Federated Learning Under Attack

Exposing Vulnerabilities Through Data Poisoning Attacks in Computer Networks

Nowroozi, Ehsan; Haider, Imran; Taheri, Rahim; Conti, Mauro

DOI
10.1109/TNSM.2025.3525554

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Network and Service Management

Citation (APA)
Nowroozi, E., Haider, I., Taheri, R., & Conti, M. (2025). Federated Learning Under Attack: Exposing Vulnerabilities Through Data Poisoning Attacks in Computer Networks. *IEEE Transactions on Network and Service Management*, *22*(1), 822 - 831. https://doi.org/10.1109/TNSM.2025.3525554

Federated Learning Under Attack: Exposing Vulnerabilities Through Data Poisoning Attacks in Computer Networks

Ehsan Nowroozi, *Senior Member, IEEE*, Imran Haider, *Member, IEEE*, Rahim Taheri, *Senior Member, IEEE*, and Mauro Conti, *Fellow, IEEE*

***Abstract*—Federated Learning is an approach that enables multiple devices to collectively train a shared model without sharing raw data, thereby preserving data privacy. However, federated learning systems are vulnerable to data-poisoning attacks during the training and updating stages. Three data-poisoning attacks—label flipping, feature poisoning, and VagueGAN—are tested on FL models across one out of ten clients using the CIC and UNSW datasets. For label flipping, we randomly modify labels of benign data; for feature poisoning, we alter highly influential features identified by the Random Forest technique; and for VagueGAN, we generate adversarial examples using Generative Adversarial Networks. Adversarial samples constitute a small portion of each dataset. In this study, we vary the percentages by which adversaries can modify datasets to observe their impact on the Client and Server sides. Experimental findings indicate that label flipping and VagueGAN attacks do not significantly affect server accuracy, as they are easily detectable by the Server. In contrast, feature poisoning attacks subtly undermine model performance while maintaining high accuracy and attack success rates, highlighting their subtlety and effectiveness. Therefore, feature poisoning attacks manipulate the server without causing a significant decrease in model accuracy, underscoring the vulnerability of federated learning systems to such sophisticated attacks. To mitigate these vulnerabilities, we explore a recent defensive approach known as Random Deep Feature Selection, which randomizes server features with varying sizes (e.g., 50 and 400) during training. This strategy has proven highly effective in minimizing the impact of such attacks, particularly on feature poisoning.**

***Index Terms*—Federated learning, causative attacks, adversarial machine learning, corrupted training sets, cybersecurity, data-poisoning.**

I. Introduction

IN THIS modern era of technology, in which other scientific disciplines are advancing swiftly, Federated learning (FL) is also keeping pace and progressing rapidly. FL is a form of machine learning (ML) that enables the training of a model using data from data sources without centralizing the data. Unlike conventional methods, which involve relaying data to a central server, FL performs the learning process directly on each device individually. Local data is employed to update the local models, and these updates are then combined to create a unified global model on the server. This decentralized method not only maintains data confidentiality but also facilitates the interaction and sharing of information across devices. FL has the potential to be highly effective in areas such as mobile and IoT devices, where data privacy is vital [1], [2].

FL has demonstrated significant potential in Intrusion Detection Systems (IDSs). Within this framework, participating devices, including phones and desktop computers, pool their individual intrusion detection models to construct a global model while simultaneously ensuring the confidentiality of their sensitive data [3], [4], [5]. By facilitating interactive model training while maintaining data decentralization and security on individual devices, FL-based IDS presents a desirable framework [6], [7]. Nevertheless, similar to other ML models, FL-based IDS models are vulnerable to adversarial attacks, such as poisoning attacks, in which malevolent data is injected during training to influence the behavior of the model. These attacks have the potential to compromise the detection systems and bring significant security risks. Consequently, it is essential to provide defense systems against such attacks [8], [9], [10].

The objective of this study is to investigate the effectiveness of data poisoning attacks in the computer network realm, as they are simple to set up, yet challenging to detect. Label flipping (LF), feature poisoning (FP), and a novel attack technique against FL called VagueGAN, and use a unique technique to apply them. In LF, we randomly altered the labels of benign data before training the model on the altered data. For FP, we randomly altered the highly contributing features identified

Received 13 July 2024; accepted 11 October 2024. Date of publication 3 January 2025; date of current version 14 March 2025. The associate editor coordinating the review of this article and approving it for publication was K. Xue. *(Corresponding author: Ehsan Nowroozi.)*

Ehsan Nowroozi is with the Centre for Sustainable Cyber Security (CS2), University of Greenwich, SE10 9LS London, U.K. (e-mail: e.nowroozi@greenwich.ac.uk).

Imran Haider is with the Department of Natural Engineering and Sciences, Bahcesehir University, 34349 Istanbul, Türkiye (e-mail: imran.haider@bahcesehir.edu.tr).

Rahim Taheri is with the School of Computing, Faculty of Technology, University of Portsmouth, PO1 2UP Portsmouth, U.K. (e-mail: rahim.taheri@port.ac.uk).

Mauro Conti is with the Department of Mathematics, Security and Privacy Research Group, University of Padua, 35121 Padua, Italy, and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: mauro.conti@unipd.it).

Digital Object Identifier 10.1109/TNSM.2025.3525554

using the Random Forest method. For VagueGAN, an adversary simply executes a data-poisoning attack by employing a Generative Adversarial Network (GAN). This study used computer network datasets from the CIC and UNSW. However, the use of FL in cybersecurity, particularly with realistic datasets, such as CIC [11] and UNSW [12], is underexplored. These datasets contain a variety of characteristics and forms of attacks that mimic real-world settings, providing the opportunity to investigate the effects of data-poisoning attacks in a federated setting. Besides, they closely mimic actual operational networks, simulating authentic network environments with a wide range of applications, real user behaviors, and a wide variety of normal and malicious activities. To make the FL experimental setup sufficiently realistic, we considered 10 clients and applied the attacks to only one client. We generated adversarial samples using the two aforementioned methods, which were applied to a small percentage of the datasets. Next, we trained and evaluated the accuracy of the model on adversarial datasets. We report the results for both benign and manipulated datasets and find significant differences in the accuracy of the models across the datasets. The results of the study clearly revealed that the LF attack failed; however, the FP attack achieved successful outcomes, demonstrating its importance in deceiving a server.

The implementation code for this study is available at [13], and the RDFS methodology is available on [14].

