future internet

Article

Federated Adversarial Training Strategies for Achieving Privacy and Security in Sustainable Smart City Applications

Sapdo Utomo [1,2,†], Adarsh Rouniyar [3], Hsiu-Chun Hsu [4] and Pao-Ann Hsiung [3,*,†]

[1] Graduate Institute of Ambient Intelligence and Smart Systems, National Chung Cheng University, Chiayi 621301, Taiwan; sapdo.utomo@brin.go.id
[2] Research Center for Smart Mechatronics, National Research and Innovation Agency (BRIN), Bandung 40135, Indonesia
[3] Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621301, Taiwan; adarsh110m@cs.ccu.edu.tw
[4] Department of Information Management, National Chung Cheng University, Chiayi 621301, Taiwan; admhchsu@ccu.edu.tw
* Correspondence: pahsiung@ccu.edu.tw
† These authors contributed equally to this work.

Abstract: Smart city applications that request sensitive user information necessitate a comprehensive data privacy solution. Federated learning (FL), also known as privacy by design, is a new paradigm in machine learning (ML). However, FL models are susceptible to adversarial attacks, similar to other AI models. In this paper, we propose federated adversarial training (FAT) strategies to generate robust global models that are resistant to adversarial attacks. We apply two adversarial attack methods, projected gradient descent (PGD) and the fast gradient sign method (FGSM), to our air pollution dataset to generate adversarial samples. We then evaluate the effectiveness of our FAT strategies in defending against these attacks. Our experiments show that FGSM-based adversarial attacks have a negligible impact on the accuracy of global models, while PGD-based attacks are more effective. However, we also show that our FAT strategies can make global models robust enough to withstand even PGD-based attacks. For example, the accuracy of our FAT-PGD and FL-mixed-PGD models is 81.13% and 82.60%, respectively, compared to 91.34% for the baseline FL model. This represents a reduction in accuracy of 10%, but this could be potentially mitigated by using a more complex and larger model. Our results demonstrate that FAT can enhance the security and privacy of sustainable smart city applications. We also show that it is possible to train robust global models from modest datasets per client, which challenges the conventional wisdom that adversarial training requires massive datasets.

Keywords: sustainable smart cities; federated learning; adversarial attack; privacy protection; robust model

Citation: Utomo, S.; Rouniyar, A.; Hsu, H.-C.; Hsiung, P.-A. Federated Adversarial Training Strategies for Achieving Privacy and Security in Sustainable Smart City Applications. *Future Internet* **2023**, *15*, 371. https://doi.org/10.3390/fi15110371

Academic Editor: Christos Kalloniatis

Received: 15 October 2023
Revised: 13 November 2023
Accepted: 15 November 2023
Published: 20 November 2023

1. Introduction

The United Nations (UN) has advocated for the Sustainable Development Goals (SDGs), which have since become an indispensable cornerstone for research frameworks within the academic and scientific communities. The Sustainable Development Goals (SDGs) are an all-encompassing collection of seventeen goals, of which SDG 11 addresses the critical challenge of promoting sustainable cities and communities worldwide. Conversely, contemporary trends underscore the increasing demand for secure systems and user privacy. This imperative did not emerge spontaneously; it is rooted in a multitude of documented cases where users suffered adverse consequences due to insecure systems and data breaches. These users encompass a wide spectrum, including corporations, governmental entities, and private citizens [1,2].

The authors of [2] have documented that between 2018 and 2019, the United States experienced an alarming number of data breaches exceeding 10,000 cases. Notably, among

these, 430 incidents were categorized as major data breaches [3]. In the European Union, since May 2018, over 160,000 cases have been officially reported. One particularly noteworthy instance revolves around the activities of Cambridge Analytica [4], which exploited fifty million Facebook profiles associated with American voters. Their objective was to develop an AI system capable of influencing electoral outcomes among targeted demographic groups.

The Equifax data breach in 2017 was a significant incident that affected nearly half of the United States' population. This breach compromised over 143 million data records [5,6]. Similarly, Orvibo, a company specializing in Internet of Things (IoT) management platforms, exposed over 2 billion records of private information, including camera recordings of conversations, on the Internet, without any password or security measures [7]. These incidents, among others, highlight the need for robust security solutions to protect user privacy.

Smart city applications often require access to sensitive user information, making it imperative to safeguard this data. Federated learning (FL), also known as "privacy by design", is a promising paradigm that preserves user privacy by keeping personal data on local devices, also known as federated clients. Implementing FL can be seen as adding additional security layers to a system.

Previous studies have identified the limitations of machine learning in terms of preserving data privacy [8]. Adversaries can potentially derive training data from the gradients of machine learning models. Deep leakage from gradients (DLG) [9] and similar attacks [10,11] can recover information pixel by pixel for images and token by token for text. However, these attacks still face challenges in consistently converging and uncovering true labels.

Alternative research has proposed a method for reconstructing images from the parameter gradients of models by inverting gradients [12] using adversarial attacks. One limitation of this approach is its high computational cost—reconstructing a single image requires 24,000 iterations. Shen et al. [13] conducted a comprehensive study of adversarial threats to distributed machine learning and federated learning. Their findings suggest that FL has several advantages over distributed machine learning. The lack of communication requirements with other clients in FL significantly reduces the risk of privacy breaches. Additionally, distributed machine learning requires more collaboration with other clients or system nodes than FL.

It is worth noting that experts in the field have hypothesized that FL systems may offer enhanced security compared to centralized systems. The insights gained from our previous research, as cited in reference [14], establish the foundation for the subsequent analysis presented in this paper. This analysis primarily focuses on identifying potential vulnerabilities and areas where data could be compromised in the event of a security breach. The scope of our investigation includes a comparative assessment of attack probabilities and the extent of potential losses between centralized systems and the FL paradigm.

Figure 1 illustrates a generalized cloud-based architecture for smart cities, along with potential attack scenarios. This architecture typically collects data from a variety of IoT devices, such as wearable devices, smart sensors, and data loggers, which are used to support applications in areas such as intelligent monitoring, intelligent transportation, and intelligent healthcare. The collected data is then transmitted to edge computing servers, which significantly reduces data transport latency to and from centralized cloud data centers.

Edge computing plays a crucial role in facilitating real-time data processing and decision-making, a key requirement for applications demanding immediate responses, such as autonomous vehicles, robotics, and monitoring systems [15–20]. By employing edge processing and filtering, only relevant information is transmitted to the cloud, thereby reducing network traffic [20,21]. This optimization is particularly beneficial for applications with limited bandwidth.

Furthermore, edge computing enhances data security and privacy by limiting data exposure to external cloud services and keeping sensitive data within a local network. This approach aligns with best practices to safeguard data privacy [16,17,21]. The concept of edge AI has gained prominence with the advent of edge AI devices, such as the NVIDIA Jetson Xavier AGX (NVIDIA, Beijing, China), NVIDIA Jetson TX2 (NVIDIA, China), NVIDIA

Jetson Xavier NX (NVIDIA, China), Google Coral Dev Board (Google Coral, Taiwan, China), and Intel NUC (Intel, Beijing, China). These devices are equipped with graphical processing units (GPUs) and substantial computational capabilities, enabling advanced AI applications at the network's edge [22–26].

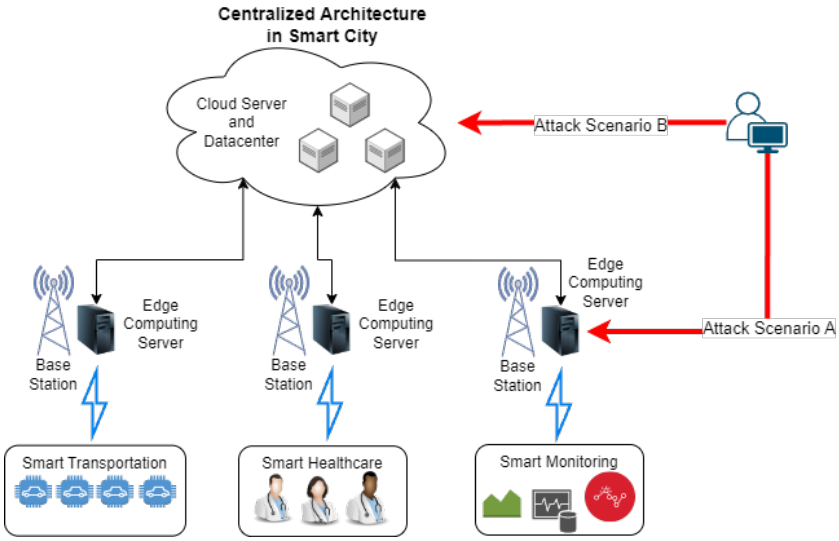

Figure 1. Attack scenarios in cloud-based architecture.

Two potential attack scenarios that may occur in cloud architecture are illustrated in Figure 1. Attack scenario A targets the edge server, which plays a critical role as a data storage and processor for a wide variety of IoT devices. This data often includes confidential and sensitive information [16,17,21]. If an attack is successful, unauthorized actors could gain access to the confidential information of numerous clients, which could have significant financial consequences.

Attack scenario B focuses on compromising the cloud server, which stores a significantly larger amount of data than the edge server. The cloud server acts as a centralized repository for all data collected from interconnected edge servers. The Orvibo incident [7] serves as a prime example, demonstrating how an attacker was able to access over two billion records from a cloud server with relative ease.

In Section 3.5, a thorough explanation of FL will be presented. This section will conduct a thorough examination of potential attack scenarios in order to evaluate the data leakage risk associated with the FL system. FL computational tasks are primarily performed on nodes called federated clients (FCs). This approach is designed to prioritize user privacy and minimize the amount of data shared between devices. Consequently, FL has the potential to significantly improve the security and privacy of the overall system [8,13,27–29]. To facilitate understanding of how FL reinforces security and privacy protocols, we refer to Figure 2. Figure 2 illustrates two distinct attack scenarios:

- Scenario A: An adversary attempts to compromise an FC. FCs can be a wide variety of devices, including personal computers, IoT devices, smartphones, and minicomputers (e.g., Raspberry Pi (Raspberry Pi, Wales, UK) and NVIDIA Jetson Nano (NVIDIA, China)). If the attack is successful, only the affected client's data will be compromised; the data of other clients will remain unaffected [13,14]. This is a significant advantage over cloud-based architectures, which are vulnerable to large-scale data breaches.
- Scenario B: An adversary attempts to gain unauthorized access to the federated server (FS). However, since FL data are primarily stored on FCs, a successful attack on the FS would not expose any FL data. Therefore, FL can significantly enhance the security and privacy of conventional IoT systems. By adopting FL, organizations can improve data protection and reduce the risk of data intrusions similar to the one that occurred at Orvibo [7].

Although federated learning (FL) can improve security and privacy compared to centralized systems, it is still vulnerable to adversarial attacks, as shown in previous research [13,27,29–36]. If this vulnerability is not properly addressed, it could lead to data leakage [8,13,28]. However, the aggregation procedure used in FL makes it difficult to extract private data from gradients. Therefore, the most common attack method in FL is to join as a client and send malicious updates to the server, which can lead to incorrect model decisions and reduced accuracy [27,30,33–36]. If an AI model used in a sensitive system, such as loan evaluation or healthcare applications, is successfully attacked, the consequences for users could be severe.

