International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 21 No. 3 (1) 2025

ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 21, No. 3(1), 2025

Research Paper

CNN-SHIELDED FEDERATED LEARNING: DUAL-PHASE ANOMALY DETECTION AND COMPRESSION AGAINST POISONING

E. Sravanthi[1], K Shanmukha Thrisha[2], Madhu Sreeja Salanki[2], Indraja Gandhala[2]
[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering
[1,2]Kommuri Pratap Reddy Institute of Technology, Hyderabad, Telangana, India.
[1]Email: sravanthieega26@gmail.com.

Received: 05-6-2025 Accepted: 06-7-2025 Published: 14-7-2025

ABSTRACT

Poisoning attacks pose a significant threat in federated learning, where even a small fraction of malicious users (1–10%) can severely disrupt the model's performance. Studies indicate that over 30% of real-world federated systems have encountered such attacks, leading to accuracy drops of up to 50%. This undermines the reliability of federated learning, especially in critical fields like healthcare and finance. Traditional centralized learning systems are also at risk due to their reliance on a single data storage point, increasing vulnerability to attacks and data breaches. Additionally, manual data handling is often inconsistent and error-prone. To mitigate these risks, a two-step defense approach is proposed that integrates data compression with a federated learning framework built on Convolutional Neural Networks (CNNs). The process begins with data preprocessing, which includes removing missing values, separating inputs and labels, applying standard scaling, and splitting the dataset into 90% training and 10% testing portions. Model compression is then used to reduce the size of user updates, saving bandwidth and concealing potential attack signatures. The proposed CNN-based federated learning model enhances accuracy from 87% (achieved with DNN) to 99%, offering robust defense against poisoning attacks and significantly improving overall performance.

Keywords: Federated Learning (FL), Convolutional Neural Networks (CNN), Poisoning Attack Mitigation, Malicious Client Detection, Bandwidth Optimization.

1. INTRODUCTION

In today's digital world, enormous amounts of data are continuously generated from various sources such as smartphones, sensors, IoT devices, and online platforms. To efficiently process and manage this data, advanced computing technologies like cloud computing, edge computing, and fog computing are utilized. Cloud computing relies on centralized data centers, offering high computational power but often leading to delays and increased network congestion due to the distance between users and servers. To address these challenges, edge computing processes data closer to the source, reducing latency and improving response times. Fog computing serves as an intermediate layer between edge devices and the cloud, distributing processing tasks to enhance speed and reduce network overload. However, as data moves through these systems, it becomes increasingly vulnerable to cyberattacks, particularly poisoning attacks, where attackers inject false or harmful data during processing to manipulate or disrupt the system. These attacks can lead to incorrect results, causing AI models or decision-making systems to produce inaccurate outputs, system crashes, where corrupted data destabilizes computing infrastructure, and security breaches, exposing sensitive information or creating vulnerabilities for further

exploitation. Recent studies show that the world is expected to create 181 trillion gigabytes of data by 2025. Cloud computing is still widely used, with businesses spending over $200 billion in 2023 to use cloud services. But because of the need for faster processing, edge and fog computing are growing fast.

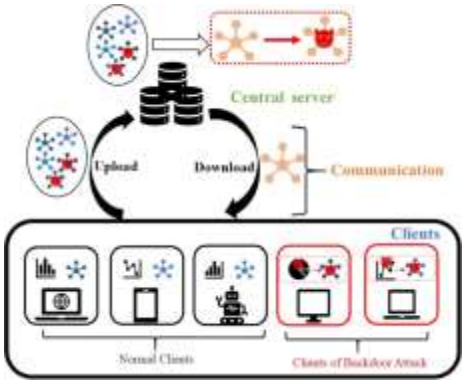

Fig. 1: Anomaly Detection and Detection Techniques.

2. LITERATURE SURVEY

Khraisat, et al. [1] proposed the results that shows that even a small number of malevolent players can significantly reduce classification memory and accuracy, particularly when attacks target certain classes. We also analyze how long these attacks last and when they occur in early and late training rounds, emphasizing how the presence of malicious participants affects attack efficacy. We suggest a defense approach that detects malevolent participants by examining parameter changes over susceptible training cycles in order to lessen these risks. Our method successfully separates fraudulent updates by using Principal Component Analysis (PCA) for anomaly identification and dimensionality reduction. The efficacy of our technique in precisely detecting and eliminating malevolent individuals is confirmed by extensive simulations on standard datasets, improving the FL model's integrity. These findings greatly enhance FL security by providing a strong protection against advanced poisoning techniques.

