Journal of Internet Services and Applications, 2025, 16:1, doi: 10.5753/jisa.2025.5055

Plant Disease Detection Using Federated Learning and Cloud Infrastructure for Scalability and Data Privacy

Paulo V. Caminha [**Federal University of ABC** | ***paulo.caminha@ufabc.edu.br***]
Helder May Nunes da Silva Oliveira [**University of São Paulo** | ***helderoliveira@ime.usp.br***]

Center for Engineering, Modeling and Applied Social Sciences (CECS), Federal University of ABC, Avenida dos Estados, 5001 - Bairro Bangu, Santo André - SP, 09280-560, Brazil.

Received: 06 October 2024 • **Accepted:** 11 June 2025 • **Published:** 01 September 2025

Abstract Agriculture faces significant challenges from crop diseases, making early and accurate detection critical. Federated Learning (FL), an advancement in artificial intelligence (AI) and machine learning (ML), presents a promising solution by enabling collaborative model training on decentralized data without the need to share sensitive information. This article examines the application of FL in detecting plant diseases through image analysis, highlighting the role of cloud computing in addressing challenges related to data processing, storage, and model scalability. By leveraging decentralized data stored and processed in the cloud, FL develops robust models that not only improve detection accuracy but also generalize effectively to new data, promoting knowledge sharing while ensuring data privacy. The integration of cloud infrastructure enables FL to scale, providing resilience and productivity gains in agricultural practices. The results show that the proposed approach achieves a 99.71% accuracy using the VGG16 model after Federated Learning aggregation, while preserving data confidentiality, enhancing agricultural resilience, and benefiting from the scalability and flexibility offered by cloud computing.

Keywords: Federated Learning, Distributed Communication, Data Privacy, Disease Detection, Agricultural Productivity.

1 Introduction

Agriculture plays a crucial role in global food security but continuously faces challenges from crop diseases that threaten both yield and quality [Moreira *et al.*, 2022]. It is estimated that 20-40% of global crop yields are lost annually due to pests and diseases, directly impacting food security and national economies [CABI, 2023]. Early and accurate detection of these diseases is critical to minimize damage and safeguard agricultural output [Basseto *et al.*, 2022]. Traditional methods, such as manual inspection, are time-consuming, labor-intensive, and prone to human error, particularly in regions with limited infrastructure or connectivity [Shoaib *et al.*, 2023].

Recent advancements in artificial intelligence (AI) and machine learning (ML) have revolutionized various sectors, including agriculture [Silva and Cavichioli, 2022]. However, conventional ML approaches often rely on centralized data collection, which poses challenges related to data privacy, transfer costs, and data availability. In agriculture, where data is often proprietary and geographically distributed, these challenges are particularly pronounced. In the context of Federated Learning (FL), decentralized data is utilized to train models collaboratively across multiple clients, without sharing sensitive information [Banabilah *et al.*, 2022]. Non-IID (non-Independent and Identically Distributed) data, where each client has data with varying distributions, is a critical aspect of this process. Through FL, it is possible to improve disease detection accuracy while maintaining data privacy and addressing the scalability limitations of traditional methods [Mamba Kabala *et al.*, 2023].

Moreover, disease information varies based on geography, plant species, and weather conditions. FL facilitates the aggregation of insights from diverse sources, ensuring adaptability to regional and crop-specific nuances [Souza *et al.*, 2024]. This adaptability is crucial for creating models that generalize across different agricultural environments, thereby improving the robustness of disease detection systems [Mamba Kabala *et al.*, 2023].

By utilizing decentralized data in the cloud, FL can build robust models that enhance both detection accuracy and generalization [Mamba Kabala *et al.*, 2023]. Cloud infrastructure also enables continuous model updates and deployment, allowing real-time improvements in disease detection as new data becomes available. Furthermore, FL, supported by cloud infrastructure, reduces the need to transfer large datasets to a central location, minimizing communication overhead and dependence on robust connections [Li *et al.*, 2020]. This is especially beneficial in rural or isolated areas where connectivity is intermittent or limited.

Additionally, stringent regulations govern the protection of personal and business data in many jurisdictions. FL aligns with these regulations by enabling collaborative model training without exposing sensitive data directly [Li *et al.*, 2020]. This compliance ensures that agricultural entities can improve disease detection models while maintaining competitive advantages and adhering to legal requirements. Thus, the FL approach enhances data utilization efficiency while ensuring data security and privacy.

In this article, we investigate the communication and distribution challenges inherent to decentralized data management and processing demands. The decentralized nature of FL is particularly relevant in agricultural scenarios where data heterogeneity and information sensitivity require strategies that preserve local data privacy while benefiting from distributed knowledge across different crops, such as strawberries and peppers. This decentralized approach offers a fresh perspective on agricultural disease detection, fostering knowledge sharing among disparate data sources while ensuring data confidentiality.

1.1 Problem Statement

Crop diseases pose a significant challenge to global agricultural productivity, often resulting in substantial yield losses. Traditional disease detection methods are manual, time-consuming, and error-prone, especially in rural areas with limited connectivity and infrastructure. Cloud-based FL addresses these challenges by reducing the need for transferring large datasets, providing a scalable solution for real-time disease detection in agricultural environments. These limitations impede early detection, which is crucial for minimizing damage. Furthermore, sharing sensitive agricultural data, such as disease information, introduces risks to data privacy and security, particularly when proprietary data is involved. In this context, applying FL offers a decentralized solution that ensures data privacy while enabling collaborative model training. However, implementing FL in agricultural settings introduces challenges such as heterogeneous data sources, varying data quality, and non-IID data distributions. Moreover, cloud infrastructure integration is essential to support scalability and continuous model updates. This article addresses these challenges by proposing a cloud-based FL framework for plant disease detection, designed to handle heterogeneous and non-IID data from various agricultural regions.

1.2 Contributions

This article presents several key contributions to agricultural disease detection using FL:

- **Decentralized Model Training:** The proposed framework leverages FL to enable agricultural entities to collaboratively train disease detection models without sharing sensitive data, preserving privacy while enhancing model performance.
- **Cloud Integration for Scalability:** We integrate cloud infrastructure to manage the computational and storage demands of FL, ensuring scalable and efficient model deployment in agricultural environments.
- **Evaluation Across Multiple Models:** Comparative analysis of several machine learning models (CNN, VGG16, DenseNet121, MobileNet, and ResNet50) is provided, highlighting their performance both before and after FL aggregation.
- **Handling of Non-IID Data:** Our methodology addresses the challenge of heterogeneous and non-IID data from various agricultural environments through weighted averaging during model aggregation, with additional strategies to balance discrepancies in data quality.
- **Practical Application in Diverse Crops:** The framework is applied to real-world datasets across a range of crops, including strawberries and peppers, demonstrating the generalizability and robustness of the proposed approach in diverse agricultural settings.

1.3 Organization of this Article

The article is organized as follows: Section 2 reviews methodologies for agricultural disease detection, including both conventional and advanced detection technologies. Section 3 outlines the process adopted, detailing data collection, model architecture, and training techniques. Section 4 provides details on model training and weight management. Section 5 presents experimental results on disease detection accuracy. Finally, Section 6 discusses study limitations and suggests future directions for FL in plant disease detection.

2 Related Works

Advancements in data mining and machine learning have revolutionized crop disease detection and management. FL has further enhanced these capabilities by addressing critical issues related to data privacy, availability, and transfer costs associated with centralized data collection. Despite growing interest, research on FL for plant disease identification remains limited. This section reviews the literature on machine learning for plant disease detection and highlights the emerging role of FL in this field.

Mamba Kabala *et al*. [2023] investigated the application of FL in crop disease detection using image analysis. Their work focuses on developing and analyzing Convolutional Neural Network (CNN) models and attention-based models, particularly Vision Transformers (ViT). The study underscores the significant potential of FL in crop disease detection. However, our article adopts a broader perspective by comparing multiple machine learning models and their applications in FL.

