for
return s

// This method is capable of all centralized FL methods
// We take FedAvg as an example
FL(s):
for e in Epochs E **do**
  select a subset K from clients in C_s
  for k in K **in parallel do**
    $w_k \leftarrow w_k - \eta \nabla l(w_k, d)$
  end for
  $m \leftarrow \Sigma_{k \in K} n_k$
  $s \leftarrow \Sigma_{k \in K} \frac{n_k}{m} w_k$
end for
return s

Furthermore, we have put forward a differential aggregation strategy, which entails adjusting the pace of aggregation in the global and local tiers. In this strategy, we define

the global training cycle as an iteration and the local training cycle as a round. By implementing the differential aggregation strategy, multiple rounds of training are incorporated into a single iteration. Through the implementation of a differential aggregation strategy, our model effectively optimizes the sharing of features and minimizes communication costs. This allows for a harmonious balance between the utilization of shared features and the associated costs of communication within the model. Algorithm 1 details the main learning process in the ET-FL framework.

4. Experiments and Evaluation

To thoroughly evaluate the effectiveness of our proposed methodology, ET-FL, we undertook a comprehensive series of experiments involving three diverse datasets:

- Shakespeare [41]: This dataset is derived from "The Complete Works of William Shakespeare," including all the plays written by William Shakespeare. We obtained the preprocessed version from LEAF [50], which adopted the dataset for the task of next-letter prediction with an input sequence length of 80 characters. We adopt the LSTM model for the next-letter prediction task.
- Amazon Review [13]: This dataset is a collection of Amazon product details and customer reviews, encompassing both the rating and review text. To prepare the data for analysis, we truncated and tokenized the review text to a maximum length of 200 words for input while selecting the rating as the corresponding label. We chose to use a Transformer model to predict the rating
- EMNIST [7]: The EMNIST dataset, short for extended MNIST, is a comprehensive extension of the original MNIST dataset. It was derived from the NIST Special Database 19, which encompasses handwritten digits and both upper- and lower-case letters. The MNIST dataset comprises 70,000 images.

Table 1. Summary of Datasets

Dataset	# train	# test	# labels
Shakespeare [41]	606,277	202,103	80
Amazon Review [13]	418,811	139,613	6
EMNIST [7]	209,993	70,007	10

We divided these datasets into 20 non-IID sub-datasets to accommodate our specific needs. Each sub-dataset consists of a training set and a corresponding test set.

4.1. Baseline and Metrics

We selected the average training loss, test accuracy, test recall, test precision, and test F1-score for evaluating our model. The training loss is calculated as the difference between the model's predicted value (output logit) and the actual value, indicating how well the model fits the training dataset. Commonly used methods for calculating training loss are

mean square error and cross-entropy loss. In our experiment, we opted for cross-entropy loss. The cross-entropy loss formula follows, where $\mathrm{p}(x)$ represents the label encoded into a one-hot vector and $\mathrm{q}(x)$ represents the model's output logit. Applying a negative sign to the formula results in a positive value, facilitating easier comprehension. A lower value indicates effective model training, while a higher value suggests the opposite.

$$\text{Cross-Entropy Loss} = -p(x) \log q(x) \tag{9}$$

The concept of accuracy, recall, precision, and F1-score is a computed metric obtained from the confusion matrix depicted in the accompanying Figure. A true positive denotes a scenario where both the actual and predicted values are positive. On the other hand, false negatives and false positives refer to cases where the actual and predicted values are discordant, indicating either a misclassification of a positive actual value as negative or a misclassification of a negative actual value as positive, respectively. Finally, a true negative encompasses situations where the actual and predicted values are correctly identified as negative.

