ISSN: 2182-2069 / E-ISSN: 2182-2077

# FedProx: FedSplit Algorithm based Federated Learning for Statistical and System Heterogeneity in Medical Data Communication

Cina Mathew[1*], and Dr.P. Asha[2]

[1*]Sathyabama Institute of Science and Technology, Chennai, India. cinamma@gmail.com, https://orcid.org/0009-0006-8573-7379

[2]Professor, Department of Computer Science and Engineering, Sathyabama Institute of Science and technology, Chennai, India. ashapandian225@gmail.com, https://orcid.org/0000-0003-3046-8811

Received: April 06, 2024; Revised: June 13, 2024; Accepted: July 15, 2024; Published: August 30, 2024

## Abstract

Distributed machine learning offers more practical and efficient use cases than conventional centralized machine learning. Nevertheless, not all security needs can be satisfied by distributed learning. In medical industry, more and more individuals are adopting Internet of Things (IoT) devices to capture their personal data for medical diagnosis and treatment. Through the use of federated learning, it is possible to secure user data while simultaneously training on massive amounts of dispersed data. The heterogeneity of client data is a well-known difficulty in federated learning (FL) contexts. The solution that developed was customized federated learning (PFL), a framework for developing local models for clients' requirements. It is common practice in PFL to construct two models at once, one for local usage and one for global use; the first is used for generalization, while the second is used to inform and update the first one. To build better customized models, it is vital to realize that both global and local models may be enhanced to increase their generalization potential. Secure heterogeneity medical data collection and training has emerged as the top priority in FL. In order to address statistical and system heterogeneity, this work presents a novel hybrid federated learning approach that uses FedProx: FedSplit Algorithm. The presence and severity of data heterogeneity determine the kind of federated learning approaches that could be necessary. The FedProx method averages the changes to the local model if the user data is horizontally partitioned, which means that different samples have the same features. Techniques such as FedSplit may be required to align the feature spaces or separate the model layers when working with data sources that are vertically partitioned, indicating they contain unique features but overlap samples. As a result of statistical variability, learning across data from different distributions is challenging, and device-level systems limits mean that each device can only do so much work, the FedProx: FedSplit model assures convergence for our method. More specifically, as compared to the present FL model, FedProx: FedSplit shows far more steady and accurate convergence behavior in very diverse conditions, increasing overall test accuracy by an average of 35%.

**Keywords:** Statistical Heterogeneity, System Heterogeneity, Federated Learning, IoT Data, Medical Data.

*Journal of Internet Services and Information Security (JISIS)*, volume: 14, number: 3 (August), pp. 353-370.
DOI: *10.58346/JISIS.2024.I3.021*

*Corresponding author: Sathyabama Institute of Science and Technology, Chennai, India.

# 1 Introduction

Many domains have benefited from machine learning's innovations, including voice recognition, computer vision and natural language processing. A vast amount of training data is essential for machine learning techniques, especially those built on deep neural networks (DNNs), since these approaches are data-driven. Edge devices, including smartphones, sensors, cars, and medical equipment, are a typical source of this kind of data. The standard approach is that information acquired by mobile devices should be sent to a central server which is located in the cloud for processing. Uploading datasets like images, signals and text messages might be problematic because of privacy and location issues since they often include personal information (Kutlu & Camgözlü, 2021). In addition, the communication channels between the server and the edge devices might become significantly overloaded when transferring huge information. One solution to these problems is mobile edge computing, which makes use of the processing power of devices in the network's edge to conduct training locally, without requiring data sharing (Chen & Ran, 2019). Federated learning is a new approach to enable learning at the edge (Mohandas et al., 2024). It involves distributed learning with centralized aggregations, managed by one or more edge servers (McMahan et al., 2017). There has been an increase in studies on federated learning in recent years (Kairouz et al., 2019). Important and interdependent parts of these systems are server-device communication and learning from dispersed data (Li et al., 2020; Sindhusaranya et al., 2023; Torroglosa et al., 2017; Wang et al., 2020).

When it comes to medical applications, every healthcare organization has its own data and may have unique needs that necessitate a customized model. There is certain to be data and model heterogeneity in healthcare industry that build customized models for each task. Due to privacy reasons, hospitals in this setting are hesitate to provide their data and model architecture. Consequently, an extensive number of approaches have been proposed for carrying out FL on such diverse medial datasets. An iterative training method is the basis of federated learning (McMahan et al., 2017; Udayakumar et al., 2023). In order to create a single global model, the server compiles all of the incoming updates and then returns its parameters to the devices at the edge. Thus, in order to achieve federated learning, edge devices just need to communicate the parameters of their trained models, without revealing any private data. Because of this quality, federated learning is an attractive option for applications with many organizations who need to learn from data while adhering to stringent privacy regulations (Li et al., 2020). Following the recommendation of Google AI researchers (McMahan et al., 2017), federated learning has been used to improve Gboard's next-word prediction models (Hard et al., 2018; Xu et al., 2021; Qu et al., 2020; Kwon et al., 2020; Brik et al., 2020).

Although federated learning's decentralized and shared structure presents challenges not seen in traditional centralized deep learning, these challenges are closely connected to classical signal processing and communications research. A key component of federated learning's success is expected to involve the development of specialized signal processing algorithms. Actually, there have been a number of new approaches developed that are federated learning-oriented, and they all stem from well-established ideas in communications and signal processing. Methods such as reducing the number of messages sent during federated learning through quantization and compression (Lin et al., 2017; Alistarh et al., 2017; Shlezinger et al., 2017; Zheng et al., 2020), creating useful and wireless computation resources that allow efficient federated gathering over shared wireless channels (Zhu et al., 2019; Amiri & Gündüz, 2020; Sery & Cohen, 2020), and establishing resource allocation schemes that take federated learning into account to guarantee reliable communication between training entities (Chen et al., 2021; Dinh et al., 2020) are some of these approaches.

In traditional centralized deep learning, the server has access to the whole dataset. However, federated learning works differently. As a result, traditional federated learning has a number of fundamental difficulties. Due to statistical heterogeneity, it is possible for individual model instances trained on different edge devices to exhibit bias. Therefore, the global model's intended use in inference may not be reflected in the typical approach based on averaging the training updates. The availability and behaviour of the many user devices involved in federated learning might vary greatly. This kind of variation, called behaviour heterogeneity, might impact the device status-dependent learning process. This research introduced a new hybrid algorithm called FedProx: FedSplit Algorithm for safe data transmission via the cloud. Its purpose is to overcome statistical and technological heterogeneity.

**Key Contribution**

- To address heterogeneity in federated networks, we provide a novel hybrid methodology called FedProx: FedSplit Algorithm in this study.
- The novel FedProx learning approach will be proposed to avoid the convergence behaviour due to the statistical heterogeneity.
- To minimize the sum of local loss functions of devices using the Fedspilit algorithm in order to build a shared global model.
- The proposed model will be compared with current methods/ models on various available datasets.