Mehta *et al*. [2023a] addressed the challenge of managing mango leaf diseases, introducing a novel FL-CNN (Federated Learning Convolutional Neural Network) model to classify mango leaf diseases into four severity levels. In contrast, our work provides a more comprehensive comparison of multiple machine learning models and their applications within a FL context.

Aggarwal *et al*. [2023] tackled the issue of rice leaf diseases, which severely affect rice production, particularly in Asia. The authors introduced a lightweight FL architecture for classifying rice leaf diseases, prioritizing data privacy. Their study explored the performance of FL models on both independent and identically distributed (IID) and non-IID datasets. In contrast, our article offers a broader evaluation of various machine learning models and their use cases in plant disease detection.

Mehta *et al*. [2023b] presented an FL-CNN model for detecting and categorizing tomato leaf diseases into four severity levels and one healthy level. This work emphasizes the importance of early and accurate plant disease detection to mitigate damage and safeguard food security.

Table 1. State-of-the-art characteristics in the federated learning literature

Literature	FL Architecture	Aggregated Model	Accuracy	Computational Efficiency	Non-IID Data	Compression Techniques	Security
Mamba Kabala *et al*. [2023]	✓	✓	✓		✓		✓
Mehta *et al*. [2023a]	✓	✓	✓		✓		✓
Aggarwal *et al*. [2023]	✓	✓	✓	✓	✓		✓
Mehta *et al*. [2023b]	✓	✓	✓		✓		✓
Our Research	✓	✓	✓	✓	✓	✓	✓

Based on our analysis of the state-of-the-art literature, we conclude that significant progress has been made in leveraging machine learning techniques for plant disease detection, with FL emerging as a promising approach to enhance these capabilities. While individual studies offer valuable insights into specific models and applications, our comprehensive review underscores the broader potential of FL across various machine learning models and agricultural contexts.

3 Methodological Procedures

In this section, we outline the methodological procedures employed to develop the plant disease identification framework using image data, encompassing data collection, preparation, and the implementation of FL.

As depicted in Figure 1, Phase A presents a dataset consisting of approximately 54,305 images of healthy and diseased plants, spanning 14 different crop types. This dataset includes images of both healthy leaves and leaves affected by agricultural pests, covering crops such as apples, cherries, corn, grapes, peaches, potatoes, tomatoes, strawberries, and peppers. To classify these plants, five Convolutional Neural Network (CNN) models were trained: a traditional CNN, VGGNet, DenseNet121, MobileNet, and ResNet50. These models were selected for their diverse strengths in feature extraction, computational efficiency, and ability to handle complex image classification tasks.

In Phase B, the trained models are evaluated to verify their accuracy and effectiveness. Once validated, the models are deployed in real-world agricultural environments. A cloud infrastructure is essential for deploying and coordinating the models across various devices, such as mobile phones, drones, and cameras. This infrastructure ensures efficient model parameter updates and transmission, even under varying connectivity conditions.

Phase C involves capturing new plant images using mobile devices, drones, cameras, and other tools to further train the models. The cloud infrastructure supports continuous data integration from these sources, allowing the models to be updated and refined while maintaining data privacy. In this case, two clients with distinct crops—peppers and strawberries—operate independently, without sharing data or crop types, while cloud services synchronize updates system-wide.

Phase D focuses on two specific crops under study: pepper and strawberry plantations. Although the study involves only two users, the system is scalable, allowing multiple devices to be integrated within a single farm to build a comprehensive database. The cloud platform efficiently aggregates data from various sources, contributing to a robust centralized model while preserving data privacy.

In Phase E, a new dataset is generated for each user, containing images of healthy and diseased plants, forming the foundation for FL training. Dataset A corresponds to Farm A with pepper plants, while Dataset B corresponds to Farm B with strawberry plants.

In Phase F, local models are trained separately for each user: Model A on Dataset A and Model B on Dataset B, ensuring no data sharing between them. The cloud infrastructure supports this distributed training process by securely aggregating model updates, thereby preserving data privacy. After five local training rounds, the trained models are aggregated using only the model weights, without revealing the underlying datasets. The cloud infrastructure facilitates this secure aggregation process. Phase H merges the aggregated models with the initial global models.

Finally, in Phase I, the aggregated models are evaluated to determine their accuracy in classifying diseases in pepper and strawberry plantations, which were not part of the initial global training. Validation metrics such as accuracy, sensitivity, precision, and loss rate are applied, similar to the initial global model evaluation. If the new models meet or exceed the performance of the original models, they are integrated into the system through horizontal FL. These improved models, augmented with pepper and strawberry data, become the new global models, enabling secure and distributed classification across farms. If the new models do not meet performance criteria, they are discarded, and the original global models are retained until new models achieve over 95% accuracy, precision, and sensitivity, with a loss rate below 1%.

3.1 Data Collection and Preparation

Data collection and preparation are critical steps for FL-based disease detection in plant images. We utilized an online database[1] containing more than 54,305 images of leaves from 14 plant species captured under controlled conditions. The dataset includes both healthy leaves and those affected by agricultural diseases, covering a range of crops such as apples, cherries, corn, grapes, peaches, potatoes, tomatoes, strawberries, and peppers. During the data selection process, plants without representations in both categories (healthy and diseased) were excluded. This resulted in the selection of 18 diseased plant types and 9 healthy types, ensuring adequate representation of real-world conditions.

After collection, the data underwent rigorous preprocessing, including standardizing image formats and normalizing

[1] https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset

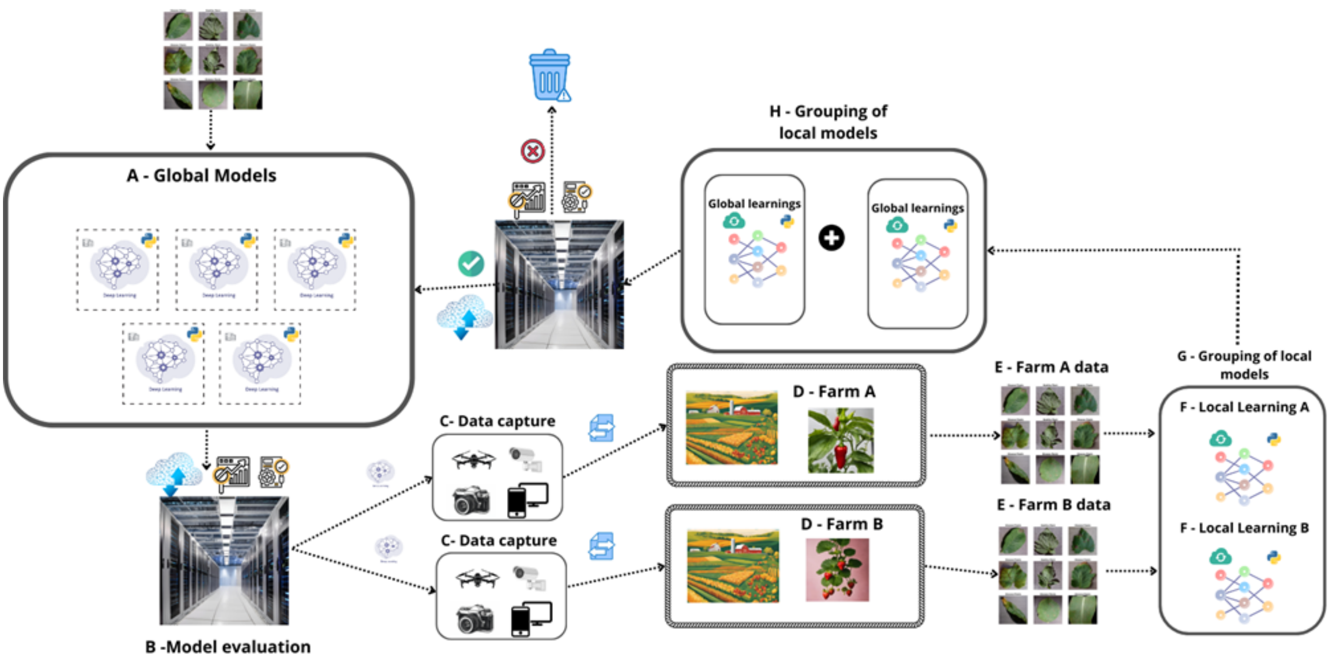

Figure 1. Flow of the applied methodology

pixel values to the range [0, 1]. Noise reduction techniques were applied to enhance data quality, and the accuracy of image annotations, indicating the presence of specific diseases, was verified to ensure training dataset reliability. The cloud infrastructure facilitated scalable data processing, ensuring rapid and efficient preprocessing, with data made accessible for training across devices. This phase was crucial for establishing a solid foundation for training models capable of effectively distinguishing between healthy and diseased plants.

