# ENHANCING IOT SECURITY: A NOVEL APPROACH WITH FEDERATED LEARNING AND DIFFERENTIAL PRIVACY INTEGRATION

Aziz Ullah Karimy, P Chandrasekhar Reddy

Department of Electronics and Communication Engineering, University College of Engineering , Science & Technology, Jawaharlal Nehru Technological University, Hyderabad-500085, Telangana, India.

## *ABSTRACT*

*The Internet of Things (IoT) has expanded to a diverse network of interconnected electronic components, including processors, sensors, actuators, and software throughout several sectors such as healthcare, agriculture, smart cities, other industries. Despite offering simplified solutions, it introduces significant challenges, specifically data security and privacy. Machine Learning (ML), particularly the Federated Learning (FL) framework has demonstrated a promising approach to handle these challenges, specifically by enabling collaborative model training for Intrusion Detection Systems (IDS). However, FL faces some security and privacy issues, including adversarial attacks, poisoning attacks, and privacy leakages during model updates. Since the encryption, mechanisms poses issues like computational overheads and communication costs. Hence, there is need for exploring of alternative mechanism such as Differential Privacy (DP). In this research, we demonstrate an experimental study aiming exploring of FL with DP to secure IoT environment. This study analyzes the effectiveness of DP in horizontal FL setup under Independent and Identically Distributed (IID) pattern. Results on MNIST dataset show promising outcomes; FL with and without employing DP mechanism achieve an accuracy of 98.92% and 98.2%, respectively. Furthermore, the accuracy rate achieved with complex cybersecurity dataset is 93% and 91% before and after employing the DP mechanism. These findings outlines the efficiency of DP in FL framework for improving security and privacy in IoT environment.*

## *KEYWORDS*

*Internet of Things; Cybersecurity; Federated Learning; Collaborative learning; Differential Privacy*

## 1. INTRODUCTION

The Internet of Things (IoT) applications are made up of various electronic components, including sensors, actuators, processors, and software, creating a complete heterogeneous interconnected network of ‘things’. These devices are equipped with sensors connected to constrained computational resources to operate efficiently in diverse environments, from healthcare, smart cities, agricultural fields, and smart grids to other industries, to facilitate tasks [1]. Therefore, IoT technologies play a very critical role in enabling automation and connectivity across a broad range of sectors and use cases. These applications of IoT offer a range of simplifications and solutions to daily lives and industrial environments. However, it also comes with significant challenges, including dealing with large volumes of data, communication issues between the devices, and the safety of the data [2]. The impact of IoT security exploitation is significant on socio-economic factors, including users' privacy and infrastructure integrity. The outcomes of the breaches lead to financial losses, exploiting personal information, and interrupting fundamental services. Consequently, the security and privacy of the IoT have

DOI: 10.5121/ijcnc.2024.16401

become the most important challenges that researchers are paying attention to. It has been very crucial to find possible methods of protecting IoT devices and the collected data from misuse and tampering by intruders [3].

For the same context, Artificial Intelligence (AI) has been proposed as a powerful method for securing IoT from intrusions. Developing an Intrusion Detection System (IDS) using Machine/Deep Learning (ML/DL) has proved to accurately detect attacks in IoT environments without regular updating of conventional signature-based IDS rules [4]-[5]. However, in the centralized method, sharing of data to central server before constructing the model is must, which causes some challenges, including data transmission and communication overhead, and security and privacy of user's data [6].

Federated Learning (FL) has arisen as a sub-field of ML concentrating on training models across various distributed collaborators at the edge without sharing private data to a central server. In FL, training takes place at local IoT devices; each device trains a local model using its own existing data and then shares only model gradients/updates [7]. This technique has proven to significantly improve model performance, reduce communication costs and computation resources, and reduces the risks of personal data leakages [8]. However, FL is not without its challenges, particularly in the realms of security and privacy. The deployment of the FL framework in various domains has exposed certain security and privacy challenges, poisoning exploitations for their normal execution [9]. Researchers have highlighted three important targets as potential security gaps. Firstly, the possible exploitation of local data by malicious users leads to generating poisoning models, impacting the integrity of the learning process. Secondly, eavesdropping on the exchange of updates between clients and servers raises privacy concerns and can be compromised through reverse engineering, in [10] researchers have proved the possibility of obtaining private training data from gradients. Finally, the server becomes the main target, poisoned local models created by exploited clients, causes to generate a tainted global model; in support of this concept, a study in [11] used generative adversarial networks to exploit model aggregation in server for stealing user data, they implemented the attack by generating similar local model updates. As a result the tainted global model, which is distributed back to all clients, consequently, causes a critical security issue in the FL framework.

Researchers have provided a couple of techniques to preserve privacy in FL framework, using cryptographic method for encrypting updates/gradients [12]. However, these techniques come with some extra challenges in constrained IoT devices, like increased computational overhead and communication costs, and make FL setup exploitable to attacks [12]. To overcome these limitations, DP has got researchers attention as robust alternative approach to preserve security and privacy in FL settings [13]. In contrast to encryption methods, which may introduce significant latency and computational overhead on IoT devices due to complex encryption and decryption processes, FL with DP offers a more lightweight and efficient solutions.The DP mechanism usually adds random noise to the intermediate output, ensuring that the change of an input element will not causes too much difference in the output distribution [13]. With adding noise at weights of each model of IoT device locally, DP reduces the need for central data aggregation and encryption, on the other hand, lower communication costs and enhance data privacy.

In this research study, we carried out an experimental study to secure IoT systems with help of FL settings, incorporating DP to enhance further the security and privacy of the IoT environment. In our experiment, we considered two datasets from cybersecurity and regular ML testing domains, and tested the effectiveness of DP in multiple FL settings. Renyi Differential Privacy (RDP) was utilized to specify more rigid privacy measures for this experimental study. Through this study, our intention was to showcase the feasibility and effectiveness of DP mechanisms in FL settings, and overall enhancing security and privacy of IoT systems.

The rest of the paper is organized as follows: Section 2 of the paper presents an extensive overview of existing approaches in IoT security, countermeasures with ML approaches, and security and privacy issues in the FL approach—section 3 delves into preliminaries of the research and overview of the proposed method. In Section 4, we elaborate on the experimental setup, ML algorithms, and utilized datasets. Section 5 offers an in-depth analysis of performance results, complemented by meaningful discussions. The paper concludes with insights into the efficacy of FL with DP mechanisms in securing IoT applications and provides recommendations for future research.