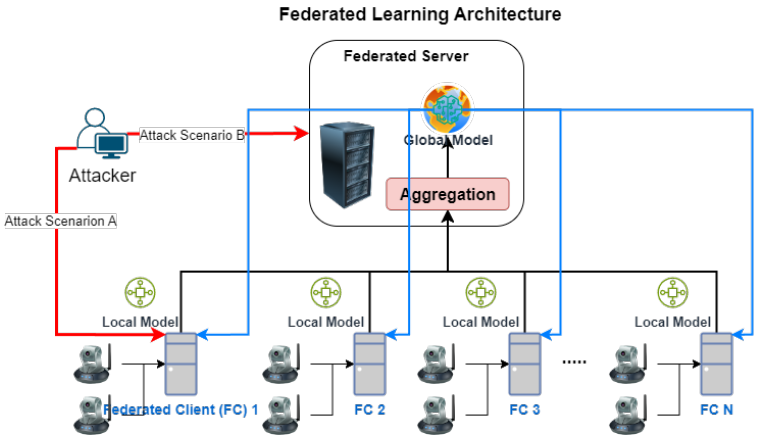

Figure 2. Attack scenarios in FL-based architecture.

To address the aforementioned challenges, we propose federated adversarial training (FAT) strategies to enhance privacy and security in sustainable smart city applications. FAT is a mechanism that has been shown to improve the robustness of FL models [30,34,36] despite the inherent complexity of this task. Our experimental results demonstrate promising outcomes, suggesting that FAT is a promising approach for practical use.

In this paper, we select an air pollution monitoring system as a case study due to its significant societal impact and its ability to exemplify the effectiveness of our proposed solution in improving privacy and security within sustainable smart city applications. The structure of our contributions in this paper is as follows:

- We conducted a thorough examination of a wide range of adversarial attack scenarios targeting federated learning.
- Our findings have been experimentally validated, leading to the presentation of an adversarial learning strategy aimed at enhancing the overall robustness of the global model.
- We have successfully demonstrated the practicality of implementing our technique in situations with limited data capacity on the client side. It is important to note that prior research has recognized the challenges related to integrating adversarial training in a federated context, primarily due to the data-intensive nature of this training process.
- Our study provides valuable insights that can serve as a foundation for the development of additional defense scenarios and countermeasures against adversarial attacks, ultimately contributing to the reinforcement of security measures for sustainable smart city applications.

The paper is structured as follows: Section 2 provides an extensive discussion of related works concerning security and privacy issues in smart city applications, along with a review of prior approaches to addressing these challenges. Section 3 offers a comprehensive account of the materials and methods employed in our proposed solution. In Section 4, we present detailed experimental results and engage in a thorough discussion of these findings. Finally, Section 5 encompasses the conclusion of this paper, summarizing the key takeaways and implications.

2. Related Work

In previous studies, diverse methods have been investigated to address security and privacy concerns within the realm of smart cities and the Internet of Things (IoT). These methods encompass the deployment of deep-learning-based intrusion detection systems, blockchain technology, and anomaly detection techniques, among others. The overarching objective of these approaches is to comprehensively fortify overall security measures. Additionally, several novel computational frameworks and architectures have been proposed to augment privacy protection. The emergence of edge GPU devices, equipped with substantial computational capabilities facilitated by GPUs, has given prominence to the concept of edge AI. Consequently, the notion of training AI models at the edge has garnered significance and viability.

In the realm of smart city applications, Singh et al. [37] introduced DeepBlockScheme, a solution that combines deep learning with blockchain technology to enhance security. To illustrate its practicality, the authors conducted a case study within the domain of vehicle manufacturing. Another study by Kumar et al. [38] employed blockchain and machine learning to enhance privacy and security in IoT-driven smart cities. In this context, a blockchain module ensures secure IoT data transmission. Principal component analysis (PCA) is applied to process raw IoT data into a new representation, and an intrusion detection system, the gradient boosting anomaly detector (GBAD), flags unauthorized access attempts. Despite the data integrity benefits offered by blockchain, its implementation introduces computational complexity, particularly in resource-constrained environments [27]. It is noteworthy that in contemporary blockchain systems like Ethereum, data stored on blockchains is commonly publicly accessible by default, posing potential privacy concerns for individuals [33].

Prior studies have introduced various frameworks and architectures aimed at supporting IoT technologies in the context of smart cities. In [39], the authors leverage blockchain and software-defined networking to ensure secure data communication, privacy, and reduced computational costs. They incorporate deep learning within the cloud architecture, enhancing smart industry production with valuable analysis. Another approach is presented in [20], where the authors utilize mobile edge computing (MEC) and the Stackelberg principle-based game theory implemented through the alternating direction method of multipliers (ADMM) within an IoT environment. This method efficiently schedules tasks for distributed devices, resulting in rapid algorithm convergence.

Addressing challenges in multimedia data management, [16] introduces the Short Supply Circuit Internet of Things (SSCIoT). This research primarily focuses on tackling data processing challenges when cloud access is limited or unavailable. The architecture emphasizes the utilization of mobile edge computing and fog computing over cloud computing to deliver quicker responses to users.

Numerous researchers have explored the integration of federated learning (FL) within the IoT and smart city domains, deviating from conventional approaches involving data transfer to multiple points. The foundational principle of FL revolves around maintaining data on the client side, as discussed earlier, aiming to fortify system security and privacy. In our prior research [40], we introduced a federated trustworthy AI (FTAI) architecture tailored to meet seven key requirements of trustworthy AI (TAI) outlined by the European Union [41]. Specifically, in this paper, the proposed method delves into TAI requirements two and three, focusing on robustness, safety, privacy, and data governance.

In a comprehensive survey [42], researchers explored FL's implementation for IoT, covering aspects such as data exchange, offloading, caching, attack detection, location services, mobile crowdsensing, privacy, and security. The study extensively discussed FL's applications across various domains, including smart healthcare, transportation, UAVs, cities, and industries. Challenges associated with FL-IoT implementations, such as adversarial attacks, non-IID data, low convergence rates, and heterogeneous client computational resources, were thoroughly outlined. In a distinct study [43], researchers integrated FL and blockchain technology to enhance privacy and scalability within smart healthcare systems.

It is noteworthy that this conceptual architecture lacks empirical evidence regarding its implementation results.

Adversarial attacks on IoT systems and their ramifications have been subject to extensive investigation in prior research. In [44], researchers delved into the impact of both nontargeted and targeted adversarial attacks, employing methods such as the fast gradient signs method (FGSM) and projected gradient descent (PGD) on CNN-based IoT device identification. The study revealed a degradation in identification accuracy with increasing perturbation and iteration step size. In [45], the authors scrutinized adversarial attacks on deep-learning-based intrusion detection within IoT networks, comparing the effectiveness of two deep learning techniques: feedforward neural networks (FNNs) and self-normalizing neural networks (SNNs). Results indicated that SNNs exhibited greater resistance to adversarial samples in IoT datasets.

Moreover, [46] introduced a partial-model adversarial attack on IoT systems using machine learning (ML). Their findings underscored the vulnerability of ML-based IoT systems to adversarial attacks, particularly when an adversary targeted only 8 out of 20 IoT devices, achieving an 83% success rate. In [47], researchers assessed the efficacy of adversarial attacks on ML-based detection of denial-of-service (DoS) attacks in IoT smart home networks. The introduction of adversarial samples into the test data resulted in a 47.2% decrease in model accuracy, with subsequent improvements observed after employing adversarial training.

As previously highlighted, adversarial attacks extend their impact to both ML and deep learning (DL) models, with federated learning (FL) not immune to such vulnerabilities [13,27,29–36]. In an endeavor to fortify defenses against poisoning attacks, [48] introduced a method utilizing generative adversarial networks (GANs) to generate auditing data during the training process. This approach incorporates a mechanism to identify and eliminate adversaries by auditing the accuracy of their models and integrating a GAN into the federated learning server. The GAN's role involves constructing an auditing dataset capable of detecting adversaries by evaluating the correctness of participant models.

In the pursuit of robust federated learning, [29] introduces the robust framework for federated learning (RFFL). This framework is crafted to adeptly identify and eliminate malicious behaviors in a cautious and iterative manner before aggregating model updates. By adopting this approach, RFFL establishes a robust learning framework, effectively mitigating the risks associated with data poisoning and model update poisoning attacks. This methodology shares commonalities with the work presented in [48], where a similar notion of implementing filtering on the federated server before aggregation is employed. However, both approaches raise concerns regarding the overall robustness of the global model against adversarial attacks, as they do not incorporate adversarial samples into their model training, thereby questioning the ability to create a truly robust model.

Addressing adversarial attacks in federated learning poses a significant challenge, with [30,34,36] highlighting the complexities involved. Notably, [30] acknowledges the inherent difficulties of implementing adversarial training in a federated setting, primarily attributed to the data-intensive requirements of such training. Previous research endeavors have been dedicated to identifying effective approaches to overcome these challenges.

The integration of FL and IoT has transitioned from a conceptual possibility to a practical reality, facilitated by the emergence of mini-computers equipped with GPUs, commonly known as "edge GPUs". These devices enable the processing of multimedia data at the edge, giving rise to the concept of "edge AI" [22–26]. This evolution has spurred increased research initiatives exploring the application of AI at the edge. For instance, in a study conducted by [22], researchers utilized the NVIDIA Jetson Nano to develop a model for detecting vehicles and pedestrians on rural roads. Through extensive testing of well-established models, including MobileNetv1, MobileNetv2, Inceptionv2, Pednet, and Multiped, they demonstrated the effective functionality of these models on the NVIDIA Jetson Nano.

Another exploration by Mathur et al. [24] delved into the feasibility of on-device FL. In this investigation, Android smartphones and the Nvidia Jetson TX2 played a pivotal role. These devices were leveraged to train a model following the federated learning methodology through the utilization of the Flower framework. The training process involved the CIFAR-10 dataset for the Nvidia Jetson TX2 and the Office-31 dataset for Android clients. The findings from this study provide compelling evidence for the viability of training AI models on edge devices, especially with lightweight models such as ResNet18. Moreover, Truong et al. [26] harnessed the power of FL to train a lightweight anomaly detection model tailored for industrial control systems. By embracing FL, the learning process was significantly expedited, with completion times reduced to a matter of minutes. The research involved the use of four NVIDIA Jetson Nanos as federated clients, demonstrating the potential for efficient AI training on edge devices.

In order to address the aforementioned security and privacy issues associated with sustainable smart city applications, adversarial training strategies have been implemented within the framework of federated learning. These methodologies have been specifically developed to overcome the challenges that arise from clients having limited datasets, with the ultimate goal of developing a robust global model. This approach exhibits potential for improving the security and privacy aspects of smart city applications and is viable when federated clients are outfitted with GPU support.