A. Contributions

The following points highlight the contributions of this study:

- This study presents an innovative approach for performing data-poisoning attacks in the realm of computer networks, focusing on the impacts of three types of data-poisoning attacks: LF, FP, and VagueGAN. The implementation of an FP attack is novel, using the Random Forest method to locate and change highly contributing features that affect a classifier's decision. This targeted strategy improves our understanding of how such attacks affect the FL models.
- This study utilizes two well-known datasets, CIC and UNSW, both of which are highly relevant in computer networks. Experimentation with these datasets provides significant insights into the limitations of FL models.The originality lies in the choice and combination of these specific datasets for training and testing neural networks (NNs) in the context of data poisoning attacks, which leads to more realistic and practical findings.
- The FL setup with ten clients and a server explores how data poisoning threats affect the server's accuracy. By attacking one client, the study reveals the FL framework's weaknesses and mitigations. The novelty here lies in the analysis of the impact of the attack on the FL system's performance and the resilience of the model.
- In our investigation, we recorded the server accuracy and Attack Success Rate (ASR) across varied poisoning scenarios for both datasets. Our findings underscore the effectiveness of data poisoning attacks when implemented at different percentages, revealing their impact on the accuracy of the server. Moreover, recording losses for clients under different poisoning scenarios reveals the effects of these attacks.
- Develop a novel technique of Random Deep Feature Selection (RDFS) to enhance server security against FP attacks. This method demonstrates significant enhancements in model robustness and resilience to poisoning attacks across several datasets. The research study indicates that when the RDFS method is employed on the server, the ASR falls considerably when feature random sizes are between 50 and 400, demonstrating that the model is more robust.

B. Organization

We outline the rest of our paper as follows: In Section II, we provide an overview of related works that discuss data-poisoning attacks and defenses using different datasets. In Section III, we discuss the datasets and the network architecture of our BAU1 model. This section also includes the experimental setup and techniques that we applied to perform both data-poisoning attacks and RDFS defense strategy. Section IV presents the results of our experiments performed under different scenarios. In Section VI we summarize our study and its limitations.

II. Related Works

Data poisoning refers to the modification of data by adversaries using various types of attacks. This section provides an in-depth review of existing literature, organizing its contents into three sections: GAN, LF, and FP.

Label Flipping (LF): The authors in [15], [16], [17], [18], investigated extensive strategies for FL with a special focus on applying and tackling the issue of LF. These studies aim to strengthen FL models against malicious label alterations by using a mix of label noise assessment and adversarial training methods. Xu et al. [19], Zeng et al. [20], and Tsouvalas et al. [21] employed knowledge obtained from label noise analyses to detect and measure the influence of contaminated labels on the efficiency of models. These methods demonstrate subtle and cooperative security mechanisms, thereby providing a promising path toward minimizing label-poisoning attacks in federated environments. The development of defense tactics incorporates the objective of minimizing both conventional classification loss and a regularization term to prevent adversarial label modifications [22], [23]. These studies provide valuable insights into the subtle connection between label noise, adversarial training, and challenges posed by federated systems.

Feature Poisoning (FP): LF attacks are mostly concerned with the manipulation of labels, whereas FP attacks specifically target features present in the data. Several studies have investigated the vulnerability of FL models to FP. Raza et al. [24] employed different data poisoning techniques, including random LF and FP attacks. Furthermore, more sophisticated forms of data-poisoning attacks, such as GAN attacks [25], have been employed. In this form of attack, the

TABLE I
COMPARING PROPOSED METHODS IN THIS PAPER WITH OTHER RESEARCH, FL (FEDERATED LEANING), D (DEFENCE METHOD), LF (LABEL FLIPPING), FP (FEATURE POISONING), GAN (GENERATIVE ADVERSARIAL NETWORK)

Ref.	Technique	LF	FP	D	GAN
[15]–[17], [30]	FL+ LF+ noise assess	✓	–	✓	–
[19]–[21]	FL+ LF + noise assess	✓	–	–	–
[22], [23]	FL+ LF + minimizing loss	✓	–	✓	–
[24]	FL+ LF + FP	✓	✓	–	–
[25]	FP + GAN	–	✓	–	–
[27]	FL + FP + AE	–	✓	–	–
[28]	FL + FP + contaminate data	–	✓	–	–
[26], [29]	FL + LF + PoisonGAN	✓	–	–	✓
Our's	FL + LF + FP + RDFS	✓	✓	✓	✓

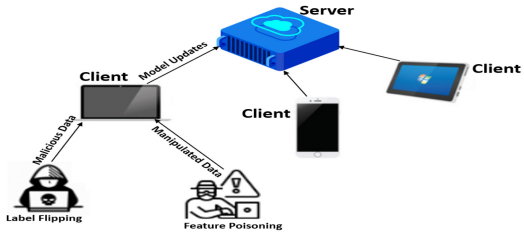

Fig. 1. Illustration of Data Manipulation: A Server and ten Clients. Trained Client 1 (CL1) on manipulated data with LF, FP, and VagueGAN attacks. Other Clients (CL2 to CL10) were trained with the original dataset without any manipulation.

adversary develops data that closely matches the real data within a collaborative learning setting. However, there are several constraints to this approach. The attacker must have prior knowledge of the victim's data and all members of the class must exhibit similarity [26]. Another challenging form of data-poisoning attack employs auto-encoders (AE). AEs are NNs that replicate input data. These AEs can be exploited by adversaries to create poisonous samples that can be employed for training purposes [27]. Nguyen et al. [28] demonstrated that IDS for the Internet of Things (IoT) based on FL are vulnerable to backdoor attacks. Their proposed attack technique demonstrates how malicious actors can deceive the detection model by utilizing compromised IoT devices to add small amounts of contaminated data during the training process without being detected.

Generative Adversarial Networks (GAN): In recent years, researchers in the field of ML have shown that GANs are capable of generating pseudo-examples that are effective in extracting private features and enhancing data poisoning attacks. This is the first time that a GAN model is proposed in an article [26] to derive the private features of local data from GAN-generated pseudo-data. In [29], researchers recommended that PoisonGAN, a standard GAN model, be employed to augment standard LF attacks by utilizing a discriminator that is set by the global model. This model is specifically employed to increase the size of local datasets using authentic pseudo-examples.

In Table I, we list and compare our study with prior studies on FL, for example, the types of attacks, and defenses.

III. METHODOLOGY

In this study, the CIC and UNSW datasets have been chosen due to their comprehensive and detailed nature, covering various network traffic types and attack scenarios. Unlike other datasets such as the NSL-KDD [31], the DARPA 1998 [32], or those datasets are discussed in [33], which may focus on a small number of features or simpler attack types, the CIC and UNSW datasets offer a comprehensive range of features and include a variety of attack types that correspond to real-world network scenarios. This wide range provides a more comprehensive and realistic evaluation of data-poisoning attacks on FL. We trained several DL models on these two datasets separately and captured their accuracy. In this FL setup, we considered ten clients and a server, applying different data-poisoning attacks, such as LF, FP, and VagueGAN, to only one client (Client1, CL1). In DL, an attacker can modify the training dataset based on their knowledge and information. According to the existing literature, there are three scenarios that an adversary can consider to conduct adversarial attacks: **open-box**, **gray-box**, and **closed-box** attacks. The adversary possesses perfect knowledge (PK) about the training data and model in a *open-box* setting and can generate adversarial instances for training and modify the model updates. In the case of a *gray-box* scenario, the threat actor has limited knowledge (LK) of the training data and model. The adversary does not know the internal information of the system in a *closed-box* setup, which is a more viable and complex case than the other scenarios. Consequently, the attacker employs recurring inquiries to gather such sensitive data. We carry out this experiment in a *open-box* scenario, as we have access to both the data and the model. Additionally, a very simple illustration of our methodology is given in Figure 1.