## 2 Related Study

An essential and often used technique in FL, FedAvg (McMahan et al., 2017), determines the global model by averaging the client-side local models. One of its variations, FedAvgM (Hsu et al., 2019), improves the global model update by adding server-side Nesterov momentum. Based on the quantity of local processing, FedNova (Kutlu & Camgözlü, 2021) normalizes aggregate weights. Global aggregation is explicitly not applied to the parameters of the batch normalization layer by FedBN (Li et al., 2021). To reduce the impact of client drift, FedProx (Li et al., 2020) uses a proximal term in local training loss, while SCAFFOLD (Karimireddy et al., 2020) uses variance reduction and a control variate approach. Clients may bring their local training loss regularization up to date using FedDyn (Acar et al., 2021) so it matches the global empirical loss more closely. As an alternative, FedDC (Gao et al., 2022) proposes including a drift variable into model learning in order to actively reduce disparities between regional and global characteristics. To make the global and local feature representations more comparable, MOON (Li et al., 2021) use model-contrastive regularization. On the other hand, FedDF (Lin et al., 2020) and FedBE (Chen & Chao, 2020) both aim to incorporate information into the global model via knowledge distillation-based model fusion and Bayesian model ensemble, respectively. With FedGen (Zhu et al., 2021), clients may learn from the server-side generator model, this eliminates the requirement for an unlabelled transfer dataset.

Communication effectiveness, confidentiality preservation, attack defence, and federated fairness are some of the open issues and difficulties surveyed (Kairouz et al., 2019), who also cover current improvements in federated learning. An approach-and-challenge classification methodology is provided (Wahab et al., 2021). Federation learning systems are categorized (Li et al., 2021) according to six factors: data distribution, machine learning model, privacy method, communication architecture, federation size, and federated motivation. In their research on federated learning in mobile edge networks, (Lim et al., 2020) classified current approaches as either addressing the core issues with federated learning or finding solutions to difficulties with edge computing via federated learning. In their description of the primary challenges encountered by federated learning in wireless communication contexts, (Niknam et al., 2020) primarily list and explore several potential uses of federated learning in 5G networks. A number of studies have investigated federated learning for Internet of Things networks (Nguyen et al., 2021). Data offloading and caching, threat detection, smart healthcare, smart transportation, unmanned aerial vehicles, and other Internet of Things (IoT) services and applications are surveyed and analyzed (Nguyen et al., 2021). Yang et al., (2019) classify federated learning as either horizontal, vertical, or federated transfer learning based on the data distribution properties. However, they fail to provide a comprehensive overview and classification of current approaches, instead focusing on introducing the idea and practical use of federated learning. Each kind of FL heterogeneity is addressed in (Gao et al., 2022), which classifies and introduces situations, aims, and approaches for data space, statistics, systems, and models, respectively. Despite surveying the literature on managing Non-IID data and analysing its influence on federated learning, (Zhu et al., 2021) fail to address additional heterogeneity concerns and relevant studies. To better understand how to train customized models to deal with statistical heterogeneity, (Tan et al., 2022) undertake a systematic assessment of current approaches in the area of personalized federated learning. However, the difficulties of federated learning are not thoroughly examined, and a thorough taxonomy is missing. Intelligent Internet of Things (IoT) applications may benefit from the cloud-edge architecture proposed (Wu et al., 2020), which offers a customizable federated learning framework. Although diverse federated learning is vast, their categorization of current approaches is inadequate. Communication efficiency, privacy preservation, assault defense, and federated fairness are some of the open issues and difficulties surveyed Kairouz et al., 2019), who also cover current improvements in federated learning. An approach-and-challenge categorization methodology is provided (Wahab et al., 2021). federation learning systems are categorized (Li et al., 2021) according to six factors: data distribution, machine learning model, privacy method, communication architecture, federation size, and federated motivation. In their research on federated learning in mobile edge networks, (Lim et al., 2020) classified current approaches as either addressing the core issues with federated learning or finding solutions to difficulties with edge computing via federated learning. In their description of the primary challenges encountered by federated learning in wireless communication contexts, (Niknam et al., 2020) primarily list and explore several potential uses of federated learning in 5G networks. A number of studies have investigated federated learning for Internet of Things networks (Nguyen et al., 2021). Data offloading and caching, threat detection, smart healthcare, smart transportation, unmanned aerial vehicles, and other Internet of Things (IoT) services and applications are surveyed and analyzed (Nguyen et al., 2021). Yang et al., (2019) classify federated learning as either horizontal, vertical, or federated transfer learning based on the data distribution properties. However, they fail to provide a comprehensive overview and categorization of current approaches, instead focusing on introducing the idea and practical use of federated learning. Each kind of FL heterogeneity is addressed in (Gao et al., 2022), which classifies and introduces situations, aims, and approaches for data space, statistics, systems, and models, respectively. Despite surveying the literature on managing Non-IID data and analysing its influence on federated learning, (Zhu et al., 2021) fail to address additional heterogeneity concerns and relevant studies. To better understand how to train customized models to deal with statistical heterogeneity, (Tan et al., 2022) undertake a systematic assessment of current approaches in the area of personalized federated learning. However, the difficulties of federated learning are not thoroughly examined, and a thorough taxonomy is missing. Intelligent Internet of Things (IoT) applications may benefit from the cloud-edge architecture proposed (Wu et al., 2020), which offers a customizable federated learning framework. Although diverse federated learning is vast, their categorization of current approaches is inadequate. It is difficult for readers to stay updated on developments in this sector since the current approaches are various and differ greatly in their own settings without a standard setting.

### Challenges of Federated Learning in Medical Application

**Statistical Heterogeneity:** When there is statistical heterogeneity, it means that distinct groups of healthcare organization generate data in different ways. Since the medical data accessible at each user device is likely to be customized towards the unique user, this is usually the case with federated learning. Due to statistical heterogeneity, it is possible for individual model instances trained on different edge devices to show bias. Therefore, the traditional approach that uses an average of the training updates may not be the best way to use the global model for inference.

**System Heterogeneity:** The availability and behavior of the many healthcare organization devices involved in federated learning might vary greatly. The device-dependent learning process may be impacted by this kind of heterogeneity, which is called behavior heterogeneity. As an example, when healthcare organization's devices are idle, charging, or linked to an unmetered network like Wi-Fi, the federated learning process might be designed to just participate individuals. Therefore, there may be inconsistency in the training group's involvement due to patients potential unreliability and the fact that they can drop out at any moment. In federated learning, the healthcare organization's devices involved differ in terms of processing capacity and energy resources, as opposed to centralized learning, which uses a computationally capable server to conduct training.

## 3 Basics of Federated Learning in Signal Transmission

There are primarily three phases to the federated learning process: model distribution, local training, and global aggregation. However, as mentioned earlier, the last step of global aggregation is primarily responsible for the specific difficulties of federated learning. Specifically, when comparing the transmission from users to the server to the distribution phase, the last stage is far less impacted by heterogeneity concerns since it comprises broadcasting the global model from the server to the users. Additionally, unlike the aggregation step, this broadcasting occurs on the downlink channel, which usually has lower capacity limits than the uplink channel and is therefore less impacted by communication limitations. In the local training stage, which updates the model using the data and the prior model, traditional optimizers like SGD and its variations are usually used. The majority of problems with federated learning therefore occur during the global aggregate phase. It is possible to think of each of the three primary steps in the global aggregate as a separate signal processing and/or communication task.

The three steps involved in the global learning technique are as follows: 1) edge user processing and encoding of the local training result; 2) edge user transmission of the outcomes via shared wireless channels; and 3) server processing and merging of the received signals. Figure 1 shows a schematic of the federated learning process that incorporates this stage-by-stage analysis.

