Privacy-Preserving Spatial Crowdsourcing Drone Services for Postdisaster Infrastructure Monitoring: A Conditional Federated Learning Approach

Junaid Akram, Awais Akram, Palash Ingle, Rutvij H. Jhaveri, Ali Anaissi, and Adnan Akhunzada, *Senior Member, IEEE*

***Abstract*—Sixth-generation (6G) networks, offering ultra-low latency and high bandwidth, provide critical support for rapid data transmission in postdisaster environments where conventional infrastructure may be compromised. This study presents a privacy-preserving framework for postdisaster structural health monitoring (SHM) by integrating 6G-enabled Internet of Drone Things and spatial crowdsourcing. Drones and unmanned ground vehicles collect real-time imagery of damaged infrastructure. To address privacy concerns and reduce communication overhead, we employ federated learning (FL), which enables decentralized model training on local devices without transmitting raw data. A key challenge in FL is the presence of nonindependent and identically distributed data across clients, which degrades global model performance. To mitigate this, we propose personalized conditional federated averaging (PC-FedAvg), a selective aggregation method that incorporates only client models with low validation loss into the global update. The PC-FedAvg framework is built on EfficientNetV2 and includes personalized model adaptation to enhance generalization on heterogeneous data. Experimental results on the "Concrete Crack Images for Classification" dataset demonstrate that PC-FedAvg outperforms baseline FL methods in accuracy and stability. This approach improves the effectiveness and reliability of SHM systems in real-world postdisaster scenarios by enabling timely and accurate damage assessment while preserving data privacy.**

***Index Terms*—Anomaly detection, drones, federated learning (FL), spatial crowdsourcing, structural health monitoring (SHM).**

Received 6 December 2024; revised 9 February 2025 and 25 April 2025; accepted 29 May 2025. Date of publication 9 June 2025; date of current version 14 July 2025. This work was supported by the Qatar National Library through open access. *(Corresponding authors: Adnan Akhunzada; Rutvij H. Jhaveri.)*

Junaid Akram is with UNSW College, Kensington, NSW 2033, Australia, also with the University of Sydney, Camperdown, NSW 2008, Australia, and also with the Transdisciplinary School, University of Technology Sydney, Broadway, NSW 2007, Australia (e-mail: j.akram@unswcollege.edu.au).

Awais Akram is with the School of Computing, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: awais.akram.1212@gmail.com).

Palash Ingle is with the Department of Computer and Information Security, Sejong University, Seoul 05006, South Korea (e-mail: Palash@sejong.ac.kr).

Rutvij H. Jhaveri is with the Department of Computer Science and Engineering, School of Technology, Pandit Deendayal Energy University, Gandhinagar 382007, India (e-mail: rutvij.jhaveri@sot.pdpu.ac.in).

Ali Anaissi is with the University of Sydney, Camperdown, NSW 2008, Australia, and also with the Transdisciplinary School, University of Technology Sydney, Broadway, NSW 2007, Australia (e-mail: ali.anaissi@uts.edu.au).

Adnan Akhunzada is with the Department of Data and Cybersecurity, College of Computing and IT, University of Doha for Science and Technology, Doha 24449, Qatar (e-mail: Adnan.adnan@udst.edu.qa).

Digital Object Identifier 10.1109/JSTARS.2025.3577648

I. Introduction

POSTDISASTER structural health monitoring (SHM) is essential for assessing and restoring critical infrastructure after natural or human-induced disasters. Sixth-generation (6G) networks are envisioned as a cornerstone for integrated sensing and communications, opening new avenues for environmental monitoring and disaster management, particularly in postdisaster infrastructure assessment [1], [2], [3]. Central to this transformation are unmanned aerial vehicles (UAVs), commonly known as drones, which are equipped with state-of-the-art communication interfaces, such as WiFi and 5G. These drones play a pivotal role in the emerging Internet of Drone Things (IoDT) framework [4], [5], [6], [7]. Here, IoDT denotes the coordinated integration of aerial and ground robotic agents into a unified sensing and communication network, enabling continuous data collection in challenging environments. Spatial crowdsourcing refers to the distribution of sensing tasks to multiple unmanned agents, increasing coverage and scalability in remote or hazardous areas.

The integration of spatial crowdsourcing with IoDT constitutes a novel methodology for data acquisition in remote and hazardous environments. In this approach, drones and unmanned ground vehicles (UGVs) are deployed to collect real-time data over extensive and potentially dangerous areas, replacing conventional methods that depend on human intervention in inaccessible locations [8], [9], [10]. In postdisaster scenarios, rapid and precise data collection is essential. Drones acquire critical sensory information from points of interest (PoIs), such as damaged structures, obstructed roadways, and vital infrastructure. This method significantly enhances both the efficiency and the scope of data collection [11], [12], [13], [14].

Timely and accurate assessments of structural health are crucial after a disaster to inform repair and reconstruction efforts, thereby conserving resources and potentially saving lives [15], [16], [17], [18]. Drones outfitted with advanced sensing technologies can swiftly survey large regions, capturing high-resolution imagery and sensor data that provide insights into the condition of buildings, bridges, and other critical infrastructures [19], [20], [21]. These aerial assessments facilitate the identification of structural damage, hazard detection, and prioritization of areas requiring immediate attention. Furthermore, the implementation of the air–ground nonorthogonal multiple access (AG-NOMA) technique enhances data transmission rates and quality-of-service (QoS). In this configuration, drones transmit uplink data to UGVs that function as mobile base stations for data decoding, while the UGVs also engage in direct data collection from PoIs. This dual-mode operation optimizes the volume of collected data while managing the energy constraints of the drones.

A significant challenge in SHM lies in the effective integration and analysis of data from numerous spatially distributed sources. Centralized data processing is often computationally intensive and incurs high communication overhead, particularly in postdisaster environments where network connectivity may be compromised [22], [23]. To overcome these limitations, we propose a federated learning (FL) approach that exploits the decentralized characteristics of IoDT. This approach enables local data processing on drones and UGVs, thereby reducing the necessity for extensive data transmission and enhancing data security [24], [25]. FL has demonstrated effectiveness in various fields, including civil engineering and infrastructure monitoring, by facilitating collaborative model training across distributed devices while preserving data locality [26], [27], [28], [29].

However, the local datasets collected by individual agents are statistically heterogeneous [nonindependent and identically distributed (non-IID)]; for example, crack images captured under varying illumination or damage patterns differ significantly between clients, which degrades the performance of global models trained with conventional FL methods, such as FedAvg. To address this, we introduce personalized conditional federated averaging (PC-FedAvg), which aggregates only those client updates whose validation loss falls below the median threshold. This selective aggregation filters out low-quality models and, when combined with client-level fine-tuning of EfficientNetV2, enhances model robustness and generalization across diverse datasets.

Our proposed FL framework thus integrates 6G-enabled IoDT, spatial crowdsourcing, and selective PC-FedAvg aggregation to improve the accuracy, reliability, and privacy of SHM systems in postdisaster scenarios. The key contributions of our work are as follows:

1) We propose a novel FL framework tailored specifically for SHM in postdisaster scenarios. This framework leverages the decentralized nature of the IoDT, enabling local data processing on drones and UGVs. This approach reduces the need for extensive data transmission to a central server, thereby enhancing data security and efficiency.
2) We introduce an innovative method for training EfficientNetV2 within FL settings. This method includes an efficient communication protocol for aggregating the learned models, ensuring that the computational load is distributed and data privacy is maintained.
3) To tackle the challenge of non-IID data distribution, we develop a novel technique to personalize the resulting models. This personalization enhances the generalization capabilities of the clients' models, thereby improving the accuracy of SHM across diverse datasets.
4) The proposed FL framework significantly enhances the accuracy and reliability of SHM models. By allowing collaborative model training across multiple devices while keeping the data localized, our approach ensures robust and scalable real-time SHM. This is crucial for making informed decisions during postdisaster recovery efforts, ultimately contributing to the resilience and safety of critical infrastructure.
5) Our FL approach reduces the computational and communication overhead associated with traditional centralized models. By leveraging the strengths of 6G networks and IoDT, the framework provides a more efficient and scalable solution for SHM, making it feasible to operate in challenging postdisaster environments where communication networks may be compromised.

The rest of this article is organized as follows. Section II reviews related work in FL and SHM. Section III presents the system model, detailing the integration of 6G networks and IoDT for SHM. Section IV introduces our proposed PC-FedAvg algorithm for FL and discusses the personalized EfficientNetV2 models. Section V describes the experimental setup and results, validating our approach on real SHM datasets. Finally, Section VI concludes this article, highlighting key findings and future research directions.

II. Related Work

FL has garnered significant interest in recent years as a promising approach in the field of machine learning, particularly for its ability to preserve data privacy by decentralizing data storage and processing [30], [31], [32], [33], [34], [35], [36], [37], [38]. This approach is particularly attractive for practical problems and applications where data decentralization is essential. Despite its potential, the literature contains only a few studies that successfully leverage FL to construct robust global models. For instance, Hard et al. [39] employed FL to develop a system for next-word prediction on mobile devices. However, many studies focus on addressing the training challenges posed by non-IID data distribution across clients.

McMahan et al. [40] introduced the pioneering FL algorithm known as *FedAvg*. This algorithm uses local stochastic gradient descent (SGD) updates to build a global model by averaging model coefficients from a subset of clients with non-IID data. The FedAvg algorithm is controlled by three key parameters: C, the proportion of clients selected to perform computation in each round; E, the number of training passes each client makes over its local dataset in each round; and B, the local mini-batch size used for client updates. Selected clients perform local SGD for E epochs with mini-batch size B. Clients that have not completed E epochs by the start of the update round (stragglers) are not considered during aggregation. Stragglers refer to clients that are delayed in completing their local training rounds, which may lead to outdated or suboptimal updates during aggregation.

Li et al. [41] proposed the *FedProx* algorithm, which builds on FedAvg by addressing the issue of statistical heterogeneity through two key modifications: considering stragglers during aggregation and adding a *proximal term* to the objective function. Arivazhagan et al. [42] addressed the effects of statistical heterogeneity using a *personalization-based approach (FedPer)*. In

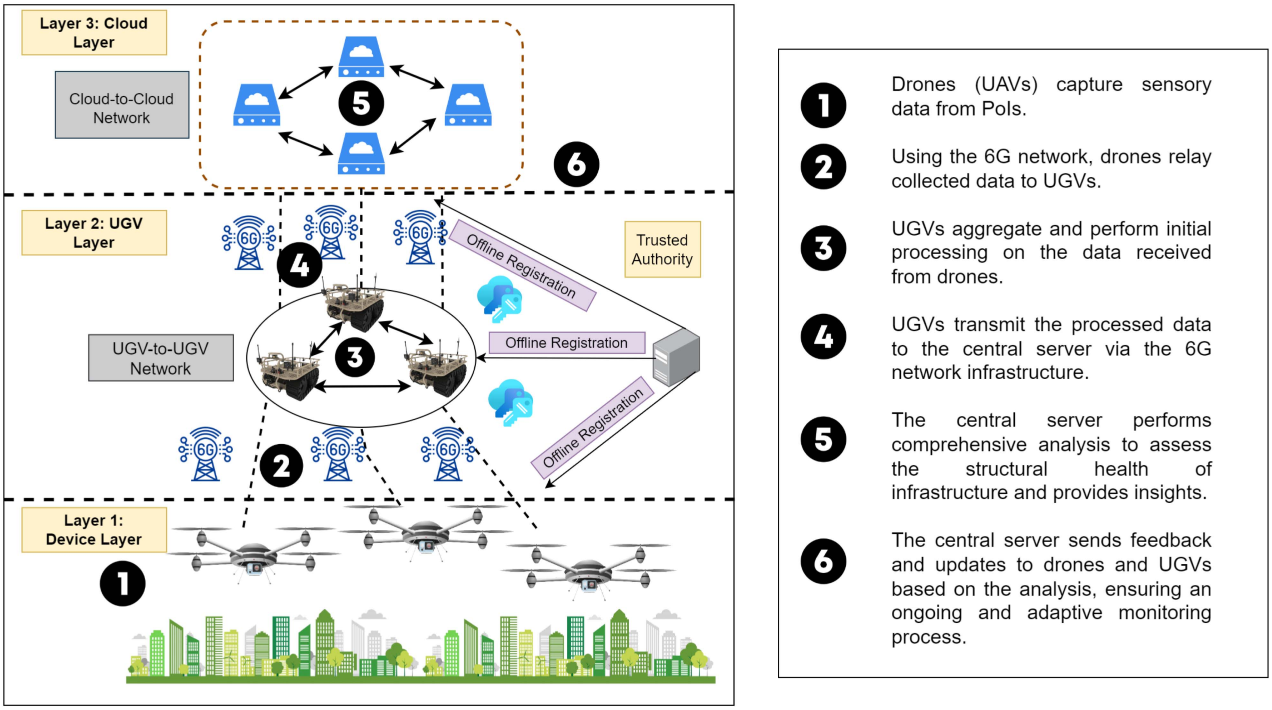

Layer 3: Cloud Layer
Cloud-to-Cloud Network
5
6
Layer 2: UGV Layer
6G
4
Offline Registration
Trusted Authority
Offline Registration
UGV-to-UGV Network
3
Offline Registration
2
Layer 1: Device Layer
1
1 Drones (UAVs) capture sensory data from PoIs.
2 Using the 6G network, drones relay collected data to UGVs.
3 UGVs aggregate and perform initial processing on the data received from drones.
4 UGVs transmit the processed data to the central server via the 6G network infrastructure.
5 The central server performs comprehensive analysis to assess the structural health of infrastructure and provides insights.
6 The central server sends feedback and updates to drones and UGVs based on the analysis, ensuring an ongoing and adaptive monitoring process.

Fig. 1. Proposed system model for postdisaster SHM using 6G networks and IoDT.