## 2. RELATED WORK

In this section, we briefly overview some of the most significant research works on IoT security and privacy, current machine learning approaches, and their limitations.

Research studies in [14] and [15] highlighted that IoT systems' constrained characteristics, such as limited processing power and memory, restrict robust security measures. An insecure heterogeneous network utilized in an IoT environment causes several security and privacy challenges, making it exploitable for cyber-attacks. Different ML solutions have been proposed to tackle these security challenges in the field. [16] and [17] reviewed these solutions, which rely on conventional ML approaches. Although these approaches have shown promise, they face challenges such as high computational overhead and the risk of data privacy breaches.

McMahan et al. [18] suggested a federated averaging algorithm to schedule clients in a synchronized way, average weights for updating gradients, and generate the global model. In this approach, the client’s data is exploited by the server and susceptible to inference intrusions. Nguyen et al. [19] introduced DIoT a self-learning system to detect Mirai attacks in smart home networks; the system was implemented using Python’s Flask and Tensorflow for a federated learning global model. The system architecture consists of two parts: the security gateway acts as an access point between IoT devices and the internet, and IoT services store device-specific anomaly models. This approach is able to reduce false alarms, but it is designed only for Mirai attack detection and lacks a dedicated FL deep learning framework.

Rahman et al. [20] proposed an IDS based on FL for IoT networks, which maintains the privacy of data and detects attacks. They tested their method with various use cases to simulate real-world scenarios. Based on the experimental study, they concluded that federated learning could perform the same as a centralized approach in accuracy. Mowla et al. [21] suggested a FL based approach to detect jamming attacks formed by Unmanned Aerial Vehicles (UAVs) in Flying Ad Hoc Networks; they implemented the Dempster–Shafer theory to prioritize client groups and find better groups for calculating global gradients. Wei Ou et al. [22] introduced a privacy-preserving vertical FL framework for Bayesian ML with the help of HE; the experimental study shows a performance accuracy of 90% in a single union server training model. Fang and Qian [23], with the help of Homomorphic Encryption (HE), suggested a multiparty privacy-preserving ML approach in FL settings, which utilizes HE-encrypted gradients to protect the security of clients' private information. However, this framework needs more communication overhead and low scalability. Similarly in other research study Zhang et al. [24] used the Homomorphic Encryption (HE) technique to encrypt local gradients and introduced a FL approach to preserve privacy. To minimize computation and communication costs, they implemented distributed selective SGD. To make a diverse attack scenario, they used a generative adversarial network (GAN). Bagdasaryan et al. [25] carried out research on the impact of gradient exploitation by malicious devices and noticed the negative effect on main model performance. Moreover, the authors outlined the potential for backdoor updates; adversaries are capable of tampering with client model updates by exploiting some of the participants. Zhu et al. [10] carried out an interesting study on deep learning called deep leakage from gradients; they used an optimization algorithm to rebuild training data samples and their labels. They could successfully highlight the reconstructing images and texts used in deep learning model training.

Based on the aforementioned studies in FL settings of [10]–[12], [14]-[25], we notice that most of them are susceptible to a variety of attacks or require more computation resources and communication costs. Hence, the FL framework requires further research to develop a secure mechanism for implementation, particularly in the IoT environment.

## 3. PRELIMINARIES AND PROPOSED METHOD

In this section, we mainly introduce the algorithms, technologies, and structure for this experimental study.

### 3.1. Federated Learning (FL)

FL is a sub-domain of Machine Learning (ML), which has achieved considerable attention from researchers, having the capability of handling privacy-related issues for network settings where sensitive data is dispersed among devices/edge nodes, reducing communication costs and improving overall model performances[7]. FL is a creative ML approach with decentralized model training capability across several edge devices. In contrast to conventional centralized ML model training techniques, this method gets rid of raw data sharing to a central location that is susceptible to significant security and privacy risks by transmitting data from numerous sources to a central server for model training [26]. The steps involved in FL are mainly divided into three stages: system initialization and device selection, local training and update, and model aggregation [27]; each step is explained as follows:

**System Initialization and Device Selection:** At this stage, the server initializes learning parameters to train the model for performing some selected tasks. The server also chooses clients who can take part in the FL process and updates the local process from each client.

**Local Training and update:** When the process is initialized with configuration and learning parameters, a new model is initialized by the server, let it be $w_G^0$, and model data transmitted to client during training process, each client does local training using their existing dataset $D_c$, and updates training data $w_c$to minimize loss function $F\ (w_c)$, optimization process is calculated as follow:

$$W_c^* = argmin_c F(w_c), p\varepsilon C \qquad (1)$$

Where $W_c^*$the optimal training data is for client c, $F\ (w_c)$ is the loss function associated with the training data $w_c$, C is the set of all clients taking part in FL process. Loss function varies in different FL processes, and for each process client c updates their computed weights $w_c$ in server for aggregation.

**Model aggregation:** After local training and model updates by clients, weights, gradients are sent to the server; models are aggregated at the server and create a new global model, in the following way:

$$w_G = \frac{\sum_{c=1}^{|C|} |D_c| w_c}{\sum_{c \in C} |D_c|} \qquad (2)$$

Where, $w_G$ is the updated global model,$|D_c|$ is size of dataset $D_c$ of client c, and $w_c$ is updated model of client c.

By the end of aggregation process, server dispense updated global weights $w_g$ to all the clients. Local models are optimized at their stages. This process is repeated until the optimal global model is obtained for the selected task to achieve the desired accuracy.

### 3.2. Stochastic Gradient Descent (SGD)

The high performance of deep learning mostly relies on the use of stochastic gradient descent for optimization. Several improvements have been made to adapt the model structure to work better with SGD-based optimization techniques [28]. Hence, it is suggested that SGD-based algorithms be used for federated optimization; here, a single batch gradient calculation occurs per communication round. SGD operates by iteratively updating the weights and biases of local models on IoT devices to reduce the loss with respect to a training dataset. Algorithms 1 describe the entire SGD process.

***Algorithm 1:*** *Stochastic Gradient Descent (SGD)*

```
1: Input: learning rate$\eta_t > 0$
2: Initialize:$w_0 \in R^d, t = 0$
3: while stopping_criteria_not_met do:
4:     Sample $\xi_i \sim P$ with $i \in \{1, 2, ..., n\}$
5:     $\zeta_i = \nabla f_i(w_t; \xi_i)$
6:     $w^+ = w - \eta t \cdot \zeta i$
7:     $t = t + 1$
8: end while
```

Similarly, instead of computing the gradient of the whole dataset at once, it computes the gradient using a randomly selected subset of local data samples (mini-batch) at each epoch as detailed in Algorithm 2 [28]. This stochastic data sampling aids SGD in combining quickly and managing large datasets efficiently. Despite the computational effectiveness of this technique, several training rounds are needed to give satisfactory models.