A. Threat Model

The threat model in Adversarial Machine Learning typically involves the attacker's knowledge, capability, and goal. In our problem, these are considered as follows:

Attackers Goal: The attacker's goal is to introduce malicious data during the training phase in order to disrupt the reliability and precision of the global FL model.

Attackers Knowledge: The adversary performs in a open-box scenario, ensuring that they have PK of the training data, model architecture, and training procedure.

Attackers Capability: An attacker is a causative adversary who has the capability to modify training data and model updates by employing innovative methods to subtly alter significant features, thereby influencing the model's behavior.

B. Network Architecture

We adopted a single deep-learning model for both clients and the server. Initially, this same model was utilized for our server as well. This model comprises an input layer, where the number of neurons is set to the feature size. The feature size represents the number of columns in the dataset minus one, as one column represents the output label and is not part of the feature columns. The model also contains two hidden layers: the first hidden layer has 2048 neurons, and the second

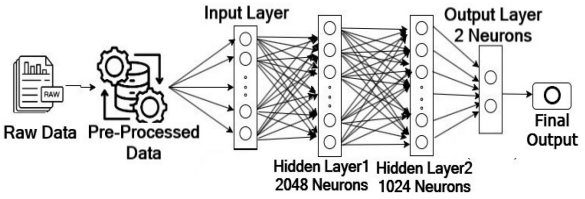

Fig. 2. Architecture of the Neural Network Model (Known as BAU1).

TABLE II
ATTACK SCENARIOS ON BAU1 NETWORK WITH DIFFERENT DATASETS

Scenarios	Dataset	Attack Strategy
$N^{UNSW}_{BAU1-LF}$	UNSW	Label Flipping (LF)
$N^{UNSW}_{BAU1-FP}$	UNSW	Feature Poisoning (FP)
$N^{UNSW}_{BAU1-GAN}$	UNSW	Generative Adversarial Network (GAN)
$N^{CIC}_{BAU1-LF}$	CIC	Label Flipping (LF)
$N^{CIC}_{BAU1-FP}$	CIC	Feature Poisoning (FP)
$N^{CIC}_{BAU1-GAN}$	CIC	Generative Adversarial Network (GAN)

hidden layer has 1024 neurons. Additionally, there are two output neurons in the third hidden layer, corresponding to the number of output classes. The model includes two activation functions: the Rectified Linear Unit (ReLU) activation function, referred to as ReLU1, applied after the first hidden layer, and ReLU2, applied after the second hidden layer. The architecture of the neural network model, named BAU1, can be easily obtained from Figure 2. Furthermore, batch normalization applies to both hidden layers. Normalization is performed during pre-processing to help the neural network become stable and generalize well. A dropout layer with a probability (self.dropout_p) is included in the NN architecture to overcome overfitting. The output activation function is the log softmax function, which computes the natural logarithm (base e) of softmax probabilities. The cross-entropy loss function is used in this network, which is commonly employed for classification problems to compute the loss between the predicted probability distribution and the true class labels. The output layer of our model comprises two neurons, with each neuron representing one of the binary classes: 0 for the benign class and 1 for the malicious class.

C. Experimental Setup

To build the model, we consider 838,861 samples for training, 104,857 for validation, and 104,857 for the test set. The dataset is divided as follows: 80% of the data for training and 20% of the data for testing and validation. Furthermore, we partition the training dataset into ten segments, allocating one segment to Client1 (CL1), and the other segments to Client1 (CL1), and the other to Client2 (CL2),. . . , Client10 (CL10). In Table II, different attack scenarios are presented. For instance, $N^{UNSW}_{BAU1-LF}$ represents the NN model named BAU1, which is trained with the UNSW dataset, to which the LF attack is applied. Similarly, $N^{CIC}_{BAU1-FP}$ indicates that we trained BAU1 with the CIC dataset that is poisoned using the FP attack. Another important scenario, $N^{CIC}_{BAU1-LF}$, points to the LF attack on the CIC dataset, which is used to train the BAU1 model, whereas $N^{UNSW}_{BAU1-FP}$ denotes the FP attack on the UNSW dataset that is fed to the BAU1 model for training. Additionally, we consider the same scenario for the VagueGAN approach, such as $N^{CIC}_{BAU1-GAN}$ and $N^{UNSW}_{BAU1-GAN}$. The number of samples in both datasets is equal, with 1,048,575 examples in each dataset. Most DL models are designed to accept images as input, which are usually three-dimensional, whereas the IDS dataset we use is one-dimensional. Therefore, to make the model feasible, two additional dimensions were added. After adding these extra dimensions, the shape of the np array was changed, which was done for the clients' training, testing, and validation data. We develop an NN using PyTorch, a popular Python library. Our analysis is performed using the hardware specification MSI GF65. For the NN model, we consider 20 epochs for training using the Stochastic Gradient Descent (SGD) optimizer with a random learning rate between 1×10^{-4} and 9.9×10^{-3} and a momentum of 0.9. The batch size for training and validation is set to 1000. For the ten clients, we employ the Federated Averaging (FedAvg) algorithm, a common setup in FL, where multiple clients with local datasets collaboratively train models while preserving data privacy.

D. Empirical Study

In this experiment, we performed three data-poisoning attacks on several NN models trained on two different datasets, namely CIC and UNSW. First, an LF attack was applied. Before performing this attack, we trained the model with a benign dataset, captured the results, saved the models, and tested the accuracy of the saved models using test data. Then, we poisoned the data by flipping the labels for 1% of the data. The model was retrained with this malicious data, and the results and models were saved. To compute the *ASR*, we flipped the labels of the complete test data and noted the accuracy, indicating the strength of the attack. We repeated the experiment with 2%, 3%, 4%, 5%, 7%, 10%, 15%, 20%, and 25% poisoned data, documented the results for all these attack percentages, and computed the ASR for each of them.

Our code is written in such a generic manner that by simply changing the name of the dataset, the rest of the processes—such as pre-processing of data, splitting the training data for ten clients, training the model with/without attack, computing ASR, and saving the results in an Excel file—are performed for both datasets without any user interaction. For the FP attack, we introduced some changes in the code and employed a random forest algorithm to compute feature importance. From our perspective, this innovative approach helps identify which features contribute the most to the classifier. Based on the random forest outcome, in the CIC dataset, the first column has the highest feature importance, which we computed using permutation on the full model. Permutation-based feature importance involves logically rearranging the values of each feature, analyzing the influence on the model's performance, and determining the impact intensity of each feature in the decision-making process.

We used a similar strategy to determine the feature importance in the UNSW dataset and found that the second column has the highest feature importance.