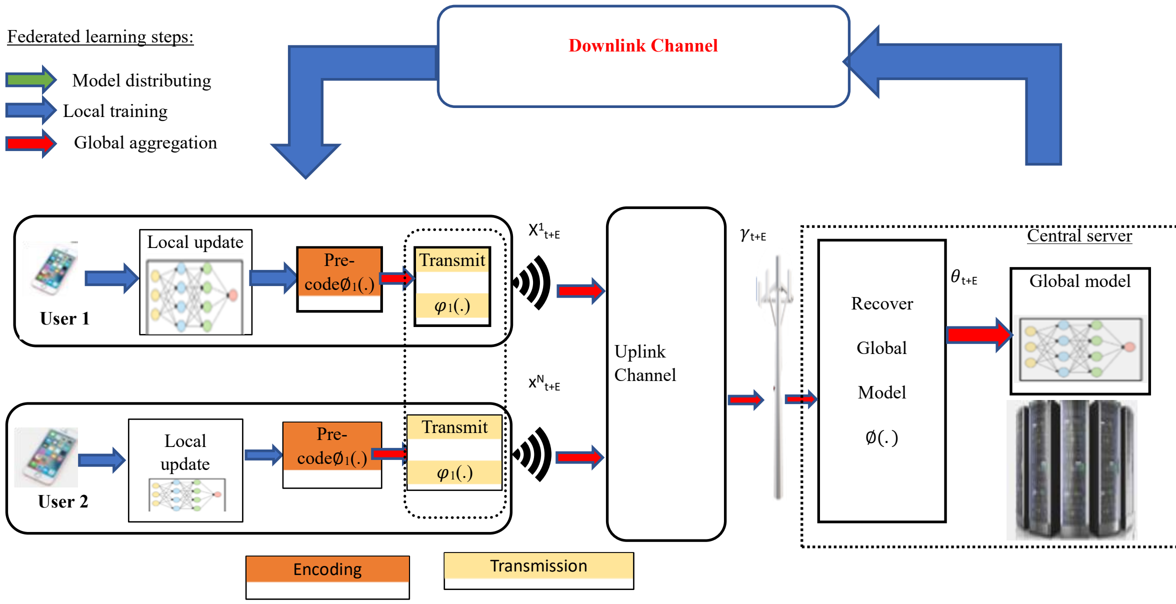


Figure 1: Federated Learning Process in Signal Transmission

**1) Local Updating and Encoding:** In the first phase, prior to transmission, the results of the training operation are processed locally on the users' end. This function is crucial for securely and reliably transmitting model changes to the server via a rate-limited shared connection. The local processing, as shown in Figure 1, involves taking the local model for user $i$ at time $t$, $\theta_t^i$, and using it to create the channel input in the transmission stage, which is represented by the mapping in equation (1):

$$S_t^i = \emptyset_i\left(\theta_t^i\right) (1)$$

The need to encode and compress the model updates should be addressed in the design of the mapping (9). This may be accomplished by implementing stochastic quantization and sparsification of the model updates to reduce the communication burden by setting $\emptyset_i$ (). In addition, $\emptyset_i$ () may be designed to enhance data privacy preservation by using the algorithmic basis of differential privacy.

**2) Transmission of Uplink:** It is common practice to use a wireless channel to transfer the processed changes to the server for global updating. In order to achieve this goal, the mapping requires each user to create their own channel input, $x_t^i$, in order to transmit $S_t^i$. Equation (2) shows the relation between $x_t^i$ *and* $S_t^i$.

$$x_t^i = \varphi_i\left(S_t^i\right) (2)$$

As shown in Figure 1, the conditional distribution $P(y_t|\{x_t^i\}_i)$ directs the statistical connection between the channel input $x_t^i$ and the output $y_t$ received at the server side. So, the users transmit channel inputs to the server, which gets a noisy version of those inputs. The server's channel output $y_t$ and the encoded model updates $S_t^i$ are related according to the transmission mapping and channel characteristics. Federated learning via shared wireless channels is usually at the center of transmission-phase approaches. The goal here is to provide dependable connectivity and high throughput without significantly slowing down the learning process overall. To do this, two primary factors must be considered: Priority one is determining how the channel's capacity and transmission time will be distributed. When deciding which users to include in the current round Gt, user selection and scheduling is a subset of channel resource division. Additionally, users may take use of the interference that results from using the wireless media to their greatest potential by taking advantage of its shared nature as a kind of over-the-air functional computing.

**3) Global Merging:** If the server wants to update the global model, it utilizes the information on $\{\theta_t^i\}_{i=1}^N$ found in its observed channel output $y_t$. The server performs this action and Equation (3) represents the mapping function.

$$\theta_t = \phi\,(y_t) \quad (3)$$

Here, as shown in Figure 1, the server estimates a combined global model by using the channel output, which includes information about the individual model updates. This mapping can be used to counteract channel noise and fading, reduce the impact of statistical heterogeneity on the global model during inference, and protect against malicious devices using Byzantine-robust aggregation.

# 4 FEDPROX: The FEDSPLIT Algorithm

As far as we are aware, the current approaches attempt to adequately handle the dual issues of system heterogeneity and statistical heterogeneity of medical data in FL. This study presents a technique called FedProx:FedSplit that takes use of both statistical and system heterogeneity of medical data at the same time. Its purpose is to improve the global model's efficiency in FL by addressing both issues. The proposed hybrid model has two advantages: By limiting the local updates to be more similar to the original (global) model, it solves the problem of statistical heterogeneity (1) without requiring the manual setting of the number of local epochs using the FedSplit method. (2) Using the FedProx method, it enables the safe incorporation of varying quantities of local work brought about by system heterogeneity. In the section that follows, we provide a summary of the FedProx:FedSplit algorithm's phases.

## 4.1 FedProx Based Averaging Procedure

Similar to FedAvg, our proposed architecture, FedProx, selects a selection of devices for local updates, which are then averaged to provide a global update at each round. Nevertheless, FedProx implements the following crucial but straightforward changes, which lead to significant empirical gains and enable us to ensure convergence for the system heterogeneity.

### Tolerating Partial Work

As was previously mentioned, various devices in federated networks often have various resource limitations with regard to their battery life, network connectivity, and computational capabilities. Therefore, requiring every device to run the same number of local epochs, $E$, or accomplish a same amount of work, as in FedAvg, is impractical. By permitting varying amounts of work to be completed locally across devices according to their available systems resources, we generalize FedAvg in FedProx and then aggregate the partial solutions received by the stragglers (as opposed to discarding these devices). To put it another way, FedProx implicitly accounts for varied $\gamma$ for various devices and iterations rather than assuming a constant $\gamma$ for all devices throughout the training phase. For device $k$ at iteration t below, we define $\gamma_t^k$ -inexactness clearly.

### Proximal Term

Although accepting uneven workloads across devices might mitigate the adverse effects of system heterogeneity, an excessive number of local updates could possibly lead to method divergence because of the heterogeneous data below. To effectively restrict the influence of variable local updates, we propose expanding the local subproblem by adding a proximal term. Specifically, device $k$ employs its preferred local solution to roughly minimize the goal $h_k$, rather than just minimizing the local function $F_k(\cdot)$. Equation (4) shows the details of this function.

$$\min_{w} h_k(w; w^t) = F_k(w) + \frac{\mu}{2} \|w - w^t\|^2 \quad (4)$$

We see that proximal words, like the one above, are often used tools in data optimization. One key difference in the proposed use is that we indicate, investigate, and evaluate this term with the intention of addressing heterogeneity in federated networks. We consider a subset of devices active at each round and deal with non-IID partitioned data in a distributed setting, which is one of the distinguishing characteristics of our study. A number of assumptions are necessary for describing such a framework in real-world federated settings. By modifying the local subproblem in FedProx, we achieve more stable and robust convergence for heterogeneous datasets than the original FedAvg. Our results demonstrate that enabling partial work is beneficial in systems with heterogeneity (Section 5). Method 1 demonstrates the FedProx approach for handling system heterogeneity.