3.2 Federated Learning Implementation

Model training was conducted using Horizontal FL, enabling distributed training across multiple locations without sharing raw data. This approach is crucial for ensuring data privacy in sensitive agricultural environments. The cloud infrastructure facilitates secure model parameter aggregation from different locations. Initially, global models were trained using aggregated data from several plantations, allowing them to learn general patterns in leaf images from various crops.

Next, local training was conducted on two datasets—one for strawberry cultivation and another for pepper cultivation. Local models were trained on data specific to each crop, preserving local data privacy. The cloud infrastructure securely aggregated updates from the locally trained models, updating the global models without sharing raw data. This approach allowed global models to integrate knowledge from various agricultural contexts, effectively improving their ability to detect diseases in new plant images while maintaining data confidentiality. The cloud infrastructure also enables continuous updates to the global models, ensuring they remain accurate as new data is introduced.

3.3 Model Architecture e Evaluated Learning Models

Five machine learning models based on CNNs were selected to evaluate the proposed framework:

Convolutional Neural Networks (CNNs) are well-suited for analyzing plant images due to their ability to learn hierarchical visual representations. They utilize convolutional layers to extract features from images, pooling layers to reduce dimensionality, and fully connected layers for classification. This approach is effective for identifying complex patterns in plant leaves, essential for automated plant disease diagnosis [**?**].

VGGNet, known for its depth and performance, uses deep convolutional layers with small filters (3x3) and max pooling to analyze detailed features in plant leaves. This architecture excels in identifying disease symptoms and nutritional deficiencies [**?**]. DenseNet121, featuring densely connected layers, balances depth and efficiency. It fosters high information flow between layers, enabling detailed analysis of intricate leaf features [**?**].

MobileNet, optimized for mobile devices, uses depth-separable convolutional layers to minimize computational costs while retaining the ability to extract important features. This model is ideal for real-time diagnostics in mobile and embedded systems [**?**]. ResNet, with its ”residual connections,” addresses the challenge of training deep networks and is effective for visual learning, even with limited data. It excels at detecting subtle features in plant images, aiding in accurate diagnoses [**?**].

Each model was trained and evaluated for its accuracy in detecting pests, allowing for an objective performance comparison.

Based on the strengths observed in these architectures, a custom CNN-based model was developed to further explore the capabilities of deep learning in plant disease detection. The adopted architecture integrates core principles from the evaluated models to optimize feature extraction and classification performance.

The initial layers normalize the input data and prepare it for training. Normalizing pixel values to [0, 1] accelerates model convergence, while convolutional layers (*Conv2D*) detect distinctive image features. The *ReLU* activation function introduces non-linearity, enabling the model to learn complex patterns in leaf images.

Pooling layers (*MaxPooling2D*) reduce the dimensionality of the extracted features while retaining the most relevant information for classification. Fully connected (*Dense*) layers with *ReLU* activation are employed to learn more abstract representations of the extracted features. Regularization techniques such as *Dropout* are applied to prevent overfitting, ensuring the model generalizes well to new data.

The model's output layer uses sigmoid activation for binary classification, determining whether a leaf is healthy or affected by pests. This architecture was optimized for accuracy and efficiency in detecting various pests in plant images.

4 Federated Learning: Model Training and Weight Management

In this section, we detail how the weights of federated models were managed, from initialization to aggregation, using advanced techniques to ensure both effectiveness and fairness in the combination process. Each step follows specific and consistent standards across local models and the global model to guarantee the integrity and robustness of the FL process.

4.1 Mathematical Representation of Federated Learning Model

In this article, FL is employed to detect diseases in plant images across different agricultural environments. FL allows multiple farms (clients) to collaboratively train a global machine learning model without sharing their raw data, thus preserving data privacy and accommodating the diverse agricultural conditions in different locations.

Mathematically, the objective of FL in this context is to minimize a global loss function $\mathcal{L}(w)$, representing the model's error across all participating farms. Each farm k, from a total of K farms, holds its own local dataset $\mathcal{D}_k$, containing images of healthy and diseased plants specific to the crops in that location (e.g., peppers and strawberries). The global optimization problem is defined as:

$$\min_w \mathcal{L}(w) = \sum_{k=1}^{K} \frac{n_k}{n} \mathcal{L}_k(w) \tag{1}$$

where:

- w represents the parameters of the global machine learning model.
- $\mathcal{L}_k(w)$ is the local loss function computed on the dataset $\mathcal{D}_k$ of farm k, measuring the model's error in predicting plant health or disease.
- n_k is the number of images in farm k's dataset, and $n = \sum_{k=1}^{K} n_k$ is the total number of images across all farms.

During each round of training, each farm k independently minimizes its local loss function $\mathcal{L}_k(w)$ using its specific dataset, reflecting the unique agricultural conditions and disease patterns at that location. The farm computes the gradient $\nabla\mathcal{L}_k(w)$, which indicates the direction in which the model parameters should be adjusted to reduce the local loss.

These locally computed gradients are then sent to a central server, which aggregates them from all participating farms. The aggregation is performed using a weighted average that accounts for the size of each farm's dataset:

$$w^{(t+1)} = w^{(t)} - \eta \sum_{k=1}^{K} \frac{n_k}{n} \nabla\mathcal{L}_k(w^{(t)}) \tag{2}$$

Here, η is the learning rate, controlling the step size of the parameter updates, and t denotes the current training iteration. The updated global model parameters $w^{(t+1)}$ are then distributed back to the farms, where they are further refined using local data in subsequent training rounds.

This federated approach is particularly beneficial in agriculture, where data privacy is critical, and disease patterns can vary significantly between different farms and regions. By enabling each farm to contribute to a shared model without exposing its raw data, FL enables the creation of a robust global model that generalizes well to diverse plant diseases across various environments. This model can then be used to improve the accuracy of disease detection in new plant images, enhancing both local and global agricultural productivity while preserving the confidentiality of each farm's data.

4.2 Initialization, Update, and Aggregation of Weights

All models, whether local or global, were initialized leveraging TensorFlow's default initialization mechanism. This process assigns random weights to the parameters of convolutional and dense layers, setting the groundwork for learning meaningful data representations.

Throughout the training process, local and global models were iteratively updated using the Adam optimizer, configured with a learning rate of 0.0001 to ensure stable convergence and optimal performance. Adam was selected due to its dynamic adjustment of learning rates and proven effectiveness in various deep learning tasks. Training was conducted over 20 epochs, utilizing the 'BinaryCrossentropy' loss function, which is well-suited for binary classification tasks like distinguishing between healthy and diseased plants. Performance metrics such as accuracy, precision, and recall were continuously monitored throughout training.

The combination of local model weights with the global model was performed using horizontal aggregation, implemented in the function 'combine_horizontal_models'. This function ensures that the weight combination follows the same standards as the global model. In cases where the dimensions of local layer weights did not match those of the global model, the local weights were concatenated with the global weights. This adjustment, primarily applied to dense layers, resized the global layer to incorporate the local layer weights, ensuring that information from local models is adequately integrated into the global model.

For layers with matching dimensions, the weights were summed using the Flower weighted averaging technique. Each client's contribution was adjusted based on the number of samples provided, assigning greater weight to clients with more data. This allowed for a more balanced and representative combination of local models, taking into account the amount of data available from each client during global weight updates.

These techniques were consistently applied to both local and global models, ensuring that the weight updates remained harmonious and effective throughout the training process.