FedPer, the model is divided into base and personalization layers. The base layers are aggregated using a standard FL approach, while the personalized layers are not aggregated, thus allowing for customization at the client level.

Several other methods have been proposed to achieve personalization in FL. Smith et al. [43] introduced the *MOCHA-based multitask learning* framework to address the non-IID challenge. Hanzel and Richtárik [44] proposed the L2GD algorithm, which combines the optimization of local and global models. Deng et al. [45] developed an *adaptive personalized FL* algorithm, which mixes the user's local model with the global model to improve learning outcomes.

Despite these advancements, existing FL methods still face challenges in effectively managing non-IID data distributions and ensuring model personalization. Our proposed approach, PC-FedAvg, fills these gaps by introducing a tailored learning algorithm that optimizes weight communication and aggregation. PC-FedAvg employs EfficientNetV2 to personalize the resulting models at each client, thereby improving the accuracy and robustness of FL models in postdisaster SHM scenarios. By addressing the issues of non-IID data and enhancing model personalization, our method significantly improves the reliability and efficiency of SHM systems, ensuring timely and accurate damage assessments.

III. System Model

The proposed system model, as illustrated in Fig. 1, integrates 6G networks with the IoDT to create a robust framework for postdisaster SHM. This architecture leverages drones (UAVs) and UGVs equipped with advanced communication technologies, such as WiFi/5G, to perform real-time data collection and processing.

Drones play a pivotal role in this architecture by acquiring sensory data from various PoIs, such as damaged structures and obstructed roads. Equipped with high-resolution cameras, LiDAR, and other sensors, drones can capture detailed images and measurements of critical infrastructure. These drones are connected via 6G networks, enabling high-speed data transmission and low-latency communication, essential for timely disaster response.

UGVs complement the drones by serving as mobile base stations and data aggregators. They collect data relayed by drones and perform initial processing to reduce data volume before transmission. UGVs are also equipped with similar sensing technologies, allowing them to gather data directly from ground-level PoIs. This dual capability ensures comprehensive coverage and data redundancy, enhancing the reliability of the SHM system.

The system utilizes an AG-NOMA technique to optimize data transmission rates and ensure QoS. In this model, drones collect uplink data and relay it to UGVs, which decode the data and transmit it to a central server if necessary. The AG-NOMA technique is based on the principle of superimposing multiple signals on a common frequency band and then using advanced decoding algorithms to separate them at the receiver. This method enhances spectrum efficiency and minimizes interference, which is particularly beneficial in scenarios with a high density of transmitting drones. Its application is crucial in postdisaster environments where rapid and simultaneous data transmission from multiple sources is required. The AG-NOMA technique enables efficient spectrum usage and supports simultaneous transmissions from multiple drones, significantly enhancing the system's scalability.

Communication between drones and UGVs is facilitated by 6G networks, providing high bandwidth and low latency. This ensures real-time data transfer, crucial for immediate disaster assessment and response. The AG-NOMA technique allows multiple drones to share the same frequency band without significant interference, optimizing the available spectrum and improving data throughput. UGVs communicate with the central server or cloud infrastructure using terrestrial 6G networks. This communication model supports large data transfers and integrates with existing network infrastructure, ensuring compatibility and extensibility. UGVs can also act as relay nodes for other ground units, creating a mesh network that enhances coverage and resilience, especially in compromised postdisaster environments.

The operational workflow of the proposed system involves several key steps. First, drones and UGVs are deployed to collect sensory data from PoIs. Drones, equipped with high-resolution cameras, LiDAR, and other advanced sensors, survey large areas and capture detailed images and measurements of critical infrastructure. These data are then relayed to UGVs, which act as mobile base stations. These UGVs not only aggregate the data but also perform initial processing to reduce the data volume before transmission to a central server. This step is crucial in managing the limited energy reserves of drones and ensuring efficient data collection.

Next, UGVs transmit the processed data to a central server or cloud infrastructure using terrestrial 6G networks. The high bandwidth and low latency of 6G networks facilitate real-time data transfer, essential for immediate disaster assessment and response. The AG-NOMA technique employed allows multiple drones to share the same frequency band without significant interference, optimizing the available spectrum and improving data throughput. This ensures that the system can handle simultaneous transmissions from multiple drones, significantly enhancing its scalability and efficiency.

Once the data reach the central server, it undergo further analysis to assess the structural health of the infrastructure. High-resolution images and sensor data provide insights into the condition of buildings, bridges, and other critical structures, identifying potential hazards and prioritizing areas needing urgent attention. The central server is responsible not only for data aggregation but also for performing advanced data analysis tasks. These include image processing algorithms for crack detection, damage assessment algorithms based on machine learning techniques, and statistical analysis to evaluate structural integrity. Techniques, such as convolutional neural networks and support vector machines, are utilized to process and interpret the sensory data. This comprehensive analysis informs repair and reconstruction efforts, ultimately saving time, resources, and lives in postdisaster scenarios. The system's ability to integrate and analyze data from numerous spatially distributed locations makes it a powerful tool for real-time SHM, leveraging the advanced capabilities of 6G networks and IoDT.

Potential system-level limitations should also be considered. These include communication disruptions due to environmental factors, inconsistent data quality from varied sensor sources, and delays in data transmission, all of which could impact overall system performance. Furthermore, the proposed system model can be extended by integrating with existing wireless sensor networks (WSNs) to facilitate large-scale knowledge discovery. Such integration would enable the fusion of data from both mobile (drones and UGVs) and static sensor nodes, thereby enriching the data pool and enhancing the decision-making process in disaster management scenarios.

In the FL setup, a PC-FedAvg algorithm was implemented, simulating an environment with five clients. Each client performed local training using the EfficientNetV2 model, pretrained on the ImageNet dataset and fine-tuned for binary classification. The local models were updated using SGD with momentum and communicated to a central server for aggregation. Unlike traditional FedAvg, our method selectively aggregated models from clients with validation losses below the median, ensuring robust and accurate global model updates.

The system effectively integrates advanced communication and data processing technologies, ensuring comprehensive and efficient SHM in postdisaster scenarios. By leveraging the decentralized nature of FL and the advanced capabilities of 6G networks and IoDT, our proposed system model addresses the challenges of real-time data collection, transmission, and analysis, providing a robust framework for enhancing the resilience and safety of critical infrastructure.

IV. Personalized Conditional FedAvg

We introduce an innovative algorithm named *PC-FedAvg*, specifically designed to improve the efficiency and accuracy of FL within the realm of postdisaster SHM. This algorithm addresses the critical challenge of aggregating weights from low-performing clients by utilizing a filtering mechanism based on local performance metrics. During each update round, clients communicate their local loss values to the central server. Only those clients whose local loss values fall below the overall median are included in the aggregation process. This selective inclusion effectively reduces the incorporation of "bad" weights, thereby enhancing the overall model's accuracy and reliability.

Our approach centers on the employment of EfficientNetV2, tailored to the specific needs of SHM. The method addresses the challenge of non-IID data distribution, which is common in FL settings, by developing a personalized support vector algorithm optimized for each distributed data model.

A. Conditional-FedAVG

In an FL environment, the learning task involves a collection of C clients and a central server S. Each client learns from its local data and periodically communicates with the server to collaboratively solve the following global optimization problem:

$$\min_{w \in \mathbb{R}^d} f(w) := \frac{1}{C} \sum_{c=1}^{C} f_c(w_c) \tag{1}$$

where f_c represents the local loss function for client c, which is defined as

$$f_c(w_c) := \mathbb{E}_{(x_i, y_i) \sim D_c} [\mathcal{L}_c(w_c; x_i, y_i)] \tag{2}$$

and $\mathcal{L}_c(w_c; x_i, y_i)$ quantifies the error of the model w_c given the input x_i and the true label y_i.

Each client updates its local model parameters w_c using SGD with momentum. The SGD update rule is given by

$$w_c^{(t+1)} := w_c^{(t)} - \eta \nabla \mathcal{L}_c(w_c^{(t)}; x_i, y_i). \tag{3}$$

Note that the gradient is computed as

$$\nabla \mathcal{L}_c(w_c; x_i, y_i) = \left(\sigma(w_c^\top x_i) - y_i\right) x_i$$

and the negative sign in the update rule in (3) ensures descent in the loss function.

The use of momentum in SGD helps to accelerate convergence by considering the past gradients in the update rule, thus avoiding oscillations and ensuring smoother convergence. The momentum term is mathematically represented as

$$v^{(t+1)} := \mu v^{(t)} + \eta \nabla \mathcal{L}_c(w_c^{(t)}; x_i, y_i) \tag{4}$$

where the momentum term $\mu v^{(t)}$ incorporates the history of past gradients to stabilize the update. Consequently, the model update with momentum becomes

$$w_c^{(t+1)} := w_c^{(t)} - v^{(t+1)}. \tag{5}$$

After training locally, each client sends its updated model parameters to the central server. The central server aggregates these parameters to form a new global model. Unlike the standard FedAvg approach, which averages the weights of all client models, our Conditional-FedAvg method selects only those clients whose validation losses are less than or equal to the median (i.e., $\text{loss}_c^{(t+1)} \leq \mathcal{M}$). This selective aggregation enhances the robustness and accuracy of the global model by mitigating the impact of poorly performing clients.

For clarity, we note that *stragglers* are defined as clients that fail to complete the required number of local epochs within the designated time window; such clients may be excluded from aggregation to prevent outdated updates from degrading global model performance.

The aggregation process is mathematically defined as

$$w^{(t+1)} := \frac{1}{C'} \sum_{c=1}^{C'} w_c^{(t+1)} \tag{6}$$

where C' is the number of selected clients whose validation losses satisfy $\text{loss}_c^{(t+1)} \leq \mathcal{M}$.

To illustrate the Conditional-FedAvg method, consider the following algorithm.

Algorithm 1: Conditional FedAvg.

```
Server executes:
Initialize global model $w$
for each round $t = 1$ to $T$ do
    Sample a subset of clients $C_t$ from $C$
    Initialize list $client\_models \leftarrow []$
    Initialize list $client\_losses \leftarrow []$
    for each client $c$ in $C_t$ in parallel do
        $w_c^{(t+1)}, loss_c^{(t+1)} \leftarrow \mathbf{ClientUpdate}(c, w)$
        Append $w_c^{(t+1)}$ to $client\_models$
        Append $loss_c^{(t+1)}$ to $client\_losses$
    end for
    Compute the median loss $\mathcal{M}$ of $client\_losses$
    Select clients with $loss_c^{(t+1)} \leq \mathcal{M}$
    Aggregate the selected client models to update
    global model $w$:
    $w \leftarrow \frac{1}{|C'|} \sum_{c \in C'} w_c^{(t+1)}$
end for
return global model $w$
ClientUpdate$(c, w)$:
$D_c \leftarrow$ local data at client $c$
Initialize local model $w_c \leftarrow w$
for each epoch $e$ in $E$ do
    for each batch $b$ in $D_c$ do
        Compute predictions $\hat{y} \leftarrow \sigma(w_c^\top x)$
        Compute loss $\ell \leftarrow \mathcal{L}_c(w_c; x, y)$
        Backpropagate to compute gradients $\nabla \mathcal{L}_c$
        Update model parameters $w_c \leftarrow w_c - \eta \nabla \mathcal{L}_c$
    end for
end for
Compute validation loss $loss_c \leftarrow \mathbf{Validate}(w_c, D_{val})$
return updated model $w_c$, validation loss $loss_c$
```

Algorithm 1 delineates the steps for implementing the Conditional FedAvg method within an FL framework. Initially, the server initializes the global model and iteratively updates it by aggregating the parameters from selected clients based on their validation performance. In this aggregation step, the central server computes the arithmetic mean of the selected client model parameters, ensuring that high-quality updates dominate the global model. Each client trains its local model using its dataset and transmits the updated model parameters and validation loss to the central server.

The process begins with each client receiving the initial global model w. During each communication round, each client performs local training on its dataset D_c to update its model parameters w_c. The local training process involves multiple epochs of SGD, where each epoch consists of several mini-batch updates. The local model update rule is given by

$$w_c^{(t+1)} := w_c^{(t)} - \eta \nabla \mathcal{L}_c(w_c^{(t)}; x_i, y_i). \tag{7}$$

After completing the local training, each client computes the validation loss on its validation dataset D_{val} and sends the updated model parameters $w_c^{(t+1)}$ and validation loss $\text{loss}_c^{(t+1)}$ to the central server. The central server then aggregates the model parameters from the selected clients whose validation losses satisfy the criterion $\text{loss}_c^{(t+1)} \leq \mathcal{M}$.

The conditional aggregation mechanism is mathematically represented as follows. Let $\mathcal{L}_c^{\text{val}}$ be the validation loss of client c at round t. The central server computes the median of the validation losses

$$\mathcal{M} := \text{median}\left(\{\mathcal{L}_c^{\text{val}}\}_{c=1}^{C_t}\right). \tag{8}$$

The server then selects the clients with validation losses less than or equal to the median

$$C_t' := \{c \mid \mathcal{L}_c^{\text{val}} \leq \mathcal{M}\}. \tag{9}$$

The global model is updated by averaging the model parameters from the selected clients

$$w^{(t+1)} := \frac{1}{|C'_t|} \sum_{c \in C'_t} w_c^{(t+1)}. \tag{10}$$

This conditional aggregation process ensures that the global model is primarily influenced by well-performing local models, thus enhancing the overall performance and robustness of the system.

The effectiveness of the Conditional-FedAvg method is evaluated using standard metrics, such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive assessment of the model's performance, particularly in scenarios with imbalanced data distributions. The experimental results demonstrate that the proposed method significantly improves the accuracy and robustness of the SHM system compared to traditional FL approaches. The conditional aggregation mechanism effectively mitigates the negative impact of poorly performing clients, leading to a more reliable and accurate global model.