---

$\boldsymbol{Algorithm}$ **1**: $Fedprox\ for\ System\ Hetereogenity$

---

$\boldsymbol{Input}$: $System\ parameter, channey\ parameter\ and\ devices$

$\boldsymbol{Output}$: $FedProx\ solver\ (Prox_{update_j})$

$\boldsymbol{for}\ t = 0, \ldots T-1\ do$

1: $Server\ selects\ a\ subset\ S_t\ of\ K\ devices\ at\ random\ (each\ device\ k\ is\ hosen\ with\ probability\ p_k)$

2: $Server\ Sends\ w^t to\ all\ chosen\ devices$

3: $Each\ chosen\ device\ k\ \in S_t finds\ a\ w_k^{t+1}\ which\ is\ a\ \gamma_k^t - inexact\ minimizer\ of\ :\ w_k^{t+1}$

$$\approx \arg\min_{w}(w: w^t) = F_k(w) + \frac{\mu}{2} \|w - w^t\|^2$$

4: $Each\ device\ k\ \in S_t\ sends\ w_k^{t+1}\ back\ to\ the\ server$

5: $Server\ aggregats\ the\ w's\ as\ w^{t+1} = \frac{1}{K} \sum_{k \in S_t} w_k^{t+1}$

$\boldsymbol{end}\ for$

---

## 4.2 FedSplit for Splitting Procedure

### Theoretic View

To start, we remember the problem's consensus formulation (2) in terms of a block-partitioned vector $x = (x1, \ldots, xm) \in (R^d)^m$, and the function $F : (R^d)^m \to R$. For issue (2), the feasible subspace is denoted as $E \cdot := \{x \mid x_1 = x_2 = \cdots = x_m\}$, and the function $R$ is defined as $F(x) = \sum_{j=1}^{m} f_j(x_j)$. To solve issue (2) according to the first-order optimality criteria, we need to identify a vector $x \in (R^d)^m$ where

$\nabla F(x)$ is either a member of the normal cone of the constraint set E or, alternatively, where $\nabla F(x) = 2E^T$. Similarly, if we classify a set-valued operator $N_E$ as shows in Equation (5).

$$N_E(x) \coloneqq \begin{cases} E^T, x_1 = x_2 = \cdots = x_m, \\ \emptyset, else \end{cases} \quad (5)$$

Next, it's the same as finding a vector $x \in \left(R^d\right)^m$ that works for inclusion.

$$0 \in \nabla F(x) + N_E(x)$$
$$where\ \nabla F(x) = (\nabla F_1(x_1) \ldots., \nabla F_m(x_m)) \quad (6)$$

When $f_j : R^d \rightarrow R$ is a convex loss function, then $\nabla F$ and $N_E$ are monotone operators on $\left(R^d\right)^m$ too (Bauschke & Combettes, 2019). The result is a monotone inclusion issue, as shown in equation (6). The optimization and applied mathematics literatures have a rich history of studying methods for addressing monotone inclusions (Ryu & Boyd, 2016). Now that we have this framework, we can create and evaluate algorithms to address the statistical heterogeneity federated challenges.

**Procedures**

We proceed to detail one approach that is based on splitting into the relation and whose zeroes indeed match the distributed problem's global minima. It is a distributed version of the Peaceman Rachford splitting algorithm, which is known as the FedSplit algorithm (Peaceman & Rachford, 1955). Algorithm 2 explore the FedSplit process for minimizing statistical heterogeneity.

---

**Algorithm 2: FedSplit for Statistical Heterogeneity**

---

$Input : Initialization\ of\ x \in R^d. FedProx\ solver\ Prox_{update_j}: R^d \rightarrow R^d$

$Initialize\ x^1 = z_1^1 = \cdots = z_m^1 = x$

$\boldsymbol{for}\ t = 1,2, \ldots$:

1. $\boldsymbol{for}\ j = 1, \ldots m$:

$a. Local\ prox\ step : set\ z_j^{(t+1/2)} = prox\_update_j\left(2x^t - z_j^t\right)$

$a. Local\ centering\ step : set\ z_j^{(t+1)} = z_j^t + 2\left(z_j^{t+1/2} - x^t\right)$

$\boldsymbol{end\ for}$

2. $Compute\ global\ average : set\ x^{t+1} = \overline{z^{t+1}}$

$\boldsymbol{end\ for}$

---

A parameter vector $z_j^t \in R^d$ is therefore kept for each device j ϵ [m] by the FedSplit process. The parameter estimations at each machine are averaged out by the central server, which keeps a parameter vector $x(t) \in R^d$. For a proper step size s > 0, the local update at device $j$ is defined as the product of the following: proximal solver $prox\_update_j\ (\cdot)$, which are usually approximate proximal updates $prox\_update_j(x) \approx prox_{sfj}\ (x)$, distributed uniformly in $x \in R^d$. When we report our convergence findings in next section, we clarify the meaning of this approximation precisely. Compared to existing FL algorithm, FedSplit has the right fixed points for the distributed issue, which is an advantage.

# 5 Result and Discussion

We test FedProx: FedSplit on various workloads, models, and federated datasets from the real world. We further assess on a dataset of synthetic data, which allows more exact manipulation of statistical heterogeneity and investigate its impact on convergence. We model system heterogeneity by making devices do varying degrees of local labour.

## 5.1. Dataset Description

**Datasets:** Table 1 summarizes the statistics from our study of four real datasets. Both older and more contemporary federated learning benchmarks (McMahan et al., 2017; Caldas et al., 2018) are used to compile these datasets. Using multinomial logistic regression, we investigate a convex classification issue using MNIST. In order to introduce statistical heterogeneity, we split the data across 1,000 devices with two-digit samples and a power-law distribution for the number of samples per device. Using the same approach, we then examine the FEMNIST dataset, which has 62 classes and is more complex than the original. We take into consideration a text sentiment analysis task using an LSTM classifier on Sentiment140 tweets in the non-convex context, where each device is represented by a Twitter account. We further explore the challenge of next-character prediction using the dataset of William Shakespeare's The Complete Works (McMahan et al., 2017). (Shakespeare).

**Baselines:** We evaluate proposed hybrid model in comparison to FedAvg (McMahan et al., 2017) and other aggregate-then-adapt baselines, such as FedProx (Li et al., 2021), FedBN (Li et al., 2021), MOON (Li et al., 2021), FedDyn (Acar et al., 2021), and FedGen (Zhu et al., 2021), all of which are state-of-the-art FL algorithms developed to deal with data heterogeneity. In order to prove that our suggested proposed for local-global knowledge matching work and collaborative data condensation, we research previous work that uses aggregation-free FL, namely FedDM (Xiong et al., 2023).