		Actual Class	
		Positive (**P**)	Negative (**N**)
Predicted Class	Positive (**P**)	True Positive (**TP**)	False Positive (**FP**)
	Negative (**N**)	False Negative (**FN**)	True Negative (**TN**)

Fig. 4. The node structure

Accuracy is the ratio of correctly predicted data to the overall data. It is the ratio of the true positive and the true negative data to the overall data. The formula is as follows:

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

Recall is the ratio of corrected predicted positive data to the overall data whose true label is positive. It provides the performance on the true label side. The formula is as follows:

$$\text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{11}$$

Precision is the ratio of corrected predicted positive data to the overall data whose predicted label is marked as positive. It provides the performance on the predicted output's side. The formula is as follows:

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{12}$$

To identify a model that performs well, it should not only have good performance on either recall or precision, but on both. The F1 score therefore takes into account both recall and precision simultaneously. The formula is as follows:

$$\text{F1-score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{13}$$

To mitigate the impact of heterogeneous datasets, we averaged the training loss and the accuracy achieved during the training iterations as the reference metrics for comparison.

As for the baseline methods, we carefully chose several FL algorithms to serve for evaluation, and compared these baseline models with our proposed methodology using specific evaluation metrics. The evaluation methods we chose are test loss and accuracy. The algorithms we selected for comparison are integral to our research and will provide valuable insights into the effectiveness of our proposed approach.

- ML (single): This baseline collects all the datasets into one server to undergo the training process. It represents the traditional machine learning training process, which provides a comparison of the traditional machine learning training process and the FL approaches.
- FedAvg [34]: FedAvg, which stands for Federated Averaging, is a foundational methodology in FL. This approach revolutionizes traditional training architectures by introducing two essential roles: the server and the client. The server orchestrates the collaborative model training process, while the client devices actively participate in model training, all while ensuring that data privacy is securely maintained.
- FedProx [10]: FedProx is a novel approach designed to deal with diverse and varying datasets. This is achieved by incorporating a hyperparameter that plays a crucial role in determining the direction of model optimization. Additionally, FedProx ensures that the model does not stray too far from the central server during the optimization process, thus maintaining stability and consistency.
- SCAFFOLD [21]: SCAFFOLD introduces the server and client variates in the training process. Every gradient will get a correction, which is the difference between the server and client variates. This prevents the client update from drifting away from the system's optimal point.
- Per-FedAvg [10]: Per-FedAvg adopts the Model-Agnostic Meta-Learning approach, which stands for the personalized approach in FL. The server holds a consensus model that is validated to handle all the tasks. However, applying the consensus model to any specific task will lead to poor performance. Therefore, the client model has to undergo an extra training process to fit the usage.

4.2. Performance Comparison

We evaluated our proposed method against baseline models using the Shakespeare, Amazon Review, and Loan datasets. For consistency, we reviewed each baseline's original research to apply the optimal settings. Our experimental setup included an optimizer with Stochastic Gradient Descent at a 0.01 learning rate, with 250 training rounds per server and four epochs per client per round. We adopted other specialized parameter settings from the existing methodologies.

To group clients effectively, we utilized a pre-trained Transformer model to extract data features from output hidden states. We then calculated client centroids as representative statistical data, applying K-means clustering to organize clients into distinct groups. Finally, we recalculated the centroids for each node and used the distance between nodes as a trust metric. Results are displayed in Tables 2 to 6.

Table 2 shows the cross-entropy loss metric, where centralized FL approaches face higher losses due to data heterogeneity. Per-FedAvg improves on the centralized methods, but our model surpasses even personalized models, achieving outstanding performance.

Table 2. Experiment Results of Averaged Cross-Entropy Loss

Model	Shakespeare	Amazon Review	EMNIST
ML (single)	1.798	0.232	0.000*
FedAvg	3.926	3.285	0.528
FedProx	3.982	3.475	0.243
SCAFFOLD	3.144	3.320	0.796
Per-FedAvg	1.302	0.258	0.018
ET-FL (FedAvg)	2.529	1.690	0.209
ET-FL (FedProx)	2.618	1.932	0.064
ET-FL (SCAFFOLD)	2.135	1.836	0.226

* less than 0.001

Table 3. Experiment Results of Averaged Accuracy

Model	Shakespeare	Amazon Review	EMNIST
ML (single)	0.466	0.710	0.996
FedAvg	0.208	0.457	0.882
FedProx	0.190	0.440	0.924
SCAFFOLD	0.201	0.449	0.878
Per-FedAvg	0.233	0.673	0.956
ET-FL (FedAvg)	0.323	0.679	0.936
ET-FL (FedProx)	0.315	0.644	0.978
ET-FL (SCAFFOLD)	0.359	0.658	0.930

Next, we examine the average accuracy results in Table 3, which show a pattern similar to the cross-entropy loss findings. Centralized FL methodologies perform below personalized FL methods, while our model surpasses centralized FL and achieves comparable accuracy to Per-FedAvg.