***Algorithm 2:*** *Stochastic Gradient Descent (SGD)*

```
1: Input: learning rate$\eta_t > 0$
2: Initialize:$w_0 \in R^d, t = 0$
3: while stopping_criteria_not_met do:
4:     $z_t \leftarrow \nabla f \mathcal{L}_t (w_t) = \frac{1}{|\mathcal{L}_t|} \sum_{\mathcal{L}_t} \nabla f_i(w_t; \xi_i)$
5:     $w_{t+1} = w_t - \eta_t \cdot z_t$
6:     $t = t + 1$
7: end while
```

### 3.3. Differential Privacy (DP)

The DP measures the impact of individual data on the output of a learning algorithm. This mechanism is considered as ϵ-differentially private if, when there is one difference in comparison of two datasets with a single entry, the chance of having specific output remains the same. Equation 3 mathematically defines DP [29].

$$\Pr[M(D) \in S] \leq e^{\varepsilon} \times \Pr[M(D') \in S] + \delta \qquad (3)$$

Where, M is learning algorithm, which introduces randomness in its outputs to obscure the presence or absence of any single individual's data in the dataset.
The output probability distribution$\Pr[M(D) \in S]$ indicates output of algorithm M on D lies in set S. D and D' are neighboring datasets, S is output space, and ε and δ are privacy parameters.

For query function$f$, the sensitivity $\nabla f$ is calculated as follow:

$$\nabla f = \max \| f(D) - f(D') \| \quad (4)$$

For the small positive values of ε and δ, Equation (3) suggests that the results of M will be almost untouched in distribution if one datapoint is changed in the dataset. The merit of the DP mechanism is that it is purely quantitative. Hence, it produces numerical proof of the portion of privacy that can be foreseen in the stochastic sense, where inferior ε and δ indicate that the mechanism preserves better privacy.

### 3.4. Renyi Differential Privacy (RDP)

RDP is more advanced with detailed analysis of privacy protection with the help of introducing a set of privacy parameters indexed by α. It is specifically helpful in the assessment of privacy mechanisms, such as the Gaussian mechanism often used in FL. The advantage of RDP over DP is its capability for tighter measures of privacy loss [30]. We can calculate RDP using Equation (5):

$$\mathrm{RDP}(\alpha, q, \sigma) = \frac{1}{\alpha - 1} \ln\left( \sum_{k=0}^{\alpha} \binom{\alpha}{k} (1-q)^{\alpha - k} q^{k} e^{\left(\frac{k^2 - k}{2\,\sigma^2}\right)} \right) \quad (5)$$

Where, α is Renyi parameter, indexing family of privacy parameters, q is sampling ratio, and σ is standard deviation of Gaussian noise added to mechanism.

RDP has been proven to be a better way of expressing proof of privacy-preserving algorithms and for the composition of heterogeneous mechanisms, offering a more convenient and quantitatively accurate method of standalone differentially private mechanisms. Most importantly, RDP allows for merging the intuitive and appealing concept of a privacy budget with the help of advanced composition theorems.

### 3.5. Overview of the System

In this paper, we concentrated on improving the security and privacy of the IoT environment, using FL settings and introducing Gaussian noise with the help of RDP techniques to further enhance privacy concerns in FL settings. In this approach, we added Gaussian noise to the weights of each trained model at IoT edge devices to ensure the privacy of sensitive information. Figure 1 demonstrates an overview of the proposed approach in detail.

The level of noise added to the weights is dynamically adjusted with a parameter called Sigma using RDP. This helps us to balance between privacy and model weight usability. Moreover, we also converted sigma values from RDP to DP to ensure compatibility with present privacy frameworks. Based on our evaluation, this approach showcases the effectiveness of FL in securing IoT data privacy and minimizing data utility. It can be considered a promising solution for privacy-preserving IoT systems.

As shown in Figure 1, each client maintains its local dataset, which may possess sensitive information such as sensor readings, user behavior data, or environmental variables. During training, these clients collaboratively train a global model by exchanging model updates and ensuring differential privacy. The processes take place as follows: First, a single global ML model is initialized by the server and transferred to all the clients in the network. Second, local models are trained and tested on each client using the existing available data. Third, a certain amount of Gaussian noise is added to local models to avoid privacy leakages, and the weights of local models are calculated. Further, weights are sent to the server for aggregation and global model creation. For this study, we considered a CNN multi-classifier model, which is explained in detail in section 4.1.