Algorithm 1: Feature Poisoning (FP) Attack

Data: D: Feature column values, L: Labels (0 and 1), att_per: P
Result: Transformed feature values
Step 1: Find min and max values in feature column:
- $min\_value \leftarrow \min(D)$
- $max\_value \leftarrow \max(D)$

Step 2: Find average for label classes 0 and 1:
- $avg\_zero \leftarrow \text{avg}(D \text{ where } L = 0)$
- $avg\_one \leftarrow \text{avg}(D \text{ where } L = 1)$

Step 3: Normalize the values:
- **for** *i in dataset* **do**
  - **if** $L = 0$ **then**
    - $norm\_val \leftarrow \frac{avg\_zero - min\_val}{max\_val - min\_val}$
  - **else**
    - $norm\_val \leftarrow \frac{avg\_one - min\_val}{max\_val - min\_val}$
  - Update feature value with $norm\_val$

Step 4: Modify malicious samples to benign samples:
- num $\leftarrow$ P
- $percent \leftarrow \text{int}(\text{len}(L) \times (\text{num}/100))$
- **for** $i = 1$ **to** *percent* **do**
  - $rand\_index \leftarrow \text{random}(0, \text{len}(unique\_val) - 1)$ **if** $L[i] = 1$ **then**
    - $D[i, 0, 0, \text{col\_id}] \leftarrow unique\_val[rand\_index]$

In FP, we manipulate the values of the most important features that were determined using the random forest technique. There were two label classes in both datasets: 0 and 1. First, we compute the *mean* of all values where the label is 0 and then compute the mean of values with label 1. In the next step, we found the minimum and maximum values in the feature column that are different for the two datasets. Subsequently, the values of the most important features are normalized using the *min-max* normalization technique. We changed the values of the column where the label was 0 to random unique values, where the label was 1. Details regarding FP attack techniques that we consider in this study is in Algorithm 1. In this algorithm, *D* represents a dataset that contains feature column values. *att_per* represents the attack percentage, which is also represented as P in the algorithm, which is the percentage of values that are to be manipulated. min_value and max_value represent the minimum and maximum values in the feature column, respectively. The i represents the iterator value of the loop, L represents the value of the label that may be either 0 or 1, and *percent* represents the number of values to be manipulated in the feature column. Additionally, number represents the user-defined *att_per*, and column_index is the index of the target feature column.

The novel VagueGAN approach, which was recently proposed by [34], is employed in our study as a third type of attack against an FL-IDS. This approach enables an adversary to simply execute a data-poisoning attack. In particular, the adversary initially controls the client c_j. Next, malicious client c_j generates a contaminated local dataset D_{p}^{j} from the original dataset D_{train}^{j} using VagueGAN and replaces it. Finally, client c_j trains and sends a contaminated local model $\theta_{\text{p}}^{t,j}$ to the server using D_{p}^{j}. In contrast to an LF and FP attack, which only affects specific labels, VagueGAN indirectly targets the global model, affecting all labels and classes. In this scenario, we only consider a small percentage of poisoned data, ranging from 1% to 5%, on CL1 to evaluate the robustness of the server. From our point of view, other poison percentages will exhibit the same behavior as the 1% to 5% range.

In contrast, to calculate the ASR, the value of the column where the label was 0 is replaced with the normalized average value of label 1, and vice versa. The percentage of attacks is calculated as the number of values to be manipulated according to the desired percentage. We set up an array containing integer values. In every iteration, if the iterator value is not in that array, then the iterator value is considered a percentage number. This percentage is multiplied by the number of labels in the training data of CL1, which is also equal to the number of samples/rows in the training data. In equation form, it can be written as:

$$\text{number\_of\_values} = \left\lfloor \frac{\text{len}(CL1\_Y) \cdot \text{attack\_percentage}}{100} \right\rfloor \quad (1)$$

In the above equation, CL1_Y represents the length of the array of labels of training data for CL1, where *len* in Python is used to determine the length of an array, while _of_attack represents the percentage number, and the number_of_values denotes the number of values that we have to change in the feature column. During computations in Python, this equation can return the float value, so that it changes to the integer value for which we have used the floor function.

E. RDFS Strategy Integration for Server Security Improvement

Our study addresses the vulnerability of an FL server to poisoning attacks. To mitigate this vulnerability, we applied the Random Deep Feature Selection (RDFS) technique proposed in [35]. To improve server security, we integrated the PRDF technique into the server architecture. This integration included the use of randomly chosen subsets of features with sizes of {50, 400} for training. We demonstrate that this approach enhances the robustness of the model against poisoning attacks by introducing randomness into the training process. We applied RDFS before the attack and then checked the server against the attacks to evaluate the robustness of the model against poisoning attacks. The RDFS steps are as follows:

- Let $\mathcal{N}$ represents the Neural Network (NN) architecture that includes layers $\{C_1, C_2, \ldots, C_k\}$. The W_j weight matrix and b_j bias vector define each C_j layer.
- The representation of the feature space is illustrated by a feature vector for all inputs x_i from the dataset $\{(X_i, y_i)\}_{i=1}^{m}$,

$$\phi(X_i) = (fe_1(x_i), fe_2(x_i), \ldots, fe_n(x_i)) \quad (2)$$

- By applying a random selection method, we choose a subset $F \subset \{1, 2, \ldots, n\}$ from the set {50, 400} for the

TABLE III
RESULTS WITHOUT ATTACK

Scenario	Component	Avg. Loss	Accuracy
N_{BAU1}^{CIC}	Client-1 ... 10	0.6415	-
N_{BAU1}^{CIC}	**Server**	-	0.9680
N_{BAU1}^{UNSW}	Client-1 ... 10	0.6801	-
N_{BAU1}^{UNSW}	**Server**	-	0.8027

size of F. Choose s features at random from the entire set of features for a size s that lies between the intervals {50, 400}. This implies that the features fed to the server for training were random. Therefore, a server trains with a random feature size separately. Here, we have two sets of random; therefore, we have for the server that is trained with {50, 400}.

$$Fe_s = \{fe_{i_1}, fe_{i_2}, \ldots, fe_{i_s}\} \subset \{fe_1, fe_2, \ldots, fe_n\} \quad (3)$$

- A shortened feature vector should be built for each input X_i using the given features, therefore,

$$\phi_{Fe_s}(X_i) = (fe_{i_1}(X_i), fe_{i_2}(X_i), \ldots, fe_{i_s}(x_i)) \quad (4)$$

- Build a new dataset employing the shortened feature vectors that we obtained and that we want to deliver to the server for training, therefore,

$$\{(\phi_{Fe_s}(X_i), y_i)\}_{i=1}^{m} \quad (5)$$

- Training is performed on $\mathcal{N}$ using the shorthand dataset obtained from the previous step Fe_s. The training phase minimizes the loss function $\mathcal{L}$ with respect to the network parameters Θ:

$$\min_{\Theta} \mathcal{L}(\mathcal{N}(\phi_{Fe_s}(X_i)), y_i) \quad (6)$$

The symbol Θ denotes the weights and biases of each layer.

IV. RESULTS AND DISCUSSION

In this section, we discuss the results obtained during the experiment using different scenarios, with and without an attack.

A. FL in the Absence of Attacks

We trained the BAU1 model using benign datasets. First, we trained FL on the CIC dataset in the absence of attacks and reported the results as N_{BAU1}^{CIC}. We used this notation in Table III to clarify the results. Subsequently, we trained the BAU1 model with the UNSW dataset without an attack, which is denoted as N_{BAU1}^{UNSW} in Table III, and we report the server accuracy for both scenarios and the average of losses regarding ten clients.