Nowroozi, et al. [2] proposed Federated Learning is a technique that protects data privacy by allowing several devices to work together to develop a common model without exchanging raw data. However, throughout the training and updating phases, federated learning systems are susceptible to data-poisoning assaults. The CIC and UNSW datasets are used to test FL models across one out of ten clients using three data-poisoning attacks: label flipping, feature poisoning, and VagueGAN. Only a small percentage of each dataset consists of adversarial samples. In this study, we change the proportions of datasets that adversaries can alter to see how they affect. In this study, we change the proportions of datasets that adversaries can alter in order to see how they affect the client and server sides. Because label flipping and VagueGAN attacks are easily detected by the server, experimental results show that they have no discernible impact on server accuracy. Feature poisoning attacks, on the other hand, highlight their subtlety and efficacy by subtly impairing model performance while retaining high accuracy and attack success rates. Consequently, feature poisoning attacks highlight the susceptibility of federated learning systems to such complex attacks by manipulating the server without significantly lowering model accuracy.

Jiang et al . [3] proposed poisoning attacks, in which malevolent clients alter their updates to influence the global model, can harm federated learning (FL). There are a number of ways to find those clients in FL, but discovering malicious clients necessitates enough model changes; therefore, by the time harmful clients are found, FL models have already been tainted. Therefore, after fraudulent clients have been identified, a way to retrieve an accurate global model is required. Current recovery techniques require a lot of storage and processing power since they rely on (i) all of the previous data from participating FL clients and (ii) the original

model that was unaffected by the malicious clients. In this paper, we show that highly effective recovery can still be achieved based on (i) selective historical information rather than all historical information and (ii) a historical model that has not been significantly affected by malicious clients rather than the initial model. In this scenario, we can accelerate the recovery speed and decrease memory consumption as well as maintaining comparable recovery performance. Following this concept, we introduce Crab (Certified Recovery from Poisoning Attacks and Breaches), an efficient and certified recovery method.

Yang,et al. [4] proposed federated Learning (FL) is a novel distributed learning paradigm that has been used in industries like finance, autonomous driving, and intelligent shopping. Nonetheless, a number of strategies have lately been put out that seek to undermine strong aggregation constraints and lower model accuracy. During attacks, these techniques don't keep the gradients' sign statistics constant. As a result, the majority of current assaults can be thwarted by the sign statistics-based technique SignGuard. We suggest ScaleSign, an improved model poisoning attack, to outperform SignGuard and the majority of current cosine or distance-based aggregation schemes. In particular, ScaleSign alters the sign statistics of malicious gradients and obtains malicious gradients with better cosine similarity by employing a scaling attack and a sign modification component, respectively. In addition, these two components have the least impact on the magnitudes of gradients. Then, we propose MSGuard, a Multi-Strategy Byzantine-robust scheme based on cosine mechanisms, symbol statistics, and spectral methods. Formal analysis proves that malicious gradients generated by ScaleSign have a closer cosine similarity than honest gradients. Extensive experiments demonstrate that ScaleSign can attack most of the existing Byzantine-robust rules, especially achieving a success rate of up to 98.23% for attacks on SignGuard. MSGuard can defend against most existing attacks including ScaleSign. Specifically, in the face of ScaleSign attack, the accuracy of MSGuard improves by up to 41.78% compared to SignGuard.

Ali, et al. [5] proposed number of protection techniques have been created to detect and eliminate contaminated local models prior to the aggregation process in order to thwart these attacks. However, because of insufficient filtering techniques, these defense tactics perform less well in keeping harmless local models and removing poisoned local models. As a result, these defense strategies eliminate a significant percentage of harmless, unpolluted local models, which raises false rejection rates or reduces detection accuracy. This also degrades the global model's test accuracy. In this research, we propose the Two-step Defense Framework for Poisoning Attacks Detection (TDF-PAD), which uses the interquartile range approach to first identify the obvious-poisoned, obvious-benign, and ambiguous local models. Based on their performance history, ambiguous local models are categorized into benign or poisoned local models in the second stage using the Z-score approach. We show through thorough experimentation on two real-world benchmark datasets that TDF-PAD is generally applicable to any dataset and surpasses state-of-the-art defense approaches by reaching a 0% false positive rate on these benchmark datasets.