Table 4. Experiment Results of Averaged Recall

Model	Shakespeare	Amazon Review	EMNIST
ML (single)	0.466	0.710	0.996
FedAvg	0.208	0.457	0.882
FedProx	0.190	0.440	0.924
SCAFFOLD	0.201	0.449	0.878
Per-FedAvg	0.233	0.673	0.956
ET-FL (FedAvg)	0.323	0.679	0.936
ET-FL (FedProx)	0.315	0.644	0.978
ET-FL (SCAFFOLD)	0.359	0.658	0.930

From Tables 4, 5, and 6, these metrics further confirm that centralized FL methods generally underperform compared to personalized ones, with the exception of FedAvg and FedProx on the EMNIST dataset for the precision metric. These findings underscore our model's strengths. By effectively grouping clients, our approach successfully mitigates

Table 5. Experiment Results of Averaged Precision

Model	Shakespeare	Amazon Review	EMNIST
ML (single)	0.519	0.687	0.996
FedAvg	0.338	0.585	0.988
FedProx	0.297	0.547	0.994
SCAFFOLD	0.419	0.559	0.979
Per-FedAvg	0.303	0.753	0.969
ET-FL (FedAvg)	0.425	0.783	0.990
ET-FL (FedProx)	0.372	0.727	0.995
ET-FL (SCAFFOLD)	0.528	0.735	0.989

Table 6. Experiment Results of Averaged F1

Model	Shakespeare	Amazon Review	EMNIST
ML (single)	0.426	0.696	0.996
FedAvg	0.210	0.461	0.914
FedProx	0.181	0.442	0.952
SCAFFOLD	0.212	0.453	0.911
Per-FedAvg	0.218	0.676	0.958
ET-FL (FedAvg)	0.331	0.683	0.954
ET-FL (FedProx)	0.317	0.641	0.973
ET-FL (SCAFFOLD)	0.363	0.653	0.951

the data heterogeneity challenge faced in centralized FL. Its improved performance over centralized methods and its competitive standing with personalized approaches highlight its robustness and effectiveness.

4.3. Trust Weight Influence Analysis

As previously mentioned, the initial trust weight is derived from the distance between centroids of different nodes' data. It is important to note that the trust weight of each node consists of two key components: the weights for aggregating consensus models from other servers and the weight applied to the server's consensus model. In our calculations, we assigned a specific value to the latter weight, while the undistributed weight was determined based on the distance. A more considerable distance results in a lower weight, while a shorter distance leads to a higher weight. Consequently, we experimented by assigning different values to the latter weight to test the personalized level of nodes. To evaluate the personalized level, we performed the evaluation before and after the aggregation and subtracted the value. In cross-entropy loss, a higher value means the loss increases more after the aggregation, while a lower value means a low loss increase. This means less personalized and more personalized, respectively. In the accuracy, recall, precision, and F1-score metrics, the lower the reduction in performance after the aggregation, the more personalized the aggregated model will be. In contrast, higher reduction means the aggregated model receives more features from the other consensus model and, thus, is less personalized. The results are presented in Table 7.

Table 7. Personalized levels under different trust settings

	Loss	Accuracy	Recall	Precision	F1-score
0.5	0.676	-0.196	-0.196	-0.072	-0.145
0.6	0.477	-0.130	-0.130	-0.044	-0.089
0.7	0.293	-0.074	-0.074	-0.034	-0.053
0.8	0.143	-0.029	-0.029	-0.016	-0.020
0.9	0.042	-0.004	-0.004	-0.001	-0.002

According to the results, we can discover that a lower trust weight on the server's weight leads to a greater increase in cross-entropy loss and a decrease in the other metrics. In contrast, higher trust in the server's weight results in less cross-entropy increase and less performance decrease. The higher weight means the consensus model has a better-personalized level inside the group. In comparison, a lower weight means that the consensus model receives more information from another consensus model. In conclusion, our design of trust successfully provides users with a method to control performance, whether it is more personalized or adopts more global features.