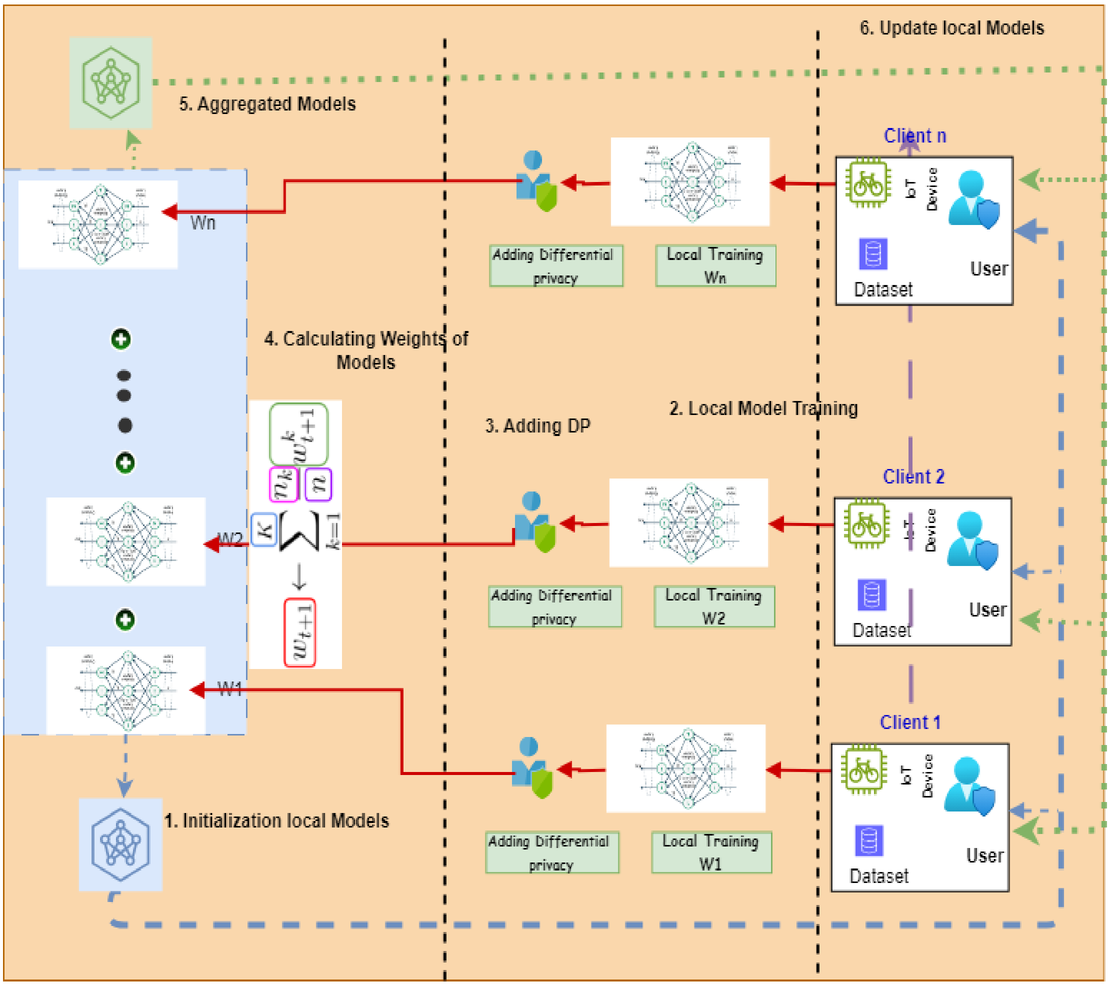

Figure 1: Architecture of FL settings for IoT environment.

## 4. Experimental Setup

In this section, we explain the experiment conducted using the proposed approach to analyze its feasibility. First, it describes experiment settings, dataset descriptions, and the environmental setup. We implemented this experiment using the Google Colab platform, including PyTorch, Scikit-learn, Numpy libraries, and CUDA for GPU acceleration. A Toshiba machine with 8GB RAM, a 750GB HDD, and an AMD FX-8800P 2.1GHz processor were the hardware parts for the experimental study.

### 4.1. Convolution Neural Networks (CNN)

For this experiment, we utilized a CNN algorithm for local and global models. CNN is one of the discriminative deep learning algorithms broadly used for handling massive datasets, having a hierarchical pattern extraction. To use input data structure, CNN networks use local connections and weights in place of fully connected networks [31]. Hyperparameter optimization of CNN is an essential part of adjustment in FL settings. This technique affects the structure and effectiveness of CNN and further improves the classification accuracy of the model. Categorized in two parts, with each category affecting the model's design and training efficiency. The first category includes obtaining a number of frozen layers and dropout ratio to avoid overfitting. Optimizing the learningrate is required for convergence during training. The second category is for improving model performance and reducing training time. The number of epochs, early-stopping, and batch size are critical in optimizing model learning capacity and computational efficiency [32].

### 4.2. Convolution Neural Networks Design

The basic architecture of the model for this study consists of two convolutional layers backed by fully connected layers. Convolutional layers employ filters to input data and to extract relevant features through some convolution operations. The output of each convolution layer is passed through an activation function; for this model, we use the hyperbolic tangent (tanh) function to introduce non-linearity. Pooling layers are placed between convolution layers for down-sampling feature maps to reduce computational complexity while obtaining important features. Max-pooling is used to choose a maximum value from a window of the input and effectively limit the spatial dimensions of data.

At the end of the architecture, after passing through convolution layers and pooling operations, feature maps are flattened and passed to fully connected layers. These layers perform linear transformations on input data, permitting networks to learn complex relationships between features. A pictorial representation of the model is shown in Fig 2.

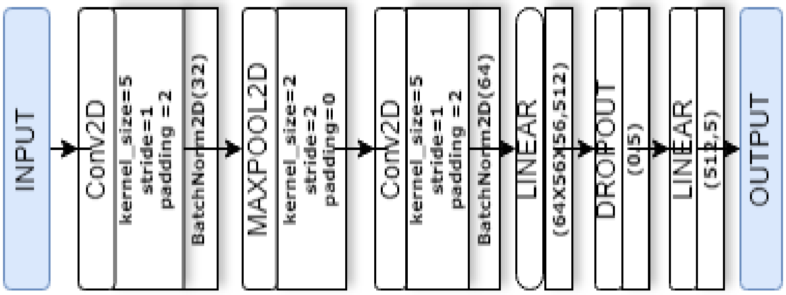

Figure 2: CNN Architecture

**Hyperbolic tangent (tanh) function:** is a widely used activation in neural networks to introduce non-linearity and enable the model to learn complex patterns from the input data.

This squashes input values between -1 and 1 using the Equation 6 [31].

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}(6)$$

Where, x is input for the function.

## 4.3. Dataset Description

In this section, we describe the two datasets used to evaluate the performance of the proposed approach in FL Settings: Car Hacking and MNIST.

### 4.3.1. CAR-Hacking Dataset

This dataset is generated in a simulated environment for cyber-attacks on control area network protocol for IoV; it consists of two main features, CAN-ID and 8-bytes of the data field (DATA[0]-DATA[7]). The dataset contains four types of cyber-attacks: DoS, which is used for overloading the network with a flood of traffic; Fuzzy for malfunctioning of the system; gear spoofing, which exploits gear display messages; and RPM spoofing, which alters engine readings [34]. We have made some series of transformations and optimizations on the dataset to make it compatible with CNN models and enhance performance [33]. First, scaled numerical features to a standard range of (0,1) with the help of a quantile transformer. This normalization phase assures the feature's consistency and helps smoother convergence during model training. Further, features are modified and scaled in the range of (0,255) to map them for pixel intensities, which is suitable for image representation.

After feature scaling, the dataset was converted into image format, with each row representing a single image. We generated images for each class with the dataset to ensure a balanced representation of classes in the training set. Additionally, the dataset is separated into two parts: train-set and test-set, and class balance is maintained across both sets to ensure evaluation of model performance. For CNN architecture, we standardized the input dimensions of the generated images and resized them to a uniform size of 224x224 pixels, which is critical for CNN model compatibility and facilitating the learning of meaningful patterns to improve classification accuracy. These preprocessing steps are demonstrated in Figure 3. Following these transformations, the preprocessed data was distributed among clients in an FL setup to train local models. For the distribution of data among clients, an IID pattern is followed for this dataset; each client poses a subset of random and homogeneous data.