Table 1: Sample Dataset Details

| Dataset | Devices | Samples | Samples/device | |
|---|---|---|---|---|
| | | | Mean | Std.Dev |
| MNIST | 1000 | 69035 | 69 | 106 |
| Shakespeare | 143 | 517106 | 3616 | 6808 |
| FEMNIST | 200 | 18345 | 92 | 159 |

## 5.2. Proposed Model Training

Client extraction gets unstable when training data is highly Non-IID because of the skewed data. Convergence is slow because the global model is unable to quickly retrieve unknown information from the local training. The present study takes a shot at speeding up the global model's convergence by using the loss function-based client selection technique $\pi_{loss}$ to pick clients to train with larger values of the loss function.

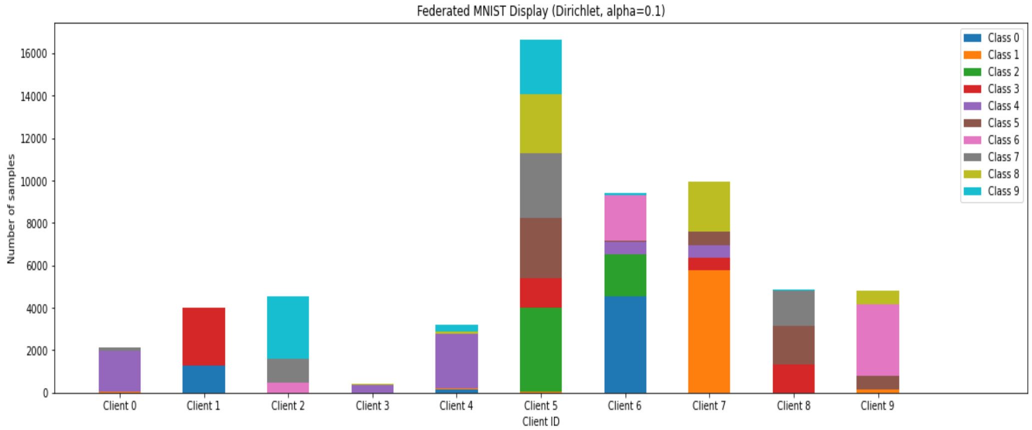


Figure 2: Federated Data from Various Client for Training Phase

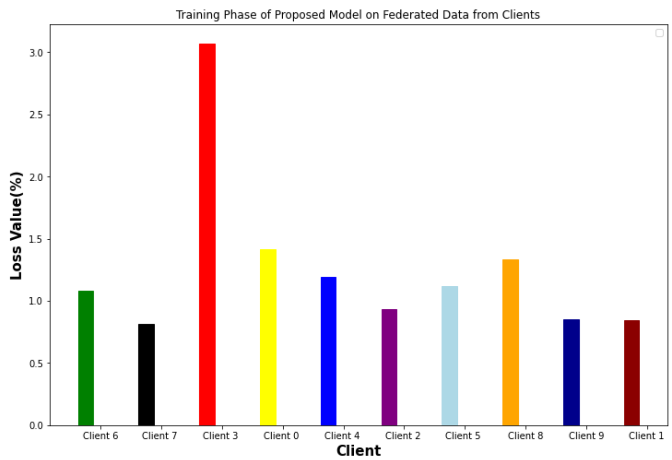


Figure 3: Training Loss Value of Various Clients

Figure 2 shows the data distribution using Non-IID, assuming there are ten clients and a single server. A higher selection probability is given to clients with a high loss by the introduction of $\pi_{loss}$, which helps to avoid the difficulty of global model convergence. By allowing the server to select a larger pool of clients with lower prediction abilities and higher loss rates for training, the convergence rate is bound to rise. When dealing with Non-IID data in particular, a high loss number can indicate that the client's training data is a small subset of the overall data. More clients with high loss values should be selected by the server to speed up convergence. The selection probability function and the importance weight of customer $v_k$ are components of the $\pi_{loss}$ approach. To start, we establish the significance of clients to training by defining the importance weight $v_k$. When choosing which clients to include in training, the server gives more weight to those with larger values of $v_k$. After that, in order to calculate the clients' selection probability, we build a selection probability function. Figure 3 shows the corresponding training loss value of the various clients with federated data.

### 5.3. Analysis of Statistical Heterogeneity

**Initialization of Hyperparameters**. In order to account for statistical heterogeneity in the tests, we follow the steps outlined in (Cohen et al., 2017) to create a subset of DomainNet that contains just the 10 most common classes across all domains. We set up six clients, each with data from a different domain, to simulate real-world situations where data heterogeneity is impacted, for example, by several hospitals employing different imaging protocols and equipment. In order to compare the accuracy of the final global model across all domains and the average accuracy across all domains, all algorithms are executed for 10 communication cycles. When it comes to the image learning rate, both FedProx and FedSplit use 1.0.

Table 2: Performance Comparison of Model Accuracy on Various FL Algorithm

| Methods | DomainNet | | | | | | |
|---|---|---|---|---|---|---|---|
| | **S** | **Q** | **P** | **C** | **R** | **I** | **Avg** |
| Proposed | **52.84** | **68.57** | **68.57** | **55.97** | **67.28** | **56.84** | **61.67** |
| FedAvg | 28.46 | 39.60 | 59.16 | 43.03 | 41.03 | 40.76 | 42.01 |
| FedProx | 29.18 | 38.13 | 60.22 | 44.81 | 41.55 | 43.76 | 42.94 |
| FedGen | 25.69 | 37.33 | 54.37 | 42.77 | 42.86 | 37.88 | 40.15 |
| FedDyn | 32.67 | 37.73 | 67.46 | 48.04 | 41.77 | 60.03 | 47.95 |
| MOON | 29.72 | 48.07 | 56.26 | 48.80 | 42.02 | 37.97 | 43.81 |
| FedDM | 46.69 | 62.37 | 60.58 | 52.28 | 52.45 | 41.38 | 52.62 |
| FedBN | 29.72 | 43.10 | 52.01 | 46.07 | 47.33 | 34.27 | 42.08 |

Global model accuracy comparison using several FL algorithms on the DomainNet dataset is illustrated in Figure 4. "Avg" indicates the average accuracy across domains. The domains are denoted by the symbols C (Clipart), P (Painting), I (Infograph), R (Real), Q (Quickdraw), and S (Sketch). The boldface and underline in each column represent the best and second-highest accuracies, respectively.

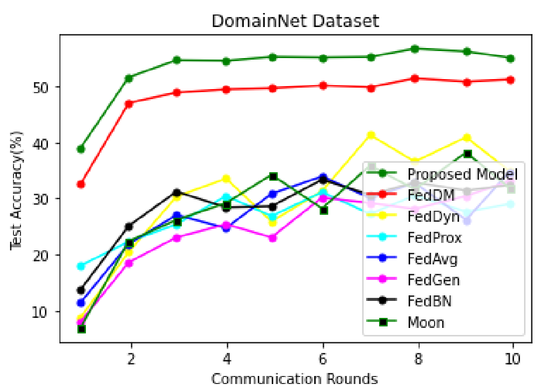


Figure 4: Performance Comparison of Convergence on DomainNet Dataset

**Performance of Convergence and Accuracy:** Figure 4 and Table 2 show that when compared to all other baseline approaches, hybrid FedProx: FedSplit has the best average accuracy and related convergence performance in all domains. In every category, this comparison shows that FedProx: FedSplit is far and away the best. In addition, as compared to current models, FedProx: FedSplit shows quicker convergence performance and more accuracy. For example, compared to FedDM, which takes ten rounds to obtain the same accuracy level, FedProx: FedSplit only takes two rounds, indicating an acceleration of 80%. Our collaborative data condensation and local-global knowledge matching methodologies validate the usefulness of integrating information from different fields, and these advantages underline their benefits.