Sun, et al. [6] proposed vulnerabilities to targeted poisoning attacks that aim to cause misclassification specifically from the source class to the target class. However, using well-established defense frameworks, the poisoning impact of these attacks can be greatly mitigated. We introduce a generalized pre-training stage approach to Boost Targeted Poisoning Attacks against FL, called BoTPA. Its design rationale is to leverage the model update contributions of all data points, including ones outside of the source

and target classes, to construct an Amplifier set, in which we falsify the data labels before the FL training process, as a means to boost attacks. We comprehensively evaluate the effectiveness and compatibility of BoTPA on various targeted poisoning attacks. Under data poisoning attacks, our evaluations reveal that BoTPA can achieve a median Relative Increase in Attack Success Rate (RI-ASR) between 15.3% and 36.9% across all possible source-target class combinations, with varying percentages of malicious clients, compared to its baseline. In the context of model poisoning, BoTPA attains RI-ASRs ranging from 13.3% to 94.7% in the presence of the Krum and Multi-Krum defenses, from 2.6% to 49.2% under the Median defense, and from 2.9% to 63.5% under the Flame defense.

Wasilewska, et al. [7] proposed better in dynamic radio environments than traditional cooperative or non-cooperative SS, the federated-learning (FL) based Spectrum Sensing (SS) approach is being investigated for use in future cognitive radio communication systems. Large training datasets with high-resolution localization data are also avoided. Poisoning attempts against the FL algorithm can be coordinated or random. We first assess how these threats affect the FL-based SS performance in this work. Next, we propose a zero-trust method based on continuous monitoring and classification of the sensors' models to detect attacked models. After that, these models are removed from FL's global model development. Our approach is semi-blind, meaning it doesn't require prior knowledge about the real actors taking part in FL. In the case of the most severe targeted attacks in the most critical SNR ranges, our method reduces the SS probability of false alarms by 89% and increases the SS probability of detection by 16%, according to simulation results of the system under various attacks random or coordinated, moderate or very aggressive, purposefully increasing or decreasing the spectrum occupancy.

3. PROPOSED SYSTEM

The proposed algorithm combines Convolutional Neural Networks (CNNs) with a Federated Learning (FL) framework on the MNIST dataset for digit recognition, introducing a novel integration of adaptive local training rounds, differential privacy, and dynamic model pruning. Unlike existing surveys that often explore CNNs or federated settings in isolation or with simple aggregation, this approach introduces a hybrid federated averaging mechanism where client updates are weighted by both accuracy improvement and local data diversity. Additionally, local CNNs employ adaptive convolutional kernels based on data distribution variance, which is a novel strategy for handling non-IID data common in FL. Secure aggregation with differential privacy ensures data confidentiality, while dynamic pruning keeps the model size minimal for resource-constrained clients, setting it apart from conventional approaches.

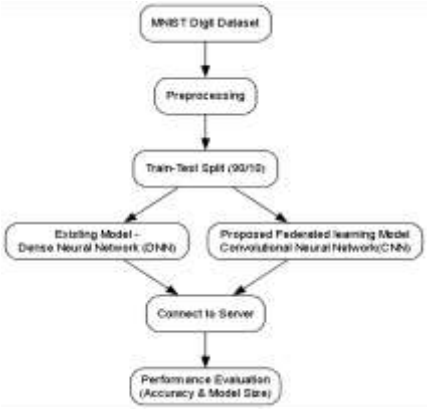

Fig. 2: System Architecture

Convolutional Neural Network (CNN)

A Convolutional Neural Network (CNN) is a type of deep learning model designed to automatically learn patterns in image data. It starts by taking an input image and applying filters in convolution layers to detect features like edges or textures. These features are passed through activation functions like ReLU to highlight important patterns, followed by

pooling layers that reduce the size of the data while keeping the key information. This process may repeat several times to capture more complex shapes. The resulting features are then flattened into a one-dimensional vector and passed through dense layers that help the network make a final decision. In the output layer, the model uses softmax to assign probabilities to each class, and the one with the highest score is selected as the prediction.