4.4. Iteration and Round Ratio Analysis

In designing the differential aggregation strategy, we considered the number of features exchanged in aggregation and the resource cost, trying to find a balance point between rounds executed in one iteration. Therefore, we conducted this experiment to explore the influence of different ratio settings on the rounds and iterations. In the FL methodologies, the non-IID datasets lead to fluctuation in performance. To eliminate the variation factor, we divided the clients into 20 nodes. We randomly picked one node to be the observation target. The target node's model converges at around 20 epochs if trained using the traditional machine learning approach. As a result, we executed our framework with a total of 20 epochs, with one epoch in one round, and conducted 2, 5, 10, and 20 rounds in one iteration. In addition, we added a test set for executing 30 rounds in one iteration. Collecting these test scenarios allowed us to observe the difference between pre-convergence, convergence, and post-convergence situations. To evaluate the outcome of each node's collection of sufficient information for global tier aggregation, we used the feature gained from other nodes as the evaluation metric. To measure the feature gain in one iteration from other nodes, we evaluated the target node using other nodes' test datasets, subtracted the results before and after the aggregation, and averaged the value from different test datasets. For those settings that are executed over one iteration, we averaged the result by the aggregation times. The experiment results are shown in Table 8.

According to the results, we can discover that some values are negative. This is because the global consensus model mixes multiple models simultaneously. However, with the mixing in model weight, some features will be less significant, causing the performance to decrease when evaluating using the corresponding test dataset. In addition, we discovered that fewer rounds in one iteration led to better performance in feature gain. Therefore, doing a global aggregation more frequently is a better option. However, our differential aggregation strategy still provides users with an option to leverage the feature gain and the communication cost for tuning the best ratio for different users.

Table 8. Averaged feature gain on the different ratios between iterations and rounds

Rounds	Accuracy	Recall	Precision	F1-score
2	0.011	0.011	0.050	0.014
5	-0.003	-0.003	-0.014	-0.002
10	-0.006	-0.006	0.049	0.001
20	-0.007	-0.007	0.004	-0.001
30	-0.012	-0.012	0.073	-0.007

4.5. Ablation Study

We conducted a series of experiments to assess how well our model's various components performed when we turned off specific components. The situations are listed below:

- w/o global tier: We deemed all the nodes to have equal authority over other nodes and complete control of the model and dataset. Thus, we adopted the decentralized FL to eliminate the possibility of any client having control over others. By removing the global tier, our architecture retains the local tier. In addition, due to the absence of connection between different nodes for feature exchange, the clients should be grouped into one node, which is a centralized FL architecture.
- w/o local tier: We designed the client grouping strategy in the local tier and adopted centralized FL architecture inside nodes. By removing the local tier, our framework remains a decentralized network in the global tier. Thus, every client is equivalent to a node located in the global tier, aggregating different nodes' features according to the trust.
- w/o trust: The trust mechanism is designed for a user-controllable parameter to determine the weight to aggregate different consensus models. While removing the trust from our model, we used the weight calculated by the client train data size, a method mentioned in [2]. Within the group level, we calculate the weight from the total size of training data inside nodes instead of the selected clients' training data size.
- w/o differential aggregation strategy: The differential aggregation strategy indicates that the training process inside the global and local tiers is asynchronous. While one training iteration performs in the global tier, the local tier may perform several training rounds. Our orientation in designing the feature is to find a balance between the feature aggregation and the resource cost. If this strategy is disabled, only one round will be performed in one iteration.