In this table, we report the loss from Client-1 to Client-10 (or CL1 to CL10) for the CIC dataset, where the average of the losses is 0.6415, and save the values obtained after the last epoch. In the same scenario, the server accuracy was 96.80%. The high accuracy of the server is due to the fact that we did not apply any attack and trained the BAU1 model on a clean dataset. In the second scenario without an attack, where the model is the same but the dataset is different, specifically UNSW, the average loss from Client-1 to Client-10 is 0.6801, and the server accuracy is approximately 80.27%. The purpose of these scenarios is to establish basic performance metrics for the BAU1 model on pristine datasets. This enabled us to evaluate the effects of different types of data poisoning attacks on the performance of the model.

TABLE IV
LF POISON ATTACK FOR THE SCENARIO N_{BAU1}^{CIC-LF} ON CIC DATASET

Poison	Component	Avg. Loss	Acc.	ASR
1%	Client-1 ... 10	0.7360	-	-
1%	**Server**	-	0.0428	0.9564
2%	Client-1 ... 10	0.7323	-	-
2%	**Server**	-	0.0537	0.9457
3%	Client-1 ... 10	0.6777	-	-
3%	**Server**	-	0.968	0.0329
4%	Client-1 ... 10	0.6717	-	-
4%	**Server**	-	0.9486	0.0539
5%	Client-1 ... 10	0.6798	-	-
5%	**Server**	-	0.7739	0.2292
7%	Client-1 ... 10	0.7086	-	-
7%	**Server**	-	0.1256	0.8720
10%	Client-1 ... 10	0.7499	-	-
10%	**Server**	-	0.032	0.9670
15%	Client-1 ... 10	0.7150	-	-
15%	**Server**	-	0.0447	0.9543
20%	Client-1 ... 10	0.7096	-	-
20%	**Server**	-	0.1281	0.8718
25%	Client-1 ... 10	0.6706	-	-
25%	**Server**	-	0.9204	0.0797

B. Results With Label Flipping (LF) Attack on CIC Dataset

The LF attack on the CIC dataset is shown in Table IV. The results show how the attack affected the effectiveness of the FL model and how easy it was to spot. When only 1% of the data are tampered with, the server's accuracy drops to 0.0428, and the ASR increases to 0.9564. The large drop in accuracy and high ASR show that even a small amount of corrupted data may disrupt the model, which makes the attack very easy to spot. The server's precision remains low at 0.0537, and the ASR remains high at 0.9457 when 2% of the data have been contaminated in the same way. Given that it has such a large effect on model performance, these data prove that the LF attack is easy to detect at low percentages. As the amount of poisoned data increases to between 3 and 4 percent, an interesting change occurs. While the ASR dropped drastically to 0.0329 and 0.0539, the server accuracy improved to 0.9680 and 0.9486, respectively. The fact that precision returns to normal and ASR values are low indicates that the FL model is strong enough to withstand these low levels of LF attacks, making the attack useless. However, when there is 5% poisoned data, the server's precision falls to 0.7739, and its ASR increases to 0.2292. By this point, the attack is more apparent but still not very problematic for the model. It can be seen that, but not as serious as at lower levels. The results were quite distinct when the amount of poisoned data increased to 7%, 10%, 15%, 20%, and 25%. In some situations, such as the 10% attack, the server accuracy remains very low at 0.0320, but the ASR is high at 0.9670, which means that the attack fails because it is easy to spot. In other cases, the results are

TABLE V
LF POISON ATTACK FOR THE SCENARIO $N_{BAU1}^{UNSW-LF}$ ON UNSW DATASET

Poison	Component	Avg. Loss	Acc.	ASR
1%	Client-1 ... 10	0.6789	-	-
1%	**Server**	-	0.8554	0.1423
2%	Client-1 ... 10	0.7313	-	-
2%	**Server**	-	0.0951	0.9052
3%	Client-1 ... 10	0.7264	-	-
3%	**Server**	-	0.1000	0.8997
4%	Client-1 ... 10	0.6759	-	-
4%	**Server**	-	0.8072	0.1918
5%	Client-1 ... 10	0.7044	-	-
5%	**Server**	-	0.2815	0.7193
7%	Client-1 ... 10	0.6777	-	-
7%	**Server**	-	0.8253	0.1757
10%	Client-1 ... 10	0.6692	-	-
10%	**Server**	-	0.8816	0.1181
15%	Client-1 ... 10	0.6844	-	-
15%	**Server**	-	0.6793	0.3186
20%	Client-1 ... 10	0.7045	-	-
20%	**Server**	-	0.1795	0.8212
25%	Client-1 ... 10	0.6841	-	-
25%	**Server**	-	0.7097	0.2920

more mixed, with server accuracy sometimes improving and ASR values not meeting the requirements for an attack to be successful. For example, when 25% of the data are poisoned, the server accuracy increases to 0.9204, but the ASR decreases to 0.0797. This indicates that even high percentages of LF attacks may not be able to deceive the system without being detected.

These findings verify the robustness of the FL model to the LF data percentages. Low-percentage attacks are readily apparent owing to significant decreases in server accuracy, whereas in-between percentage attacks highlight the model's durability, regaining accuracy, and low ASR. A high percentage of attacks fail because they have either too low accuracy, making them traceable, or the ASR is too low. Thus, the LF attack on the CIC dataset fails to degrade the models sustainably and undetectably, demonstrating the necessity for more advanced attack tactics, such as FP or VagueGAN, to have a significant impact.

C. Results With Label Flipping (LF) Attack on UNSW Dataset

The outcomes of the LF attack on the UNSW dataset, shown in Table V, demonstrate the influence of the attack on the accuracy and detectability of the FL model. Starting with a 1% LF attack, the server's accuracy is quite acceptable at 0.8554, but the ASR is reduced to 0.1423, suggesting a failed attack because the ASR does not satisfy the criteria for success. As the attack strength increases to 2% or 3%, the server accuracy falls to approximately 0.1, but the ASR rises to approximately 0.9, indicating that the attack is readily apparent owing to the considerable reduction in accuracy and high ASR. At 4%, the server accuracy slightly increases to 0.8072, but the ASR decreases to 0.1918, failing to fulfill the requirements for a successful attack. A 5% LF attack decreased the server accuracy to 0.2815 and the ASR to 0.7193, showing an evident but less severe attack compared

TABLE VI
FEATURE POISON (FP) ATTACK FOR THE SCENARIO N_{BAU1}^{CIC-FP} ON CIC DATASET

Poison	Component	Avg. Loss	Acc.	ASR
1%	Client-1 ... 10	0.6507	-	-
1%	**Server**	-	0.9642	0.9628
4%	Client-1 ... 10	0.6754	-	-
4%	**Server**	-	0.8611	0.8616
20%	Client-1 ... 10	0.6787	-	-
20%	**Server**	-	0.7427	0.7763
25%	Client-1 ... 10	0.6321	-	-
25%	**Server**	-	0.9680	0.9671

to lower percentages. Higher attack percentages (7, 10, 15, 20, and 25%) resulted in different outcomes. The 7% and 10% attacks have significant server accuracy but low ASR, suggesting unsuccessful attacks, whereas the 15% attack has moderate effects. The 20% attack reduces the server accuracy to 0.1795 and has an ASR of 0.8212, indicating that it is noticeable, but not completely successful. Furthermore, a 25% attack increases the server accuracy to 0.7097, resulting in an ASR of 0.2920, indicating an unsuccessful attack.