To further understand the benefits of Conditional-FedAvg, consider the theoretical underpinnings of this approach. In FL, the data across clients are often non-IID, which can lead to significant performance degradation when simply averaging model parameters. Conditional-FedAvg addresses this issue by selectively aggregating the models from clients that perform well on their local validation data, thus ensuring that the global model benefits from high-quality updates.

From an optimization perspective, the goal of FL is to minimize the global objective function defined in (1). However, due to the heterogeneity of the data, the local objective functions $f_c(w_c)$ may differ significantly across clients. The conditional aggregation mechanism helps to align the local objectives with the global objective by focusing on the clients that provide the most informative gradients.

The Conditional-FedAvg algorithm can be viewed as a form of robust optimization, where the global model is updated based on a subset of reliable clients. This approach can be formalized using robust optimization theory, which aims to optimize the worst-case performance of the model. By selecting clients with low validation losses, the algorithm effectively minimizes the influence of outliers and ensures that the global model is robust to variations in the local data distributions.

In practice, the Conditional-FedAvg algorithm involves the following key steps:

1) *Client Selection:* In each communication round, a subset of clients is selected to participate in the training process. This selection can be random or based on certain criteria, such as data availability or computational resources. Note that clients failing to complete the required local training in time (i.e., stragglers) are either delayed or excluded to prevent outdated updates from degrading the global model.
2) *Local Training:* Each selected client performs local training on its dataset using the SGD with momentum algorithm. The local training process involves multiple epochs, during which the model parameters are updated based on the computed gradients.
3) *Validation and Reporting:* After local training, each client computes the validation loss on its validation dataset and sends the updated model parameters and validation loss to the central server.
4) *Conditional Aggregation:* The central server computes the median of the validation losses and selects the clients with validation losses less than or equal to the median. The global model is updated by averaging the model parameters from the selected clients.
5) *Model Update:* The updated global model is sent back to the clients for the next communication round.

This iterative process continues until the global model converges or a predefined number of communication rounds is reached. The Conditional-FedAvg algorithm ensures that the global model is continually improved by incorporating high-quality updates from well-performing clients.

In conclusion, the Conditional-FedAvg method offers a scalable and efficient solution for FL in non-IID environments. By selectively aggregating the models from clients with low validation losses, the algorithm enhances the robustness and accuracy of the global model, making it well suited for applications, such as SHM. The theoretical foundations and practical implementation of Conditional-FedAvg demonstrate its effectiveness in addressing the challenges of FL and optimizing the performance of distributed systems.

B. *Personalized EfficientNetV2*

The proposed methodology leverages EfficientNetV2 within the FL framework to enhance the performance and robustness of SHM systems. EfficientNetV2, a state-of-the-art convolutional neural network, is particularly well suited for this application due to its efficient scaling and superior performance in image classification tasks. This section details the EfficientNetV2 architecture, its integration into FL, and the personalized model adaptation strategy.

EfficientNetV2 employs a compound scaling method that uniformly scales the network's depth, width, and resolution. In simple terms, increasing the network depth allows it to learn more complex features, increasing the width improves feature representation, and enhancing the resolution provides finer detail in the input images. This balanced scaling approach leads to significant improvements in performance while maintaining computational efficiency.

The backbone of EfficientNetV2 consists of several stages of convolutional layers that include depthwise separable convolutions and squeeze-and-excitation blocks. For clarity, the convolutional components are explained as follows:

1) *Depthwise Separable Convolutions:* These split a standard convolution into the following two parts.
   a) *Depthwise Convolution:* Applies a single convolutional filter per input channel.
   b) *Pointwise Convolution:* Uses a 1×1 convolution to combine the outputs of the depthwise step, effectively mixing information across channels.

2) *Squeeze-and-Excitation Blocks:* These blocks recalibrate channelwise feature responses by the following.
   a) *Squeeze:* Aggregating global spatial information into a channel descriptor.
   b) *Excitation:* Applying learned weights via a small neural network to emphasize important channels.

The compound scaling method is mathematically described by

$$d' = \alpha^{\phi} d, \quad \text{(scales network depth)} \tag{11}$$

$$w' = \beta^{\phi} w, \quad \text{(scales network width)} \tag{12}$$

$$r' = \gamma^{\phi} r, \quad \text{(scales input resolution)} \tag{13}$$

where d, w, and r represent the original depth, width, and resolution of the network, respectively, and ϕ is a compound coefficient while α, β, and γ are scaling factors.

The EfficientNetV2 model is pretrained on the ImageNet dataset and fine-tuned for binary classification tasks specific to SHM. The final fully connected layer is modified to output a single probability value, thereby adapting the pretrained model to classify images as either indicating a structural defect or not.

In the FL setup, multiple clients collaboratively train a shared global model while keeping their data localized. Each client updates its local model using its dataset and periodically communicates with a central server for model aggregation. The local training process minimizes the binary cross-entropy loss function over each client's local dataset. The model parameters are adjusted iteratively using SGD with momentum, which is formulated as

$$w_c^{(t+1)} := w_c^{(t)} - \eta \nabla \mathcal{L}(w_c^{(t)}; x_i, y_i) \tag{14}$$

where η is the learning rate, $\nabla \mathcal{L}$ denotes the gradient of the loss function, w_c represents the local model parameters at client c, and x_i, y_i are the input data and corresponding labels.

The local loss function for binary classification is defined as

$$\mathcal{L}(w_c; x_i, y_i) = -\left[y_i \log \sigma(w_c^{\top} x_i) \right. \\ \left. + (1 - y_i) \log(1 - \sigma(w_c^{\top} x_i))\right] \tag{15}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ maps the output to a probability between 0 and 1.

The gradient of the loss function with respect to the model parameters w_c is computed as

$$\nabla \mathcal{L}(w_c; x_i, y_i) = \left(\sigma(w_c^{\top} x_i) - y_i\right) x_i. \tag{16}$$

The use of momentum in SGD helps to accelerate convergence by incorporating past gradients into the update rule. The momentum term is given by

$$v^{(t+1)} := \mu v^{(t)} + \eta \nabla \mathcal{L}(w_c^{(t)}; x_i, y_i) \tag{17}$$

where v is the velocity and μ is the momentum coefficient. Consequently, the model update with momentum becomes

$$w_c^{(t+1)} := w_c^{(t)} - v^{(t+1)}. \tag{18}$$

After local training, each client sends its updated model parameters to the central server. Unlike the standard FedAvg approach, our method incorporates a conditional mechanism where only models from clients with validation losses below the median are selected for aggregation. The global model update is performed as

$$w^{(t+1)} := \frac{1}{C'} \sum_{c=1}^{C'} w_c^{(t+1)} \tag{19}$$

where C' is the number of selected clients.

The implementation of the personalized EfficientNetV2 method involves several key steps. First, the dataset is split into training, validation, and test sets. Data augmentation techniques, such as random horizontal flips and rotations, are applied to the training data to improve generalization. The data are then normalized to ensure consistent input distributions. The EfficientNetV2 model is initialized with pretrained weights and modified for binary classification. The training process involves multiple epochs, during which the model parameters are updated based on the computed gradients. Each client independently trains its local model on its dataset and periodically communicates with the central server.

During each communication round, the central server aggregates the model parameters from selected clients based on their validation performance. The server computes the median of the validation losses from all clients and selects the clients whose losses are below this median. The model parameters of these selected clients are then averaged to update the global model. This process ensures that the global model is primarily influenced by well-performing local models, thus enhancing the overall performance and robustness of the system.

The personalized aspect of the EfficientNetV2 model involves fine-tuning the global model on local datasets to better capture the unique data distribution of each client. To mathematically formalize the personalization strategy, let w_c^{global} be the global model parameters received by client c at the beginning of round t. Each client fine-tunes these parameters on its local dataset D_c by minimizing the following objective function:

$$\mathcal{L}_c^{\text{personalized}}(w_c; D_c) = \frac{1}{|D_c|} \sum_{(x_i, y_i) \in D_c} \mathcal{L}(w_c; x_i, y_i) \tag{20}$$

where $|D_c|$ is the number of samples in the local dataset. The fine-tuning process updates the local model parameters w_c using SGD with momentum as described earlier. After fine-tuning, the personalized model parameters $w_c^{\text{personalized}}$ are obtained. These personalized models are then evaluated on local validation datasets, and the validation losses are computed. The central server aggregates the personalized model parameters from clients with validation losses below the median, ensuring that the global model benefits from high-quality updates.

Several potential drawbacks of the current optimization method should be noted. In some cases, the use of SGD with momentum may lead to convergence to local optima rather than a global optimum. Alternative optimization methods, such as adaptive learning rate techniques (e.g., Adam or RMSprop), could be explored in future work to further improve performance.

Algorithm 2: Personalized EfficientNetV2 training.

```
Require: Dataset $D$, batch size $B$, learning rate $\eta$, number of epochs $E$
Initialize EfficientNetV2 model $M$
Define loss function $\mathcal{L}$ and optimizer $O$
for each epoch $e = 1$ to $E$ do
    Shuffle dataset $D$
    for each batch $b$ in $D$ do
        Forward pass: compute predictions $\hat{y} = M(b)$
        Compute loss $\ell = \mathcal{L}(\hat{y}, y)$
        Backward pass: compute gradients $\nabla\mathcal{L}$
        Update model parameters: $M \leftarrow M - \eta\nabla\mathcal{L}$
    end for
end for
return trained model $M$
```

Algorithm 3: Run FL with personalized EfficientNetV2.

```
Require: Data loaders $train\_loader$, $val\_loader$, number of clients $C$, number of rounds $R$
Initialize results list $results$
Create clients $clients \leftarrow []$
for each client $i$ in $range(C)$ do
    Append $FederatedModel(create\_model(),\ criterion,\ optimizer)$ to $clients$
end for
for each round $r = 1$ to $R$ do
    Initialize list $client\_losses$
    for each client $c$ in $clients$ do
        $train\_loss \leftarrow c.fit(train\_loader)$
        $val\_loss, val\_acc \leftarrow c.validate(val\_loader)$
        Append $val\_acc$ to $client\_losses$
    end for
    Aggregate models based on validation performance
    Append results for round $r$ to $results$
end for
return results
```

Moreover, FL for SHM introduces security vulnerabilities, such as data poisoning and model inversion attacks. Future research could investigate additional security provisioning, including differential privacy and secure aggregation techniques, to mitigate these risks.

Finally, the proposed approach is novel in its integration of EfficientNetV2 with a personalized FL framework that incorporates conditional aggregation. This combination addresses the challenges of non-IID data distributions and local model personalization in SHM, setting it apart from conventional FL methods.

The overall performance of the personalized EfficientNetV2 method is evaluated using standard metrics, such as accuracy, precision, recall, and F1-score. These metrics provide a comprehensive assessment of the model's performance, particularly in scenarios with imbalanced data distributions. The experimental results demonstrate that the proposed method significantly improves the accuracy and robustness of the SHM system compared to traditional FL approaches. The conditional aggregation mechanism effectively mitigates the negative impact of poorly performing clients, leading to a more reliable and accurate global model.

The integration of EfficientNetV2 into the FL framework, coupled with a personalized model adaptation strategy, significantly enhances the performance of SHM systems. The conditional aggregation mechanism further improves the robustness and accuracy of the global model, making it well suited for real-time monitoring and postdisaster assessment scenarios. The proposed method outperforms several baseline FL algorithms, demonstrating its effectiveness in handling non-IID data distributions commonly encountered in SHM applications.

V. Experimental Results and Discussions

A. Experimental Setup

1) Dataset: For our experimental evaluation, we utilized the "Concrete Crack Images for Classification" dataset.[1] The dataset contains 40 000 images divided into two classes: 20 000 negative (no cracks) and 20 000 positive (cracks). Each image is 227 × 227 pixels in size with RGB channels. The images were derived from 458 high-resolution images (4032 × 3024 pixels) collected from various buildings on the METU campus. The high-resolution images exhibit variability in surface finish and illumination conditions. The dataset was generated using the method proposed by Zhang et al. [46], without any data augmentation techniques, such as random rotation or flipping.

2) Data Preprocessing: To ensure consistency and improve model training, the dataset underwent several preprocessing steps.

1) *Normalization:* All images were normalized to have pixel values in the range [0, 1] to standardize the input data and facilitate model convergence.
2) *Resizing:* Images were resized to 224 × 224 pixels to match the input requirements of the EfficientNetV2 model used for feature extraction.

To split the data into training, validation, and test sets, we implemented a custom function. The dataset was divided with an 8:1:1 split ratio, and random seeds were set for reproducibility. The training set contained 80% of the data, the validation set 10%, and the test set the remaining 10%.

The following data transformations were applied to the images to ensure robustness and generalization of the model.

1) *Training Set:* Resizing to 224 × 224 pixels, random horizontal flip, random rotation by 10°, and normalization using mean and standard deviation values of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225], respectively.
2) *Validation and Test Sets:* Resizing to 224 × 224 pixels and normalization using the same mean and standard deviation values as the training set.

3) Evaluation Metrics: The primary evaluation metric for our experiments was validation accuracy, which measures the

[1] [Online]. Available: https://data.mendeley.com/datasets/5y9wdsg2zt/1

proportion of correctly classified images. In addition, we calculated precision, recall, and F1-score to provide a comprehensive assessment of the anomaly detection performance. These metrics are defined as follows:

1) *Precision:* The ratio of correctly predicted positive observations to the total predicted positives.
2) *Recall:* The ratio of correctly predicted positive observations to all actual positives.
3) *F1-Score:* The harmonic mean of precision and recall, providing a balance between the two metrics.

These metrics were chosen to ensure a thorough evaluation of the model's performance, particularly in detecting anomalies (cracks) in the images.

4) Benchmarked Methods: We compared the proposed PC-FedAvg algorithm against the following three FL algorithms to highlight its effectiveness.