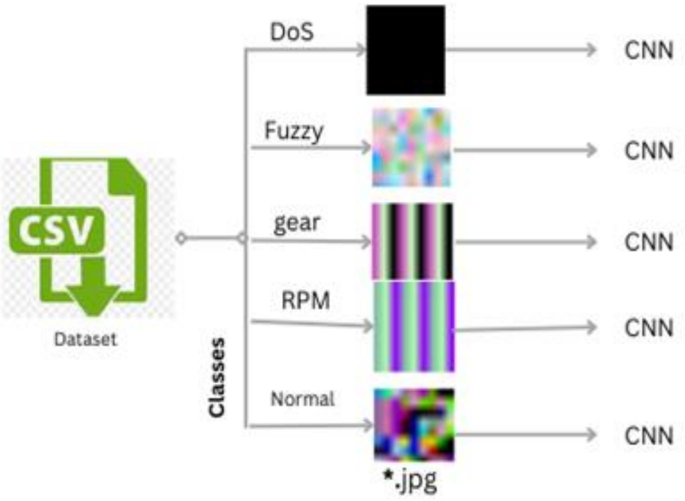

Figure 3: Data Preprocessing for Car-hacking dataset

### 4.3.2. MNIST Dataset

This is a broadly utilized dataset in ML that contains a collection of grayscale images of handwritten digits (0-9) in 28x28 pixels. This dataset is labeled to make it suitable for various classification tasks. Here, each image belongs to a handwritten digit [35].

For this experimental study, we used this dataset to evaluate proposed FL settings with non-cybersecurity environments. This logic is implemented to create a real-time IoT-based scenario where each IoT device has specific data. The number of shards was determined based on the number of clients in FL settings. Subsequently, all images within each shard were stored with their corresponding labels to ensure different representations of classes. Clients possessed various classes, demonstrating real-world scenarios.

## 4.4. Model Training

The training process for local and global models is explained in Algorithm 3 and parameters are described in Table 1. Both models are trained iteratively to leverage the collaborative learning of selected clients in FL settings while preserving their privacy. During the training process, each client’s data is split into batches, followed by gradient descent optimization to update the model’s parameters. To improve privacy, Gaussian noise is added to weights of gradients of model parameters; we have also used the gradient clipping technique to control the magnitude of noise. After completing the training process for a chosen number of epochs, model weights are updated based on gradients.

The global model aggregates the weights from all the clients to learn the knowledge. This aggregation process is assured with a privacy-preserving technique employing DP at the local training phase.

Table 1: Parameters description of Algorithm1

| Ser # | Parameter | Meaning |
|---|---|---|
| 1 | t | Iteration number |
| 2 | i | Initial global model parameters |
| 3 | lr | Learning rate |
| 4 | $w_0$ | Initial global model parameters |
| 5 | $w_i^t$ | Local model parameters for client $i$ at epoch $t$ |
| 6 | bs | Batch-size |
| 7 | $\nabla J(w_i^t)$ | Gradients of  loss function with respect to the local model parameters |
| 8 | $\theta$ | Gaussian noise |
| 9 | $w^t$ | Global model parameters at epoch $t$ |

## 5. Results and Discussion

In this section, we analyze the performance of our proposed approach utilizing two benchmark datasets of cybersecurity and image classifications as explained in section 4.3, in two approaches, both datasets are analysed with and without employing DP at local training phases. Furthermore, we compared the outcomes of our approach with the state-of-the-art methods in Table 2.

The performance evaluation of proposed FL approach utilizing the CNN algorithm and employed DP to preserve privacy of clients' data has shown a promising results. In this setup, the data is first converted to colour images as explained in section 4.3.1and than divided among clients.

***Algorithm 3:*** *FL Training Process*

```
Initialize global model parameters: $w_0$
Set hyperparameters: clients, epochs, lr, bs
for t in range (ecpochs) do
    local _weights : ← [ ]
    for i in range (clients) do
        for mini_batch j in client_dataset[i] do
            Calculate gradients: $\nabla J\left(w_i^t\right) = \frac{1}{|mini-batch|} \sum_{x \in mini-batch} \nabla J\left(w_i^t, x\right)$
            Add Gaussian noise to gradients: $noisy - grad = \nabla J\left(w_i^t\right) + \theta$
            Apply gradient clipping: $\text{clipped} - \text{grad} = \text{clip(noise\_grad)}$
            Update local model weights: $w_i^t = w_i^t - lr \times \text{clipped} - \text{grad}$
        End for
        Add local model weights to local-weights list: local_weights.append($w_i^t$)
    End for
    Aggregate local models weights to get global model weights: $w_i^t = \frac{1}{clients} \sum_{i=1}^{clients} w_i^t$
    Update global model: $w_{t+1} = w^t$
    Evaluate global model performance:  accuracy, loss
End for
```

In the local training carried out for clients, the models have shown a consistent decrease in loss over epochs with both datasets across all clients in the FL setup. This suggests adequate model learning progress and convergence over time, as demonstrated in Figure 4 and Figure 5.

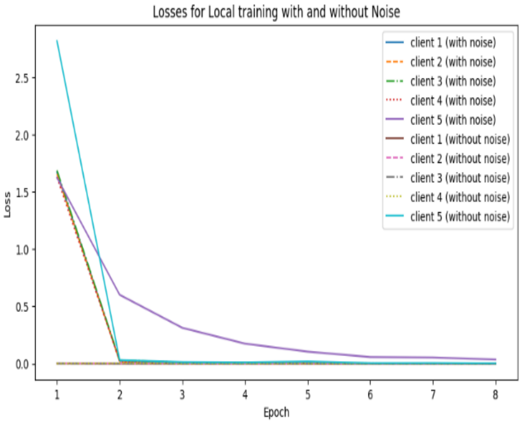

Figure 4. Loss decrease during local training of Car-Hacking over epochs

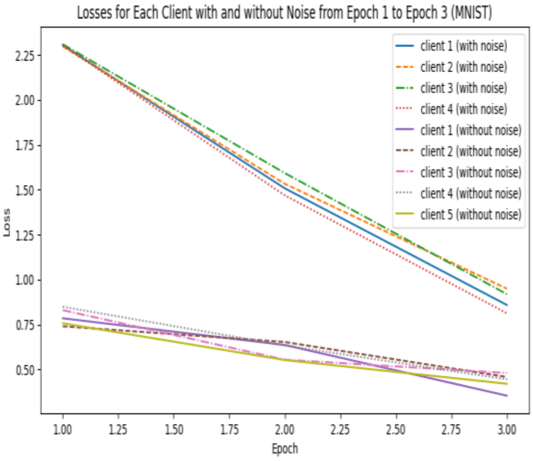

Figure 5. Loss decrease during local training of MNIST over epochs

In the global model training aggregated from clients' weights at server, from Table 2, the performance of the proposed approach on the Car-Hacking dataset without employing DP mechanisms shows an accuracy of 93%, demonstrating its ability to classify images at a high level. Similarly, considering precision, recall, and f1-score rate, achieving 0.8608, 0.9276, and 0.8929, respectively, and the average loss decreased from 2.3946 up to 0.0041 same has been shown in Figure 7 indicates the effectiveness of FL Setup to preserve privacy and security of IoT environment [36].