3. Materials and Methods

3.1. Overall Research Workflow

The research workflow, illustrated in Figure 3, was systematically organized to enhance repeatability. The sequential steps are detailed as follows:

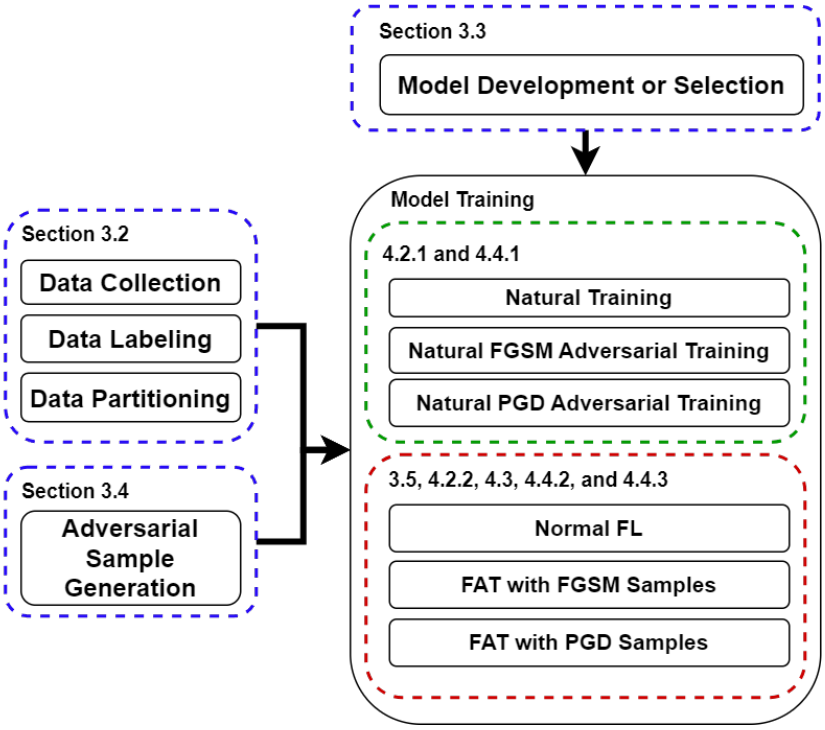

Figure 3. The overall research workflow in this work.

1. **Data Collection:**
   - The dataset utilized in this study was accessible for download from our Kaggle repository [49]. Alternatively, researchers had the flexibility to use their own dataset tailored to specific applications, as the proposed method was application-agnostic.
   - A comprehensive explanation of the dataset processing was available in Section 3.2. The set of images utilized in this research is illustrated in Figure 4.

2. **Data Labeling:**
   - Immediate labeling of the dataset occurred right after capturing the images or videos to ensure accuracy.
   - Given that the dataset's primary purpose was air quality measurement, label values were extracted from official website data, providing air quality parameters at the time of data capture.
   - For alternative datasets, meticulous attention to correct labeling was imperative.
3. **Data Partitioning:**
   - Recognizing that federated client computational power is typically less powerful than that of the central server, handling only a limited amount of data is feasible.
   - To showcase the proposed method, the dataset was partitioned into smaller subsets, each containing 1224 images. These subsets were then distributed across all clients and servers for experimentation.
   - If federated clients lacked GPU support, it was advisable to reduce the number of subsets to prevent extended processing times, ensuring a more efficient execution of the entire process.
4. **Model Development or Selection:**
   - In this step, we employed a model derived from our previous research [50], adjusting it to suit the current study's goals. The model architecture in this study is presented in Figure 5.
   - Researchers could choose any model, whether self-developed or established (e.g., MobileNet, ResNet, VGG16), as long as it met their study's objectives and enhanced the global model's robustness.
   - The detailed architecture of our chosen model is provided in Section 3.3.
5. **Adversarial Sample Generation:**
   - For adversarial training to be conducted, adversarial samples were required.
   - In order to generate adversarial samples, the original dataset was utilized. A variety of attack techniques were capable of producing adversarial samples.
   - In this research, we specifically employed the FGSM and PGD methods, each meticulously explained in Section 3.4.
   - The data label for the adversarial samples was the same as the original dataset used.
   - It was recommended to perform adversarial sample generation after data partitioning for ease of data verification, considering the time-intensive nature of the process.
6. **Natural Training:**
   - Natural (conventional) model training was performed using a dataset comprising normal data devoid of adversarial samples.
   - The objective was to obtain key metrics, particularly model accuracy, serving as a baseline for comparison with subsequent experimental results.
   - Further details on this training step were elaborated in Section 4.2.1, with hyperparameters specified in Table 1.
7. **Natural FGSM Adversarial Training:**
   - This phase mirrored natural training, with the distinction that the dataset incorporated both normal and FGSM adversarial samples. The dataset composition was detailed in Section 4.4.1.
8. **Natural PGD Adversarial Training:**
   - Similar to natural training, this step involved utilizing a dataset comprising normal data alongside PGD adversarial samples (see Section 4.4.1 for dataset details).

9. **Normal FL:**
    - This step employed one federated server and four federated clients (with the option to add more based on available resources in replication) and focused on observing global model performance under non-attack conditions.
    - The primary objective was to examine the process under ideal circumstances, establishing a baseline for global model accuracy. The FL architecture is depicted in Figure 6, and a comprehensive explanation of FL is provided in Section 3.5. FedAvg (Algorithm 1) served as the aggregation method, and the FL hyperparameters are detailed in Table 2 (refer to Section 4.2.2).
10. **FAT with FGSM Samples:**
    - This step encompassed various experimental scenarios, exploring distinct client settings, dataset distributions, and the number of attacks. Scenario names included fl-fgsm-25, fl-fgsm-50, fl-fgsm-75, fl-mixed-fgsm, and fat-fgsm (refer to Table 3 for clarity).
    - FGSM adversarial samples were employed as a dataset for conducting attacks. The attacking process is illustrated in Figure 7 (Section 4.3). A comprehensive presentation of results and discussions for this step is available in Section 4.4.2.
11. **FAT with PGD Samples:**
    - Similar to the previous step, this step utilized PGD adversarial samples for attack scenarios. The scenario names encompassed fl-pgd-25, fl-pgd-50, fl-pgd-75, fl-mixed-pgd, and fat-pgd (refer to Table 3 for clarity).
    - The comprehensive results and discussions pertaining to this step are provided in Section 4.4.3.

Figure 4. Image samples in this study.

Table 1. The parameter settings in natural model training.

Hyperparameter	Value
Batch Size	32
Maximum Epochs	100
Early Stopping Patience	7
Early Stopping Monitor	Validation Loss
Model Checkpoint Monitor	Validation Loss
Model Checkpoint Trigger	Minimum
Save Best Weight Only	True
Loss Function	Categorical Crossentropy
Optimizer	ADAM
Learning Rate	0.0001 (1×10^{-4})

Table 2. The parameter settings in normal FL and FAT.

Hyperparameter	Value
Communication Round	30
Aggregation	FedAvg
Client's Local Epoch	35 if server round < 6 else 15
Client's Batch Size	16
Client's Early Stopping Patience	7
Client's Early Stopping Monitor	Validation Loss
Client's Model Checkpoint Monitor	Validation Loss
Client's Model Checkpoint Trigger	Minimum
Client's Save Best Weight Only	True
Client's Loss Function	Categorical Crossentropy
Client's Optimizer	ADAM
Client's Learning Rate	0.0001 (1×10^{-4})
Number of Clients	4
Client Selection	All

Table 3. List of federated training scenarios in our experiments.

Scenario Name	Datase Used in Clients	Purpose of Experiments
fl-normal	All clients used a normal dataset	To obtain baseline accuracy
fl-fgsm-25	1 client used full FGSM adversarial samples dataset; 3 clients used a normal dataset	To observe the attack's effect on the global model
fl-fgsm-50	2 clients used full FGSM adversarial samples dataset; 2 clients used a normal dataset	To observe the attack's effect on the global model
fl-fgsm-75	3 clients used full FGSM adversarial samples dataset; 1 client used a normal dataset	To observe the attack's effect on the global model
fl-mixed-fgsm	1 client used full FGSM adversarial samples dataset; 1 client used a normal dataset; 2 clients used mixed 50% normal dataset and 50% FGSM adversarial samples dataset	To observe the attack's effect on the global model as well as the feasibility of this scenario for defense mechanisms
fat-fgsm	All clients used mixed 50% normal dataset and 50% FGSM adversarial samples dataset	Defense mechanisms to train a robust global model
fl-pgd-25	1 client used full PGD adversarial samples dataset; 3 clients used a normal dataset	To observe the attack's effect on the global model
fl-pgd-50	2 clients used full PGD adversarial samples dataset; 2 clients used a normal dataset	To observe the attack's effect on the global model
fl-pgd-75	3 clients used full PGD adversarial samples dataset; 1 client used a normal dataset	To observe the attack's effect on the global model
fl-mixed-pgd	1 client used full PGD adversarial samples dataset; 1 client used a normal dataset; 2 clients used mixed 50% normal dataset and 50% PGD adversarial samples dataset	To observe the attack's effect on the global model as well as the feasibility of this scenario for defense mechanisms
fat-pgd	All clients used mixed 50% normal dataset and 50% PGD adversarial samples dataset	Defense mechanisms to train a robust global model

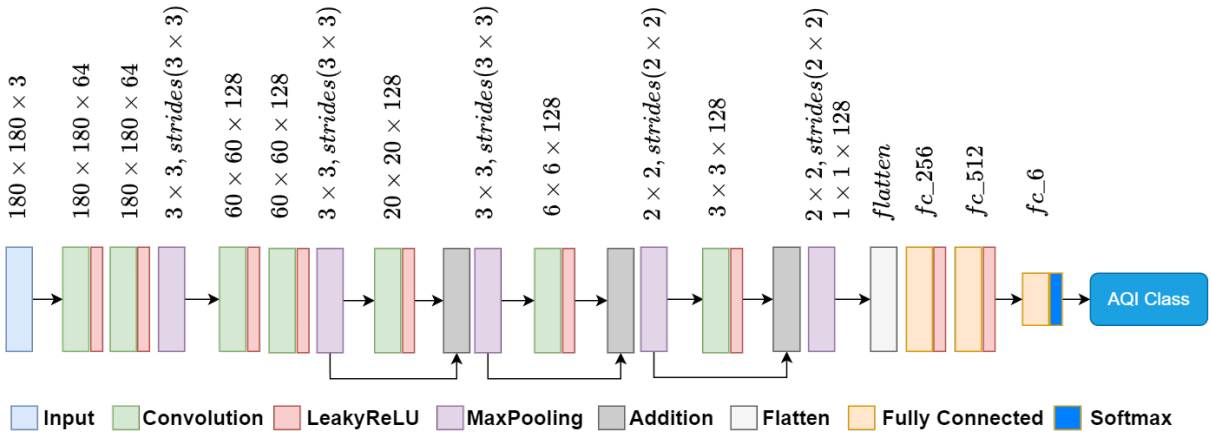

Figure 5. Model architecture.