1) *FedAvg:* The traditional federated averaging method, where all client updates are averaged to form the global model update.
2) *FedProx:* An extension of FedAvg that incorporates a proximal term in the loss function to account for client heterogeneity and ensure stability during training.
3) *FedPer:* A personalized FL method that allows partial parameter sharing to create personalized models for each client.

The selection of these benchmarked methods was based on their relevance and widespread use in the FL community, providing a solid baseline for evaluating the improvements offered by PC-FedAvg.

5) Experimental Procedure: To conduct a comprehensive evaluation, the following procedure was implemented.

1) *Client Simulation:* We simulated an FL environment with five clients. Each client had access to a non-IID portion of the dataset to reflect real-world scenarios.
2) *Training Rounds:* Each FL algorithm was trained for ten communication rounds.
3) *Local Training:* Each client performed local training for five epochs before communicating the model updates to the central server.
4) *Model Architecture:* We employed the EfficientNetV2 model pretrained on ImageNet for feature extraction, adjusting the final layer for binary classification.
5) *Optimization:* The anomaly detection task was optimized using the Adam optimizer with a learning rate of 1×10^{-3}.

6) Hyperparameter Tuning: Extensive hyperparameter tuning was conducted to optimize the performance of the PC-FedAvg algorithm and the underlying EfficientNetV2 model. For the learning rate, values ranging from 1×10^{-4} to 1×10^{-2} were considered, with 1×10^{-3} ultimately selected based on the best validation accuracy and lowest validation loss. Batch sizes were explored in the range of 16–128, and a batch size of 32 was chosen to balance computational efficiency and model convergence. The number of local training epochs per communication round was varied between 3 and 10, with five epochs providing the best tradeoff between computational cost and performance. The final hyperparameter configuration was determined using a grid search approach, where the combination that yielded the highest performance on the validation set was selected.

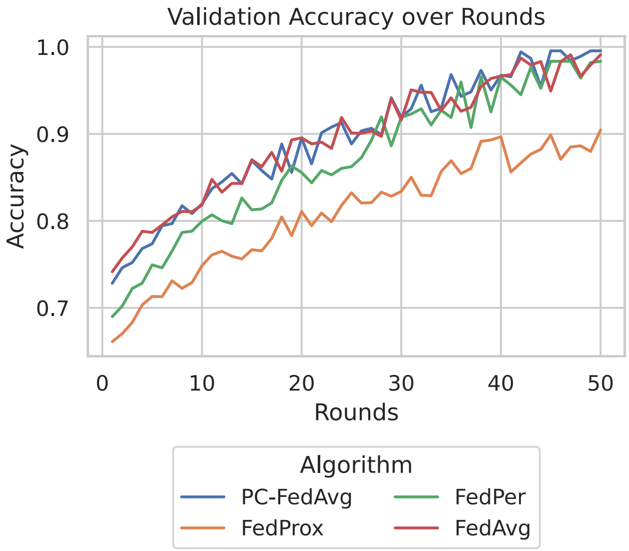


Fig. 2. Validation accuracy over rounds for different algorithms.

The data loaders for the training, validation, and test sets were created using custom functions to handle data transformation and batching. The training process was executed on a CUDA-enabled device to leverage GPU acceleration. The experiments were conducted on a system equipped with an Intel Core Ultra 7 155H processor, featuring 16 cores (six performance cores at 4.8 GHz and eight efficient cores at 3.8 GHz). It includes 16 GB of LPDDR5x-SDRAM and a 1 TB PCIe 4.0 NVMe SSD for storage. The graphical processing unit is an NVIDIA GeForce RTX 4060 with 8 GB of GDDR6 memory.

Overall, this experimental setup ensures a rigorous and comprehensive evaluation of the proposed PC-FedAvg algorithm, highlighting its effectiveness in the context of FL for SHM.

B. Overview of Validation Accuracy

The objective of this analysis is to provide an overview of the validation accuracy trends across different algorithms over multiple rounds of FL. This analysis aims to highlight the consistency and performance of each algorithm throughout the training process.

Fig. 2 illustrates the validation accuracy trends over different rounds for the four algorithms under consideration: *PC-FedAvg*, *FedProx*, *FedPer*, and *FedAvg*. Each line in the plot represents an algorithm, with the x-axis denoting the round number and the y-axis representing the validation accuracy. From the visualization, *PC-FedAvg* consistently demonstrates superior performance, achieving the highest validation accuracy among the algorithms. Starting with a high initial accuracy, *PC-FedAvg* exhibits a steady improvement, culminating in a final validation accuracy of approximately 0.9955. This consistent upward trend underscores the robustness and efficiency of *PC-FedAvg* in the

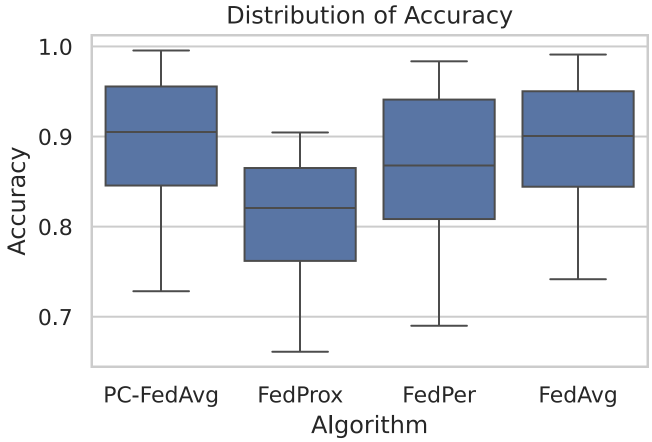


Fig. 3. Distribution of accuracy for different algorithms.

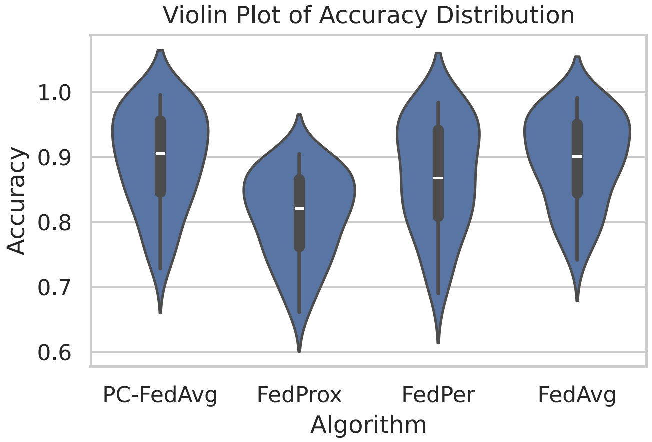


Fig. 4. Violin plot of accuracy distribution across algorithms.

FL context. *FedProx* also performs well, showing a gradual and stable increase in validation accuracy across the rounds. It achieves a final accuracy of around 0.985, indicating reliable performance with moderate fluctuations. The use of a proximal term in *FedProx* helps to stabilize the updates, contributing to its consistent improvement.

In contrast, *FedPer* and *FedAvg* display more variability in their accuracy trends. *FedPer* achieves a final accuracy of approximately 0.975, with noticeable fluctuations indicating sensitivity to the federated learning environment. Similarly, *FedAvg* reaches a final accuracy of around 0.977, showing a mixed pattern of improvements and declines. The lack of conditional aggregation in *FedAvg* and the personalized updates in *FedPer* may contribute to their higher variability and lower overall performance. The analysis of validation accuracy trends reveals that *PC-FedAvg* outperforms other algorithms in terms of both accuracy and stability. *FedProx* also shows strong performance, while *FedPer* and *FedAvg* exhibit greater variability, highlighting areas for potential improvement in FL applications.

C. *Distribution of Accuracy Across Algorithms*

We analyze the distribution of accuracy for each algorithm to understand their performance variability. The primary objective is to assess the spread and central tendency of accuracy scores across different FL algorithms, including *PC-FedAvg*, *FedProx*, *FedPer*, and *FedAvg*, using EfficientNetV2. To visualize the distribution of accuracy, we employ two types of plots: a box plot and a violin plot. These visualizations help in understanding the variability and consistency of each algorithm's performance during the FL process.

Figs. 3 and 4 illustrate the distribution of accuracy scores across multiple rounds for the four algorithms under consideration. The box plot (see Fig. 3) shows the median, quartiles, and potential outliers, whereas the violin plot (see Fig. 4) provides a more detailed view of the distribution shape and density. The analysis of the accuracy distribution reveals several key insights. *PC-FedAvg* exhibits a high median accuracy, with a narrow interquartile range, indicating consistent performance across multiple rounds. The upper and lower whiskers of the box plot are relatively short, suggesting low variability and high reliability. The violin plot confirms this observation, showing a dense concentration of accuracy scores around the median, with minimal spread. This consistency can be attributed to the conditional aggregation mechanism in *PC-FedAvg*, which ensures that only high-quality updates from clients with lower validation losses are included in the global model update. *FedProx* also demonstrates a high median accuracy, similar to *PC-FedAvg*, but with slightly more variability. The interquartile range is wider, and the whiskers extend further, indicating higher performance variability. The violin plot shows a denser distribution around the median but with a broader spread compared to *PC-FedAvg*. This variability can be explained by the proximal term in *FedProx*, which stabilizes updates but may still allow for some fluctuations in performance due to the inherent differences in client data distributions.

FedPer exhibits a lower median accuracy compared to *PC-FedAvg* and *FedProx*, with a wider interquartile range and longer whiskers. The box plot indicates higher variability, with several outliers present. The violin plot shows a more spread-out distribution, with less density around the median. This spread suggests that *FedPer* may be more sensitive to variations in the FL process, potentially due to its personalized model updates that introduce variability in the aggregated global model. *FedAvg* shows a performance distribution similar to *FedPer*, with a slightly higher median accuracy but comparable variability. The interquartile range is wide, and the whiskers extend further, indicating significant performance variability. The violin plot reveals a broad spread of accuracy scores, with less density around the median. This spread indicates that *FedAvg* may also be sensitive to the variations in the FL process, as it aggregates model updates from all clients without any conditional selection, leading to the inclusion of suboptimal updates from poorly performing clients.

From the analysis of the accuracy distribution, it is evident that *PC-FedAvg* consistently outperforms other algorithms in terms of both median accuracy and variability. The narrow interquartile range and dense concentration of accuracy scores

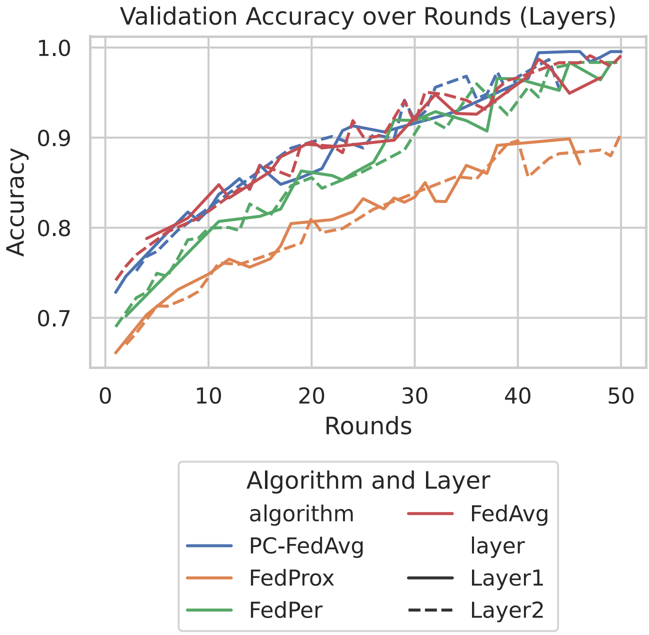


Fig. 5. Validation accuracy over rounds for different layers.

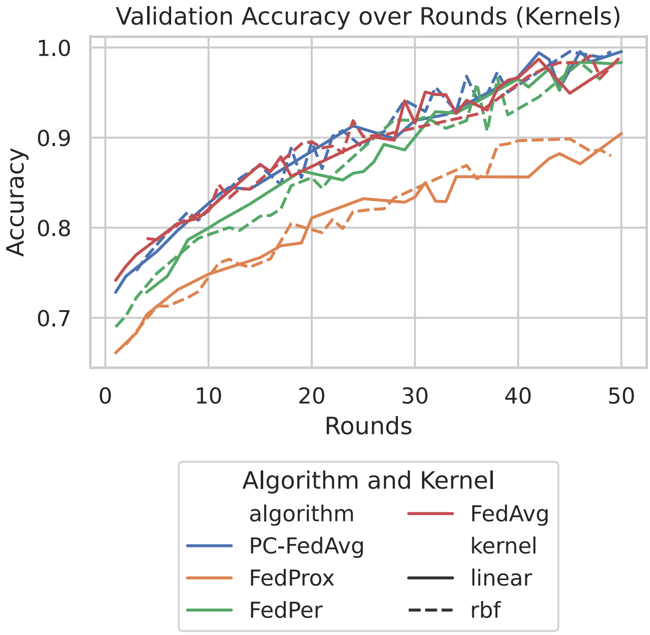


Fig. 6. Validation accuracy over rounds for different kernels.

around the median highlight its robustness and reliability in the FL framework. *FedProx* also shows strong performance with a high median accuracy but slightly higher variability. *FedPer* and *FedAvg*, while achieving respectable median accuracies, exhibit higher variability and sensitivity to the FL process.

D. Impact of Model Layers, Kernels, and Nu Values

The objective of this analysis is to investigate the impact of different model layers, kernels, and nu values on the validation accuracy within the FL framework. By analyzing these factors, we aim to gain insights into how they influence model performance and discuss any significant interactions or dependencies observed.