On the other hand, when we employ the DP mechanisms with parameters values of differential privacy budget ε=1.0, probability of privacy breach δ= 1e-5, learningrate = 0.01, and dynamically calculated collaborative sigma value using RDP σ = 0.9886978, to further enhance privacy concerns in suggested approach. From evaluation values in Table 2, the model's performance has demonstrated a slight decrease in accuracy rate. Yet, despite this, the model obtained a relatively high accuracy of 91%, displaying its stability to noise and its capability to perform well in existing privacy-preserving measure mechanisms. Considering other used evaluation metrics rate for the experiment, precision, recall, and f1-score achieved promising values of 0.9361, 0.8979, and 0.914, respectively. Similarly, the loss has also decreased from 2.9700 up to 0.1109, overall indicating their effectiveness in detecting true positive samples under noise conditions introduced by the DP mechanisms.

Furthermore, we evaluated this experimental study using the MNIST dataset to illustrate its performance with the normal dataset. We considered five clients. Following the training process global model with multiple epochs without employing the DP mechanisms, analyzing is done on a centralized testset. Considering the accuracy rate, model performances consistently improve accuracy over epochs, obtaining an accuracy of 98.92% on testset after 15 epochs. At the same time, the model is also evaluated using precision, recall, and f1-score metrics and achieved 98.51%, 98.73%, and 98.61%, respectively, after 15 epochs; the average loss also decreased progressively from 0.7706 up to 0.0570 same is plotted on Figure 9, demonstrating robustness and balanced classification capabilities.

Similarly, when we employed the DP mechanisms with parameters values of DP budget ε=1.0, probability of privacy breach δ= 1e-2, learningrate = 0.02, and collaborative sigma value using RDP was calculated to be σ = 1.2074424, to enhance the security and privacy issues in suggested approach. Based on the evaluation value from Table 2, the model's performance decreased in accuracy rate. The model obtained a high accuracy of 98.28%, demonstrating its capability to handle noise addition and perform well in the presence of privacy-preserving measures. Considering other evaluation metrics, the model achieved promising results of precision 98.38%, recall 98.4%, and F1-score 98.39% in the final epoch; similarly, loss decreased from 0.6638 up to 0.0590, overall indicating balanced performance across all classes.

In the analysis conducted on the CAR-Hacking dataset, the proposed approach obtained promising results in terms of evaluation metrics, as shown in Figure 6 and Figure 7, illustrating its ability to classify attacks and, at the same time, preserve privacy and security of IoT environments. Applying the DP mechanisms has made a slight change in the evaluation metrics rate but improved preserving privacy issues in the FL setup, demonstrating the model's strength to noise and its effectiveness in performing well under privacy-preserving mechanisms. Similarly, the analysis carried out on the MNIST dataset, as shown in Figure 8 and Figure 9, further proved the robustness of the proposed approach, with consistent enhancement in evaluation metrics over epochs and balanced classification capabilities. At the same time, applying the DP mechanism with this dataset showed a small quantity of change in evaluation metrics, in contrast to improving privacy issues in the FL setup; this indicates good performances of the model and outlines the model's ability to maintain privacy-preserving measures efficiently.
Overall, this experimental study outlines the effectiveness and resilience of the suggested FL approach in Car-Hacking and MNIST datasets, illustrating its potential for real-world scenarios in IoT systems.

Table 2. Comparison of Proposed approach with latest state-of-the-art approaches

| **Algorithms** | **Datasets** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1 (%)** | **Reference** |
|---|---|---|---|---|---|---|
| FL | CAR-H | 93 | 86.08 | 92.76 | 89.29 | |
| | MNIST | 98.92 | 98.51 | 98.73 | 98.619 | |
| FL-DP | CAR-H | 91 | 93.61 | 89.79 | 91.4 | |
| | MNIST | 98.28 | 98.38 | 98.4 | 98.38 | |
| FL | Custom | 95.6 | --- | --- | --- | [19] |
| FL | KDD, NSL-KDD, UNSW | 95.5,9379,95.6 | 85.16 (PPV) | 84.98(TPR) | --- | [37] |
| FL | NSL-KDD | 98.73 | 85.35 | 73.49 | 78.98 | [38] |
| FL | Edge-IIoTset | 91.87 | --- | --- | --- | [39] |