These findings demonstrate that data poisoning ratios considerably affect the efficacy and detectability of LF attacks, even in the UNSW dataset. Low percentages have no effect on the model, while other percentages degrade the performance and make the attack more detectable. High percentages indicate varying results, with low ASR attacks failing and others resulting in reduced accuracy.

D. Results With Feature Poisoning (FP) Attack on CIC Dataset

In this section, the FP Attack on CIC Dataset and the results are presented in Table VI, which shows the effectiveness as well as the effect of FP attacks on the FL model. The experiment demonstrated that even small amounts of poisoned data can have significant effects on the performance of the model. For instance, even with only 1% of the data poisoned, the server's precision remains high at 0.9642 and the ASR remains high at 0.9628, implying a successful and undetectable attack. This pattern remained in the 4% poisoned scenario, which had a server accuracy of 0.8611 and an ASR of 0.8616, demonstrating the efficacy of the attack. However, when the amount of poisoned data increased to 20% and 25%, the server accuracy and ASR continued to represent successful attacks, with values of 0.7427 and 0.7763 for 20% and 0.9680 and 0.9671 for 25%, respectively. These findings indicate the model's sensitivity to FP attacks because attacks remain successful and difficult to detect as the amount of poisoned data increases.

These studies employ the CIC dataset to determine how FP attacks affect FL models and their detectability. The research effort evaluates server precision and ASR at different data poisoning rates to determine how vulnerable the model is to attacks like this and how well they may compromise model consistency without detection. Previously, LF attacks on the CIC dataset induced large accuracy decreases, making them more apparent, whereas FP attacks maintained high accuracy and ASR, making them more invisible and successful on

TABLE VII
FEATURE POISON ATTACK FOR THE SCENARIO $N_{BAU1}^{UNSW-FP}$ ON UNSW DATASET

Poison	Component	Avg. Loss	Acc.	ASR
1%	Client-1 ... 10	0.6779	-	-
1%	**Server**	-	0.8195	0.8231
2%	Client-1 ... 10	0.6769	-	-
2%	**Server**	-	0.8527	0.8722
3%	Client-1 ... 10	0.6720	-	-
3%	**Server**	-	0.8433	0.8194
5%	Client-1 ... 10	0.6728	-	-
5%	**Server**	-	0.8620	0.8491
7%	Client-1 ... 10	0.6690	-	-
7%	**Server**	-	0.9017	0.9009
10%	Client-1 ... 10	0.6720	-	-
10%	**Server**	-	0.8725	0.8944
15%	Client-1 ... 10	0.6458	-	-
15%	**Server**	-	0.9066	0.9086
20%	Client-1 ... 10	0.6939	-	-
20%	**Server**	-	0.4682	0.4824
25%	Client-1 ... 10	0.6644	-	-
25%	**Server**	-	0.9018	0.9064

the CIC dataset. This comparison shows the need for better defenses against the subtlety of FP assaults.

E. Results With Feature Poisoning (FP) Attack on UNSW Dataset

Table VII provides the findings of the FP attacks on the UNSW dataset, demonstrating the different consequences of the attack on the performance and visibility of the FL model. With a 1% FP attack, the server precision is 0.8195 and the ASR is 0.8231, revealing that the attack is successful and subtle. As the amount of poisoned data increases to 2% and 3%, the server precision increases to 0.8527 and 0.8433, respectively, while the ASR values remain high at 0.8722 and 0.8194, confirming successful attacks that are difficult to detect. Even at higher percentages, such as 5%, 7%, and 10%, the server precision remains above 0.86, and ASR values are continually high, proving that the attack is effective. In particular, for 20% and 25% poisoned data, the server accuracy also reveals an effective impact with values of 0.4682 and 0.9018, as well as ASR values of 0.4824 and 0.9064, respectively. These findings highlight the efficacy and stealthiness of FP attacks, which pose a significant threat to model precision.

The objective of this evaluation is to see how varying percentages of data poisoning from FP attacks affect the performance and detectability of FL models on the UNSW dataset. By studying server precision and ASR, this study aims to achieve a better understanding of the model's sensitivity to FP attacks and how these attacks might subtly but effectively impact model integrity. A comparison of the FP attack results (Table VII) to the LF attack results (Table V) on the UNSW dataset proves that FP attacks are typically more successful and difficult to detect. Although LF attacks frequently cause significant decreases in server precision, making them identifiable, FP attacks maintain high accuracy and ASR, resulting in successful but less apparent disruptions. This comparison emphasizes the need for improved protection systems that safeguard against the subtleties of FP attacks.

TABLE VIII
VAGUEGAN ATTACK FOR THE SCENARIO $N_{BAU1}^{CIC-GAN}$ ON CIC DATASET

Poison	Component	Avg. Loss	Acc.	ASR
1%	Client-1 ... 10	0.6581	-	-
1%	**Server**	-	0.1961	0.9752
2%	Client-1 ... 10	0.7101	-	
2%	**Server**	-	0.2872	0.9213
3%	Client-1 ... 10	0.7227	-	-
3%	**Server**	-	0.1765	0.8193
4%	Client-1 ... 10	0.7195	-	-
4%	**Server**	-	0.2801	0.8095
5%	Client-1 ... 10	0.7034	-	-
5%	**Server**	-	0.1201	0.7820

TABLE IX
VAGUEGAN ATTACK FOR THE SCENARIO $N_{BAU1}^{UNSW-GAN}$ ON UNSW DATASET

Poison	Component	Avg. Loss	Acc.	ASR
1%	Client-1 ... 10	0.6890	-	-
1%	**Server**	-	0.1760	0.9872
2%	Client-1 ... 10	0.6387	-	-
2%	**Server**	-	0.2310	0.9020
3%	Client-1 ... 10	0.7139	-	-
3%	**Server**	-	0.3141	0.8493
4%	Client-1 ... 10	0.7090	-	-
4%	**Server**	-	0.2628	0.8193
5%	Client-1 ... 10	0.6812	-	-
5%	**Server**	-	0.1787	0.8957

F. Results With VagueGAN Attack on CIC and UNSW Datasets

As shown in Tables VIII and IX, The VagueGAN attack has a significant effect on the performance of the FL models on the CIC and UNSW datasets. Low server accuracies (0.1961 at 1% poisoning and 0.1201 at 5% poisoning) while maintaining high ASR (0.9752 and 0.7820, respectively) are the outcomes of VagueGAN attacks on the CIC dataset. Similarly, with high ASR values of 0.9872 and 0.8957, the UNSW dataset reveals that server accuracies declined to 0.1760 at 1% poisoning and remained low up to 5%. With such a high ASR, these findings demonstrate that VagueGAN successfully creates adversarial attacks that significantly affect the performance of the model. A server can recognize a VagueGAN attack as it impacts all features, unlike FP and LF attacks, which change only one feature and may remain undetected. When comparing the VagueGAN findings to FP and LF attacks, it is evident that VagueGAN attacks are more generally effective and have a higher ASR, but they are also easier to identify.