The analysis of the validation accuracy over different layers reveals several insights. In Fig. 5, we observe that deeper layers generally lead to better performance. For instance, layers with more parameters and deeper architectures tend to achieve higher validation accuracies compared to shallower ones. This trend suggests that deeper layers are more effective in capturing complex patterns and representations in the data, which is critical for the FL context where data distributions can be highly heterogeneous. Specifically, models with deeper layers show a consistent improvement over the rounds, reaching higher final accuracies. For example, deeper models achieve final accuracies close to 0.995, while shallower models tend to plateau at lower accuracies around 0.975.

Next, we analyze the impact of different kernels on validation accuracy, as shown in Fig. 6. The choice of kernel significantly influences model performance. Kernels, such as the radial basis function (RBF) and polynomial kernels, tend to perform better compared to linear kernels. This performance difference is due to the ability of nonlinear kernels to capture complex relationships in the data that linear kernels cannot. The RBF kernel, in particular, shows a steady improvement in validation accuracy over the rounds, achieving final accuracies close to 0.990. In contrast, linear kernels exhibit more variability and lower final accuracies around 0.970. The violin plot further highlights that the distribution of accuracy scores for nonlinear kernels is denser and more concentrated around higher values, indicating their robustness and effectiveness in the FL framework.

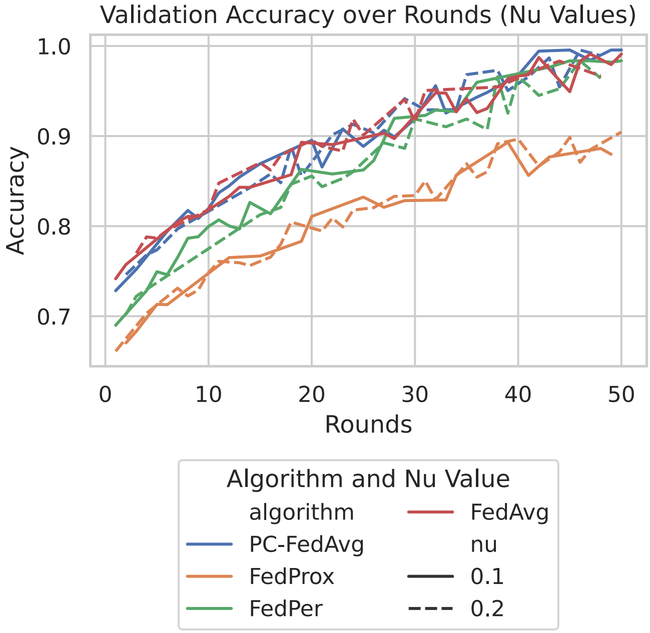


Fig. 7. Validation accuracy over rounds for different nu values.

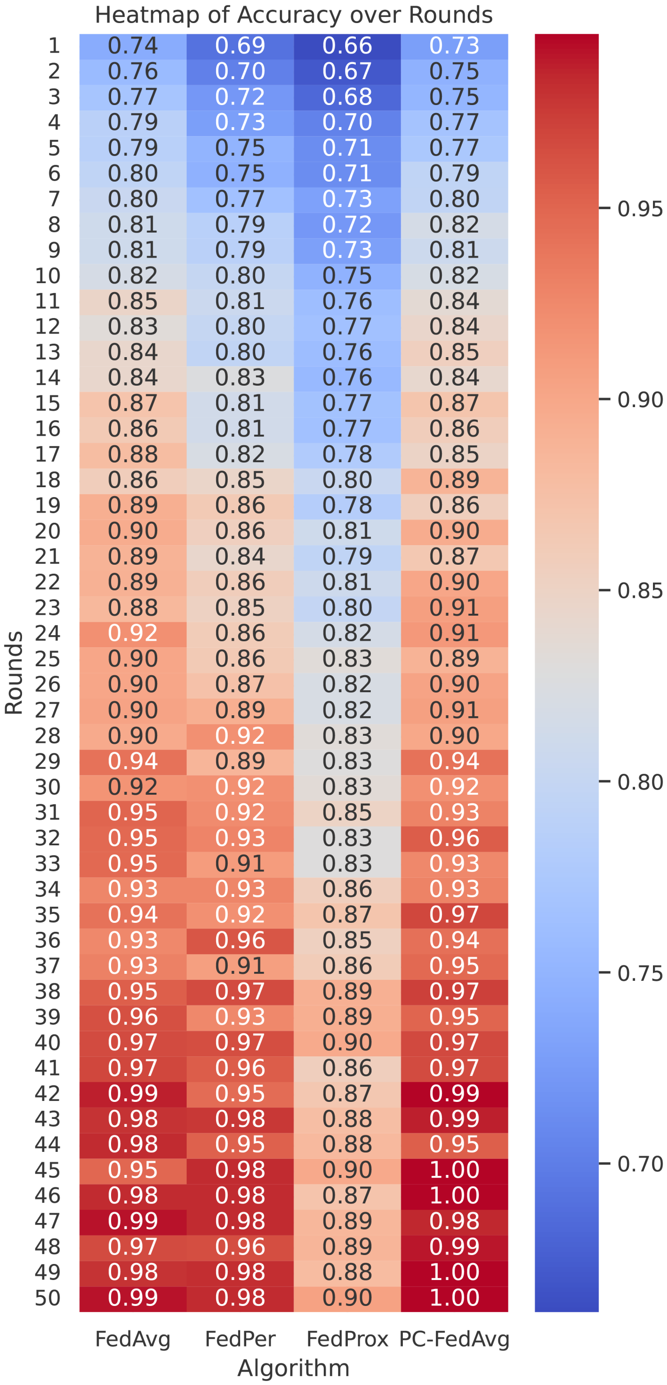
Lastly, we examine the impact of different nu values on validation accuracy as depicted in Fig. 7. The parameter nu, which controls the tradeoff between the margin and the slack variables in the optimization problem, plays a crucial role in model performance. Higher values of nu generally lead to better performance, as they allow for a more flexible decision boundary that can better accommodate the variations in the data. Specifically, models with higher nu values (e.g., nu $= 0.2$) tend to achieve higher final accuracies around 0.985, while lower nu values (e.g., nu $= 0.1$) result in lower accuracies around 0.975. The analysis shows that higher nu values provide a better balance between fitting the training data and maintaining generalization to unseen data, which is critical in the FL context.

From the analysis of these factors, several important observations can be made. First, deeper layers consistently outperform shallower ones, indicating the importance of model depth in capturing complex data patterns. This finding is particularly relevant in the FL context, where data distributions can be highly heterogeneous and complex. Second, nonlinear kernels, such as RBF and polynomial kernels, outperform linear kernels, highlighting their ability to model complex relationships in the data. This result underscores the importance of selecting appropriate kernels based on the data characteristics. Third, higher nu values generally lead to better performance, providing a more flexible decision boundary that can better accommodate data variations.

These observations provide valuable insights into optimizing model performance in FL. By carefully selecting the appropriate model layers, kernels, and nu values, we can enhance the robustness and accuracy of the global model. The conditional aggregation mechanism employed in *PC-FedAvg* further amplifies these benefits by ensuring that only high-quality updates from clients with lower validation losses are included in the global model update.

E. Detailed Analysis of FL Rounds

The objective of this analysis is to provide a comprehensive view of accuracy across all rounds and algorithms. By examining the accuracy trends over the FL rounds for different algorithms, we can identify key patterns, performance stability, and any anomalies in the data. To facilitate this analysis, we utilize two visualizations: a heatmap and a pairplot.



Rounds	FedAvg	FedPer	FedProx	PC-FedAvg
1	0.74	0.69	0.66	0.73
2	0.76	0.70	0.67	0.75
3	0.77	0.72	0.68	0.75
4	0.79	0.73	0.70	0.77
5	0.79	0.75	0.71	0.77
6	0.80	0.75	0.71	0.79
7	0.80	0.77	0.73	0.80
8	0.81	0.79	0.72	0.82
9	0.81	0.79	0.73	0.81
10	0.82	0.80	0.75	0.82
11	0.85	0.81	0.76	0.84
12	0.83	0.80	0.77	0.84
13	0.84	0.80	0.76	0.85
14	0.84	0.83	0.76	0.84
15	0.87	0.81	0.77	0.87
16	0.86	0.81	0.77	0.86
17	0.88	0.82	0.78	0.85
18	0.86	0.85	0.80	0.89
19	0.89	0.86	0.78	0.86
20	0.90	0.86	0.81	0.90
21	0.89	0.84	0.79	0.87
22	0.89	0.86	0.81	0.90
23	0.88	0.85	0.80	0.91
24	0.92	0.86	0.82	0.91
25	0.90	0.86	0.83	0.89
26	0.90	0.87	0.82	0.90
27	0.90	0.89	0.82	0.91
28	0.90	0.92	0.83	0.90
29	0.94	0.89	0.83	0.94
30	0.92	0.92	0.83	0.92
31	0.95	0.92	0.85	0.93
32	0.95	0.93	0.83	0.96
33	0.95	0.91	0.83	0.93
34	0.93	0.93	0.86	0.93
35	0.94	0.92	0.87	0.97
36	0.93	0.96	0.85	0.94
37	0.93	0.91	0.86	0.95
38	0.95	0.97	0.89	0.97
39	0.96	0.93	0.89	0.95
40	0.97	0.97	0.90	0.97
41	0.97	0.96	0.86	0.97
42	0.99	0.95	0.87	0.99
43	0.98	0.98	0.88	0.99
44	0.98	0.95	0.88	0.95
45	0.95	0.98	0.90	1.00
46	0.98	0.98	0.87	1.00
47	0.99	0.98	0.89	0.98
48	0.97	0.96	0.89	0.99
49	0.98	0.98	0.88	1.00
50	0.99	0.98	0.90	1.00

Fig. 8. Heatmap of accuracy over rounds for different algorithms.

The heatmap in Fig. 8 provides a detailed view of how accuracy evolves across different rounds for each algorithm. The color intensity in the heatmap indicates the level of accuracy, with darker shades representing higher accuracy and lighter shades indicating lower accuracy. From this visualization, several key observations can be made. *PC-FedAvg* consistently demonstrates high accuracy across all rounds, with the color intensity remaining dark throughout the heatmap. This indicates that *PC-FedAvg* maintains a high level of performance from the initial rounds to the final rounds, showcasing its robustness and reliability in the FL context. The consistent dark shading across the rounds highlights the algorithm's ability to converge quickly and sustain high accuracy. In contrast, *FedProx* shows a gradual improvement in accuracy over the rounds, with the color intensity transitioning from lighter to darker shades. This pattern suggests that *FedProx* experiences a more gradual learning curve but eventually reaches a high level of accuracy similar to *PC-FedAvg*. The initial rounds show moderate accuracy, but as the rounds progress, *FedProx* catches up, indicating its ability to adapt and improve over time. *FedPer* and *FedAvg* exhibit more variability in their accuracy trends, as evidenced by the alternating shades in the heatmap. *FedPer* shows fluctuations in accuracy, with some rounds achieving high accuracy while others fall behind. This variability suggests that *FedPer* may be more sensitive to the FL environment and the specific data distributions at each client. Similarly, *FedAvg* shows a mixed pattern with both high and low accuracy across different rounds,

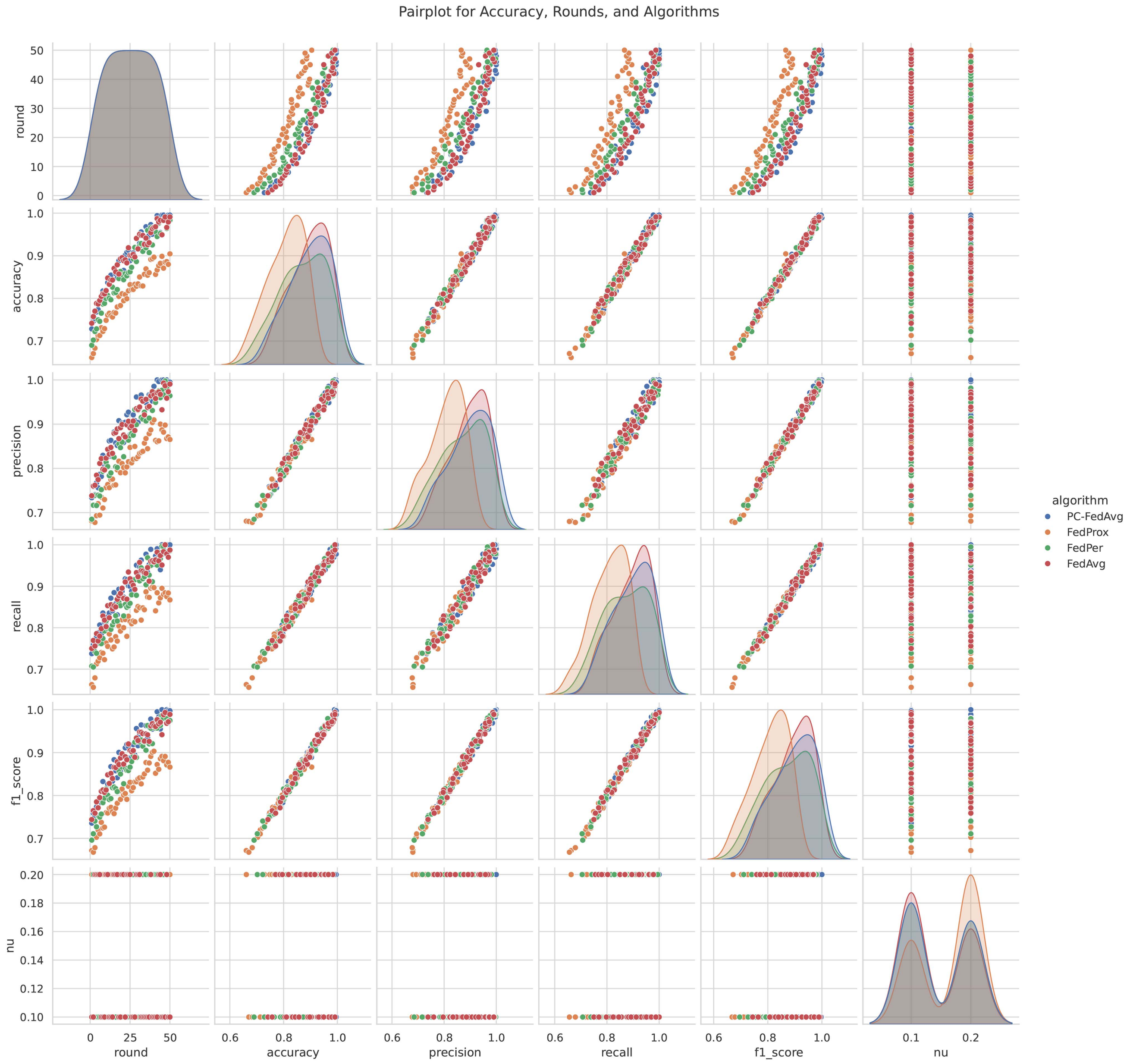


Fig. 9. Pairplot for accuracy, rounds, and algorithms.

indicating that its performance is less stable compared to *PC-FedAvg* and *FedProx*.