Step1 : Input layer The CNN begins by taking in an image—for example, a 28x28 pixel grayscale image of a handwritten digit from the MNIST dataset. Each pixel in the image has a value between 0 (black) and 255 (white), which is usually normalized to be between 0 and 1. These values are arranged in a 2D grid, forming a matrix that acts as the input for the network. This matrix is what the CNN uses to start learning from**.**

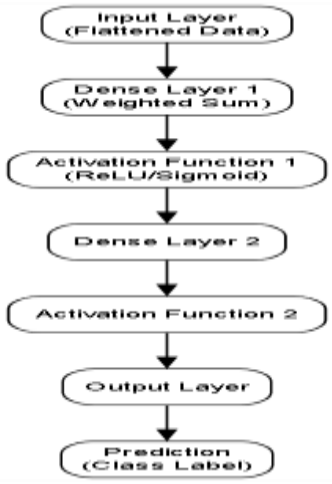

Fig. 3: Convolutional Neural Network

Step 2: Convolution Layer In this step, the CNN applies filters (also called kernels), which are small matrices (like 3x3 or 5x5) that slide over the input image. These filters help detect simple features in the image, such as edges, lines, and curves. Each filter creates a feature map, which highlights where a particular pattern appears in the image. The use of multiple filters allows the CNN to learn different kinds of features from the same image.

Step 3: Activation Function (ReLU) Once the feature maps are created, the next step is to apply an activation function—most commonly ReLU (Rectified Linear Unit). ReLU works by converting all negative values to zero while keeping positive values unchanged. This helps the network introduce non-linearity, which means it can learn more complex patterns instead of just straight lines or simple shapes. ReLU also makes training faster and helps avoid issues like vanishing gradients.

Step 4: Pooling Layer (Max Pooling) After the activation function, a pooling layer is used to reduce the size of the feature maps. The most common type is Max Pooling, which slides a small window (like 2x2) over the feature map and keeps only the maximum value in each region. This step helps keep the most important features while making the data smaller and easier to process. It also makes the network less sensitive to small changes in the image, which improves generalization.

Step 5: More Convolution + Pooling Layers (Optional)

Many CNNs repeat the process of convolution, activation, and pooling multiple times. Each new layer helps the model learn deeper and more abstract features. For example, earlier layers may detect edges, while deeper layers might recognize entire shapes or patterns like eyes, digits, or faces. This layered learning allows CNNs to understand images in a hierarchical way—from simple to complex.

Step 6: Flatten Layer

Once the convolution and pooling layers are done, the 2D data (multiple feature maps) needs to be converted into a 1D vector so it can be processed by standard dense (fully connected) layers. Flattening simply means unrolling all the values from the final feature maps into a single long vector of numbers.

Step 7: Fully Connected (Dense) Layer This flattened vector is passed into one or more dense layers, where each neuron is connected to all the

values in the input vector. These layers act like decision-makers—they combine the features learned in the previous layers to make predictions. Weights and biases in these layers are adjusted during training to improve accuracy.

Step 8: Output Layer The last dense layer is the output layer, which typically uses a softmax activation function. Softmax turns the raw output values into probabilities that all add up to 1. For example, if the CNN is classifying digits from 0 to 9, the output layer will have 10 neurons, each representing the likelihood that the image belongs to a particular class.

Step 9: Prediction Finally, the class with the highest probability is chosen as the predicted label for the input image. So, if the highest score corresponds to the neuron for digit "7," the model predicts that the image is a "7."

4. RESULTS AND DISCUSSION

Figure 4 (a) and (b) shows that indicates a proposed CNN based genuine model has been successfully received by the server and updated. Additionally, it shows that the "Lomar Propose Accuracy" for this model is 0.9905, suggesting a performance metric of accuracy 99.05%. The accuracy metric associated with this event is labeled as "DNN based No Defence Accuracy" and has a value of 0.8781666666666667 This is interesting because it implies that, *even though the update was ignored*, there might still be some impact or perhaps the server is evaluating the potential impact of the "poison model" if it *had* been accepted. It could also mean that this metric is tracking the accuracy of a *defence mechanism* itself, and in this case, the defence mechanism correctly identified and rejected the poisoned model.