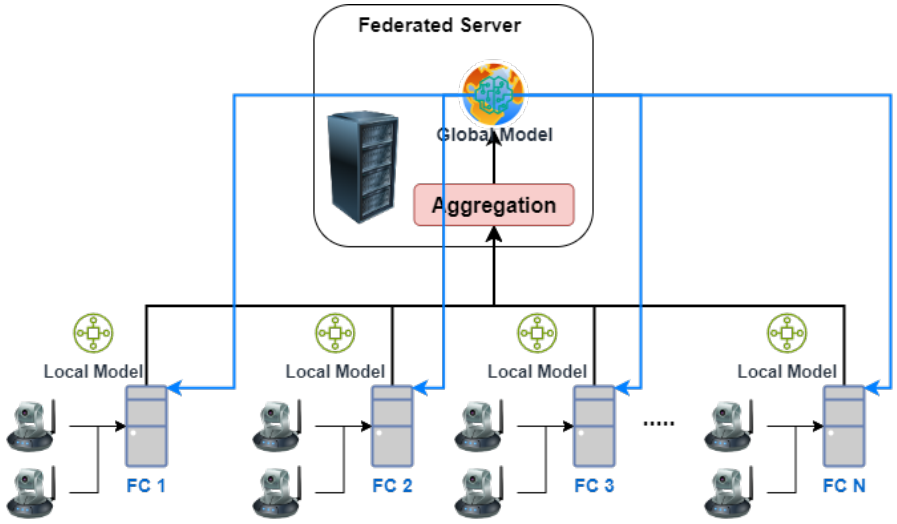

Figure 6. Federated learning architecture in general.

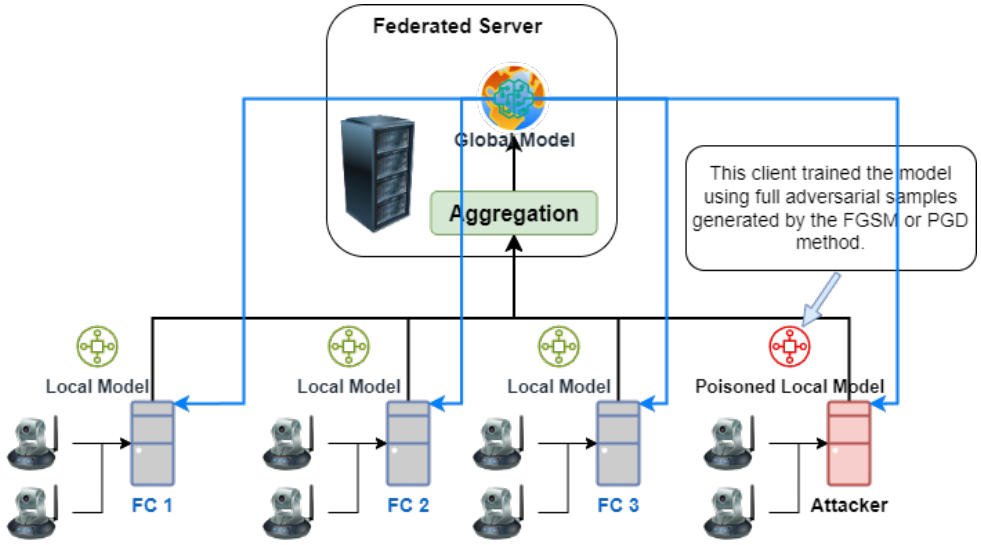

Figure 7. Attack scenario in federated learning.

3.2. Dataset

This investigation utilized a proprietary image dataset [49] designed for air pollution classification, which was conveniently accessible through the Kaggle platform. Captured at seven distinct locations in India—ITO in Delhi, Dimapur in Nagaland, Spice Garden in Bengaluru, Knowledge Park III in Greater Noida, New Ind Town in Faridabad, Borivali East in Mumbai, and Oragadam in Tamil Nadu—along with one site in Biratnagar, Nepal, the dataset comprised a total of 12,240 photos depicting various scenes. All images were taken during daylight hours under natural lighting conditions. Air pollution labels were sourced from the official website of the Central Pollution Control Board (CPCB) [51,52]. Figure 4 provides an overview of the image dataset utilized in this research.

The air quality index (AQI) encompassed six distinct classifications: good, moderate, unhealthy for sensitive groups, unhealthy, very unhealthy, and severe. The dataset was subsequently partitioned into multiple subsets, each containing 1224 images. These subsets were then distributed to all federated clients and servers to evaluate experimental outcomes.

3.3. Model

We utilized the Eff-AQI model [50] from our previous study for this research. Initially, its purpose was to rapidly calculate PM_{10}, $PM_{2.5}$, and air quality index (AQI) values from image inputs. Our commitment to advancing AI for social good and SDG 17 served as the driving force behind the model development. Also, it was developed as part of the national project known as Integrated Command and Control Center for a Sustainable Smart City.

The efficacy, suitability for deployment on edge GPUs, and compatibility with the study's goals were the deciding factors in choosing this model. In order to reduce the size of the model and alter its function from value estimation to AQI classification, specific model layers were modified. The overall model size was reduced without affecting the level of accuracy that was considered acceptable. The decrease in model size had positive effects in the FL environment, as it led to reduced communication costs throughout each round of the FL aggregation procedure.

It is important to acknowledge that although our model selection was based on these considerations, it is worth noting that other established models like MobileNet, ResNet, VGG16, and others could be employed. However, hardware specifications may differ as a result of model architectures and size variations.

The adapted model architecture employed in this study was presented in Figure 5. Specifically designed for processing RGB images with dimensions of $180 \times 180 \times 3$, the model comprised distinct blocks and layers for effective feature extraction and classification. The initial model block encompassed two convolutional neural network (CNN) layers, each featuring 64 filters, along with a max-pooling layer. Subsequently, the second model block consisted of two CNN layers, each incorporating 128 filters, and a single max-pooling layer.

Notably, the modified residual blocks in the third through fifth stages consisted of a singular CNN layer. The output of this CNN layer was added to the preceding max-pooling output before being forwarded to the subsequent max-pooling layer. The first three max-pooling layers employed a 3×3 kernel and stride size, while the fourth and fifth layers used a 2×2 kernel and stride size. The leaky rectified linear unit (LeakyReLU) was adopted as the activation function for all CNN layers, replacing the traditional rectified linear unit (ReLU). This decision was motivated by the ability of LeakyReLU to address the issue of ReLU termination [53] and its demonstrated effectiveness in enhancing model accuracy [54].

The five blocks dedicated to feature extraction were followed by a flattening process, and the resultant features were conveyed to sets of fully connected layers. The initial fully connected layer housed 256 neurons, followed by a layer with 512 neurons. Both fully connected layers utilized LeakyReLU as their activation function. The final, fully connected layer, encompassing six neurons, employed the softmax activation function for AQI classification.

A deliberate choice was made to incorporate residual blocks, which are known to mitigate gradient disappearance issues and enhance feature extraction efficacy [55,56]. To manage computational complexity during the initial stages of the extraction process, residual blocks were excluded from the first two blocks, where a reduced number of CNN filters was deemed sufficient. The overall model comprised 870,598 parameters, which were equivalent to 3.54 MB. The adoption of LeakyReLU and residual blocks aligned with the intention to optimize model performance in feature extraction and classification tasks.

3.4. Adversarial Attacks

In the realm of FL, adversarial attacks were broadly categorized into insiders and outsiders. While Section 1 delved into outsider attacks (refer to Figure 2), the subsequent discussion focused specifically on insider attacks, as depicted in Figure 7. This paper concentrated on white-box attacks, wherein the attacker possessed comprehensive knowledge of the model's architecture. Notably, in the FL scenario, the global model was shared among all clients, potentially granting attackers access to the model architecture upon joining as clients.

The adversarial samples in this study were generated through the utilization of the fast gradient sign method (FGSM) and projected gradient descent (PGD). These methods had been widely adopted in prior research for both attacking and training models [31,57–60]. The proposed approaches were subsequently assessed through adversarial sample attack scenarios and federated adversarial training scenarios. The FGSM, initially suggested by Goodfellow et al. [61], was employed, and the formula governing the sample generation was articulated as follows:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)) \tag{1}$$

where x_{adv} is the adversarial example, x is the original input, y is true lable of input image, ϵ is the perturbation strength, J is the loss function, $\nabla_x J$ is the gradient of the loss with respect to the input, θ is our model, sign is the sign function.

For the generation of FGSM adversarial samples, we adopted epsilon (ϵ) = 0.01, requiring only one iteration to produce a sample image. Similarly, the generation of adversarial samples for PGD also utilized epsilon (ϵ) = 0.01. Unlike FGSM, PGD perturbation is executed iteratively, and for this paper, we employed 20 iterations (t) with a step size (α) of 0.005. The PGD method was introduced by Madry et al. [59], and the formula governing the sample generation is expressed as follows:

$$x_{\text{adv}}^{(0)} = x \quad \text{initialization} \tag{2}$$

$$x_{\text{adv}}^{(t+1)} = \text{clip}_{x,\epsilon}\left(x_{\text{adv}}^{(t)} + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_{\text{adv}}^{(t)}, y)), x \pm \epsilon\right) \tag{3}$$

where $x_{\text{adv}}^{(t)}$ is the adversarial example at iteration t, x is the original input, ϵ is the perturbation strength, α is the step size, J is the loss function, $\nabla_x J(\theta, x_{\text{adv}}^{(t)}, y)$ is the gradient of the loss of model θ with respect to the input for the target label y, sign is the sign function, $\text{clip}_{x,\epsilon}$ is the function that clips values to ensure they stay within the range $x \pm \epsilon$.

3.5. Federated Learning

The overall federated learning procedure is illustrated in Figure 6. The procedural steps are outlined as follows:

1. The server distributed the global model or global weight to each client (blue line);
2. The client utilized its own data to locally retrain the global model;
3. Each client transmitted its trained local model to the server (black line);
4. The server aggregated all the models into a new global model;
5. The server updated each client's model with a new global model or weight (blue line) and repeated steps 1 through 5.

Algorithm 1, referred to as Federated Average (FedAvg) [62], stood as an aggregation algorithm extensively employed in FL. The algorithm incorporated various parameters, including the number of clients (K), local mini-batch size (B), number of local epochs (E), and learning rate (η). At the commencement of the training process, the server initialized the global model parameters (w_0). In each round, the server selected either a random set of clients or all clients for participation in the training, ensuring a minimum level of client involvement (m). Subsequently, the server distributed the current global model parameters (w_t) to the participating clients in parallel.

Algorithm 1 Federated Learning (FedAvg) [62]

Input:

- K: Number of clients, B: Local minibatch size
- E: Number of local epochs, η: Learning rate

Server executes:

initialize w_0
for each round $t = 1, 2, \ldots$ **do**
  for each client $k \in N$ **in parallel do**
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  end for
  $w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$
end for

ClientUpdate(k, w): // Run on client k

$B \leftarrow$ split P_k into batches of size B
for each local epoch i from 1 to E **do**
  for each batch $b \in B$ **do**
    $w \leftarrow w - \eta \nabla l(w; b)$
  end for
end for
return w to server

On the client side, each participating client (k) received the global model parameters and performed local updates. The client divided its local data (P_k) into mini-batches of size B and iterated over the data for a specified number of local epochs (E). Within each epoch, the client processed each batch (b) and updated its local model parameters (w) using the learning rate (η) and the gradient of the loss function ($\nabla l(w, b)$). After completing the local updates, the client returned its updated model parameters (w) to the server.

The server aggregated the received model updates from all participating clients by computing the weighted average based on the number of samples (n_k) used by each client. The aggregated model parameters (w_{t+1}) became the updated global model for the next round. This training process continued for multiple rounds until convergence or a predefined stopping criterion was met.

The FedAvg [62] algorithm ensured that all clients contributed to the global model while preserving data privacy and computational efficiency. It served as a fundamental component in federated learning, facilitating distributed training across a network of clients while maintaining a shared global model.