### 5.4. Analysis of System Heterogeneity

To model the local training dataset for each client, we take into account K = 10 clients and divide the training split of each benchmark dataset into numerous data parts. In order to divide up the data among the customers, we use Dirichlet distribution. We take into account three levels of data heterogeneity, denoted as α=0.02, α=0.05, and α=0.1, for each benchmark dataset. The controllable parameter α determines the degree of heterogeneity. It should be noted that a lower α indicates a greater level of non-Independent and Identically Distributed (IID) in the data distribution across customers. In order to resemble challenging situations of data heterogeneity that may be faced in practical applications, we have selected these α values.

We use a local learning rate of 0.01 and a local batch size of 64 for 10 local epochs in our aggregate-then-adapt baseline approaches. To condense local data for FedProx: FedSplit, we use a batch size of 256, 1000 local update steps, and 50 images per class (IPC). To train the global model, we use 500 epochs, a batch size of 256, and a learning rate of 0.001. For each kind of data, we use the mean of the original, locally-sampled data as an initialization. The data learning rate for FedProx is configured as 0.2 for FMNIST, 1.0 for MINIST, and 0.1 for Shakespeare. To make sure FedSplit is stable during local data condensation, we use a data learning rate of 1.0 and clip the norm of gradients at 2.0. With ρ=5, we have set the global model re-sampling coefficient γ at 0.9.

## Model Accuracy

We begin by comparing the algorithms' top global model accuracy within 20 communication cycles. With respect to mean accuracy and variance, FedProx: FedSplit considerably surpasses all aggregate-then adapt baselines in a number of contexts, as shown in Table 3 and Figure 5, Figure 6 and Figure 7 for various dataset. The results show that on MINIST, Shakespeare, and FMNIST, respectively, FedProx: FedSplit improves performance by up to 25.44%, 17.91%, and 31.03% when compared to FedAvg. On the same datasets, FedProx: FedSplit still has a lead of up to 19.43%, 13.70%, and 17.74% compared to FedDyn, the top aggregate-then-adapt baseline. Also, on MNIST, Shakespeare, and FMNIST, respectively, FedAF achieves an accuracy advantage of 4.87%, 4.56%, and 2.17% over FedDM. Our collaborative data condensation and local global knowledge matching techniques are shown to be successful by FedAF's more noticeable performance with higher data heterogeneity, such as at α = 0.02.

Table 3: Performance Comparison of Model Accuracy on FMINIST, MINIST and Shakespeare Dataset

| **Methods** | **FMNIST** | **MNIST** | **Shakespeare** | **FMNIST** | **MNIST** | **Shakespeare** | **FMNIST** | **MNIST** | **Shakespeare** |
|---|---|---|---|---|---|---|---|---|---|
| | **α =0.02** | | | **α =0.05** | | | **α =0.1** | | |
| FedAvg | 56.50±5.55 | 39.71±1.15 | 30.80±2.20 | 69.14±5.84 | 46.51±3.07 | 33.37±0.75 | 82.19±5.67 | 56.15±4.62 | 39.97±1.53 |
| FedProx | 60.38±5.00 | 36.46±5.39 | 30.82±0.80 | 69.33±4.12 | 45.83±2.23 | 36.61±1.44 | 81.56±4:52 | 58.54 ±1.87 | 40.45±1.53 |
| FedGen | 61.44±2.07 | 36.61±1.06 | 29.20±2.09 | 75.48±1.83 | 42.72±2.11 | 33.56±3.91 | 82.29±2.53 | 58.17±2.84 | 40.23±1.06 |
| FedDM | 85.36±0.96 | 60.28 ± 0.82 | 44.15±0.30 | 86.08±0.68 | 62.97±0.96 | 46.27±0.98 | 86.65±0.31 | 64.88±0.35 | 47.05±0.13 |
| FedDyn | 69.79±5.04 | 45.73 ±3.98 | 35.01±2.07 | 75.19±5.49 | 57.68 ±1.84 | 39.10±0.34 | 84.73±2.7.74 | 59.97±2.20 | 41.81±1.46 |
| MOON | 51.33±7.00 | 33.32±1.13 | 33.41±0.70 | 71.41±4.08 | 47.41±4.59 | 37.90±0.80 | 81.61±2.68 | 57.62+4.99 | 40.24±0.68 |
| FedBN | 58.26±4.28 | 36.53±2.52 | 29.73±1.73 | 72.91±4.69 | 45.13. ± 22.18 | 33.73±2.15 | 77.33±3.07 | 57.67+3.21 | 39.84±0.20 |
| **Proposed Model** | **91.28 ±0.22** | **72.58±0.73** | **59.16±0.21** | **93.85±0.18** | **71.83 ± 0.54** | **53.86±0.22** | **92.68 ±0.31** | **72.58±0.56** | **65.27±0.13** |

**Convergence Performance.** In addition, we look at how fast FedProx: FedSplit converges compared to baselines; Figure 5, Figure 6 and Figure 7 shows the learning curves from three benchmark data sets of MNIST, Shakespeare and FEMNIST respectively. This shows that FedProx: FedSplit consistently outperforms other baseline methods when it comes to convergence speed, especially when dealing with highly heterogeneous data. As an example, FedProx: FedSplit outperforms other aggregate-then-adapt baselines in terms of accuracy in just two rounds when α = 0.02. On MINIST with α = 0.02, FedProx: FedSplit likewise maintains an advantage over FedDM. In contrast, FedDM takes fifteen rounds to obtain a mean accuracy of 62%, whereas FedProx: FedSplit only takes three rounds, indicating a speed improvement of 82% in convergence.