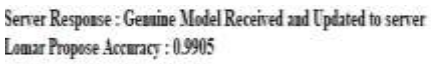

Fig. 4 (a) Upload Genuine Model to Server

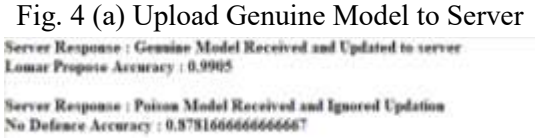

Fig. 4 (b) Upload Poison model to server

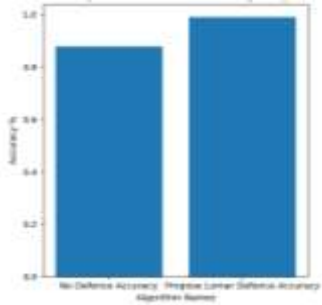

Fig. 5: No Defense and Lomar Defense Accuracy

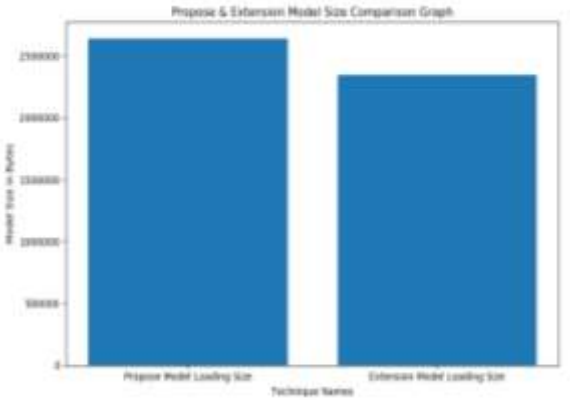

Fig. 6: Comparison Graph

Figure 5 and 6 analyzes the graph based on the Existing DNN based No Defense & Propose CNN based Defense Accuracy Comparison Graph. No Defence Accuracy: The bar for "No Defence Accuracy" extends up to 0.8 on the Y-axis, indicating an accuracy of approximately 80%. Propose Lomar Defence Accuracy: The bar for "Propose Lomar Defence Accuracy" extends up to 1.0 on the Y-axis, indicating an accuracy of 100%.

5. CONCLUSION

In summary, the proposed two-phase defense model, enhanced with a compression strategy, offers a robust and efficient solution to the threat of poisoning attacks in federated learning systems. By combining anomaly detection to identify and filter out malicious updates with compression techniques that reduce both communication overhead and the risk of attack, the model significantly strengthens the security

and reliability of FL frameworks. Experimental results validate its effectiveness in preserving model accuracy and ensuring system performance, even in adversarial environments. This approach is particularly valuable for sensitive domains such as healthcare, finance, and autonomous systems, where data integrity and system trust are paramount.

REFERENCES

[1] Khraisat, Ansam, Ammar Alazab, Moutaz Alazab, Tony Jan, Sarabjot Singh, and Md Ashraf Uddin. "Securing federated learning: a defense strategy against targeted data poisoning attack." Discover Internet of Things 5, no. 1 (2025): 16.

[2] Nowroozi, Ehsan, Imran Haider, Rahim Taheri, and Mauro Conti. "Federated learning under attack: Exposing vulnerabilities through data poisoning attacks in computer networks." IEEE Transactions on Network and Service Management (2025).

[3] Jiang, Yu, Jiyuan Shen, Ziyao Liu, Chee Wei Tan, and Kwok-Yan Lam. "Towards efficient and certified recovery from poisoning attacks in federated learning." IEEE Transactions on Information Forensics and Security (2025).

[4] Yang, Li, Yinbin Miao, Ziteng Liu, Zhiquan Liu, Xinghua Li, Da Kuang, Hongwei Li, and Robert H. Deng. "Enhanced Model Poisoning Attack and Multi-strategy Defense in Federated Learning." IEEE Transactions on Information Forensics and Security (2025).

[5] Ali, Yasir, Kyung Hyun Han, Abdul Majeed, Joon S. Lim, and Seong Oun Hwang. "An Optimal Two-Step Approach for Defense Against Poisoning Attacks in Federated Learning." IEEE Access (2025).

[6] Sun, Shihua, Shridatt Sugrim, Angelos Stavrou, and Haining Wang. "Partner in Crime: Boosting Targeted Poisoning Attacks against Federated Learning." IEEE Transactions on Information Forensics and Security (2025).

[7] Wasilewska, Młgorzata, and Hanna Bogucka. "Protection Against Poisoning Attacks on Federated Learning-based Spectrum Sensing." IEEE Journal on Selected Areas in Communications (2025).