G. RDFS Defense Strategy for Improving Server Security

We prove that FP is more effective than LF and VagueGAN attacks. Therefore, regarding the defense system, we consider the recent strategy RDFS, which is based on random deep feature selection. This method has recently proven quite effective in different domains, such as multimedia forensics and security [36], and in computer networks [35]. Applying the RDFS defense technique to the CIC dataset with random feature sizes of 50 and 400 shows significant effectiveness against diverse percentages of poisoned data (Table X). For 1% poisoned data, both feature sizes retain high server

TABLE X
PERCENTAGE OF RANDOM INDEX RESULTS FOR N_{BAU1}^{CIC-FP} ON CIC DATASET

Poison	RFS	Acc.	ASR	RFS	Acc.	ASR
1%	50	0.9016	0.3423	400	0.9550	0.1430
4%	50	0.9020	0.2816	400	0.9601	0.1209
20%	50	0.9127	0.2627	400	0.9780	0.1022
25%	50	0.9042	0.2711	400	0.9817	0.0117

TABLE XI
PERCENTAGE OF RANDOM INDEX RESULTS FOR $N_{BAU1}^{UNSW-FP}$ ON UNSW DATASET

Poison	RFS	Acc.	ASR	RFS	Acc.	ASR
1%	50	0.8290	0.2516	400	0.8763	0.1371
2%	50	0.8422	0.2301	400	0.8809	0.1916
3%	50	0.8228	0.3012	400	0.9056	0.1837
5%	50	0.8631	0.2501	400	0.8820	0.1400
7%	50	0.8305	0.2012	400	0.9166	0.1677
10%	50	0.8930	0.1900	400	0.8956	0.1562
15%	50	0.8421	0.2385	400	0.9054	0.1920
25%	50	0.8729	0.2078	400	0.9026	0.1757

accuracy and low ASR, proving that RDFS is effective. When the poisoning percentage increases to 4%, 20%, and 25%, the server accuracy improves, while the ASR decreases slightly, showing RDFS defensive resistance. Both feature sizes of 50 and 400 in the UNSW dataset (Table XI) at 1% poisoned data exhibit high server accuracy and low ASR, comparable to the CIC dataset, indicating good early defense. As poisoning grows to 2%, 3%, 5%, 7%, 10%, 15%, and 25%, the server accuracy remains high and ASR remains low, demonstrating the strength of the RFS strategy.

V. DISCUSSION AND LIMITATIONS

Our research provided novel insights into the vulnerability of FL models to data poisoning attacks by emphasizing the more effective nature of FP attacks compared to LF and VagueGAN attacks. Our research highlighted the importance and details of these attacks in the context of FL-based IDS by employing them specifically on computer network datasets, namely CIC and UNSW. We proposed an innovative method to conduct FP attacks by employing Random Forests to identify and manipulate highly contributing features. The experimental results demonstrated the significant vulnerability of FL models to FP attacks, as well as the efficacy and subtle nature of these attacks. Model precision was significantly reduced by even small modifications to features. In contrast, LF attacks were significantly easier to identify than other types of attacks, despite their significant impact, due to the substantial reductions in precision they caused. Additionally, the VagueGAN attack, which utilized Generative Adversarial Networks (GANs) to introduce adversarial examples into datasets, significantly reduced model accuracy while maintaining a high Attack Success Rate (ASR). The controlled experimental configuration involved ten clients, enabling a comprehensive analysis of the attack structure. This analysis revealed that FP and VagueGAN attacks were more capable of evading detection than LF attacks. This innovative application of feature importance analysis in FP attacks has enabled a more thorough understanding of how adversaries may exploit specific data attributes to undermine FL models. Moreover, our study developed the Random Deep Feature Selection (RDFS) strategy, which randomizes server features during training. This strategy is an effective defense mechanism against FP attacks and, as we demonstrated, is also effective against LF and VagueGAN attacks, proving its efficacy in maintaining model integrity under attack conditions.

This study has some limitations:

Attack Scope: This study targeted LF, FP, and VagueGAN attacks. Although these are prevalent, more complex attacks, such as backdoors and evasion techniques, were not considered. Future research should investigate a broader range of attack vectors.

Defense Mechanisms: While the study emphasized the vulnerabilities and efficacy of three different types of attacks, it also developed and evaluated RDFS defense mechanisms against FP. Future research should investigate a general defense model against all possible poisoning and evasion attacks.

Generalizability: These studies were conducted using specific real-world datasets related to computer networks. Although these datasets are important, the generalizability of the findings to other fields is unknown. Future research should use a variety of datasets to confirm the strength of the results.

VI. CONCLUSION REMARKS

In this study, we evaluated the efficacy of the RDFS defense system in the computer network domain and examined the vulnerability of FL models to LF, FP, and the recently introduced VagueGAN attacks. Our experiments focused on these attacks on the CIC and UNSW datasets. They demonstrated that LF attacks were ineffective due to visible accuracy drops, whereas FP attacks significantly deceived the server in various scenarios. Furthermore, our results underlined the potential threat to the integrity of FL models posed by VagueGAN attacks. The analysis of feature importance revealed that dataset features are vulnerable to manipulation in FP attacks, highlighting the significant effectiveness of FP attacks. The integrated defense mechanism of the RDFS defense strategy is demonstrated by its promising performance in mitigating some FP attacks. The objective of future research should be to develop a comprehensive defense strategy that can mitigate a broader range of attack vectors and enhance the robustness of FL models in practical settings by incorporating more complex threats such as backdoors and evasion techniques.

REFERENCES

[1] S. Agrawal et al., "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Comput. Commun.*, vol. 195, Nov. 2022, pp. 346–361.
[2] R. Shrestha et al., "Anomaly detection based on LSTM and autoencoders using federated learning in smart electric grid," *J. Parallel Distrib. Comput.*, vol. 193, Nov. 2024, Art. no. 104951. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731524001151
[3] M. J. Idrissi et al., "Fed-ANIDS: Federated learning for anomaly-based network intrusion detection systems," *Expert Syst. Appl.*, vol. 234, Dec. 2023, Art. no. 121000.
[4] D. Javeed, M. S. Saeed, M. Adil, P. Kumar, and A. Jolfaei, "A federated learning-based zero trust intrusion detection system for Internet of Things," *Ad Hoc Netw.*, vol. 162, Sep. 2024, Art. no. 103540.