4.3 Handling Imbalanced and Non-IID Data with Wasserstein Distance and Flower Weighted Averaging

Federated Learning environments often face considerable challenges due to data distribution discrepancies among clients, leading to imbalanced and non-IID (non-independent and identically distributed) datasets. To address discrepancies in data distributions among clients, a combined approach was adopted, integrating Wasserstein distance and the Flower weighted averaging technique.

Wasserstein distance, also known as the "Earth Mover's Distance," was used to measure the difference between each client's data distribution and an ideal uniform distribution, where all classes would have equal representation. This technique helped identify and quantify imbalances in each client's local data distributions.

During the local model aggregation process, Wasserstein distance was applied to weigh the contribution of each client to the global model. Clients whose data distributions were farther from the ideal uniform distribution had their contributions reduced proportionally. This helped prevent models from clients with imbalanced data from excessively influencing the global model, which could harm the model's ability to generalize to other clients.

In addition, the Flower weighted averaging technique was used to further adjust the impact of each client in the model aggregation. The weighted averaging considers the number of samples from each client, adjusting the contribution of each local model to the global model according to the size of their data. This helped balance the influence of clients, especially in cases where some clients had more data than others, preventing clients with fewer data from having disproportionate influence.

By combining Wasserstein distance with Flower weighted averaging, the approach ensured a more robust and adaptive aggregation, taking into account both the imbalance in data distributions and the relative size of each client's data. This combination resulted in a more stable global model, with less variability between training rounds, and a better ability to generalize, even with significantly different local data distributions.

This approach proved effective in handling imbalanced and non-IID data in federated learning scenarios. To simulate a realistic non-IID environment in the scenario with five users, we adopted a label-skewed partitioning strategy during the data distribution phase. Specifically, the dataset was divided among five clients such that each user received a different proportion of the two main classes: healthy and diseased crops. This was implemented using a Dirichlet distribution with a concentration parameter $\alpha = 0.3$, which controlled the degree of heterogeneity. Lower α values resulted in higher non-IIDness, concentrating more samples of a specific class per client. For instance, clients 2 and 5 received over 90% of diseased samples, while clients 1 and 3 had nearly balanced datasets. This method ensures that the local datasets reflect real-world disparities, where certain regions or farms might predominantly observe one condition over another.

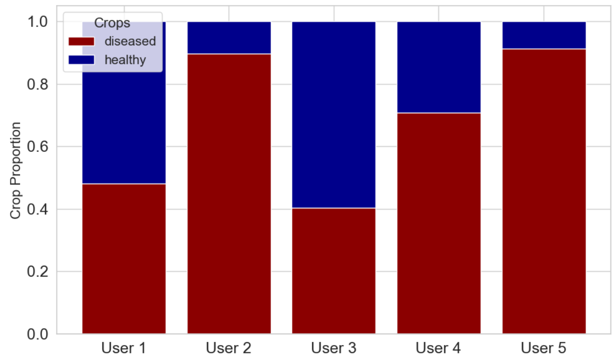

Figure 2. Crop Distribution per User

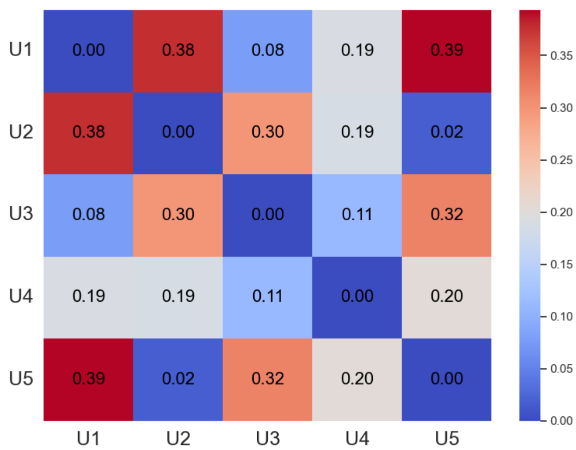

Figure 3. Wasserstein Distance Matrix Between Users

Figure 2 presents the proportional distribution between diseased and healthy crops for each of the five users in the system. It is observed that users 1 and 3 have more balanced distributions, with proportions close to 50% for both categories. On the other hand, users 2 and 5 show a significant predominance of diseased crop images, with more than 90% of the samples belonging to this class, while user 4 exhibits an intermediate distribution, with 71% of the samples being diseased. This heterogeneity in the distributions reveals a fundamental characteristic of the federated learning scenario: the non-independence and non-identical nature (non-IID) of the data across clients. The disparity in class representation among users is a critical factor, as it can affect the overall performance of the global model and contribute to biases during the aggregation of local models.

This form of non-IID partitioning, based on label-skewed distribution, is illustrated in Figure 2 and reflects real-world scenarios in which different regions exhibit varying prevalence of certain classes. Such heterogeneity directly impacts the performance of the federated model. For example, clients such as users 2 and 5, who possess more than 90% of data from the ”diseased” class, tend to bias the global model toward this class, potentially compromising accuracy in more balanced contexts. In contrast, clients like users 1 and 3, with more evenly distributed data, provide a broader representation of the problem space, contributing positively to generalization. As the disparity between local distributions increases, the risk of bias in the global model and instability during training also rises. Therefore, mechanisms such as the use of Wasserstein distance are essential, as they quantify these disparities and enable dynamic adjustment of each client's contribution during model aggregation. This helps mitigate the adverse effects of heterogeneity, promoting a fairer balance between local updates and enhancing the robustness and generalizability of the global model.

It is important to note that the application of Wasserstein distance in this context does not modify the size or composition of each client's local dataset. Instead, it operates solely at the model aggregation level by adjusting the weight of each client's contribution to the global model based on the degree of imbalance in their data distribution. Clients with distributions closer to the reference (e.g., a balanced 50/50 class split) are assigned higher weights, while those with highly skewed distributions receive lower influence during aggregation. This approach preserves data privacy and decentralization, while mitigating the adverse effects of statistical heterogeneity without discarding or resampling any local data. As a result, the technique enhances fairness in the training process and improves the robustness of the global model without requiring any modifications to the underlying datasets.

Figure 3 illustrates the Wasserstein distance matrix, quantifying the divergence between the crop distributions of different users. This metric is particularly useful for assessing the degree of dissimilarity between the probability distributions of local data in a federated learning context. It is noticeable that the smallest distances occur between users with similar distributions, such as between users 2 and 5 (distance of 0.02), both of whom have a strong predominance of diseased crops. In contrast, the largest distances are observed between users with opposing profiles, such as users 1 and 5 (0.39), and users 3 and 5 (0.32). These differences highlight the statistical variability among the network nodes, reinforcing the importance of robust aggregation techniques to mitigate the effects of class imbalance and ensure fairness in global model training. The matrix analysis also serves as a diagnostic tool for personalization strategies or regrouping of users with similar profiles.

To mitigate the impacts of heterogeneity in local user distributions and improve the performance of the global model in a federated scenario with non-IID data, specific balancing and regularization strategies were adopted. Initially, a weighting mechanism was implemented in local updates, assigning greater importance to the contributions of users with more representative or balanced class distributions. Additionally, regularization techniques based on Wasserstein distances were applied, allowing the penalization of local updates that deviated excessively from the global average distribution. This approach aimed to smooth out discrepancies among client data, reducing the negative influence of highly skewed distributions, such as those of users 2 and 5. As a result, the model's convergence became more stable and fair, ensuring better generalization and performance across the diverse crop scenarios.

4.4 Regularization and Prevention of Overfitting

To address potential overfitting, a Dropout layer was incorporated into the model's architecture with a rate of 0.2, effectively reducing model complexity during training. The Dropout technique randomly disables 20% of the units during training, encouraging the model to learn more robust and generalizable representations. This technique was consistently applied in both local and global models, ensuring that all models were trained to avoid overfitting to specific client data and to improve generalization.

In this study, we detailed the methodology for training and combining weights in federated models, ensuring that all models—both local and global—followed the same initialization, updating, and aggregation standards. The consistent application of weighting, regularization, and combination techniques helped ensure effectiveness and fairness in the FL process. While the approach provides a solid foundation, future work may explore advanced techniques to enhance resistance to non-IID data and optimize weight combination in federated scenarios.