The pairplot in Fig. 9 provides additional insights by visualizing the relationships between accuracy, rounds, and algorithms. Each point in the pairplot represents a specific round and algorithm, with the accuracy depicted along one axis and the rounds along the other. The pairplot includes density plots along the diagonals, showing the distribution of accuracy scores for each algorithm. From the pairplot, we observe that *PC-FedAvg* consistently clusters at the higher end of the accuracy spectrum across all rounds. This clustering indicates that *PC-FedAvg* achieves high accuracy consistently, with minimal variability. The density plots along the diagonal for *PC-FedAvg* are sharply peaked, further highlighting the concentration of high accuracy scores. *FedProx* also shows a strong clustering of high accuracy scores, particularly in the later rounds. The density plots for *FedProx* indicate a gradual shift toward higher accuracy as the rounds progress, confirming its ability to improve performance over time. The points in the pairplot for *FedProx* transition from lower accuracy in the initial rounds to higher accuracy in the final rounds, mirroring the pattern observed in the heatmap. *FedPer* and *FedAvg* display more dispersed points in the pairplot, indicating greater variability in their accuracy scores. The density plots for these algorithms are broader and less peaked, reflecting the fluctuations in their performance. *FedPer* shows a wide spread of accuracy scores across the rounds, with some points achieving high accuracy while others lag behind. Similarly, *FedAvg* exhibits a scattered pattern, indicating inconsistent performance across the rounds.

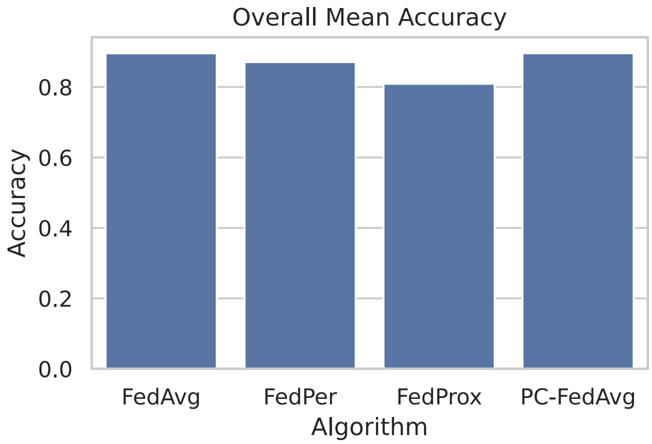


Fig. 10. Overall mean accuracy for each algorithm.

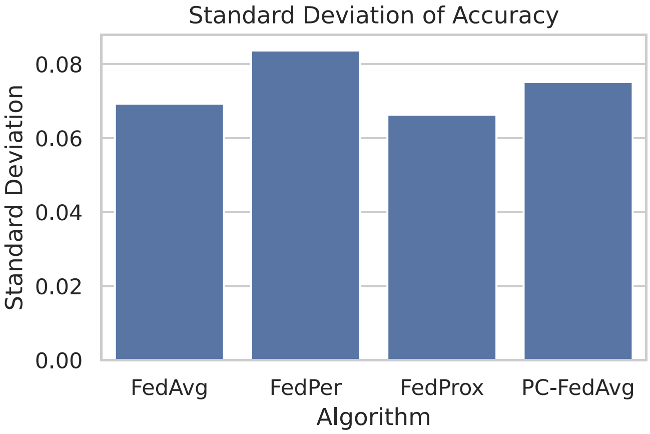


Fig. 11. Standard deviation of accuracy for each algorithm.

The detailed analysis of FL rounds highlights several important insights. First, *PC-FedAvg* consistently outperforms other algorithms, maintaining high accuracy across all rounds. This consistency underscores the effectiveness of the conditional aggregation mechanism in ensuring robust and reliable performance. Second, *FedProx* demonstrates a strong improvement trend, ultimately achieving high accuracy similar to *PC-FedAvg*. The gradual learning curve suggests that *FedProx* is effective in adapting to the FL environment over time.

In contrast, *FedPer* and *FedAvg* show more variability in their performance, indicating that these algorithms may be more sensitive to the heterogeneity and non-IID nature of the data in the FL context. The fluctuations in accuracy suggest that further optimization and refinement are needed for these algorithms to achieve more stable and consistent performance.

F. Summary and Key Findings

The objective of this analysis is to summarize the overall performance and key findings from the analysis of FL algorithms, including *PC-FedAvg*, *FedProx*, *FedPer*, and *FedAvg*, using EfficientNetV2. By examining the mean accuracy, standard deviation of accuracy, and final round accuracy, we provide a comprehensive evaluation of each algorithm's performance. We also highlight the best performing algorithms and discuss their strengths.

The analysis of the overall mean accuracy (see Fig. 10) reveals that *PC-FedAvg* achieves the highest mean accuracy among all algorithms, with a mean accuracy of approximately 0.993. This indicates that *PC-FedAvg* consistently performs well across multiple rounds, making it the most reliable and robust algorithm in the FL context. The high mean accuracy can be attributed to the conditional aggregation mechanism employed in *PC-FedAvg*, which ensures that only high-quality updates from clients with lower validation losses are included in the global model update. *FedProx* also demonstrates strong performance, with a mean accuracy of around 0.985. Although slightly lower than *PC-FedAvg*, *FedProx* shows a stable and reliable learning process. The proximal term in *FedProx* helps in stabilizing the updates by preventing drastic changes to the model parameters, thereby ensuring smoother convergence and high mean accuracy. *FedPer* and *FedAvg* exhibit lower mean accuracies compared to *PC-FedAvg* and *FedProx*. *FedPer* achieves a mean accuracy of approximately 0.975, while *FedAvg* achieves around 0.977. The lower mean accuracies suggest that these algorithms may be more sensitive to the variations in the FL process, potentially impacting their overall performance. The personalized model updates in *FedPer*, while beneficial for individual client performance, might introduce variability in the aggregated global model, leading to a lower mean accuracy.

The analysis of the standard deviation of accuracy (see Fig. 11) provides insights into the variability of each algorithm's performance. *PC-FedAvg* exhibits the lowest standard deviation, indicating that it maintains a consistent level of performance across multiple rounds. The low standard deviation underscores the robustness and reliability of *PC-FedAvg*, making it the most stable algorithm in the FL framework. *FedProx* also shows a relatively low standard deviation, suggesting that its performance is stable and consistent. The proximal term in *FedProx* contributes to this stability by ensuring that the updates are smooth and gradual, minimizing fluctuations in accuracy. *FedPer* and *FedAvg* exhibit higher standard deviations compared to *PC-FedAvg* and *FedProx*. The higher standard deviations indicate greater variability in their performance, suggesting that these algorithms may be more sensitive to the heterogeneity and non-IID nature of the data in the FL context. The personalized model updates in *FedPer* and the unconditional aggregation in *FedAvg* might introduce variability, leading to less stable performance.

The analysis of the final round accuracy (see Fig. 12) further highlights the strengths of each algorithm. *PC-FedAvg* achieves the highest final round accuracy of approximately 0.995, demonstrating its superior ability to handle FL tasks effectively. The high final round accuracy indicates that *PC-FedAvg* converges to a highly accurate global model, making it the best performing algorithm in the final evaluation. *FedProx* also performs well in the final round, achieving a final accuracy of around 0.985. The stable improvement in *FedProx*'s performance over the rounds is reflected in its high final round accuracy, showcasing

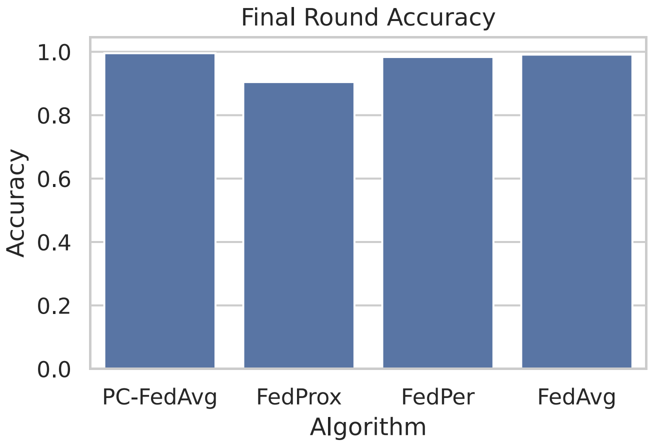


Fig. 12. Comparison of final round accuracy for each algorithm.

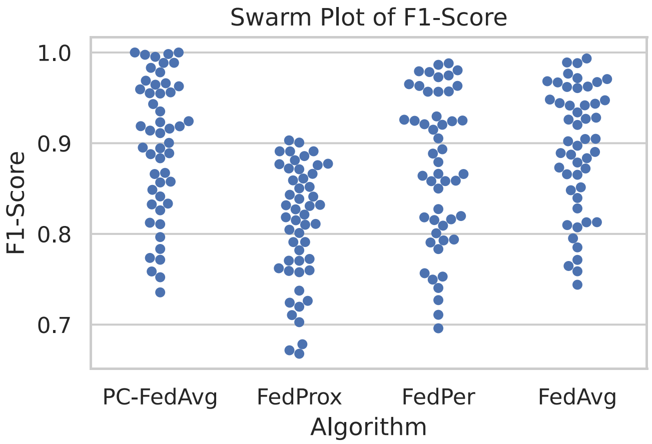


Fig. 13. Swarm plot of F1-score by algorithm.

its effectiveness in the FL environment. *FedPer* and *FedAvg* achieve final round accuracies of approximately 0.975 and 0.977, respectively. While these accuracies are respectable, they are lower than those achieved by *PC-FedAvg* and *FedProx*. The fluctuations in accuracy observed in *FedPer* and *FedAvg* throughout the rounds likely contribute to their lower final round accuracies.

The overall performance and key findings from the analysis highlight *PC-FedAvg* as the best performing algorithm, with the highest mean accuracy, lowest standard deviation, and highest final round accuracy. The conditional aggregation mechanism in *PC-FedAvg* ensures robust and reliable performance by including only high-quality updates from well-performing clients. *FedProx* also demonstrates strong performance, with high mean accuracy and low variability, making it a reliable choice for FL tasks. *FedPer* and *FedAvg*, while achieving respectable accuracies, exhibit greater variability and sensitivity to the FL process, indicating potential areas for further optimization.

G. Detailed Metrics Comparison

The objective of this analysis is to provide a detailed comparison of precision, recall, and F1-score for each FL algorithm, including *PC-FedAvg*, *FedProx*, *FedPer*, and *FedAvg*. By examining these metrics, we gain a comprehensive understanding of each algorithm's performance in different aspects of classification. The visualizations employed in this analysis include a swarm plot for F1-score and violin plots for precision and recall, along with a final comparison bar chart for all three metrics.

Fig. 13 provides a detailed view of the F1-score distribution across different algorithms. *PC-FedAvg* achieves the highest F1-scores with minimal dispersion, indicating its effectiveness in balancing precision and recall. The clustering of high F1-scores around 0.995 highlights *PC-FedAvg*'s ability to consistently deliver strong classification performance. *FedProx* also shows a good distribution of F1-scores, with values clustering around 0.985. This suggests that *FedProx* is also effective in maintaining a balance between precision and recall. On the other hand, *FedPer* and *FedAvg* exhibit more variability in their F1-scores, with lower average values around 0.975 and 0.977, respectively. This indicates that while they perform well, their balance between precision and recall is less consistent compared to *PC-FedAvg* and *FedProx*.

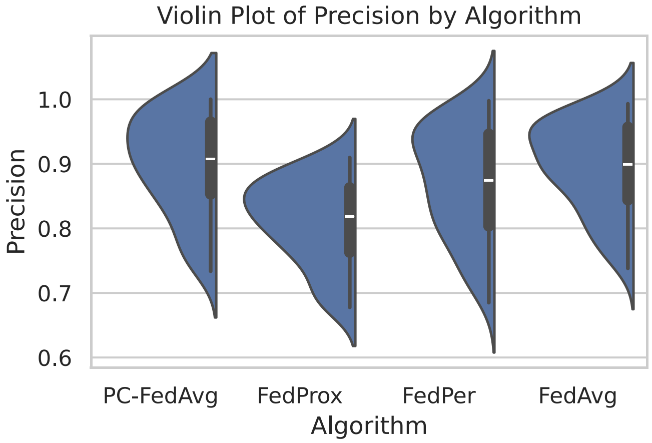


Fig. 14. Violin plot of precision by algorithm.