4. Experiment Results and Discussion

4.1. Experimental Environment

This study made use of a GPU server with the following specifications: The system configuration featured an Intel (R) Xeon (R) Silver 4110 CPU (Intel, Chandler, AZ, USA) operating at a frequency of 2.10 GHz, accompanied by a substantial 128 GB of RAM. Additionally, the setup incorporated five NVIDIA GeForce RTX 2080 Ti (NVIDIA, China) graphics processing units, each equipped with 11 GB of memory. Among these GPUs, one was allocated for the federated server, while the remaining four were dedicated to

individual clients. The operating system employed in this study was Ubuntu 16.04.6 LTS, 64-bit. The programming environments utilized included Python 3.8, TensorFlow 2.8.0, and Flower 1.3.0 [63] (Federated Learning Framework). However, for practical deployment, edge GPUs such as the NVIDIA TX2 and Intel NUC with GPUs were deemed sufficient.

4.2. Experimental Parameters

4.2.1. Natural Model Training Parameters

Natural model training involved training a model conventionally without utilizing FL. The objective of this training session was to assess the accuracy of the model, establishing it as a baseline for comparison with the global model trained using FL. In natural model training, a larger dataset was typically employed for training, validation, and testing. Conversely, FL operated in a distributed manner, with participating clients often having limited data and computing power in real-world scenarios.

For training and testing the model, the parameter settings outlined in Table 1 were applied. Categorical crossentropy served as the loss function during both training and validation. The training configuration included a batch size of 32 and a maximum of 100 epochs. The training process was halted if the loss failed to decrease after 7 iterations. The model weight with the minimum loss value was retained at the conclusion of training. The optimizer chosen for this process was Adam, with a learning rate of 0.0001.

4.2.2. Normal FL and FAT Parameters

The same hyperparameter configuration was applied for both normal FL and FAT in our study, as illustrated in Table 2. Normal FL involved the absence of adversarial examples during the local training process of the client, and this global model did not ensure its robustness. Conversely, the FAT approach incorporated adversarial samples into the training data with the goal of enhancing the robustness of the global model. In this study, the predetermined number of communication rounds was established at 30. Numerous experiments have shown that increasing the number of communication rounds did not significantly improve the accuracy of the global model. We opted for FedAvg as our aggregation method, a widely recognized approach that has proven effective in confirming the utility of our proposed methodologies.

During the initial five communication rounds, all clients were requested to train the local model for a maximum of 35 epochs. Subsequently, the local epochs for the remaining communication rounds were reduced to 15. After five rounds, it was observed that increasing the number of epochs beyond 15 did not result in improved accuracy of local models. Also, this approach reduced the overall duration of the FL training process. The batch size for the client was set at 16 due to the client's limited local data within the actual application. Additionally, since the input image dimensions were significantly larger than those of datasets such as MNIST and CIFAR, a larger batch size would result in out-of-memory (OOM) problems. The early stopping threshold was established at 7 epochs. This was to ensure the training process terminated if the loss value did not decrease within that timeframe. Furthermore, it was sufficient to achieve the optimal local gradient for every client and reduce power consumption during training. The remaining hyperparameters aligned with the process of training natural models.

4.3. Attacks in FL and FAT Strategies for Defense

To illustrate insider attacks, as discussed in Section 3.3, we considered a scenario where attackers had knowledge of the IP address of the federated server and the SSL certificate, enabling them to participate in the federated learning (FL) system as clients. The attackers aimed to engage in data poisoning by transmitting corrupted gradients to the federated server, which were compromised by FGSM and PGD adversarial samples. The goal was to manipulate the global model into misclassifying accurate inputs once the aggregation process was completed. This scenario was depicted in Figure 7. To assess the impact of this attack on global models, we used three attackers (representing 75% of the total clients),

two attackers (50% of the total clients), and one attacker (25% of the total clients) in our experimental setup.

Table 3 contains an exhaustive list of federated training scenarios utilized in our experiments. It also included scenarios to observe the impact of the attacks, as well as to train a robust global model for FGSM and PGD attacks by implementing FAT defense mechanisms.

4.4. Performance Evaluation and Discussion

4.4.1. Natural Model Training and Natural Model Adversarial Training Results

Three training methodologies were implemented during the investigative phase to determine the baseline accuracy of the employed model. The hyperparameters used in this training effort are meticulously outlined in Table 1. The first approach involved training the model on a normal dataset of 7344 RGB images with a size of 180 × 180 pixels. Of these, 5140 images were used for training, and 1102 images were used for both validation and testing. The subsequent method entailed training the model using an adversarial technique, incorporating FGSM adversarial samples into a normal dataset. This composite dataset consisted of 2448 FGSM adversarial samples and 4896 normal RGB images, totaling 7344 images. The images were randomized and partitioned into separate datasets for training, validation, and testing, following the same distribution as the first approach. The third method replicated the scenario of the second, with the only distinction being the use of PGD adversarial samples instead of FGSM adversarial samples to populate the dataset.

Subsequently, these three trained models were evaluated using an undisclosed dataset that remained untouched during the aforementioned phases. As a reminder, the original dataset [49] contained a total of 12,240 images. Therefore, 1224 normal images were selected to assess the accuracy of these models. The results of this evaluation are presented in Table 4.

Table 4. Test accuracy of natural training and natural adversarial training accuracy.

Methods	Test Result	Test Result in Unseen Data
Normal data	91.92%	90.36%
Mixed normal data and FGSM samples	91.65%	91.01%
Mixed normal data and PGD samples	94.28%	92.16%

The test accuracies for the first, second, and third methodologies were 91.92%, 91.65%, and 94.28%, respectively, as shown in Table 4. The corresponding test accuracies for unseen data were 90.36%, 91.01%, and 92.16% for the first, second, and third methodologies, respectively. Notably, the third methodology exhibited the highest level of accuracy, in contrast to the accuracies observed in the first and second methodologies. These findings suggest that adversarial training utilizing both FGSM and PGD adversarial samples yields more robust models in the context of the natural model training paradigm.

4.4.2. FAT with FGSM Samples Results

Numerous experiments on adversarial attacks and federated adversarial training were conducted using FGSM adversarial samples. As explained in Section 4.3, the attack strategy involved white-box insider attacks on the FL. The primary goal was to gain a comprehensive understanding of the impact of poisoning attacks on the classification performance of the global model. In these experiments, each client received 1224 images for training, validation, and testing. The client datasets contained a variety of compositions, including mixed data, adversarial data, or exclusively normal data, as detailed in Table 3. The accuracy of the results obtained from these investigations is graphically depicted in Figure 8.

The accuracies across all scenarios were found to be comparable, as shown in Table 5. The baseline accuracy, represented by the fl-normal scenario (generating a global model without any attackers), was 91.34%. Notably, this accuracy was identical to that observed in fl-fgsm-50, where 50% of all clients were categorized as attackers. The accuracies for fat-fgsm, fl-mixed-fgsm, fl-fgsm-25, and fl-fgsm-75 were 88.55%, 88.15%, 91.18%, and 88.53%, respectively. Based on these empirical findings, we conclude that adversarial attacks utilizing FGSM adversarial samples with epsilon (ϵ) = 0.01 were unable to inflict catastrophic damage on the global model.

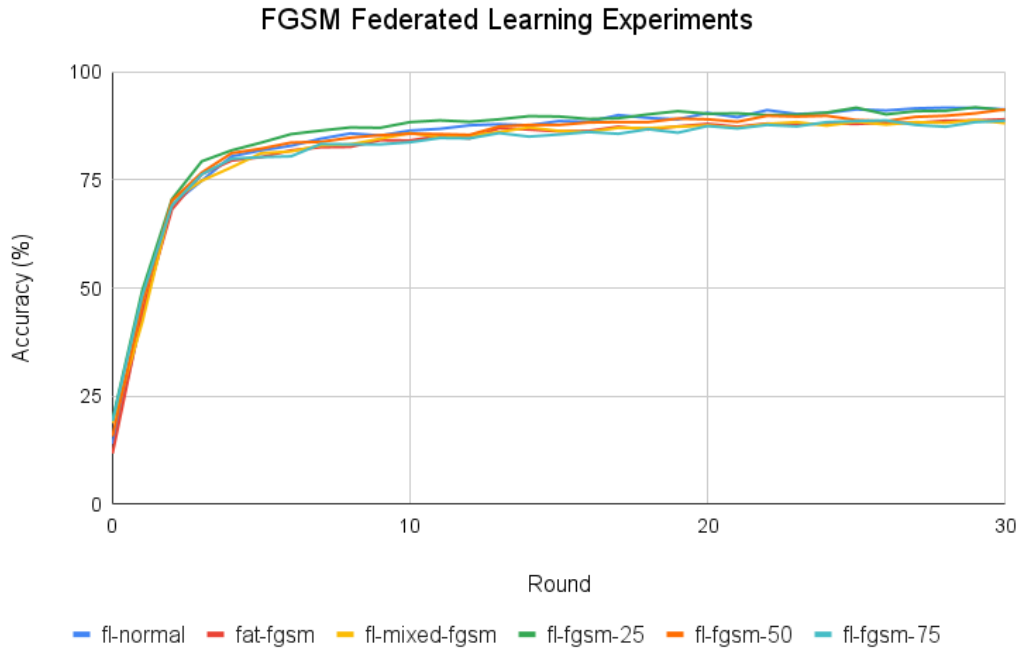

Figure 8. FAT results for different scenarios involving FGSM samples.

Table 5. The result of the test accuracy of FAT scenarios with FGSM samples.

Scenario Name	Accuracy (%)
fl-normal	**91.34**
fat-fgsm	89.05
fl-mixed-fgsm	88.15
fl-fgsm-25	91.18
fl-fgsm-50	**91.34**
fl-fgsm-75	88.73

4.4.3. FAT with PGD Samples Results

This section examines similar FAT scenarios as the preceding section, albeit using adversarial samples generated with PGD. Table 2 details the hyperparameters used in these experiments. The accuracy results for the fl-normal, fat-pgd, mixed-pgd, fl-pgd-25, fl-pgd-50, and fl-pgd-75 scenarios are graphically depicted in Figure 9. Unlike the use of FGSM adversarial samples, which yielded nearly identical results for all six experiments, the results of the experiments in this section vary, as shown in Table 6.

The accuracy of the fl-normal model remained the same as in the previous section, at 91.34%. In the fat-pgd scenario, where 50% of the data in each participating client dataset was normal and 50% was PGD adversarial samples, the accuracy dropped to 81.13%. This represents a 10% reduction in the accuracy from adversarial attacks, demonstrating the effectiveness of the fat-pgd adversarial training method in producing a robust global model. The accuracy reduction from the baseline is relatively small, at approximately 10%, and can be mitigated by using a stronger baseline model architecture with more layers and

neurons or by using a more complex CNN architecture. Another approach to enhancing the robustness of a global model is to increase the number of clients engaged in FL. Assuming a greater volume of data becomes available, the global model will exhibit enhanced accuracy and robustness.