5 Experiments and Results

This section presents the results obtained from two distinct evaluation scenarios: *(i)* a performance comparison among the primary machine learning models based on Convolutional Neural Networks (CNNs) and *(ii)* an in-depth analysis of the FL process, emphasizing model performance before and after horizontal aggregation. Several traditional machine learning techniques were considered, including CNN, VGGNet, DenseNet121, MobileNet, and ResNet50. The experiments were conducted with federated data distributed among 2 and 5 users, allowing the evaluation of models' behavior and performance.

5.1 Models Before and after Federated Learning Aggregation

During the training phase, a dataset comprising 14,592 images of leaves infected by pathogens and 4,950 images of healthy leaves was utilized. This dataset, containing 19,542 samples from seven distinct plant species, was horizontally partitioned for evaluation in FL. Two experimental scenarios were designed: (i) a two-user configuration with pepper and strawberry crops, and (ii) a five-user setup with different crop assignments. Initially, the global model was pre-trained using data from the remaining plant types, which were not shared with the users. These retained datasets were allocated separately. We then applied FL to enable collaborative training between the two users and performed model aggregation using standard FL techniques. Subsequently, we extended the experiment to a five-user setting, where each user had access to data from a unique crop: apple, pepper, strawberry, tomato, or grape. In this second scenario, we further limited the model's initial exposure to the datasets, simulating a more distributed.

Figure 4 presents leaf samples used for training and validating the machine learning model in the context of plant disease detection. The image features nine leaves arranged in three rows and three columns, with each leaf classified as either healthy or diseased. The leftmost column displays examples of healthy leaves labeled as "healthy plant," characterized by the absence of stains or discoloration. The other columns show diseased leaves labeled as "diseased plants," exhibiting symptoms like dark spots, necrosis, yellowing, or damaged edges.

This dataset is crucial in building robust and accurate models, especially within FL frameworks where different users contribute localized data. By incorporating both diseased and healthy leaves from diverse plant types, the dataset ensures that the models generalize effectively to new data in real-world applications.

The parameters presented in Table 2 were selected based on practical testing, and both centralized and federated models share the same values, reflecting an effort to maintain a consistent approach across both scenarios. A learning rate of 0.0001 and a batch size of 32 proved to be the most effective in both cases, ensuring good convergence and training stability. The image resolution of 160x160 pixels with 3 color channels (RGB) was sufficient to capture essential information, while the model architecture, composed of Conv2D, MaxPooling2D, Dense, and Dropout layers, was kept identical, indicating a robust and efficient structure. The number of training steps was slightly adjusted for the federated models (30 steps, compared to 25 in the centralized case) to accommodate the effects of decentralization. Although this standardization facilitates direct comparison between scenarios, it may constrain the optimal performance of federated models, which could naturally benefit from scenario-specific hyperparameter tuning.

5.2 Results of Training and Validation Before Federated Learning Aggregation

Evaluating individual models trained on centralized data is crucial before the FL aggregation process to establish a reliable baseline. This step enables observation of each machine learning model's effectiveness when trained independently, providing a reference for subsequent federated learning enhancements.

In this section, we present the training and validation results for several CNN-based models, highlighting their performance under consistent conditions, including identical dataset sizes, epochs, and learning rates. By doing so, we can better understand how each model performs in a controlled environment and establish a reference point to later assess the impact of FL aggregation on their performance.

Table 3 summarizes the training and validation results of various machine learning models before FL aggregation. ResNet50 performed the best with an accuracy of 99.80% during training and 99.44% in validation, along with low loss values. In contrast, MobileNet, despite being faster, showed the lowest performance with a validation accuracy of 97.12%. The other models, VGG16 and DenseNet121, demonstrated balanced performance, with VGG16 achieving a slight advantage in terms of accuracy and loss, showcasing its reliability for plant disease detection.

Table 4 presents the same scenario previously discussed, but with a smaller initial dataset. During training in a more distributed environment, the ResNet50 architecture achieved the best performance across all evaluation metrics, standing out as the most robust model. The CNN also yielded satisfactory results, although its performance was slightly lower than that of ResNet50. These findings provide a relevant basis for analyzing the impact of different architectures in the context of federated learning.

5.3 Results before and After Federated Learning Aggregation

FL introduces an additional layer of complexity to traditional machine learning models by decentralizing data distribution across multiple users. This setup allows for privacy-preserving training, as each user maintains data locally. After independent local training, model parameters are aggregated into a global model. This process effectively leverages distributed data without centralized collection, enhancing data privacy while boosting global model robustness.

Table 5 presents a detailed comparison of training time and accuracy across various models in centralized versus FL environments. A clear trade-off emerges between training speed and accuracy, particularly with lightweight models like MobileNet. In the two-user scenario (centralized dataset size: 19,552), FL reduced MobileNet's training time to just 6 minutes, compared to 40 minutes in centralized training, with a minor accuracy difference (98.85% in FL vs. 98.80% centralized). The trade-off becomes even more evident with complex models like DenseNet121 and VGG16, where FL reduces training time significantly—from hours to minutes—but with observable accuracy drops, especially as the number of users increases. For example, in the five-user scenario (centralized training dataset size: 8,445), DenseNet121's accuracy drops from 99.47% in the centralized model to 95.47% in the federated model. This suggests that while federated learning offers substantial improvements in speed, it may compromise model precision due to issues like data heterogeneity and the aggregation process. As the number of users rises, the loss in accuracy becomes more noticeable, highlighting the challenge of maintaining high model performance while benefiting from faster training times. These results underscore the need for further investigation into methods that can better balance the trade-off between speed and accuracy in federated learning, potentially through enhanced aggregation techniques, improved data partitioning strategies, and model regularization, ensuring that federated sys-

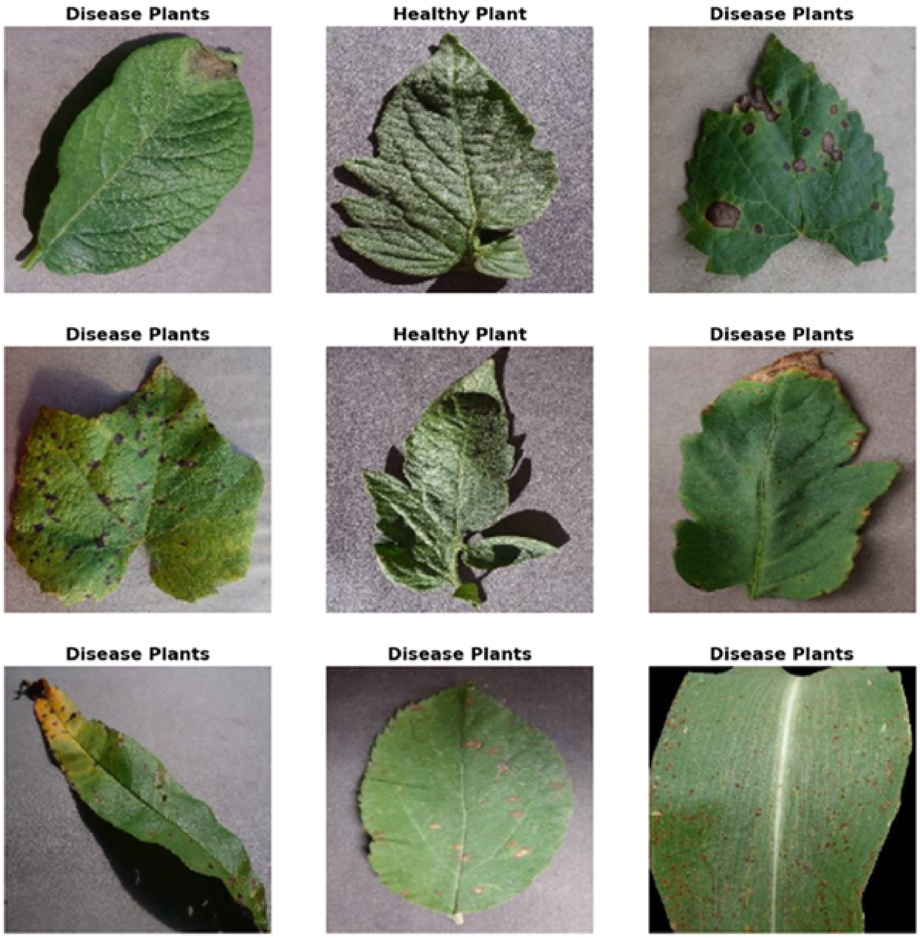

Figure 4. Classification of plants as healthy or diseased

Table 2. Comparison of Parameters in Centralized and Federated Scenarios

Category	Parameter	Centralized Model	Federated Models (2 and 5 users)
Training	Training Steps	25	30
	Learning Rate	0.0001	0.0001
	Batch Size	32	32
Input Data	Number of Classes	2	2
	Image Size	160×160	160×160
	Color Channels	3 (RGB)	3 (RGB)
Model	Architecture	Conv2D, MaxPooling2D, Dense, Dropout	