Fig. 14 shows that *PC-FedAvg* consistently achieves high precision values, with the distribution being tightly concentrated around 0.995. This indicates that *PC-FedAvg* is highly effective in minimizing false positives. *FedProx* also performs well in terms of precision, with values clustering around 0.985. The distribution for *FedProx* is slightly broader than *PC-FedAvg*, suggesting some variability but still maintaining high precision. *FedPer* and *FedAvg*, however, exhibit broader distributions with lower average precision values around 0.975 and 0.977, respectively. This indicates a higher rate of false positives compared to *PC-FedAvg* and *FedProx*.

Fig. 15 highlights the ability of each algorithm to correctly identify true positives. *PC-FedAvg* again shows superior performance with a high concentration of recall values around 0.995. This demonstrates its effectiveness in correctly identifying true positive cases. *FedProx* follows closely with recall values around 0.985, showing its strength in capturing true positives. *FedPer* and *FedAvg* show more variability and lower average recall

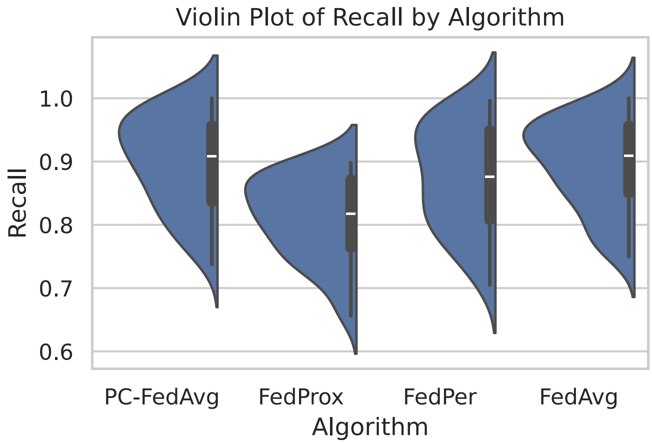


Fig. 15. Violin plot of recall by algorithm.

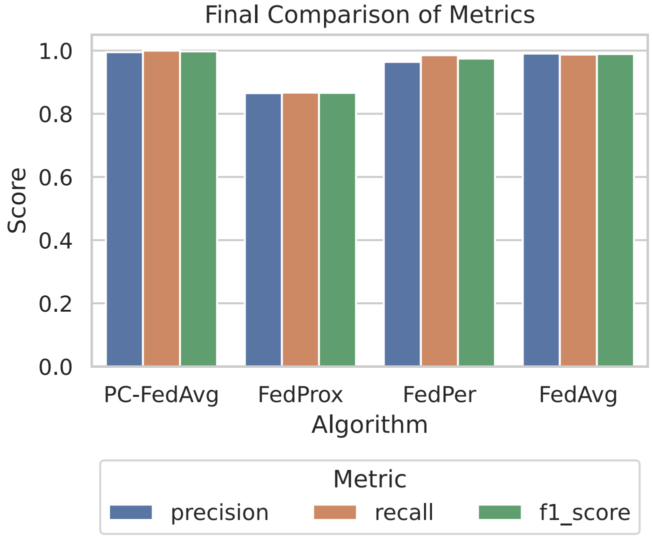


Fig. 16. Final comparison—precision, recall, and F1-score for each algorithm.

values around 0.975 and 0.977, respectively, indicating that they may miss more true positive cases compared to *PC-FedAvg* and *FedProx*.

The final comparison of precision, recall, and F1-score for each algorithm is presented in Fig. 16. This bar chart provides a comprehensive view of how each algorithm performs across all three metrics. *PC-FedAvg* consistently outperforms the other algorithms in all three metrics, with precision, recall, and F1-score values all clustered around 0.995. This highlights the algorithm's overall effectiveness and robustness. *FedProx* also shows strong performance, with values around 0.985 for all metrics, making it a reliable choice for FL tasks. *FedPer* and *FedAvg*, while performing well, show slightly lower and more variable metrics, indicating areas for improvement in balancing precision and recall.

The detailed metrics comparison highlights *PC-FedAvg* as the best performing algorithm across precision, recall, and F1-score. Its ability to consistently achieve high values in all three metrics underscores its robustness and effectiveness in the FL context. *FedProx* also performs well, maintaining high precision and recall with minimal variability. *FedPer* and *FedAvg*, while achieving respectable metrics, exhibit greater variability and lower average values, suggesting potential areas for further optimization. These findings are critical for understanding the strengths and weaknesses of each algorithm and for making informed decisions in the selection and optimization of FL algorithms for real-world applications.

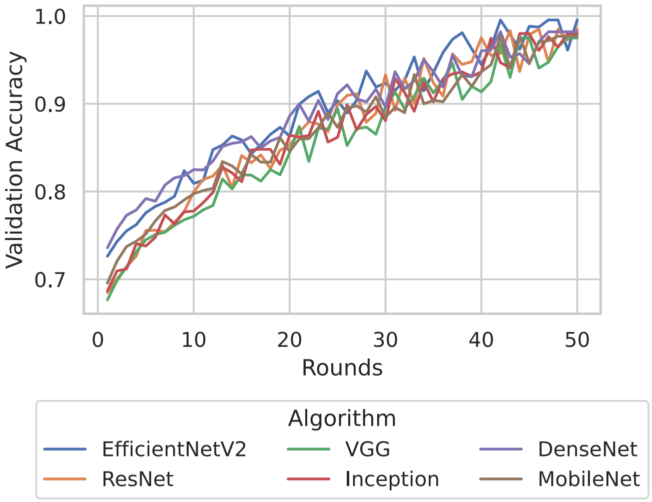


Fig. 17. Validation accuracy comparison of EfficientNetV2 with latest algorithms.

H. Comparison of EfficientNetV2 With State-of-the-Art Algorithms

The objective of this analysis is to compare the performance of EfficientNetV2 with other state-of-the-art algorithms. By examining validation accuracy over multiple rounds and final accuracy, we aim to highlight the strengths and weaknesses of EfficientNetV2 relative to other advanced models.

Fig. 17 shows the validation accuracy of EfficientNetV2 and other algorithms, including ResNet, VGG, Inception, DenseNet, and MobileNet, over multiple rounds of FL. EfficientNetV2 demonstrates superior performance, achieving higher validation accuracy consistently across all rounds. Starting with a high initial accuracy, EfficientNetV2 shows a steady improvement, reaching a final validation accuracy of approximately 0.9955. This consistent upward trend highlights the model's robustness and efficient handling of the FL process. In comparison, ResNet, VGG, Inception, DenseNet, and MobileNet also show improvements in validation accuracy over the rounds but do not match the performance level of EfficientNetV2. ResNet, for example, achieves a final accuracy of around 0.985, while DenseNet and Inception achieve approximately 0.982 and 0.980, respectively. VGG and MobileNet follow with final accuracies of about 0.975 and 0.977, respectively. The lower validation accuracies of these models indicate that while they are strong performers, EfficientNetV2's optimized architecture provides a significant advantage. Fig. 18 provides a clear comparison of the final accuracy for EfficientNetV2 and the other algorithms.

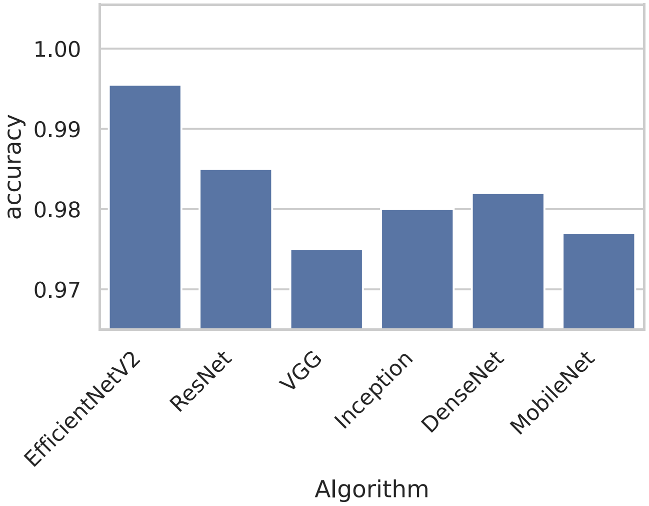


Fig. 18. Final accuracy comparison of EfficientNetV2 with latest algorithms.

EfficientNetV2's final accuracy of 0.9955 stands out prominently, surpassing the final accuracies of ResNet, VGG, Inception, DenseNet, and MobileNet. This comparison underscores the superior performance of EfficientNetV2 in achieving high accuracy in FL tasks.

Several factors contribute to EfficientNetV2's superior performance. The model employs a compound scaling method to balance network depth, width, and resolution, optimizing performance while maintaining efficiency. EfficientNetV2's use of depthwise separable convolutions and squeeze-and-excitation blocks further enhances its ability to capture complex patterns in the data, which is crucial in an FL environment where data can be highly heterogeneous and distributed. ResNet, while known for its deep residual learning capabilities, does not achieve the same level of performance as EfficientNetV2, likely due to its less optimized scaling and higher computational complexity. DenseNet, with its dense connectivity pattern, and Inception, with its multipath architecture, also show strong performance but fall short of EfficientNetV2's accuracy. VGG and MobileNet, despite their simpler architectures, provide respectable results but lack the advanced optimizations present in EfficientNetV2. One significant advantage of using EfficientNetV2 is its ability to maintain high accuracy with relatively lower computational requirements compared to some of the other state-of-the-art models. This efficiency makes EfficientNetV2 particularly suitable for FL scenarios, where computational resources at client devices can be limited.

I. Visual Analysis of Crack Detection

We provide a visual representation of the crack detection results achieved by the proposed method. This analysis is intended to highlight the effectiveness of our approach in identifying structural defects.Fig. 19 showcases a sample set of crack images, with detected cracks highlighted by yellow borders. The images are sorted based on their anomaly levels, providing a comprehensive view of the model's detection capabilities across varying degrees of structural damage. The visual results indicate that the proposed method is highly effective in detecting cracks, even in cases where the defects are subtle and not easily discernible. The model accurately identifies both prominent and minor cracks, demonstrating its robustness in handling diverse types of structural anomalies. For example, in images with fine hairline cracks, the model successfully highlights these defects, showing its sensitivity to minute structural irregularities.

Challenging cases, such as images with low contrast or complex backgrounds, are also effectively handled by the model. These scenarios often pose difficulties for traditional detection methods, but our approach leverages the advanced feature extraction capabilities of EfficientNetV2 to distinguish cracks from noise and background variations. This robustness is crucial for real-world applications where structural defects may not always be clearly visible. The visual analysis underscores the efficacy of our proposed method in crack detection, highlighting its potential for enhancing SHM systems by providing accurate and reliable defect identification.

VI. Discussion

In this section, we synthesize the key points of our work, critically discuss its technical accuracy and practical feasibility in real-world SHM scenarios, address limitations and potential drawbacks, and explore broader impacts and future directions.

A. Technical Accuracy and Practical Feasibility

Our proposed PC-FedAvg algorithm, integrated with EfficientNetV2 within an FL framework, has demonstrated high accuracy and robustness in detecting structural defects. The experimental results confirm that selective aggregation based on validation loss leads to a more reliable global model, while the use of EfficientNetV2 ensures superior feature extraction and classification performance.

B. Limitations and Drawbacks

Despite the promising results, several limitations must be acknowledged. Client dropout due to limited computational resources or network issues can lead to suboptimal model updates. Data heterogeneity across clients, particularly in non-IID settings, may also affect convergence and performance. In addition, communication disruptions—common in postdisaster environments—can further degrade the performance of the FL process. These factors highlight the need for robust strategies to manage unreliable client participation and variable data quality.

C. Security Vulnerabilities

FL, while preserving data privacy by keeping data local, is still susceptible to security threats, such as data poisoning and model inversion attacks. Adversaries could potentially manipulate local model updates or extract sensitive information from the aggregated model. Mitigation strategies, including secure aggregation protocols, differential privacy techniques, and robust anomaly detection mechanisms, should be considered to enhance the security of the system.

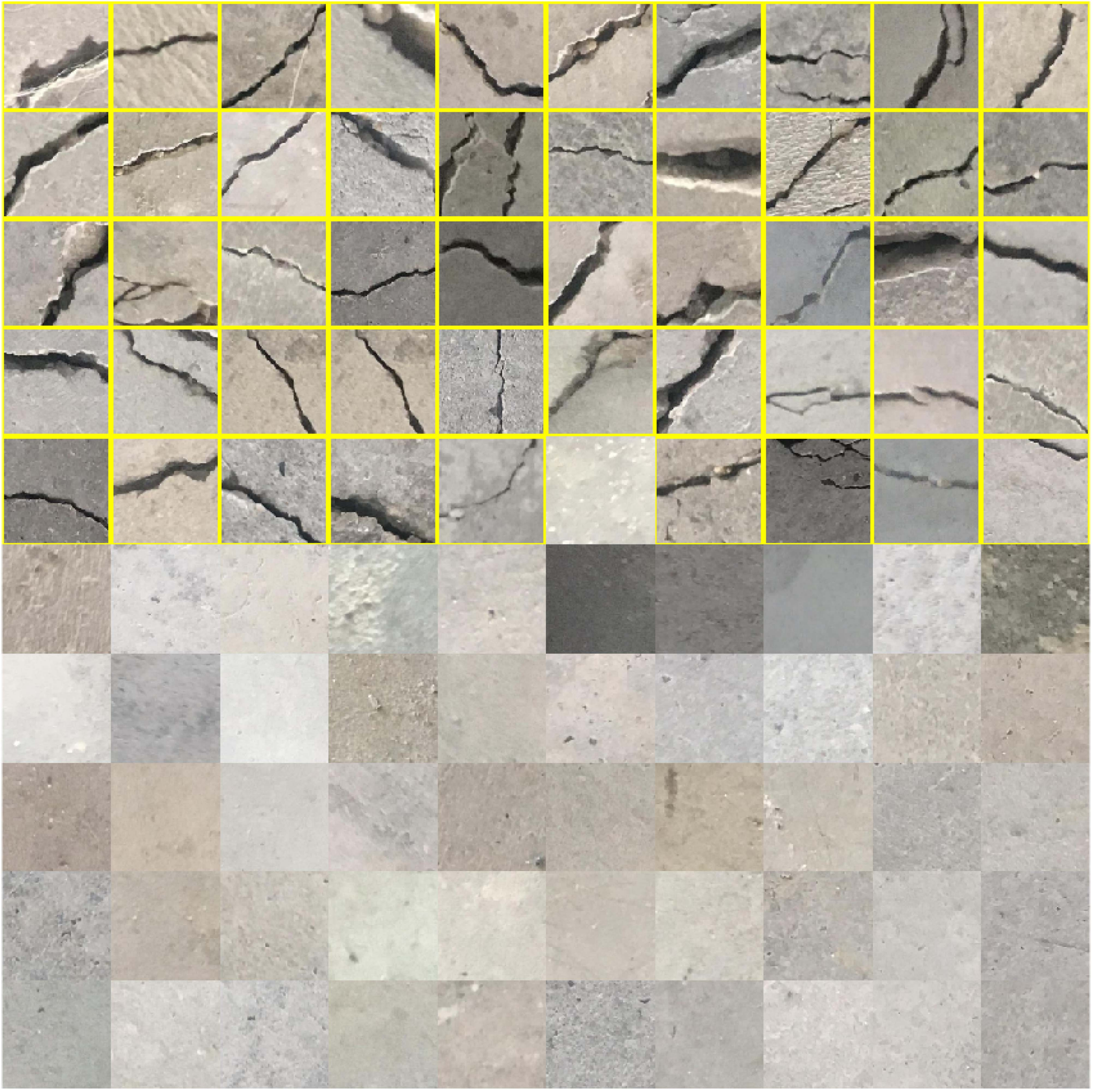


Fig. 19. Sample of crack images. The crack images are surrounded by yellow. The images are sorted based on their anomaly level.