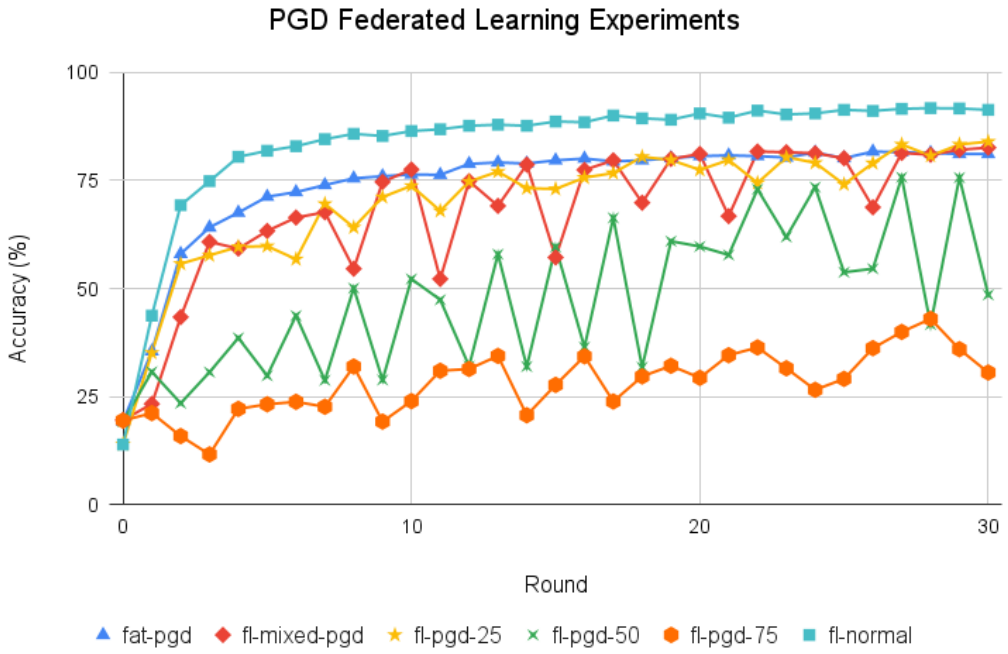

Figure 9. FAT results for different scenarios involving PGD samples.

Table 6. The result of the test accuracy of FAT scenarios with PGD samples.

Scenario Name	Accuracy (%)
fl-normal	**91.34**
fat-pgd	81.13
fl-mixed-pgd	82.60
fl-pgd-25	83.99
fl-pgd-50	48.53
fl-pgd-75	30.56

In the fl-mixed-pgd scenario, one client trained a local model using a normal dataset, two clients used a combination of 50% normal data and 50% PGD adversarial samples, and one client assumed the role of an attacker with a completely adversarial dataset. The accuracy of the global model declined significantly during some communication rounds but recovered to 82.60% after 30 rounds. However, the fat-pgd scenario exhibited greater stability in classification accuracy trends than the fl-mixed-pgd scenario (Figure 9).

Figure 10 shows the confusion matrices for the test accuracy and performance of fl-normal (left), fat-pgd (center), and fl-mixed-pgd (right). fl-normal was used as the baseline. A closer look reveals that fat-pgd has better accuracy in predicting the good air quality index (AQI) class than fl-mixed-pgd, while fl-mixed-pgd has better accuracy in predicting the unhealthy for sensitive groups (USG) and severe classes. However, both models have critical drawbacks.

For example, fat-pgd's accuracy in predicting the severe class could lead to dangerous consequences. If a severe AQI class is misclassified as a very unhealthy class, it is still considered acceptable. However, in the fat-pgd test prediction, 7.5% of the images that should belong to the severe class were predicted as USG class. This could harm people if they believe it is safe to go outside due to this misclassified information.

On the other hand, in the fl-mixed-pgd test prediction for the good class, only 57.75% of the images were predicted correctly; 16.9%, 11.97%, 5.65%, 2.11%, and 5.65% of the images were misclassified as moderate, USG, unhealthy, very unhealthy, and severe, respectively. This result could confuse some people, especially sensitive groups, about whether it is safe to perform activities outside while the air quality is actually good.

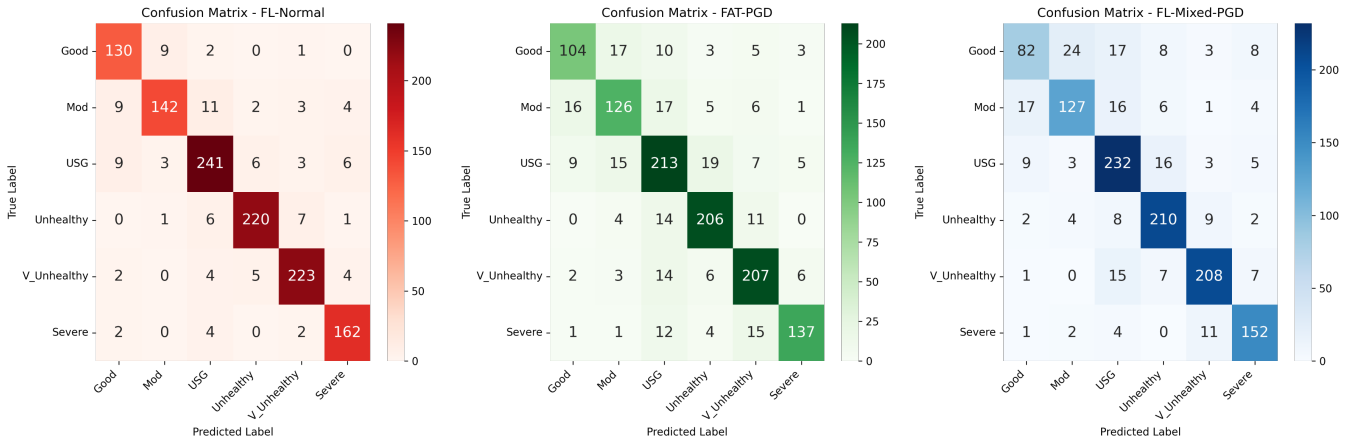

Figure 10. Confusion matrices: (**left**) fl-normal, (**center**) fat-pgd, and (**right**) fl-mixed-pgd.

Based on this explanation, we can understand how dangerous adversarial attacks are. The ability to make models make misjudgments or misclassify decisions can often endanger people.

The three remaining scenarios, denoted as fl-pgd-25, fl-pgd-50, and fl-pgd-75, were all attack scenarios. In the fl-pgd-25 scenario, one attacker joined the FL system, while the remaining three participants served as standard clients (assuming 25% of participants were attackers). The attacker executed a white-box poisoning (insider attack) by using a local model that had been trained exclusively on PGD adversarial samples. The attacker then used this method to send poisoned gradients to the federated server, which aggregated them with the three genuine gradients. This attack was effective because there was no additional filtering in the aggregation process on the federated server. However, the results showed that a 25% participant attack is insufficient to significantly compromise the accuracy of the global model. Under these conditions, the accuracy of the global model can be maintained at 83.99%. The impact of the attacks becomes visible when at least 50% of the participants are attackers, as demonstrated by the scenario results for fl-pgd-50 (50% attackers) and fl-pgd-75 (75% attackers). In scenarios fl-pgd-50 and fl-pgd-75, the global model accuracy after the attack was recorded at 48.53% and 30.56%, respectively.

To gain insights into the adversarial attack phenomenon, we scrutinized the results of the fl-pgd-50 scenario. In this scenario, Clients 1 and 2 were assigned as normal clients, while Clients 3 and 4 acted as attackers. The normal clients retrained and updated their models using a standard dataset, while the attackers used only PGD adversarial samples to retrain their models. After retraining, the local gradients from each client and attacker were sent to the federated server. The federated server used the FedAvg algorithm to aggregate all received gradients, creating a new global model, which was then evaluated using the server's own dataset.

Figure 11 shows the local accuracy of each client and attacker, as well as the updated global model accuracy for each iteration. The figure shows that the attackers achieved almost 100% accuracy in their local model updates. This high accuracy on the attacker side indicates an increased risk level, as it means that the attackers have effectively manipulated their local models. Therefore, it is reasonable to infer that including these poisoned gradients in the aggregation process would significantly reduce the accuracy of the resulting global model.

Although both of the normal clients achieved around 75% accuracy, this contribution was insufficient to counteract the negative impact of the attacker gradients on the aggregation process. As a result, the global model exhibited very low and unstable accuracy, as shown by the orange line in Figure 11. Moreover, as a consequence of this attack, Clients 1 and 2 were unable to attain their optimal accuracy in their local updates. This was attributed to the reception of compromised and erroneous gradients from the global model in each round. In contrast to the fl-normal scenario, where all participating clients were normal, this led to a relatively high overall accuracy. The impact of the attack in this scenario significantly hindered the ability of Clients 1 and 2 to achieve their highest accuracy levels during local updates.

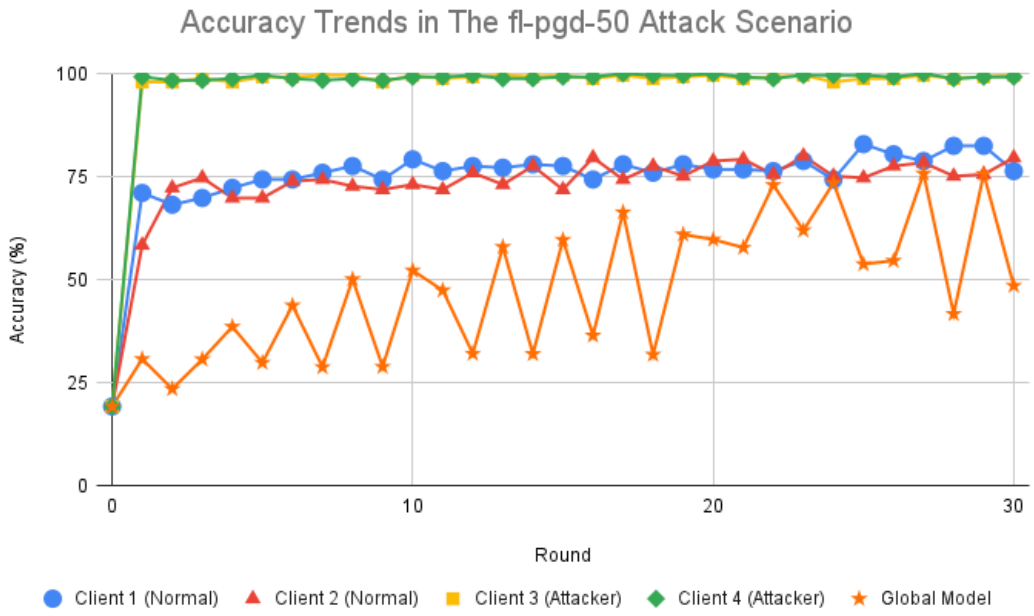

Figure 11. Accuracy trends for each client and server in fl-pgd-50 scenario.

Figures 12 and 13 provide a more comprehensive insight into how the attacker gradients detrimentally impacted the robustness of the global model. Figure 12 shows that Clients 1 (Figure 12a) and 2 (Figure 12b) were about equally good at correctly identifying the air quality index (AQI) class when their local models were tested on a normal dataset. In contrast, Client 3 (Figure 12c) predominantly classified inputs as severe when evaluating its local model on a normal dataset. Client 4's (Figure 12d) results were akin to those of Client 3, with a notable ability to correctly predict 60.08% of very unhealthy classes. However, accuracy for good, moderate, unhealthy for sensitive groups (USG), and unhealthy classes was considerably lower. Consequently, when these four local model gradients were aggregated to formulate a global model, the resultant accuracy of the new global model diminished, as evident in Figure 13.