Table 3. Two-User Scenario: Results of Training and Validation Before Federated Learning Aggregation

	Model Training				Validation Models			
Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Loss**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Loss**
CNN	99.72	99.33	99.56	0.0084	98.83	97.57	97.77	0.0383
VGG16	99.50	99.03	98.99	0.0150	99.01	97.98	98.08	0.0287
DenseNet121	99.24	98.70	98.28	0.0210	98.90	98.26	97.36	0.0297
MobileNet	98.80	98.08	97.17	0.0345	97.12	95.07	93.43	0.0733
ResNet50	99.80	99.60	99.62	0.0060	99.44	98.89	98.89	0.0200

Table 4. Five-User Scenario: Results of Training and Validation Before Federated Learning Aggregation

	Model Training				Validation Models			
Models	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Loss**	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**Loss**
CNN	99.72	99.33	99.56	0.0084	98.83	97.57	97.77	0.0383
VGG16	99.50	99.03	98.99	0.0150	99.01	97.98	98.08	0.0287
DenseNet121	99.24	98.70	98.28	0.0210	98.90	98.26	97.36	0.0297
MobileNet	98.80	98.08	97.17	0.0345	97.12	95.07	93.43	0.0733
ResNet50	99.80	99.60	99.62	0.0060	99.44	98.89	98.89	0.0200

tems can deliver both efficient training and high precision for practical applications.

Table 3 compares the models' performance after aggregation. MobileNet retains its advantage in training efficiency,

Table 5. Comparative Training and Validation Metrics Before Federated Learning Aggregation Across Different User Scenarios

	Scenario with 2 Users				Scenario with 5 Users			
	Centralized		FL - Average		Centralized		FL - Average	
Models	**Training Duration**	**Training Accuracy (%)**	**Training Duration**	**Training Accuracy (%)**	**Training Duration**	**Training Accuracy (%)**	**Training Duration**	**Training Accuracy (%)**
CNN	1h 25m	99.7	10m	98.8	38m	99.7	18m	95.9
VGG16	4h	99.5	22m	99.7	3h 41m	99.6	2h 16m	97.8
DenseNet121	2h 20m	99.2	20m	98.9	1h 29m	99.5	38m	95.5
MobileNet	40m	98.8	6m	98.9	25m	98.8	10m	94.9
ResNet50	3h	99.8	20m	99.0	1h 27m	99.8	44m	98.0

completing its training in just 6 minutes, while VGG16 and ResNet50 require around 20 minutes. The results reflect the trade-off between complexity and speed, with MobileNet being more suitable for resource-constrained scenarios.

The FL environment often introduces non-IID data, where the data distributions across different users (or clients) are not homogeneous. This can pose a significant challenge during the aggregation process, as models trained on different datasets may learn features that are highly specific to their local data. When these models are aggregated, the resulting global model may face difficulty in generalizing across the entire dataset.

In our study, this issue became evident in the performance differences observed between models after aggregation. Models such as VGG16 and DenseNet121, which possess deep architectures capable of learning complex representations, were better equipped to handle the variability introduced by non-IID data. Their ability to extract meaningful features across different datasets allowed them to maintain high levels of accuracy even after the aggregation process.

On the other hand, MobileNet, with its lightweight architecture, struggled to adapt to the non-IID nature of the data. The significant drop in accuracy highlights the challenges that simpler models face when confronted with diverse datasets in FL environments. MobileNet's reliance on fewer parameters and less complex feature extraction processes limits its ability to generalize effectively after aggregation.

The impact of non-IID data was also observed in ResNet50, albeit to a lesser extent. While ResNet50's accuracy decreased post-aggregation, it retained strong precision and recall values, indicating that the model still performed well on the data it correctly classified. However, the model's slight drop in accuracy suggests that even advanced architectures like ResNet50 can face challenges when aggregating models trained on heterogeneous data sources.

5.4 Comparison of Model Test Results Before and After Aggregation

Evaluating model performance in an FL environment requires assessing generalization capabilities both before and after aggregation. This analysis is critical to understanding FL's impact on accuracy, precision, recall, and loss—especially when aggregating models trained on diverse user data. In this section, we provide a detailed comparison of the test results obtained before and after aggregation. By analyzing key performance metrics, we aim to identify how the aggregation of local models affects the overall accuracy and robustness of the global model, particularly in the context of handling heterogeneous data. This analysis will shed light on the strengths and weaknesses of different model architectures and their adaptability to the FL framework.

Table 6 compares the models' performance before and after aggregation. VGG16 showed the most significant improvement, with accuracy rising from 99.06% to 99.71% after aggregation. ResNet50 experienced a slight decline from 99.58% to 98.27%, while CNN and MobileNet both saw reductions in accuracy after aggregation. The findings indicate that VGG16 benefitted the most from the broader dataset provided by the aggregation process.

The table also presents the results in a FL scenario with five users. Tests were conducted before and after the aggregation of local models. VGG16 achieved the best performance after aggregation, with an accuracy of 94.2% and a loss of 0.118. The other models showed a decrease in performance after aggregation, especially CNN, DenseNet121, and MobileNet. This indicates that data heterogeneity among users affected the overall performance, and the number of users may also impact the performance of the global model. ResNet50 remained stable, with similar accuracy before and after aggregation.

When compared to the table with two users, all models performed better in that scenario, with accuracies above 97% and very low losses. This shows that with fewer users and likely more homogeneous data, aggregation better preserves performance. In contrast, in the five-user scenario, the variability among data negatively impacted accuracy and increased losses, highlighting the challenges of federated learning in environments with greater data diversity.

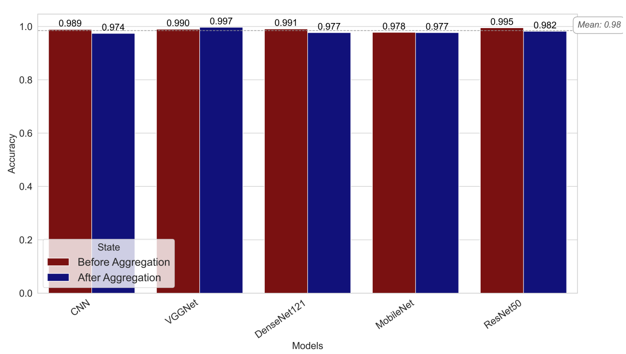

Figure 5. Two users - Comparison of Model Accuracy Before and After Aggregation

Figure 5 illustrates the performance of the CNN models (CNN, VGG16, DenseNet121, MobileNet, and ResNet50) both before and after aggregation in the FL process. The graph highlights that, despite some variability in the results, all models maintained high accuracy levels above 95% for

Table 6. Comparison of Model Test Results Before and After Aggregation in Federated Learning