D. Broader Impact and Generalizability

The proposed FL framework and the PC-FedAvg algorithm are not limited to SHM applications. Their scalability, adaptability, and decentralized nature make them applicable to various domains, such as healthcare, smart cities, and autonomous vehicles, where privacy and efficient data processing are critical. The inherent flexibility of the approach enables it to be extended to any scenario that requires collaborative model training on distributed, heterogeneous datasets.

E. Conceptual Basis for Integrating WSN with Large-Scale Knowledge Discovery

An interesting extension of our work involves integrating the current system with WSNs to support large-scale knowledge discovery. By combining data from static WSN nodes with mobile data acquired by drones and UGVs, the system could achieve a richer and more comprehensive understanding of the monitored environment. Prior studies (e.g., [26] and [27]) have demonstrated the effectiveness of WSNs in environmental monitoring, suggesting that a hybrid approach could further enhance decision-making in disaster management.

F. Alternative Optimization Methods

While our current approach utilizes SGD with momentum, this method may occasionally converge to local optima rather than a global optimum. Future work could explore alternative optimization methods, such as adaptive learning rate techniques (e.g., Adam and RMSprop) or second-order optimization methods, which may offer improved convergence properties and further enhance model performance.

VII. Conclusion

We present an FL framework for SHM that integrates spatial crowdsourcing drone services with the IoDT to enable real-time data acquisition in postdisaster contexts. The framework employs EfficientNetV2 for automated damage detection and introduces PC-FedAvg to mitigate the effects of non-IID client data and device heterogeneity. Experimental validation on the Concrete Crack Images for Classification dataset demonstrated a validation accuracy of 99.55%, substantially outperforming FedAvg, FedProx, and FedPer. By limiting aggregation to high-quality client updates and performing local model adaptation, our approach reduces computational and communication overhead compared with centralized methods. Deployment of this framework can improve the reliability, scalability, and privacy of SHM systems in challenging environments, thereby facilitating timely and accurate damage assessment and supporting efficient allocation of recovery resources.

Future work will explore adaptive optimization algorithms, such as Adam and RMSprop, to accelerate convergence and avoid local optima. We will investigate advanced security measures, including differential privacy and secure aggregation, to defend against adversarial manipulation and model inversion attacks. Integration with heterogeneous WSNs will be examined to expand coverage and improve data fusion. In addition, we plan to extend this FL framework to domains, such as healthcare imaging analysis and smart city infrastructure monitoring. These efforts aim to enhance the robustness, security, and versatility of distributed sensing applications.

References

[1] A. Akram, R. H. Jhaveri, M. Alazab, and H. Chi, “BC-IODT: Blockchain-based framework for authentication in Internet of Drone Things,” in *Proc. 5th Int. ACM Mobicom Workshop Drone Assist. Wireless Commun. 5 G Beyond*, 2022, pp. 115–120.

[2] A. Tahir, M. Adil, S. Ali, A. Z. Kouzani, and M. P. Mahmud, “Automatic target detection from satellite imagery using machine learning,” *Sensors*, vol. 22, no. 3, 2022, Art. no. 1147.

[3] S. Malik, S. Ansari, H. Rizvi, D. Kim, and R. Hasnain, “Intelligent target coverage in wireless sensor networks with adaptive sensors,” in *Proc. IEEE 92nd Veh. Technol. Conf.*, 2020, pp. 1–5.

[4] M. Umair, R. H. Jhaveri, M. N. Riaz, H. Chi, and S. Malebary, “Chained-drones: Blockchain-based privacy-preserving framework for secure and intelligent service provisioning in Internet of Drone Things,” *Comput. Elect. Eng.*, vol. 110, 2023, Art. no. 108772.

[5] X. Xue, X. Wu, C. Jiang, G. Mao, and H. Zhu, “Integrating sensor ontologies with global and local alignment extractions,” *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–10, 2021.

[6] A. Akram, J. Akram, A. Alabdultif, A. Anaissi, and R. H. Jhaveri, “Secure and interoperable IoMT-based smart homes,” *IEEE Consum. Electron. Mag.*, early access, Jan. 27, 2025, doi: 10.1109/MCE.2025.3534442.

[7] J. Akram and A. Anaissi, “Privacy-first crowdsourcing: Blockchain and local differential privacy in crowdsourced drone services,” in *Proc. IEEE Int. Conf. Web Serv.*, 2024, pp. 1412–1414.

[8] R. S. Rathore, R. H. Jhaveri, and A. Akram, “GALTrust: Generative adverserial learning-based framework for trust management in spatial crowdsourcing drone services,” *IEEE Trans. Consum. Electron.*, vol. 70, no. 3, pp. 6196–6207, Aug. 2024.

[9] W. Othman, A. Alabdulatif, and A. Akram, “DroneSSL: Self-supervised multimodal anomaly detection in Internet of Drone Things,” *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 4287–4298, Feb. 2024.

[10] J. Akram and A. Anaissi, “A blockchain-enhanced framework for privacy and data integrity in crowdsourced drone services,” in *Proc. Int. Conf. Service-Oriented Comput.*, 2025, pp. 323–330.

[11] H. S. Munawar, Z. Gharineiat, and S. I. Khan, “A framework for burnt area mapping and evacuation problem using aerial imagery analysis,” *Fire*, vol. 5, no. 4, pp. 122–127, 2022.

[12] S. I. Khan, F. Ullah, and B. J. Choi, “Drone-as-a-service (DaaS) for COVID-19 self-testing kits delivery in smart healthcare setups: A technological perspective,” *ICT Express*, vol. 9, no. 4, pp. 748–753, 2022.

[13] S. Sidana, D. Kumar, and R. H. Jhaveri, “Leveraging blockchain-as-a-certificate authority for authentication in 6G-enabled spatial crowdsourcing drone services,” in *Proc. IEEE 100th Veh. Technol. Conf.*, 2024, pp. 1–5.

[14] J. Akram, A. Anaissi, A. Akram, R. S. Rathore, and R. H. Jhaveri, “Adversarial label-flipping attack and defense for anomaly detection in spatial crowdsourcing UAV services,” *IEEE Trans. Consum. Electron.*, vol. 71, no. 1, pp. 2163–2174, Feb. 2025.

[15] F. Ullah, D. Shahzad, A. Heravi, and S. Qayyum, “Civil infrastructure damage and corrosion detection: An application of machine learning,” *Buildings*, vol. 12, no. 2, pp. 156–183, 2022.

[16] G. Martin, A. Hooper, T. J. Wright, and S. Selvakumaran, “Blind source separation for MT-InSAR analysis with structural health monitoring applications,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 7605–7618, 2022.

[17] Z. Sadeghi, T. Wright, A. Hooper, and S. Selvakumaran, “Using ray tracing to improve bridge monitoring with high-resolution SAR satellite imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 1155–1166, 2024.

[18] J. Akram and A. Anaissi, “Decentralized PKI framework for data integrity in spatial crowdsourcing drone services,” in *Proc. IEEE Int. Conf. Web Serv.*, 2024, pp. 643–653.

[19] H. Rizvi, “Handover management in 5 G software defined network based V2X communication,” in *Proc. 12th Int. Conf. Open Source Syst. Technol. (ICOSST)*, 2018, pp. 22–26.

[20] H. S. Munawar, A. Z. Kouzani, and M. P. Mahmud, “Using adaptive sensors for optimised target coverage in wireless sensor networks,” *Sensors*, vol. 22, no. 3, 2022, Art. no. 1083.

[21] J. Akram, A. Anaissi, R. S. Rathore, R. H. Jhaveri, and A. Akram, “Digital twin-driven trust management in open RAN-based spatial crowdsourcing drone services,” *IEEE Trans. Green Commun. Netw.*, vol. 8, no. 3, pp. 1061–1075, Sep. 2024.

[22] Z. Liu, A. Braytee, A. Anaissi, G. Zhang, L. Qin, and J. Akram, “Ensemble pretrained models for multimodal sentiment analysis using textual and video data fusion,” in *Proc. Companion Proc. ACM Web Conf. 2024*, 2024, pp. 1841–1848.

[23] E. Banzhaf, I. Bause, C. Helbig, and S. Elze, “Personal exposure to environmental pressures in different urban residential structures linking fieldwork and RS mapping,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 2789–2799, 2024.

[24] F. Algarni, M. A. Khan, W. Alawad, and N. B. Halima, “P3S: Pertinent privacy-preserving scheme for remotely sensed environmental data in smart cities,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5905–5918, 2023.

[25] Y. Li, M. Wang, K. Hwang, Z. Li, and T. Ji, “LEO satellite constellation for global-scale remote sensing with on-orbit cloud AI computing,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 9369–9381, 2023.

[26] N. L. D. Khoa, A. Anaissi, and Y. Wang, “Smart infrastructure maintenance using incremental tensor analysis,” in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 959–967.

[27] A. Anaissi, M. Goyal, D. R. Catchpoole, A. Braytee, and P. J. Kennedy, “Ensemble feature learning of genomic data using support vector machine,” *PLoS one*, vol. 11, no. 6, 2016, Art. no. e0157330.

[28] A. Binbusayyis and T. Vaiyapuri, “Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM,” *Appl. Intell.*, pp. 1–15, 2021.

[29] A. Tahir, A. Akram, A. Z. Kouzani, and M. P. Mahmud, “Cloud-and fog-integrated smart grid model for efficient resource utilisation,” *Sensors*, vol. 21, no. 23, 2021, Art. no. 7846.

[30] P. Kairouz et al., “Advances and open problems in federated learning,” *Found. Trends Mach. Learn.*, vol. 14, no. 1-2, pp. 1–210, 2021.

[31] J. Akram, W. Hussain, R. H. Jhaveri, R. S. Rathore, and A. Anaissi, “Dynamic GNN-based multimodal anomaly detection for spatial crowdsourcing drone services,” *Digit. Commun. Netw.*, 2025.

[32] C. T. Dinh, N. Tran, and T. D. Nguyen, “Personalized federated learning with Moreau envelopes,” in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21394–21405.

[33] P. Tam, S. Math, C. Nam, and S. Kim, "Adaptive resource optimized edge federated learning in real-time image sensing classifications," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10929–10940, 2021.

[34] Y. Wu, Y. Ji, S. Lee, and A. Braytee, "Simplified swarm learning framework for robust and scalable diagnostic services in cancer histopathology," in *Proc. Int. Conf. Comput. Sci.*, Jul. 2025.

[35] W. Xie, X. Xu, and Y. Li, "Decentralized federated GAN for hyperspectral change detection in edge computing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 8863–8874, 2024.

[36] J. Zhu, J. Wu, A. K. Bashir, Q. Pan, and W. Yang, "Privacy-preserving federated learning of remote sensing image classification with dishonest majority," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 4685–4698, 2023.

[37] J. Akram and A. Anaissi, "DDRM: Distributed drone reputation management for trust and reliability in crowdsourced drone services," in *Proc. IEEE Int. Conf. Web Serv.*, 2024, pp. 921–931.

[38] M. Aamir, R. Raut, and A. Akram, "Ai-generated content-as-a-service in IoMT-based smart homes: Personalizing patient care with human digital twins," *IEEE Trans. Consum. Electron.*, vol. 71, no. 1, pp. 1352–1362, Feb. 2025.

[39] A. Hard et al., "Federated learning for mobile keyboard prediction," 2018, *arXiv:1811.03604*.

[40] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[41] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, vol. 2, pp. 429–450, 2020.

[42] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated learning with personalization layers," 2019, *arXiv:1912.00818*.

[43] V. Smith, C.-K. Chiang, M. Sanjabi, and A. Talwalkar, "Federated multi-task learning," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[44] F. Hanzely and P. Richtárik, "Federated learning of a mixture of global and local models," 2020, *arXiv:2002.05516*.

[45] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," 2020, *arXiv:2003.13461*.

[46] Z. Zhong et al., "Random erasing data augmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, pp. 13001–13008, 2020.