Figure 13 delineates the confusion matrix of the final global model after 30 rounds of federated adversarial training (FAT) in the fl-pgd-50 scenario. Due to the influence exerted by Clients 3 and 4, 53.35% of the 1224 normal images were classified as having a severe AQI, whereas the actual correct classification was only 160 images; the other 493 images were misclassified as having a severe AQI. This outcome underscores the success of the attacks in shaping the final global model, which demonstrated a tendency to misclassify into a specific class, namely the severe class in this instance.

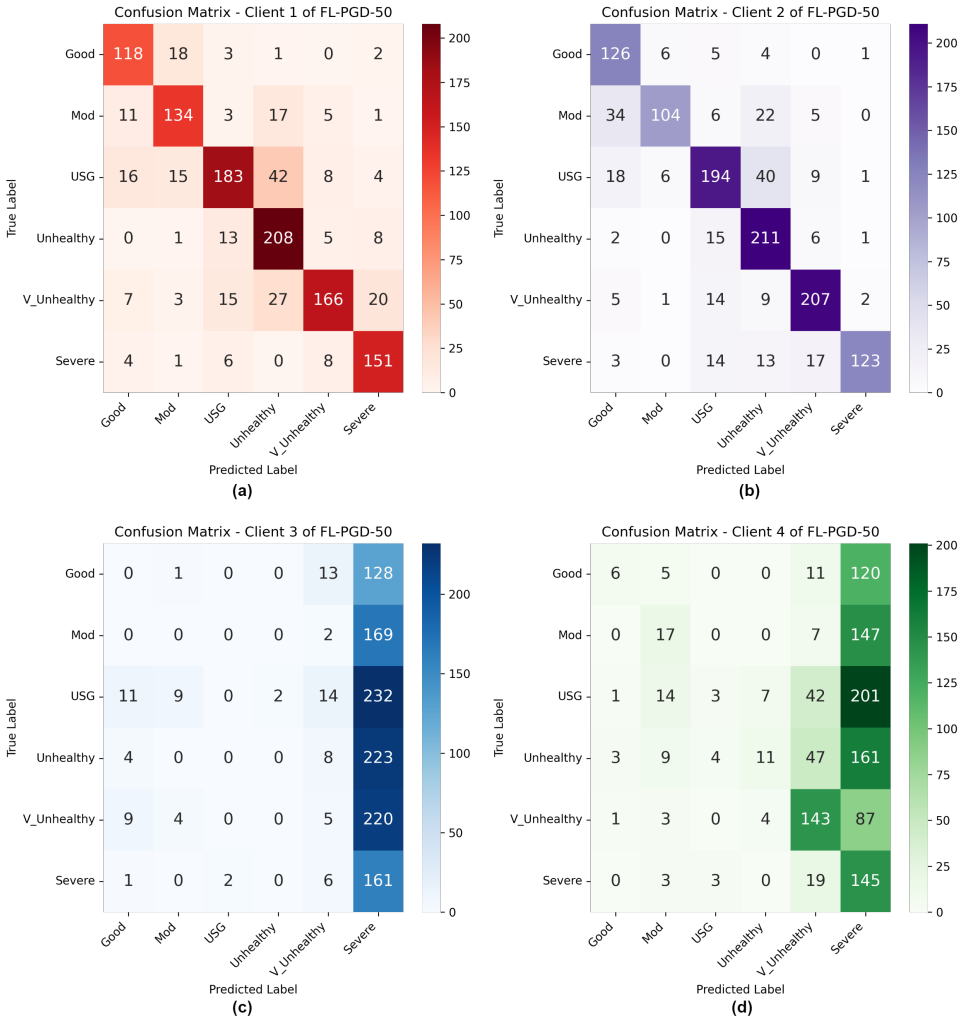

Figure 12. Confusion matrices of fl-pgd 50 scenario: (**a**) Client 1; (**b**) Client 2; (**c**) Client 3; and (**d**) Client 4.

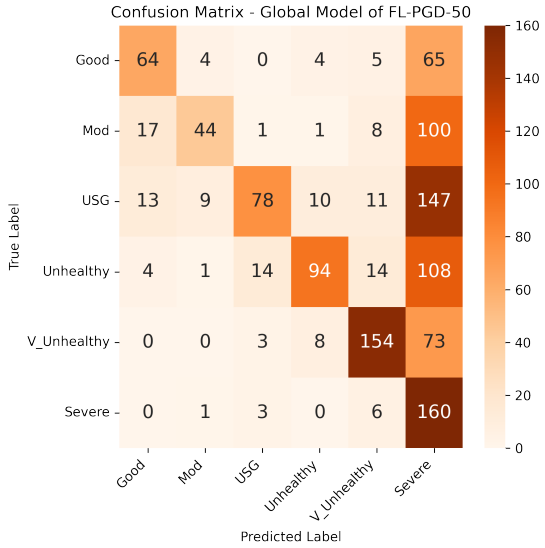

Figure 13. Confusion matrix of global model in fl-pgd-50 scenario.

4.4.4. Defense Mechanism Suggestions for Future Works

This section explores additional defense mechanisms that have the potential to improve security and privacy, building on the previously introduced and tested mechanisms.

Future research could explore the potential of implementing role-based access control (RBAC) in the FL ecosystem. RBAC could play a crucial role in managing and controlling access to the various components and processes involved in FL. By employing RBAC, it would be feasible to regulate and grant access to clients, so unauthorized clients could not join the system, thus mitigating the risk of insider attacks. Additionally, RBAC could prevent clients with limited authorization from sending their local updates to the server or receiving the updated global model from the server. This would ensure the traceability and trustworthiness of client gradients in the aggregation process, thereby enhancing the security and privacy of the FL ecosystem.

Another potential strategy is to adopt a trustworthy aggregation protocol. To facilitate trustworthy aggregation, a filtering layer can be added before the aggregation process to check for anomalous behavior in the gradients. Specifically, the filtering layer can evaluate each client's gradients against an unseen dataset that is statistically representative of the application's needs. If the filtering layer detects malicious gradients, the federated server can exclude or delete them from the aggregation process. For example, gradients from local models like those in Figure 12c,d could be excluded from the aggregation process. If the malicious gradients are not filtered out, they will not only affect the global model but will also prevent other clients from training their local models to their maximum accuracy. This approach would be optimal when combined with our proposed methodology, resulting in a final global model that is both robust and highly accurate.

5. Conclusions

This study presented a new adversarial training strategy for federated learning that is effective even with limited data. The strategy was tested against a wide range of adversarial attacks and made the global model more robust overall. PGD and FGSM techniques were used to generate adversarial samples, and the AQI classification model was used as an example, as it is easy to understand. Monitoring and mitigating air pollution is a primary objective of Sustainable Development Goal 17 (SDG 17).

Federated learning is a new learning paradigm that protects user privacy and enhances system security. It also shows promise for IoT and smart cities. Experimental findings indicate that FGSM-based adversarial attacks are less effective at reducing the accuracy of global models than PGD-based attacks.

This research challenged the assumption that adversarial training is only effective with large datasets. It provided valuable insights that can be used to develop new defense scenarios and countermeasures against adversarial attacks in federated learning. Ultimately, this research contributed to improving the security of sustainable smart city applications.

Nevertheless, the federated adversarial training strategies proposed in this study make the global model more robust to attacks. For example, the fat-pgd and fl-mixed-pgd scenarios have accuracies of 81.13% and 82.60%, respectively, which is only a 10% loss compared to the fl-normal scenario, which achieves an accuracy of 91.34%. This potential drawback may be alleviated by using a more effective baseline model and increasing the number of clients. As more clients join, more data will be involved in the training, and the quality of the global model will be improved.

Author Contributions: Conceptualization, S.U. and P.-A.H.; methodology, S.U.; software, S.U.; validation, P.-A.H. and S.U.; formal analysis, S.U.; investigation, S.U.; resources, P.-A.H.; data curation, A.R.; writing—original draft preparation, S.U.; writing—review and editing, P.-A.H. and S.U.; visualization, S.U.; supervision, P.-A.H.; project administration, H.-C.H.; funding acquisition, P.-A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partially supported by research project grants from the National Science and Technology Council, Taiwan, under the project grant numbers NSTC 111-2634-F-194-003 and NSTC 112-2927-I-194-001.

Data Availability Statement: The dataset is available at: https://www.kaggle.com/datasets/adarshrouniyar/air-pollution-image-dataset-from-india-and-nepal (accessed on 27 June 2023).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cheng, L.; Liu, F.; Yao, D.D. Enterprise data breach: Causes, challenges, prevention, and future directions. *WIREs Data Min. Knowl. Discov.* **2017**, *7*, e1211. [CrossRef]
2. Neto, N.N.; Madnick, S.; Paula, A.M.G.D.; Borges, N.M. Developing a Global Data Breach Database and the Challenges Encountered. *J. Data Inf. Qual.* **2021**, *13*, 1–33. [CrossRef]
3. Neto, N.N.; Madnick, S.; Paula, A.M.G.D.; Borges, N.M. Cyber Security Data Breaches. 2020. Available online: https://databreachdb.com/ (accessed on 1 October 2023).
4. Cadwalladr, C.; Graham-Harrison, E. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *Guardian* **2018**, *17*, 22.
5. Wang, P.; Johnson, C. Cybersecurity incident handling: A case study of the Equifax data breach. *Issues Inf. Syst.* **2018**, *19*, 150–159.
6. Zou, Y.; Mhaidli, A.H.; McCall, A.; Schaub, F. "I've Got Nothing to Lose": Consumers' Risk Perceptions and Protective Actions after the Equifax Data Breach. In Proceedings of the Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018), Baltimore, MD, USA, 12–14 August 2018; pp. 197–216.
7. Leong, Y.Y.; Chen, Y.C. Cyber risk cost and management in IoT devices-linked health insurance. *Geneva Pap. Risk Insur. Issues Pract.* **2020**, *45*, 737–759. [CrossRef]
8. Nair, A.K.; Raj, E.D.; Sahoo, J. A robust analysis of adversarial attacks on federated learning environments. *Comput. Stand. Interfaces* **2023**, *86*, 103723. [CrossRef]
9. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. *arXiv* **2019**, arXiv:1906.08935.
10. Lim, J.Q.; Chan, C.S. From Gradient Leakage To Adversarial Attacks In Federated Learning. In Proceedings of the 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 19–22 September 2021; pp. 3602–3606. [CrossRef]
11. Zhao, B.; Mopuri, K.R.; Bilen, H. iDLG: Improved Deep Leakage from Gradients. *arXiv* **2020**, arXiv:2001.02610.
12. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting gradients-How easy is it to break privacy in federated learning? In Proceedings of the 34th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 6–12 December 2020; pp. 16937–16947.
13. Shen, S.; Zhu, T.; Wu, D.; Wang, W.; Zhou, W. From distributed machine learning to federated learning: In the view of data privacy and security. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6002. [CrossRef]
14. Hsiung, P.A.; Utomo, S.; A, J.; Rouniyar, A.; Hsu, H.C.; Jiang, G.H.; Chang, C.H.; Tang, K.C. Trustworthy AI and Federated Learning for Sustainable Smart Cities, 2023. Available online: https://smartcities.ieee.org/newsletter/january-2023/trustworthy-ai-and-federated-learning-for-sustainable-smart-cities (accessed on 29 September 2023).
15. Vu Khanh, Q.; Nguyen, V.H.; Minh, Q.N.; Dang Van, A.; Le Anh, N.; Chehri, A. An efficient edge computing management mechanism for sustainable smart cities. *Sustain. Comput. Inform. Syst.* **2023**, *38*, 100867. [CrossRef]
16. Debauche, O.; Mahmoudi, S.; Guttadauria, A. A New Edge Computing Architecture for IoT and Multimedia Data Management. *Information* **2022**, *13*, 89. [CrossRef]
17. Badidi, E.; Mahrez, Z.; Sabir, E. Fog Computing for Smart Cities' Big Data Management and Analytics: A Review. *Future Internet* **2020**, *12*, 190. [CrossRef]
18. Sittón-Candanedo, I.; Alonso, R.S.; García, O.; Muñoz, L.; Rodríguez-González, S. Edge Computing, IoT and Social Computing in Smart Energy Scenarios. *Sensors* **2019**, *19*, 3353. [CrossRef] [PubMed]
19. Zhang, D.G.; Ni, C.H.; Zhang, J.; Zhang, T.; Yang, P.; Wang, J.X.; Yan, H.R. A Novel Edge Computing Architecture Based on Adaptive Stratified Sampling. *Comput. Commun.* **2022**, *183*, 121–135. [CrossRef]
20. Lv, Z.; Chen, D.; Lou, R.; Wang, Q. Intelligent edge computing based on machine learning for smart city. *Future Gener. Comput. Syst.* **2021**, *115*, 90–99. [CrossRef]
21. Li, H.; Ota, K.; Dong, M. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Netw.* **2018**, *32*, 96–101. [CrossRef]
22. Barba-Guaman, L.; Eugenio Naranjo, J.; Ortiz, A. Deep Learning Framework for Vehicle and Pedestrian Detection in Rural Roads on an Embedded GPU. *Electronics* **2020**, *9*, 589. [CrossRef]
23. Rajagopal, A.; Bouganis, C.S. perf4sight: A toolflow to model CNN training performance on Edge GPUs. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 963–971. [CrossRef]
24. Mathur, A.; Beutel, D.J.; de Gusmão, P.P.B.; Fernandez-Marques, J.; Topal, T.; Qiu, X.; Parcollet, T.; Gao, Y.; Lane, N.D. On-device Federated Learning with Flower. *arXiv* **2021**, arXiv:2104.03042.