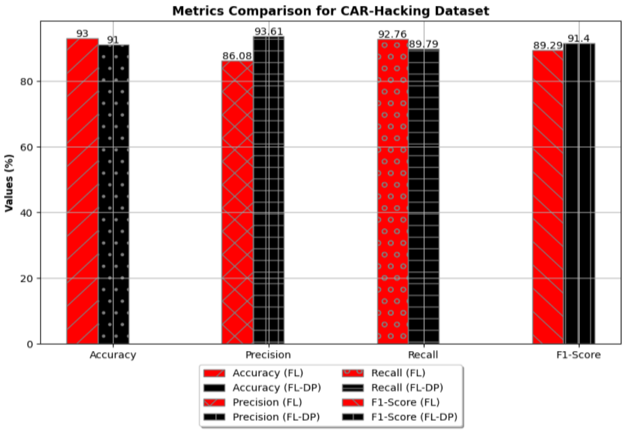

Figure6 : Evaluation Metrics on Car-Hacking

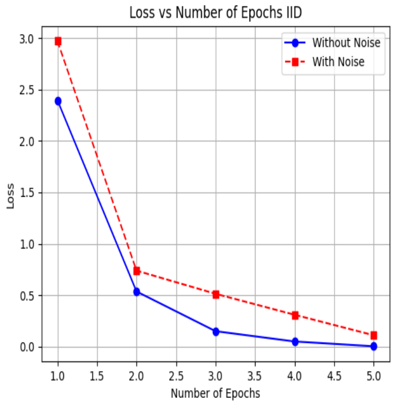

Figure7: Loss changes during training of Car-Hacking

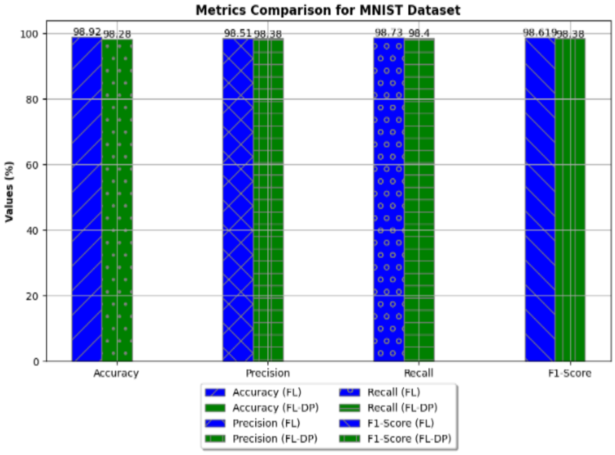

Figure8 : Evaluation Metrics on MNIST

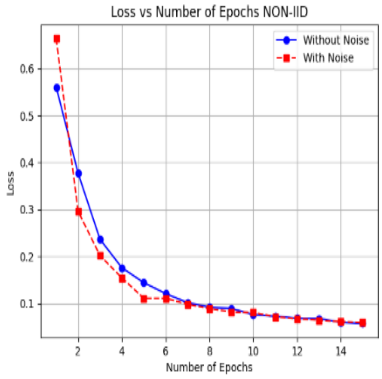

Figure 9: Loss changes during training of MNIST

## 6. CONCLUSION

In this experimental study, we analyzed the efficiency of FL in securing the IoT environment, particularly employing the DP mechanisms to improve security and privacy in the FL framework. For the broader evaluation of the suggested approach, datasets from both regular and cybersecurity, domains are considered to demonstrate real-world scenarios. Results shown by evaluation metrics (accuracy, precision, recall, f1-score) for both scenarios in terms of classifications and progressively decrease in loss during training indicate promising outcomes. Incorporating Differential Privacy mechanisms demonstrates enhancement in privacy preservation of the FL setup without notable compromises in model performances. Particularly, the FL setup achieved an accuracy of 98.92% with a well-defined MNIST dataset, and after integrating with the DP mechanisms, the accuracy slightly decreased to 98.2%. Similarly, after some transformation of the CAR-Hacking dataset into image format, the proposed approach achieved an accuracy of 93% in classification in the normal setup; while implementing the DP mechanisms, the accuracy changed slightly to 91%. These findings outline the robustness of FL incorporated with the DP mechanisms in reducing security and privacy issues within IoT environments and provide insights for real-world deployment in various IoT applications.

Regarding future research, firstly, focus on the optimization of DP parameters with more robustness and scalable FL techniques, such as vertical FL algorithms that divide features across various devices. Furthermore, analysing the effectiveness of the proposed approach in real-world IoT applications gives more valuable insights into the effectiveness and scalability of various use cases.

## CONFLICTS OF INTEREST

The authors declare no conflict of interest.

## REFERENCES

[1] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," J Big Data, vol. 6, no. 1, p. 111, Dec. 2019, doi: 10.1186/s40537-019-0268-2.

[2] A. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," IEEE Internet Things J., vol. 9, no. 12, pp. 9444–9466, Jun. 2022, doi: 10.1109/JIOT.2021.3126811.

[3] M. Bouzidi, N. Gupta, F. A. Cheikh, A. Shalaginov, and M. Derawi, "A Novel Architectural Framework on IoT Ecosystem, Security Aspects and Mechanisms: A Comprehensive Survey," IEEE Access, vol. 10, pp. 101362–101384, 2022, doi: 10.1109/ACCESS.2022.3207472.

[4] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," IEEE Commun. Surv. Tutorials, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.1109/COMST.2020.298829

[5] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine Learning in IoT Security: Current Solutions and Future Challenges," IEEE Commun. Surv. Tutorials, vol. 22, no. 3, pp. 1686–1721, 2020, doi: 10.1109/COMST.2020.2986444.

[6] L. Cui et al., "Security and Privacy-Enhanced Federated Learning for Anomaly Detection in IoT Infrastructures," IEEE Trans. Ind. Inf., vol. 18, no. 5, pp. 3492–3500, May 2022, doi: 10.1109/TII.2021.3107783.

[7] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," Int. J. Mach. Learn. & Cyber., vol. 14, no. 2, pp. 513–535, Feb. 2023, doi: 10.1007/s13042-022-01647-y.

[8] P. R. Silva, J. Vinagre, and J. Gama, "Towards federated learning: An overview of methods and applications," WIREs Data Min &Knowl, vol. 13, no. 2, p. e1486, Mar. 2023, doi: 10.1002/widm.1486.

[9] T. Zhang, L. Gao, C. He, M. Zhang, B. Krishnamachari, and S. Avestimehr, "Federated Learning for Internet of Things: Applications, Challenges, and Opportunities." arXiv, Apr. 05, 2022. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/abs/2111.07494.

[10] L. Zhu, Z. Liu, and S. Han, "Deep Leakage from Gradients." arXiv, Dec. 19, 2019. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/abs/1906.08935.

[11] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning." arXiv, Sep. 14, 2017. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/abs/1702.07464

[12] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the GDPR perspective," Computers & Security, vol. 110, p. 102402, Nov. 2021, doi: 10.1016/j.cose.2021.102402.

[13] K. Wei et al., "Federated Learning with Differential Privacy: Algorithms and Performance Analysis," 2019, doi: 10.48550/ARXIV.1911.00222.

[14] Rajmohan, T., Nguyen, P.H. & Ferry, N. A decade of research on patterns and architectures for IoT security. Cybersecurity 5, 2 (2022). https://doi.org/10.1186/s42400-021-00104-7.

[15] Abiodun, O.I., Abiodun, E.O., Alawida, M. et al. A Review on the Security of the Internet of Things: Challenges and Solutions. Wireless Pers Commun 119, 2603–2637 (2021). https://doi.org/10.1007/s11277-021-08348-9.

[16] Kuzlu, M., Fair, C. & Guler, O. Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity. Discov Internet Things 1, 7 (2021). https://doi.org/10.1007/s43926-020-00001-4.