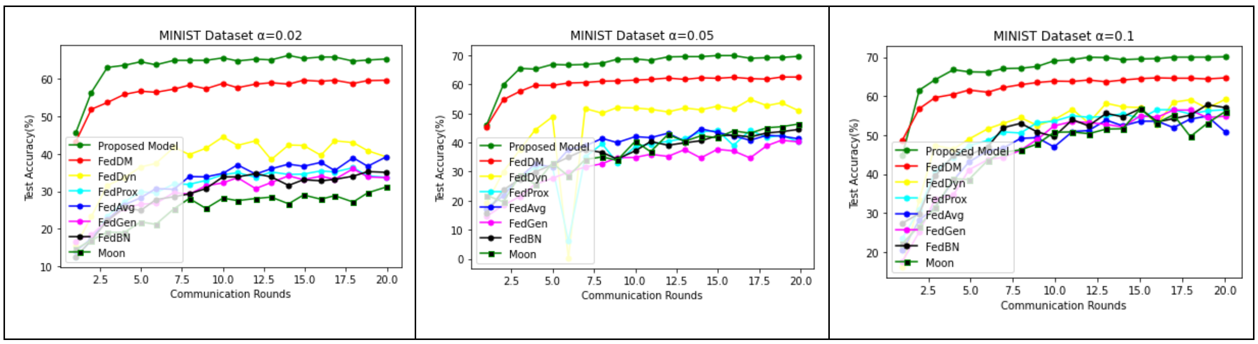


Figure 5: Performance Comparison of Convergence on MINIST Dataset

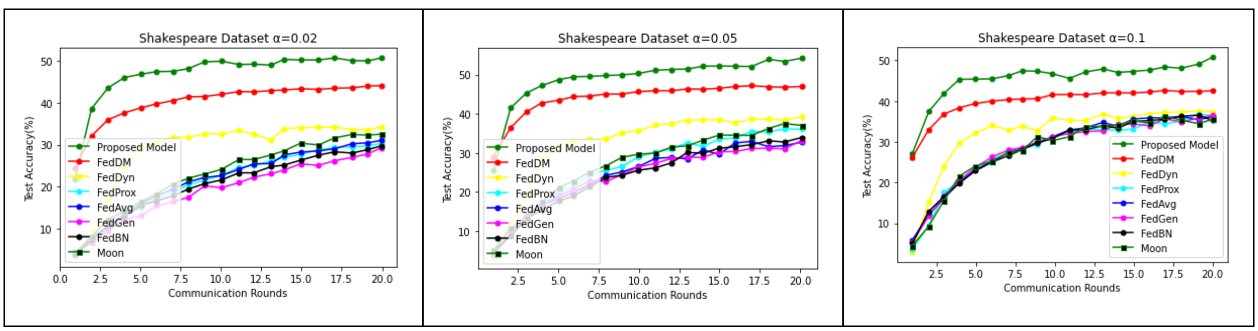


Figure 6: Performance Comparison of Convergence on Shakespeare Dataset

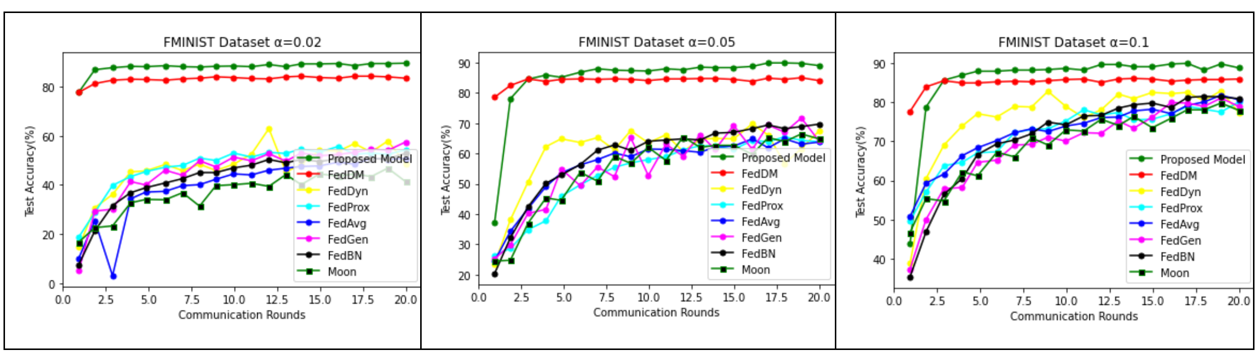


Figure 7: Performance Comparison of Convergence on FMINIST Dataset

## Analysis on Core Design

Core design's effect on learning MNIST's global model accuracy at three distinct heterogeneity levels (Table 4). This federated learning paradigm addresses heterogeneity difficulties via the use of two essential techniques: FedProx and FedSplit. In order to better understand how these two methods enhance performance, we carry out more tests on MNIST dataset. In particular, we evaluate the whole FedProx:FedSplit in comparison to two alternative setups that do not use any of the basic techniques. Additionally, we contrast it with FedDM, which does not use any of these methods. In comparison to results obtained using FedDM, the mean accuracy that may be achieved by utilizing a single basic approach alone still exhibits significant improvements (Table 4). Additionally, the mean accuracy is much better with the complete FedProx:FedSplit. These findings prove that FedProx:FedSplit can improve overall learning performance by encouraging the use of supplementary knowledge derived from information circulated across consumers.

Table 4: Analysis of Core Design on MNIST Dataset

| **Configuration** | **α=0.02** | **α=0.05** | **α=0.1** |
|---|---|---|---|
| FedProx:FedSplit | 68.98±0.54 | 69.54±0.41 | 70.48±0.38 |
| FedProx | 67.42±0.82 | 68.11±0.76 | 69.34±0.53 |
| FedSplit | 65.28±0.74 | 66.97±0.62 | 67.95±0.44 |
| FedDM | 60.28 ± 0.82 | 62.97 ±0.96 | 64.88±0.35 |

# 6 Conclusion

The practical issue of statistical and system heterogeneity in medical industry has been discovered in this study. The issue of diverse labels and heterogeneous of medical data has been addressed by introducing this proposed hybrid model. A hybrid FedProx: FedSplit was used to handle the statistical and systemic heterogeneity of medical data in federated networks, which is essential for dealing with the diverse patient data in FL. With the use of a proximal term, FedSplit can stabilize its process and distribute work among devices in varying quantities. In actual federated contexts, we provide the convergence guarantees for FedProx: FedSplit under the assumption of device dissimilarity. We have shown that the FedProx: FedSplit framework may greatly enhance the convergence behavior of federated learning in actual heterogeneous networks via our empirical assessment across a suite of federated datasets, which validates our research.

# References

[1] Acar, D. A. E., Zhao, Y., Navarro, R. M., Mattina, M., Whatmough, P. N., & Saligrama, V. (2021). Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*.

[2] Alistarh, D., Grubic, D., Li, J., Tomioka, R., & Vojnovic, M. (2017). QSGD: Communication-efficient SGD via gradient quantization and encoding. *In Advances in Neural Information Processing Systems*, 1709–1720.

[3] Amiri, M. M., & Gündüz, D. (2020). Federated learning over wireless fading channels. *IEEE Transactions on Wireless Communications*, *19*(5), 3546-3557.

[4] Bauschke, H. H., & Combettes, P. L. (2019). *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, 2nd edition.

[5] Brik, B., Ksentini, A., & Bouaziz, M. (2020). Federated learning for UAVs-enabled wireless networks: Use cases, challenges, and open problems. *IEEE Access*, *8*, 53841-53849.

[6] Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečný, J., McMahan, H. B., & Talwalkar, A. (2018). Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*.

[7] Chen, H. Y., & Chao, W. L. (2020). Fedbe: Making bayesian model ensemble applicable to federated learning. *arXiv preprint arXiv:2009.01974*.

[8] Chen, J., & Ran, X. (2019). Deep learning with edge computing: A review. *Proceedings of the IEEE*, *107*(8), 1655-1674.

[9] Chen, M., Shlezinger, N., Poor, H. V., Eldar, Y. C., & Cui, S. (2021). Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, *118*(17), e2024789118. https://doi.org/10.1073/pnas.2024789118

[10] Cohen, G., Afshar, S., Tapson, J., & Van Schaik, A. (2017). EMNIST: Extending MNIST to handwritten letters. *In International Joint Conference on Neural Networks (IJCNN)*, 2921-2926.

[11] Dinh, C. T., Tran, N. H., Nguyen, M. N., Hong, C. S., Bao, W., Zomaya, A. Y., & Gramoli, V. (2020). Federated learning over wireless networks: Convergence analysis and resource allocation. *IEEE/ACM Transactions on Networking*, *29*(1), 398-409.

[12] Gao, D., Yao, X., & Yang, Q. (2022). A survey on heterogeneous federated learning. *arXiv preprint arXiv:2210.04505*.

[13] Gao, L., Fu, H., Li, L., Chen, Y., Xu, M., & Xu, C. Z. (2022). Feddc: Federated learning with non-iid data via local drift decoupling and correction. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10112-10121.