25. Ahmed, K.M.; Imteaj, A.; Amini, M.H. Federated Deep Learning for Heterogeneous Edge Computing. In Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 13–15 December 2021; pp. 1146–1152. [CrossRef]
26. Truong, H.T.; Ta, B.P.; Le, Q.A.; Nguyen, D.M.; Le, C.T.; Nguyen, H.X.; Do, H.T.; Nguyen, H.T.; Tran, K.P. Light-weight federated learning-based anomaly detection for time-series data in industrial control systems. *Comput. Ind.* **2022**, *140*, 103692. [CrossRef]
27. Yamany, W.; Moustafa, N.; Turnbull, B. OQFL: An Optimized Quantum-Based Federated Learning Framework for Defending Against Adversarial Attacks in Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 893–903. [CrossRef]
28. Qayyum, A.; Janjua, M.U.; Qadir, J. Making federated learning robust to adversarial attacks by learning data and model association. *Comput. Secur.* **2022**, *121*, 102827. [CrossRef]
29. Hu, F.; Zhou, W.; Liao, K.; Li, H.; Tong, D. Toward Federated Learning Models Resistant to Adversarial Attacks. *IEEE Internet Things J.* **2023**, *10*, 16917–16930. [CrossRef]
30. Hong, J.; Wang, H.; Wang, Z.; Zhou, J. Federated Robustness Propagation: Sharing Robustness in Heterogeneous Federated Learning. *arXiv* **2022**, arXiv:2106.10196.
31. Zhu, J.; Yao, J.; Liu, T.; Yao, Q.; Xu, J.; Han, B. Combating Exacerbated Heterogeneity for Robust Models in Federated Learning. *arXiv* **2023**, arXiv:2303.00250.
32. Chen, Z.; Tian, P.; Liao, W.; Yu, W. Zero Knowledge Clustering Based Adversarial Mitigation in Heterogeneous Federated Learning. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1070–1083. [CrossRef]
33. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *arXiv* **2021**, arXiv:1912.04977.
34. Shah, D.; Dube, P.; Chakraborty, S.; Verma, A. Adversarial training in communication constrained federated learning. *arXiv* **2021**, arXiv:2103.01319.
35. Jere, M.S.; Farnan, T.; Koushanfar, F. A Taxonomy of Attacks on Federated Learning. *IEEE Secur. Priv.* **2021**, *19*, 20–28. [CrossRef]
36. Zizzo, G.; Rawat, A.; Sinn, M.; Buesser, B. FAT: Federated Adversarial Training. *arXiv* **2020**, arXiv:2012.01791.
37. Singh, S.K.; Azzaoui, A.E.; Kim, T.W.; Pan, Y.; Park, J.H. DeepBlockScheme: A Deep Learning-Based Blockchain Driven Scheme for Secure Smart City. *Hum. Centric Comput. Inf. Sci.* **2021**, *11*, 1–12. [CrossRef]
38. Kumar, P.; Kumar, R.; Srivastava, G.; Gupta, G.P.; Tripathi, R.; Gadekallu, T.R.; Xiong, N.N. PPSF: A Privacy-Preserving and Secure Framework Using Blockchain-Based Machine-Learning for IoT-Driven Smart Cities. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2326–2341. [CrossRef]
39. Singh, S.K.; Jeong, Y.S.; Park, J.H. A deep learning-based IoT-oriented infrastructure for secure smart City. *Sustain. Cities Soc.* **2020**, *60*, 102252. [CrossRef]
40. Utomo, S.; John, A.; Rouniyar, A.; Hsu, H.C.; Hsiung, P.A. Federated Trustworthy AI Architecture for Smart Cities. In Proceedings of the 2022 IEEE International Smart Cities Conference (ISC2), Paphos, Cyprus, 26–29 September 2022; pp. 1–7. [CrossRef]
41. Floridi, L. Establishing the rules for building trustworthy AI. *Nat. Mach. Intell.* **2019**, *1*, 261–262. [CrossRef]
42. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Vincent Poor, H. Federated Learning for Internet of Things: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* **2021**, *23*, 1622–1658. [CrossRef]
43. Singh, S.; Rathore, S.; Alfarraj, O.; Tolba, A.; Yoon, B. A framework for privacy-preservation of IoT healthcare data using Federated Learning and blockchain technology. *Future Gener. Comput. Syst.* **2022**, *129*, 380–388. [CrossRef]
44. Bao, Z.; Lin, Y.; Zhang, S.; Li, Z.; Mao, S. Threat of Adversarial Attacks on DL-Based IoT Device Identification. *IEEE Internet Things J.* **2022**, *9*, 9012–9024. [CrossRef]
45. Ibitoye, O.; Shafiq, O.; Matrawy, A. Analyzing Adversarial Attacks against Deep Learning for Intrusion Detection in IoT Networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Big Island, HI, USA, 9–13 December 2019; pp. 1–6. [CrossRef]
46. Luo, Z.; Zhao, S.; Lu, Z.; Sagduyu, Y.E.; Xu, J. Adversarial machine learning based partial-model attack in IoT. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning, New York, NY, USA, 13 July 2020; pp. 13–18. [CrossRef]
47. Anthi, E.; Williams, L.; Javed, A.; Burnap, P. Hardening machine learning denial of service (DoS) defences against adversarial attacks in IoT smart home networks. *Comput. Secur.* **2021**, *108*, 102352. [CrossRef]
48. Zhao, Y.; Chen, J.; Zhang, J.; Wu, D.; Blumenstein, M.; Yu, S. Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e5906. [CrossRef]
49. Rouniyar, A.; Utomo, S.; A, J.; Hsiung, P.A. Air Pollution Image Dataset from India and Nepal, 2023. Available online: https://www.kaggle.com/datasets/adarshrouniyar/air-pollution-image-dataset-from-india-and-nepal (accessed on 27 June 2023).
50. Utomo, S.; Rouniyar, A.; Jiang, G.H.; Chang, C.H.; Tang, K.C.; Hsu, H.C.; Hsiung, P.A. Eff-AQI: An Efficient CNN-Based Model for Air Pollution Estimation: A Study Case in India. In Proceedings of the 2023 ACM Conference on Information Technology for Social Good, Lisbon, Portugal, 6–8 September 2023; GoodIT '23; pp. 165–172. [CrossRef]
51. National Air Quality Index. Available online: https://app.cpcbccr.com/AQI_India/ (accessed on 27 June 2023).
52. Hourly Weather in Biratnagar, Nepal. Available online: https://www.tomorrow.io/weather/NP/4/Biratnagar/079711/hourly/ (accessed on 27 June 2023).
53. Zhang, K.; Chen, Z.; Xiang, Y. Vision-Based Particulate Matter Estimation. In *Deep Learning Applications: In Computer Vision, Signals and Networks*; World Scientific: Singapore, 2023; pp. 3–17. [CrossRef]

54. Utomo, S.; John, A.; Pratap, A.; Jiang, Z.S.; Karthikeyan, P.; Hsiung, P.A. AIX Implementation in Image-Based PM2. 5 Estimation: Toward an AI Model for Better Understanding. In Proceedings of the 2023 15th International Conference on Knowledge and Smart Technology (KST), Phuket, Thailand, 21–24 February 2023; pp. 1–6. [CrossRef]
55. Wang, Z.; Yang, Y.; Yue, S. Air quality classification and measurement based on double output vision transformer. *IEEE Internet Things J.* **2022**, *9*, 20975–20984. [CrossRef]
56. Zhang, Q.; Fu, F.; Tian, R. A deep learning and image-based model for air quality estimation. *Sci. Total Environ.* **2020**, *724*, 138178. [CrossRef]
57. Zhang, J.; Li, B.; Chen, C.; Lyu, L.; Wu, S.; Ding, S.; Wu, C. Delving into the Adversarial Robustness of Federated Learning. *arXiv* **2023**, arXiv:2302.09479.
58. Li, X.; Song, Z.; Yang, J. Federated Adversarial Learning: A Framework with Convergence Analysis. *Proc. Mach. Learn. Res.* **2022**, *202*, 19932–19959.
59. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. *arXiv* **2019**, arXiv:1706.06083.
60. Zhao, W.; Alwidian, S.; Mahmoud, Q.H. Adversarial Training Methods for Deep Learning: A Systematic Review. *Algorithms* **2022**, *15*, 283. [CrossRef]
61. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
62. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the Machine Learning Research (PMLR), Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
63. Beutel, D.J.; Topal, T.; Mathur, A.; Qiu, X.; Fernandez-Marques, J.; Gao, Y.; Sani, L.; Kwing, H.L.; Parcollet, T.; Gusmão, P.P.D.; et al. Flower: A Friendly Federated Learning Research Framework. *arXiv* **2020**, arXiv:2007.14390.