[5] Z. Jin, J. Zhou, B. Li, X. Wu, and C. Duan, "FL-IIDS: A novel federated learning-based incremental intrusion detection system," *Future Gener. Comput. Syst.*, vol. 151, pp. 57–70, Feb. 2024.
[6] O. Aouedi and K. Piamrat, "F-BIDS: Federated-blending based intrusion detection system," *Pervasive Mobile Comput.*, vol. 89, Feb. 2023, Art. no. 101750.
[7] M. Maddu and Y. N. Rao, "Network intrusion detection and mitigation in SDN using deep learning models," *Int. J. Inf. Secur.*, vol. 23, no. 2, pp. 849–862, 2024.
[8] Y.-C. Lai et al., "Two-phase defense against poisoning attacks on federated learning-based intrusion detection," *Comput. Secur.*, vol. 129, Jun. 2023, Art. no. 103205.
[9] Y. Miao et al., "RFed: Robustness-enhanced privacy-preserving federated learning against poisoning attack," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 5814–5827, 2024.
[10] A. Sharma and N. Marchang, "A review on client-server attacks and defenses in federated learning," *Comput. Secur.*, vol. 140, May 2024, Art. no. 103801.
[11] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data," *J. Big Data*, vol. 7, no. 1, p. 104, 2020.
[12] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.
[13] I. H. E. Nowroozi. "Federated learning under attack: Exposing vulnerabilities through data poisoning attacks in computer networks." 2023. Accessed: Jun. 16, 2024. [Online]. Available: https://github.com/ehsannowroozi/FederatedLearning_Poison_LF_FP
[14] B. T. M. Barni, E. Nowroozi, and B. Zhang. "Random Deep Feature Selection (RDFS)." 2019. Accessed: Jun. 16, 2024. [Online]. Available: https://github.com/ehsannowroozi/RDFS
[15] E. Hallaji, R. Razavi-Far, M. Saif, and E. Herrera-Viedma, "Label noise analysis meets adversarial training: A defense against label poisoning in federated learning," *Knowl. Based Syst.*, vol. 266, Apr. 2023, Art. no. 110384.
[16] D. Li, W. E. Wong, W. Wang, Y. Yao, and M. Chau, "Detection and mitigation of label-flipping attacks in federated learning systems with KPCA and k-means," in *Proc. 8th Int. Conf. Depend. Syst. Appl. (DSA)*, 2021, pp. 551–559.
[17] N. M. Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "Defending against the label-flipping attack in federated learning," 2022, *arXiv:2207.01982*.
[18] E. Nowroozi, N. Jadalla, S. Ghelichkhani, and A. Jolfaei, "Mitigating label flipping attacks in malicious URL detectors using ensemble trees," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 6, pp. 6875–6884, Dec. 2024.
[19] J. Xu, Z. Chen, T. Q. Quek, and K. F. E. Chong, "FedCorr: Multi-stage federated learning for label noise correction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10184–10193.
[20] B. Zeng, X. Yang, Y. Chen, H. Yu, and Y. Zhang, "CLC: A consensus-based label correction approach in federated learning," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 5, pp. 1–23, 2022.
[21] V. Tsouvalas, A. Saeed, T. Ozcelebi, and N. Meratnia, "Labeling chaos to learning harmony: Federated learning with noisy labels," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 2, pp. 1–26, 2024.
[22] N. M. Jebreel, J. Domingo-Ferrer, D. Sánchez, and A. Blanco-Justicia, "LFighter: Defending against the label-flipping attack in federated learning," *Neural Netw.*, vol. 170, pp. 111–126, Feb. 2024.
[23] X. Shen, Y. Liu, F. Li, and C. Li, "Privacy-preserving federated learning against label-flipping attacks on non-IID data," *IEEE Internet Things J.*, vol. 11, no. 1, pp. 1241–1255, Jan. 2024.
[24] A. Raza, S. Li, K.-P. Tran, and L. Koehl, "Using anomaly detection to detect poisoning attacks in federated learning applications," 2023, *arXiv:2207.08486*.
[25] C. Yang, Q. Wu, H. Li, and Y. Chen, "Generative poisoning attack method against neural networks," 2017,*arXiv:1703.01340*.
[26] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 603–618.
[27] J. Feng, Q.-Z. Cai, and Z.-H. Zhou, "Learning to confuse: Generating training time adversarial data with auto-encoder," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–11.
[28] T. D. Nguyen, P. Rieger, M. Miettinen, and A.-R. Sadeghi, "Poisoning attacks on federated learning-based IoT intrusion detection system," in *Proc. Workshop Decent. IoT Syst. Secur. (DISS)*, 2020, pp. 1–7. [Online]. Available: https://api.semanticscholar.org/CorpusID:216086694
[29] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3310–3322, Mar. 2021.
[30] Y. Jiang, W. Zhang, and Y. Chen, "Data quality detection mechanism against label flipping attacks in federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1625–1637, 2023.
[31] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD cup 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, 2009, pp. 1–6.
[32] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Comput. Netw.*, vol. 34, no. 4, pp. 579–595, 2000.
[33] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, Sep. 2019.
[34] W. Sun, B. Gao, K. Xiong, Y. Wang, P. Fan, and K. B. Letaief, "A GAN-based data poisoning attack against federated learning systems and its countermeasure," 2024, *arXiv:2405.11440*.
[35] E. Nowroozi, M. Mohammadi, P. Golmohammadi, Y. Mekdad, M. Conti, and S. Uluagac, "Resisting deep learning models against adversarial attack transferability via feature randomization," *IEEE Trans. Services Comput.*, vol. 17, no. 1, pp. 18–29, Jan./Feb. 2024.
[36] M. Barni, E. Nowroozi, B. Tondi, and B. Zhang, "Effectiveness of random deep feature selection for securing image manipulation detectors against adversarial examples," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 2977–2981.

Ehsan Nowroozi (Senior Member, IEEE) received the Ph.D. degree from the University of Siena, Italy. He is a Senior Lecturer with the Centre for Sustainable Cyber Security, University of Greenwich, London. He held postdoctoral positions with the University of Siena, the University of Padua, Sabanci University, and Queen's University Belfast. He was also a Senior Lecturer with Ravensbourne University, London. His research focuses on adversarial machine learning/adversarial multimedia forensics. He has been an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT since 2024.

Imran Haider (Member, IEEE) received the bachelor's degree in computer science from the National University of Computer and Emerging Sciences, Pakistan. He is currently pursuing the master's degree in cybersecurity with Bahcesehir University, Istanbul, Turkey. He possesses a strong passion for cybersecurity and artificial intelligence research. He also holds expertise in penetration testing—an essential skill for identifying and mitigating security vulnerabilities in digital systems.

Rahim Taheri (Senior Member, IEEE) received the Ph.D. degree in information technology from the Shiraz University of Technology, Iran, in 2020. He is currently a Senior Lecturer of Cyber Security and Forensics with the University of Portsmouth, U.K. Before joining the University of Portsmouth, he was a Postdoctoral Research Associate with the King's Communications, Learning, and Information Processing Lab, King's College London, U.K. His main research interests include machine learning applications in security, adversarial machine learning, and federated learning.

Mauro Conti (Fellow, IEEE) received the Ph.D. degree from the Sapienza University of Rome, Italy, in 2009. He is a Full Professor with the University of Padua, Italy. He is also affiliated with TU Delft and the University of Washington, Seattle. His research in the area of security and privacy is also funded by companies, including Cisco, Intel, and Huawei. He published more than 500 papers in the topmost international peer-reviewed journals and conferences. He is the Editor-in-Chief of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.