	Two users				Five users			
	Testing Models Before Aggregation		Testing Models After Aggregation		Testing Models Before Aggregation		Testing Models After Aggregation	
Models	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**	**Loss**	**Accuracy**	**Loss**
CNN	99.72%	0.0084	98.83%	0.0383	91.00%	0.110	88.90%	0.261
VGG16	99.50%	0.0150	99.01%	0.0287	93.70%	0.134	94.20%	0.118
DenseNet121	99.24%	0.0210	98.90%	0.0297	93.60%	0.160	89.50%	0.218
MobileNet	98.80%	0.0345	97.12%	0.0733	91.90%	0.085	89.40%	0.250
ResNet50	99.80%	0.0060	99.44%	0.0200	91.00%	0.085	90.50%	0.211

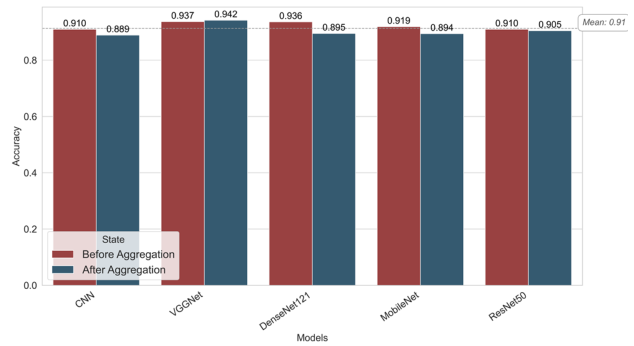

Figure 6. Five users - Comparison of Model Accuracy Before and After Aggregation

the scenario with 2 users.

Before the aggregation in Figure 5, ResNet50 emerged as the top performer with an accuracy of 99.58%, attributed to its deep architecture and the use of residual connections, which effectively address issues like the vanishing gradient problem often encountered in deep networks. Similarly, DenseNet121 and VGG16 also displayed strong performance, achieving accuracies of 99.17% and 99.06%, respectively. Meanwhile, the traditional CNN and MobileNet models recorded lower, but still respectable, accuracies of 98.96% and 97.81%.

Post-aggregation in Figure 5, VGG16 exhibited the most notable improvement, with its accuracy increasing to 99.71%. This performance boost is likely due to the aggregation process, which helps consolidate predictions by incorporating a more comprehensive dataset. ResNet50, though experiencing a slight decrease in accuracy to 98.28%, maintained robust performance overall. DenseNet121 and MobileNet both achieved identical accuracies of 97.70%, while the traditional CNN model showed a slight reduction in accuracy to 97.41%. The comparison highlights the impact of model aggregation, particularly in the case of VGG16, which benefited significantly from the FL process. Despite some models showing minor reductions in accuracy, the results indicate that aggregation, in general, improves model robustness and generalization capabilities, especially in scenarios involving diverse datasets and distributed data sources.

When comparing the results obtained with figure users 5 and 6, we observe that, although both scenarios showed robust performance, the increase in the number of users appears to have a moderate effect on the accuracies. In the two-user scenario, the models maintained high accuracies, with ResNet50 standing out at 99.58% before aggregation, and VGG16 showing a significant increase to 99.71% after aggregation. In the five-user scenario, VGG16 achieved the highest performance gain, with accuracy rising to 99.97%, while ResNet50 experienced a slight decrease, from 99.58% to 98.28%. This slight reduction in ResNet50's accuracy after aggregation in the five-user scenario can be attributed to the impact of greater data diversity, which may introduce variations in predictions. While aggregation showed a clear benefit for VGG16 in both cases, the increase in the number of users also resulted in more varied behavior among the models, suggesting that the aggregation process is more effective when more data sources are combined, helping to improve generalization but potentially introducing some degree of variability in the individual model performance.

5.5 Communication Cost and Performance in Centralized and Federated Training

Communication efficiency plays a pivotal role in machine learning model training, especially when contrasting centralized and federated approaches. Centralized training demands a single transfer of the dataset and model weights to the server, with the cost depending on the model's complexity and size. Heavier models with larger weights and datasets increase transmission costs. Conversely, in Federated Learning (FL), data remains decentralized on local devices, but model parameters are exchanged repeatedly across multiple rounds, leading to cumulative communication costs. In federated learning, however, the data remains on local devices, but the model parameters must be transmitted repeatedly over multiple rounds, resulting in a cumulative communication cost. This means that, in federated learning, communication costs increase progressively with the number of training rounds, especially for more complex models, which consequently can impact the efficiency and feasibility of training in scenarios with bandwidth and network resource limitations. [Zhuansun *et al.*, 2024].

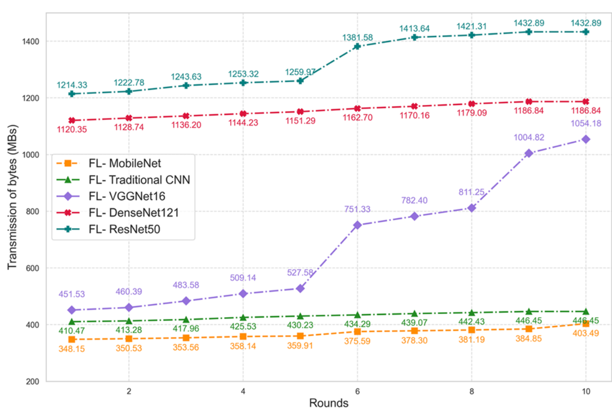

Figure 7. Comparison of Bytes per round

Figure 7 showcases the total data volume transmitted by various models over 10 FL rounds. More complex models like ResNet50 and DenseNet121 demonstrated significantly higher transmission volumes, around 1.50 GB and 1.24 GB, respectively. Due to their deeper architectures and more parameter layers, these models produce larger weight updates, increasing data traffic. Conversely, lighter models such as MobileNet and traditional CNNs transmitted substantially smaller volumes—403 MB (0.39 GB) and 468 MB (0.46 GB), respectively—while maintaining strong performance and reducing communication costs. These results highlight the advantage of compact models in bandwidth-constrained scenarios, as they not only maintain competitive performance but also make federated training more efficient in terms of network resources, reducing operational costs and improving system scalability.

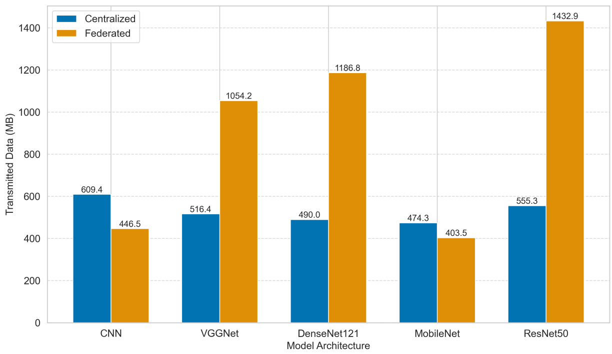

Figure 8. Transmitted Data Comparison: Centralized vs Federated Learning

Figure 8 compares the communication costs between centralized and federated learning across different models. In centralized training, the volume of data transmitted varies by model, with values like 480.5 MB for CNN and up to 584.2 MB for ResNet50. This occurs because each model requires a single transfer of its weights. In federated learning, however, the transmitted data increases progressively with each training round due to the multiple rounds of communication. Models such as ResNet50 and DenseNet121 demonstrate significantly larger data transmission volumes, reaching over 1.5 GB and 1.2 GB, respectively, while lighter models like MobileNet and VGGNet show lower transmission volumes of approximately 423 MB and 468 MB, respectively. This illustrates how federated training results in higher communication costs as the number of rounds increases, particularly for more complex models.

To mitigate communication costs in federated learning, a simple yet effective quantization technique based on binary compression was applied [Chen *et al.*, 2021]. After each round of local training, the model weights are serialized and compressed, significantly reducing the data size sent to the aggregation server. This process resulted in a noticeable reduction in transmission volume without compromising model aggregation quality, which is especially advantageous in bandwidth-constrained environments.

Over the 10 rounds required to reach maximum performance, communication efficiency varied across models. MobileNet and the traditional CNN exhibited more controlled growth in transmitted data, making them viable options for network-constrained scenarios. Conversely, heavier models became less practical due to high transmission costs. Therefore, model selection must balance performance and communication cost, with lighter models being preferable when aiming to reduce transmitted data volume without sacrificing training quality.