[17] A. Jamalipour and S. Murali, "A Taxonomy of Machine-Learning-Based Intrusion Detection Systems for the Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9444-9466, 15 June15, 2022, doi: 10.1109/JIOT.2021.3126811.

[18] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," 2016, doi: 10.48550/ARXIV.1602.05629.

[19] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "D\"IoT: A Federated Self-learning Anomaly Detection System for IoT." arXiv, May 10, 2019. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/abs/1804.07474.

[20] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of Things Intrusion Detection: Centralized, On-Device, or Federated Learning?," IEEE Network, vol. 34, no. 6, pp. 310–317, Nov. 2020, doi: 10.1109/MNET.011.2000286.

[21] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "Federated Learning-Based Cognitive Detection of Jamming Attack in Flying Ad-Hoc Network," IEEE Access, vol. 8, pp. 4338–4350, 2020, doi: 10.1109/ACCESS.2019.2962873.

[22] J. Zhang, B. Chen, S. Yu, and H. Deng, "PEFL: A Privacy-Enhanced Federated Learning Scheme for Big Data Analytics," in 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA: IEEE, Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9014272.

[23] C. Jost, H. Lam, A. Maximov, and B. Smeets, "Encryption performance improvements of the Paillier cryptosystem," Cryptology ePrint Archive, 2015.

[24] [1] H. Fang and Q. Qian, "Privacy Preserving Machine Learning with Homomorphic Encryption and Federated Learning," Future Internet, vol. 13, no. 4, p. 94, Apr. 2021, doi: 10.3390/fi13040094.

[25] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How To Backdoor Federated Learning." arXiv, Aug. 06, 2019. Accessed: Apr. 19, 2024. [Online]. Available: http://arxiv.org/abs/1807.00459.

[26] T. R. Jeter and M. T. Thai, "Privacy Analysis of Federated Learning via Dishonest Servers," in 2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), New York, NY, USA: IEEE, May 2023, pp. 24–29. doi: 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00015.

[27] K. Nakayama and G. Jeno, "Federated Learning with Python: Design and implement a federated learning system and develop applications using existing frameworks.", Packt Publishing, 2022. [Online]. Available: https://books.google.co.in/books?id=Zi2WEAAAQBAJ.

[28] Herbert Robbins. Sutton Monro. "A Stochastic Approximation Method." Ann. Math. Statist. 22 (3) 400 - 407, September, 1951. https://doi.org/10.1214/aoms/1177729586.

[29] Z. Shen and T. Zhong, "Analysis of Application Examples of Differential Privacy in Deep Learning," Computational Intelligence and Neuroscience, vol. 2021, Art. no. 4244040, 2021. doi: 10.1155/2021/4244040.

[30] I. Mironov, "Rényi Differential Privacy," in 2017 IEEE 30th Computer Security Foundations Symposium (CSF), Santa Barbara, CA: IEEE, Aug. 2017, pp. 263–275. doi: 10.1109/CSF.2017.11.

[31] [T. Mitchell, "Machine Learning.", McGraw-Hill Education, 1997.

[32] L. Chuan and F. Quanyuan, "The Standard Particle Swarm Optimization Algorithm Convergence Analysis and Parameter Selection," in Third International Conference on Natural Computation (ICNC 2007), Haikou, China: IEEE, 2007, pp. 823–826. doi: 10.1109/ICNC.2007.746.

[33] Mohamad T. Sultan, Hesham El Sayed, and Manzoor Ahmed Khan, "An intrusion detection mechanism for MANETs based on deep learning artificial neural networks (ANNs)," International Journal of Computer Networks & Communications (IJCNC), vol. 15, no. 1, pp. 1-15, Jan. 2023, doi: 10.5121/ijcnc.2023.15101.

[34] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car Hacking: Attack Defense Challenge 2020 Dataset.", IEEE Dataport, Feb. 03, 2021. doi: 10.21227/QVR7-N418.

[35] Deng, L. "The mnist database of handwritten digit images for machine learning research." IEEE Signal Processing Magazine, (2012). 29(6), 141-142.

[36] Beaton Kapito, Mwawi Nyirenda, and Hyunsung Kim, "Privacy-preserving machine authenticated key agreement for Internet of Things," International Journal of Computer Networks & Communications (IJCNC), vol. 13, no. 2, pp. 99-114, Mar. 2021, doi: 10.5121/ijcnc.2021.13206.

[37] T. V. Khoa et al., “Collaborative Learning Model for Cyberattack Detection Systems in IoT Industry 4.0,” in 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea (South): IEEE, May 2020, pp. 1–6. doi: 10.1109/WCNC45663.2020.9120761.

[38] D. Man, F. Zeng, W. Yang, M. Yu, J. Lv, and Y. Wang, “Intelligent Intrusion Detection Based on Federated Learning for Edge-Assisted Internet of Things,” Security and Communication Networks, vol. 2021, pp. 1–11, Oct. 2021, doi: 10.1155/2021/9361348.

[39] M. M. Rashid, S. U. Khan, F. Eusufzai, Md. A. Redwan, S. R. Sabuj, and M. Elsharief, “A Federated Learning-Based Approach for Improving Intrusion Detection in Industrial Internet of Things Networks,” Network, vol. 3, no. 1, pp. 158–179, Jan. 2023, doi: 10.3390/network3010008.

## AUTHORS

Aziz Ullah Karimyis currently pursuing his Ph.D. in Machine Learning andIoT security at the University College of Engineering, Science and Technology, Hyderabad (JNTUH). He obtained his Bachelor's degree in Electronics and Communication Engineering from Visvesvaraya Technological University, India, in 2016, followed by a Master's degree in Embedded Systems from Jawaharlal Nehru Technological University, Hyderabad, in 2021. His research focuses on advancing the applications of Machine Learning in cybersecurity, specifically in securing IoT environments.

Prof. (Dr.) P.Chandrasekhar Reddy is a Senior Professor at JNTUH University College of Engineering, Science and Technology, Hyderabad. He holds a Ph.D. degree, along with Master of Technology (M.Tech) and Master of Engineering (M.E), as well as a Bachelor's degree in Bachelor of Engineering (B.E). His primary areas of interest include Wireless Communication, 5G, Image Processing, Electrical Vehicles, and the Internet of Things (IoT). He has over 30 years of experience in the academic sphere, and his career has been distinguished by mentoring over 35 Ph. D. researchers and contributing approximately 150 scholarly papers to the scientific community.