Table 9. The component effectiveness in the ablation study

	Loss	Accuracy	Recall	Precision	F1-score
w/o global tier	3.926 (+1.397)	0.208 (-0.115)	0.208 (-0.115)	0.338 (-0.087)	0.210 (-0.121)
w/o local tier	0.841 (-1.688)	0.742 (+0.419)	0.742 (+0.419)	0.707 (+0.282)	0.707 (+0.376)
w/o trust	2.997 (+0.468)	0.268 (-0.055)	0.268 (-0.055)	0.391 (-0.034)	0.268 (-0.063)
w/o differential aggregation strategy	0.899 (-1.630)	0.735 (+0.412)	0.735 (+0.412)	0.695 (+0.270)	0.698 (+0.367)
ET-FL	2.529	0.323	0.323	0.425	0.331

According to the metrics, our model without a global tier loses the ability to personalize the model for each node, thus reducing performance. Our model without the local tier results in every client being a node in the global tier. This structure mitigates the loss inside the node due to only one client in each node. Thus, it can easily reach the system optimal inside the node. However, while leveling up the client tier, every client has to train every round, thus raising the computation resource cost in contrast to the random client selection strategy in centralized FL. While removing the trust mechanism, the model performance is affected by the size of the dataset nodes, losing the ability to control the convergence direction. This results in a subpar performance. The experiment without the differential aggregation strategy also results in better performance. However, exchanging the model weight in every round increases the transmission cost. In summary, our design of the hybrid framework with client grouping, the trust mechanism, and the differential aggregation strategy provides the optimal setting for resolving the data heterogeneous problem and the personalized approach with a more straightforward, user-controllable method.

5. Conclusion

ET-FL is an advanced two-layer hybrid Federated Learning (FL) framework that integrates both global and local tiers. It applies centralized FL methods at the local level and decentralized methods at the global level. To tackle data heterogeneity within nodes, we developed a unique grouping strategy that clusters clients by statistical similarity. Additionally, we introduced "Trust," a new aggregation weight that enhances both global aggregation and peer selection, enabling secure, reliable collaboration. A differential aggregation strategy further balances global feature aggregation with resource efficiency across tiers. We rigorously evaluated ET-FL on real-world datasets, comparing it to five baseline models with two primary metrics. Results showed that ET-FL consistently outperformed centralized FL methods and rivaled personalized FL approaches, achieving these outcomes with lower computational costs, making it highly efficient and cost-effective. ET-FL is especially promising for applications dealing with data heterogeneity or limited computational resources, as well as for decentralized setups organized by device ownership. This balanced, resource-efficient approach opens up new possibilities for Federated Learning applications.

Acknowledgments. The work of Yi-Cheng Chen was supported in part by the National Science and Technology Council, Taiwan, under Grant NSTC 111-2628-H-008-005-MY4 and 113-2410-H-008 -065 -MY3.

References

1. Acar, D.A.E., Zhao, Y., Navarro, R.M., Mattina, M., Whatmough, P.N., Saligrama, V.: Federated learning based on dynamic regularization. arXiv abs/2111.04263 (2021)
2. Agarwal, N., Suresh, A.T., Yu, F., Kumar, S., McMahan, H.B.: cpsgd: Communication-efficient and differentially-private distributed sgd. arXiv abs/1805.10559 (2018)
3. Arivazhagan, M.G., Aggarwal, V., Singh, A.K., Choudhary, S.: Federated learning with personalization layers. arXiv abs/1912.00818 (2019)

4. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A., Seth, K.: Practical secure aggregation for privacy-preserving machine learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. pp. 1175–1191. ACM (2017)
5. Caldas, S., Konečný, J., McMahan, H.B., Talwalkar, A.: Expanding the reach of federated learning by reducing client resource requirements. arXiv abs/1812.07210 (2018)
6. Chang, K., Balachandar, N., Lam, C.K., Yi, D., Brown, J.M., Beers, A.L., Rosen, B.R., Rubin, D., Kalpathy-Cramer, J.: Distributed deep learning networks among institutions for medical imaging. Journal of the American Medical Informatics Association 25, 945–954 (2018)
7. Cohen, G., Afshar, S., Tapson, J., Schaik, A.v.: Emnist: an extension of mnist to handwritten letters. arXiv abs/1702.05373 (2017)
8. Duan, M., Liu, D., Chen, X., Tan, Y., Ren, J., Qiao, L., Liang, L.: Astraea: Self-balancing federated learning for improving classification accuracy of mobile deep learning applications. arXiv preprint arXiv:1907.01132 (2019)
9. Eichner, H., Koren, T., McMahan, H.B., Srebro, N., Talwar, K.: Semi-cyclic stochastic gradient descent. arXiv abs/1904.10120 (2019)
10. Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In: Advances in Neural Information Processing Systems. pp. 3557–3568 (2020)
11. Gholami, A., Torkzaban, N., Baras, J.S.: Trusted decentralized federated learning. In: 2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC). pp. 1–6. IEEE (2022)
12. Hard, A., Rao, K., Mathews, R., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., Ramage, D.: Federated learning for mobile keyboard prediction. arXiv abs/1811.03604 (2018)
13. Hou, Y., Li, J., He, Z., Yan, A., Chen, X., McAuley, J.: Bridging language and items for retrieval and recommendation. arXiv abs/2403.03952 (2024)
14. Hu, C., Jiang, J., Wang, Z.: Decentralized federated learning: A segmented gossip approach. arXiv abs/1908.07782 (2019)
15. Huang, L., Yin, Y., Zhang, Z.F., Deng, H., Liu, D.: Loadaboost: Loss-based adaboost federated machine learning with reduced computational complexity on iid and non-iid intensive care data. arXiv abs/1811.12629 (2020)
16. Huang, Y., Bert, C., Fischer, S., Schmidt, M., Dörfler, A., Maier, A., Fietkau, R., Putz, F.: Continual learning for peer-to-peer federated learning: A study on automated brain metastasis identification. arXiv abs/2204.13591 (2022)
17. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.L.: Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arXiv abs/1811.11479 (2018)
18. Jeong, E., Oh, S., Kim, H., Park, J., Bennis, M., Kim, S.L.: Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. arXiv abs/1811.11479 (2023)
19. Kalra, S., Wen, J., Cresswell, J.C., Volkovs, M., Tizhoosh, H.R.: Decentralized federated learning through proxy model sharing. Nature Communications 14 (2021)
20. Kang, J., Xiong, Z., Niyato, D.T., Yu, H., Liang, Y.C., Kim, D.I.: Incentive design for efficient federated learning in mobile networks: A contract theory approach. In: 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS). pp. 1–5. IEEE (2019)
21. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T.: Scaffold: Stochastic controlled averaging for federated learning. arXiv preprint arXiv:1910.06378 (2020)
22. Khodak, M., Balcan, M.F., Talwalkar, A.: Adaptive gradient-based meta-learning methods. arXiv abs/1906.02717 (2019)
23. Konečný, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv abs/1610.05492 (2016)