[14] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., & Ramage, D. (2018). Federated learning for mobile keyboard prediction. https://doi.org/10.48550/arXiv.1811.03604

[15] Hsu, T. M. H., Qi, H., & Brown, M. (2019). Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

[16] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., & Cummings, R. (2019). *Advances and Open Problems in Federated Learning*. arXiv preprint arXiv:1912.04977

[17] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A. T. (2020,). Scaffold: Stochastic controlled averaging for federated learning. *In International Conference on Machine Learning*, 5132-5143.

[18] Kutlu, Y., & Camgözlü, Y. (2021). Detection of coronavirus disease (COVID-19) from X-ray images using deep convolutional neural networks. *Natural and Engineering Sciences*, *6*(1), 60-74.

[19] Kwon, D., Jeon, J., Park, S., Kim, J., & Cho, S. (2020). Multiagent DDPG-based deep learning for smart ocean federated learning IoT networks. *IEEE Internet of Things Journal*, *7*(10), 9895-9903.

[20] Li, L., Fan, Y., Tse, M., & Lin, K. Y. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, *149*, 106854. https://doi.org/10.1016/j.cie.2020.106854.

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

[22] Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., & He, B. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering*, *35*(4), 3347-3366.

[23] Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, *37*(3), 50-60.

[24] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, *2*, 429-450.

[25] Li, X., Jiang, M., Zhang, X., Kamp, M., & Dou, Q. (2021). Fedbn: Federated learning on non-iid features via local batch normalization. *In International Conference on Learning Representations*.

[26] Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y. C., Yang, Q., & Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *22*(3), 2031-2063.

[27] Lin, T., Kong, L., Stich, S. U., & Jaggi, M. (2020). Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, *33*, 2351-2363.

[28] Lin, Y., Han, S., Mao, H., Wang, Y., & Dally, W. J. (2017). Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887*.

[29] McMahan, B., Moore, E., Ramage, D., Hampson, S., & Y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *In Artificial Intelligence and Statistics*, 1273-1282.

[30] Mohandas, R., Veena, S., Kirubasri, G., Thusnavis Bella Mary, I., & Udayakumar, R. (2024). Federated Learning with Homomorphic Encryption for Ensuring Privacy in Medical Data. *Indian Journal of Information Sources and Services*, *14*(2), 17–23. https://doi.org/10.51983/ijiss-2024.14.2.03

[31] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated learning for internet of things: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, *23*(3), 1622-1658.

[32] Niknam, S., Dhillon, H. S., & Reed, J. H. (2020). Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, *58*(6), 46-51.

[33] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, (2019). Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977

[34] Peaceman, D. W., & Rachford, Jr, H. H. (1955). The numerical solution of parabolic and elliptic differential equations. *Journal of the Society for industrial and Applied Mathematics*, *3*(1), 28-41.

[35] Qu, Y., Pokhrel, S. R., Garg, S., Gao, L., & Xiang, Y. (2020). A block chained federated learning framework for cognitive computing in industry 4.0 networks. *IEEE Transactions on Industrial Informatics*, *17*(4), 2964-2973.

[36] Ryu, E. K., & Boyd, S. (2016). Primer on monotone operator methods. *Applied and Computational Mathematics, 15*(1), 3-43.

[37] Sery, T., & Cohen, K. (2020). On analog gradient descent learning over multiple access fading channels. *IEEE Transactions on Signal Processing*, *68*, 2897-2911.

[38] Shlezinger, N., Chen, M., Eldar, Y. C., Poor, H. V., & Cui, S. (2020). UVeQFed: Universal vector quantization for federated learning. *IEEE Transactions on Signal Processing*, *69*, 500-514.

[39] Sindhusaranya, B., Yamini, R., Manimekalai, M. A. P., & Geetha, K. (2023). Federated Learning and Blockchain-Enabled Privacy-Preserving Healthcare 5.0 System: A Comprehensive Approach to Fraud Prevention and Security in IoMT. *Journal of Internet Services and Information Security, 13*(3), 199-209.

[40] Tan, A. Z., Yu, H., Cui, L., & Yang, Q. (2022). Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, *34*(12), 9587-9603.

[41] Torroglosa-García, E., & Skarmeta-Gomez, A.F. (2017). Towards Interoperabilty in Identity Federation Systems. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 8*(2), 19-43.

[42] Udayakumar, R., Suvarna, Y.P., Yogesh, M.G., Vimal, V.R., & Sugumar, R. (2023). User Activity Analysis Via Network Traffic Using DNN and Optimized Federated Learning based Privacy Preserving Method in Mobile Wireless Networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 14*(2), 66-81.

[43] Wahab, O. A., Mourad, A., Otrok, H., & Taleb, T. (2021). Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems. *IEEE Communications Surveys & Tutorials*, *23*(2), 1342-1397.
[44] Wang, J., Liu, Q., Liang, H., Joshi, G., & Poor, H. V. (2020). Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, *33*, 7611-7623.
[45] Wu, Q., He, K., & Chen, X. (2020). Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. *IEEE Open Journal of the Computer Society*, *1*, 35-44.
[46] Xiong, Y., Wang, R., Cheng, M., Yu, F., & Hsieh, C. J. (2023). Feddm: Iterative distribution matching for communication-efficient federated learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16323-16332.
[47] Xu, J., Glicksberg, B. S., Su, C., Walker, P., Bian, J., & Wang, F. (2021). Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, *5*, 1-19.
[48] Yang, H. H., Liu, Z., Quek, T. Q., & Poor, H. V. (2019). Scheduling policies for federated learning in wireless networks. *IEEE Transactions on Communications*, *68*(1), 317-333.
[49] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(2), 1-19.
[50] Zheng, S., Shen, C., & Chen, X. (2020). Design and analysis of uplink and downlink communications for federated learning. *IEEE Journal on Selected Areas in Communications*, *39*(7), 2150-2167.
[51] Zhu, G., Wang, Y., & Huang, K. (2019). Broadband analog aggregation for low-latency federated edge learning. *IEEE Transactions on Wireless Communications*, *19*(1), 491-506.
[52] Zhu, H., Xu, J., Liu, S., & Jin, Y. (2021). Federated learning on non-IID data: A survey. *Neurocomputing*, *465*, 371-390.
[53] Zhu, Z., Hong, J., & Zhou, J. (2021). Data-free knowledge distillation for heterogeneous federated learning. *In International Conference on Machine Learning*, 12878-12889.

## Authors Biography

**Cina Mathew,** Experienced educator and researcher with 14 years of teaching experience. Holding a Master's degree in Computer Science and Engineering, she is currently pursuing a PhD in Machine Learning. Her passion lies in exploring innovative teaching methods and staying updated on the latest advancements in AI and ML. With a strong academic background and extensive teaching experience, she is dedicated to shaping the next generation of tech enthusiasts. She is excited to collaborate and contribute to the field of machine learning, driving progress and excellence in education and research."

**Dr.P. Asha** is an Professor of Computer Science and Engineering Department, Sathyabama Institute of Science and Technology, Chennai. She has 10+ years of working experience in Teaching and Research. Her area of Specialization includes Machine Learning, Internet of Things, Artificial Intelligence, Parallel Computing, Computer Graphics, Visualization and Data Mining. She has 100+ research publications and 60+ in Scopus/ Web of Science Journals and Conferences along with 6 Granted Patents and Copyrights. She serves as a reviewer for many Scopus and Web of Science Journals with High Impact factor. She has also published books on Machine Learning, Web of Things, Data Analytics with Python and Industry 4.0 support by Machine Learning.