The training involved a total of 5,593 samples distributed across five clients, directly influencing the federated process efficiency and the total data transmitted in each round.

In summary, the results indicate that while centralized training involves a single communication cost, federated training incurs progressively higher costs due to multiple rounds of data transmission. Complex models, such as ResNet50 and DenseNet121, generate significantly larger data volumes, making them less efficient in bandwidth-limited environments. On the other hand, lighter models, such as MobileNet and traditional CNNs, offer a more efficient alternative, maintaining competitive performance with reduced communication costs. The choice of model should therefore carefully balance performance and communication costs, with lighter models being preferable in network-constrained scenarios to minimize the volume of transmitted data without compromising the quality of training.

5.6 Analysis and Discussion

The results shown in Tables 3, 4, 5, and 6, along with Figures 5, 6 and 7, illustrate the comparative strengths and weaknesses of each model before and after aggregation. ResNet50, DenseNet121, and VGG16 emerged as top performers, demonstrating strong generalization abilities in both pre- and post-aggregation scenarios. Notably, VGG16 benefited the most from the aggregation process, likely due to its architecture's capacity to leverage the additional information provided by FL.

MobileNet, while the fastest among the models, consistently showed a trade-off between speed and accuracy. Although its performance remains suitable for many applications, it is limited by lower precision and recall when compared to deeper models like ResNet50 and DenseNet121. These findings highlight the critical importance of model selection in FL contexts. For resource-constrained environments where quick deployment is crucial, MobileNet is advantageous. Conversely, for scenarios demanding higher accuracy and robustness, ResNet50 and VGG16 deliver superior results, albeit with longer training durations.

Although the aggregation process generally improved model performance, some variability was observed, notably in CNN and MobileNet. This is likely due to the heterogeneous nature of client data, which can pose challenges during model aggregation. Despite this, the enhancements seen in VGG16 underscore FL's capacity to boost generalization by integrating insights from diverse datasets.

In summary, the findings demonstrate that FL is a highly effective tool for distributed model training, though careful consideration is required when selecting model architectures. Deeper models like VGG16 and ResNet50 notably benefit from the aggregation process, whereas lighter models such as MobileNet may trade accuracy for faster processing times. As FL methodologies evolve, further refinement of aggregation strategies and model designs will be crucial to optimize performance in decentralized settings.

6 Conclusions

This study comprehensively evaluated the performance, communication costs, and aggregation behaviors of different machine learning models within a FL framework for plant disease detection. The experiments involved scenarios with both 2 and 5 users, providing insights into how the number of participants and the distribution of data impact model performance, training efficiency, and communication load.

The results confirm that FL is a promising solution for decentralized training, maintaining data privacy while enabling collaborative model development across multiple clients. However, the effectiveness of FL is heavily influenced by the number of clients and the heterogeneity of local datasets, particularly when data is non-IID (non-independent and identically distributed). In scenarios with 2 users, the models exhibited strong performance, with VGG16 achieving an impressive 99.71% accuracy after aggregation. This high performance is attributed to the relatively consistent data distributions across users. In contrast, when the number of users increased to 5, performance decreased slightly across most models, including VGG16 and CNN. This drop in accuracy is mainly due to the increased variability and imbalance in local datasets, which complicated the gradient alignment during aggregation and introduced additional noise in the global model update.

To mitigate communication costs, which are often a bottleneck in FL, especially with large models, the study incorporated a binary compression technique to reduce the size of transmitted data. By compressing the model weights after each local training round, the transmission volume was significantly reduced without impacting model accuracy. This approach was particularly beneficial for lightweight models such as MobileNet and CNN, which required less than 500 MB of data transmission over 10 rounds of training. In contrast, deeper models like ResNet50 and DenseNet121, which contained more parameters, exceeded 1.5 GB of transmitted data, highlighting the trade-offs between model complexity and communication efficiency.

In conclusion, while FL offers significant potential for decentralized machine learning, especially in privacy-sensitive domains like agriculture, its performance and efficiency are contingent on careful model selection, handling of non-IID data, and strategies to minimize communication costs. The study demonstrates that lightweight models like MobileNet and CNN are more suited for resource-constrained environments, while deeper models like VGG16 and ResNet50 are more effective in scenarios where data heterogeneity is manageable. Future work should explore advanced aggregation techniques, adaptive client selection methods, and further compression strategies to enhance FL's scalability and applicability in large-scale, real-world deployments.

Declarations

Acknowledgements

This research is part of the INCT of Intelligent Communications Networks and the Internet of Things (ICoNIoT) funded by CNPq (proc. 405940/2022-0) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) Finance Code 88887.954253/2024-00.

Authors' Contributions

Each author declares substantial contributions through the conception and design of the study, as well as the analysis and interpretation of data. Additionally, each author contributed significantly to the drafting and critical review of the the article.

Competing interests

The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials

The data and materials used in the article are available upon request.

References

Aggarwal, M., Khullar, V., Goyal, N., Alammari, A., Albahar, M. A., and Singh, A. (2023). Lightweight federated learning for rice leaf disease classification using non independent and identically distributed images. *Sustainability*, 15(16):12149. DOI: 10.3390/su151612149.

Banabilah, S., Aloqaily, M., Alsayed, E., Malik, N., and Jararweh, Y. (2022). Federated learning review: Fundamentals, enabling technologies, and future applications. *Information processing & management*, 59(6):103061. DOI: 10.1016/j.ipm.2022.103061.

Basseto, V., Sanches, S., Grandi, M., Oliveira, C., Kern, H., and Silla, P. (2022). Catálogo virtual de doenças e pragas de soja. Available at:`https://www.embrapa.br/busca-de-publicacoes/-/publicacao/1147321/catalogo-virtual-de-doencas-e-pragas-de-soja`.

CABI (2023). Global burden of crop loss. Available at:`https://www.cabi.org/projects/global-burden-of-crop-loss/`.

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

Li, L., Fan, Y., Tse, M., and Lin, K.-Y. (2020). A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854. DOI: 10.1016/j.cie.2020.106854.

Mamba Kabala, D., Hafiane, A., Bobelin, L., and Canals, R. (2023). Image-based crop disease detection with federated learning. *Scientific Reports*, 13(1):19220. DOI: 10.1038/s41598-023-46218-5.

Mehta, S., Kukreja, V., and Yadav, R. (2023a). Advanced mango leaf disease detection and severity analysis with federated learning and cnn. In *2023 3rd International Conference on Intelligent Technologies (CONIT)*, pages 1–6. DOI: 10.1109/CONIT59222.2023.10205922.

Mehta, S., Kukreja, V., and Yadav, R. (2023b). A federated learning cnn approach for tomato leaf disease with severity analysis. In *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, pages 309–314. DOI: 10.1109/ICAISS58487.2023.10250571.

Moreira, A. L., Tiecher, P. R., Duarte, S. M., Samuelsson, E., Ferreira, L., and dos Santos, F. C. (2022). Joaninhas: Controle de pragas. *Revista Científica da Faculdade de Educação e Meio Ambiente*, 13(edesp-multi). Available at:`https://revista.faema.edu.br/index.php/Revista-FAEMA/article/view/969`.

Shoaib, M., Shah, B., EI-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., and Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science*, 14. DOI: 10.3389/fpls.2023.1158933.

Silva, B. F. and Cavichioli, F. A. (2022). O uso de veículos aéreos não tripulados para detecção de pragas e doenças na cultura da soja. *Revista Interface Tecnológica*, 19(1):236–247. DOI: 10.31510/infa.v19i1.1363.

Souza, J. R., Oliveira, S. Z., and Oliveira, H. (2024). The impact of federated learning on urban computing. *Journal of Internet Services and Applications*, 15(1):380–409. DOI: 10.5753/jisa.2024.4006.

Zhuansun, Y., Li, D., Huang, X., and Sun, C. (2024). Communication-efficient federated learning with adaptive compression under dynamic bandwidth. *arXiv preprint arXiv:2405.03248*. DOI: 10.48550/arxiv.2405.03248.