24. Lee, G., Jeong, M., Shin, Y., Bae, S., Yun, S.Y.: Preservation of the global knowledge by not-true distillation in federated learning. In: Advances in Neural Information Processing Systems. pp. 38461–38474 (2022)
25. Li, D., Wang, J.: Fedmd: Heterogenous federated learning via model distillation. arXiv abs/1910.03581 (2019)
26. Li, Q., He, B., Song, D.: Model-contrastive federated learning. arXiv abs/2103.16257 (2021)
27. Li, T., Sahu, A.K., Sanjabi, M., Zaheer, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. arXiv abs/1812.06127 (2018)
28. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Feddane: A federated newton-type method. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers. pp. 1227–1231. IEEE (2019)
29. Li, T., Sanjabi, M., Smith, V.: Fair resource allocation in federated learning. arXiv abs/1905.10497 (2019)
30. Li, X., Chen, B., Lu, W.: Feddkd: Federated learning with decentralized knowledge distillation. Applied Intelligence pp. 1–17 (2022)
31. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. ArXiv abs/1907.02189 (2019)
32. Liang, P.P., Liu, T., Ziyin, L., Salakhutdinov, R., Morency, L.P.: Think locally, act globally: Federated learning with local and global representations. ArXiv abs/2001.01523 (2020)
33. Masmoudi, N., Jaafar, W.: Ocd-fl: A novel communication-efficient peer selection-based decentralized federated learning. arXiv abs/2403.04037 (2024)
34. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629 (2016)
35. Mohri, M., Sivek, G., Suresh, A.T.: Agnostic federated learning. arXiv abs/1902.00146 (2019)
36. Nguyen, H.T., Sehwag, V., Hosseinalipour, S., Brinton, C.G., Chiang, M., Poor, H.V.: Fast-convergent federated learning. IEEE Journal on Selected Areas in Communications 39, 201–218 (2020)
37. Nishio, T., Yonetani, R.: Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). pp. 1–7. IEEE (2018)
38. Qi, P., Zhou, X., Ding, Y., Zhang, Z., Zheng, S., Li, Z.: Fedbkd: Heterogenous federated learning via bidirectional knowledge distillation for modulation classification in iot-edge system. IEEE Journal of Selected Topics in Signal Processing 17, 189–204 (2023)
39. Reddi, S.J., Charles, Z.B., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S., McMahan, H.B.: Adaptive federated optimization. arXiv abs/2003.00295 (2020)
40. Sattler, F., Wiedemann, S., Müller, K.R., Samek, W.: Robust and communication-efficient federated learning from non-i.i.d. data. IEEE Transactions on Neural Networks and Learning Systems 31, 3400–3413 (2019)
41. Shakespeare, W.: The Complete Works of William Shakespeare. Project Gutenberg (1994)
42. Smith, V., Chiang, C.K., Sanjabi, M., Talwalkar, A.: Federated multi-task learning. arXiv abs/1705.10467 (2017)
43. Tang, Z., Shi, S., Li, B., Chu, X.: Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication. IEEE Transactions on Parallel and Distributed Systems 34, 909–922 (2023)
44. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D., Khazaeni, Y.: Federated learning with matched averaging. arXiv abs/2002.06440 (2020)
45. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. ArXiv abs/2007.07481 (2020)
46. Xu, G., Li, H., Liu, S., Yang, K., Lin, X.: Verifynet: Secure and verifiable federated learning. IEEE Transactions on Information Forensics and Security 15, 911–926 (2020)

47. Xu, J., Glicksberg, B.S., Su, C., Walker, P., Bian, J., Wang, F.: Federated learning for healthcare informatics. arXiv abs/1911.06270 (2020)
48. Yue, H., Lanju, K., Qingzhong, L., Baochen, Z.: Decentralized federated learning via mutual knowledge distillation. In: 2023 IEEE International Conference on Multimedia and Expo (ICME). pp. 342–347. IEEE (2023)
49. Zhang, X., Hong, M., Dhople, S.V., Yin, W., Liu, Y.: Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. ArXiv abs/2005.11418 (2020)
50. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv abs/1806.00582 (2018)
51. Zhu, Z., Hong, J., Zhou, J.: Data-free knowledge distillation for heterogeneous federated learning. In: Proceedings of the 38th International Conference on Machine Learning. pp. 12878–12889. Proceedings of Machine Learning Research (2021)

Yi-Cheng Chen received his Ph.D. degree from the Department of Computer Science at National Chiao Tung University (NCTU), Taiwan, in 2012. Currently, he is a professor and chair of the Department of Information Management at National Central University (NCU), Taiwan. He has been active in international academic activities, as conference organizer and journal editor/reviewer. Dr. Chen has published a number of papers in several prestigious conferences and journals. His research interests include machine learning, social network analysis, data mining and cloud computing.

Lin Hui is currently a professor with the department of computer science and information engineering, Tamkang University, Taiwan. Her research interests include machine learning, multimedia applications, and mobile information systems. She has published some journal articles, book chapters, and conference papers related to these research fields. She had served as journal guest editor/reviewer, and program co-chair/chair for many international conferences and workshops.

Yung-Lin Chu received the M.S. degree from the Department of Information Management, National Central University, Taiwan.

Received: December 05, 2024; Accepted: May 18, 2025.