IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 10 April 2025, accepted 28 April 2025, date of publication 9 May 2025, date of current version 30 May 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3568275

RESEARCH ARTICLE

A4FL: Federated Adversarial Defense via Adversarial Training and Pruning Against Backdoor Attack

SAEED-UZ-ZAMAN[1], BIN LI[1], MUHAMMAD HAMID[2], MUHAMMAD SALEEM[3], AND MUHAMMAD AMAN[4]

[1]School of Information Engineering, Yangzhou University, Yangzhou 225002, China
[2]Department of Computer Science, Government College Women University, Sialkot 51310, Pakistan
[3]Department of Industrial Engineering, Faculty of Engineering at Rabigh, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[4]Department of Industrial Engineering, College of Engineering, University of Business and Technology, Jeddah 21448, Saudi Arabia

Corresponding authors: Bin Li (lb@yzu.edu.cn) and Muhammad Hamid (mhamid@gcwus.edu.pk)

ABSTRACT Backdoor attacks threaten federated learning (FL) models, where malicious participants embed hidden triggers into local models during training. These triggers can compromise crucial applications, such as autonomous systems, when they activate specific inputs, causing a targeted misclassification in the global model. We recommend a strong defense mechanism that combines statistical testing, model refinement, and adversarial training methods. The primary goal is to develop a robust defense against backdoor attacks in federated learning (FL), where malicious participants embed hidden triggers into local models. This defense aims to preserve the integrity of the global model and ensure high reliability in real-world FL deployments, even when facing sophisticated adversarial strategies. Our defense strategy incorporates ''Messy'' samples with obvious triggers and ''wrap'' samples with similar but nonidentical triggers during adversarial training. This dual approach enhances the model's ability to detect and resist hidden manipulations. We facilitate applying neuron pruning to remove compromised neurons, further refining the model architecture for improved security. Continuous statistical testing, including variance analysis and cosine similarity checks, ensures that only legitimate and significant updates are integrated into the global model. A key innovation of our method is a significance-based filtering mechanism that effectively identifies and excludes malicious updates, preventing backdoor triggers from affecting the global model. This iterative defense process adapts to attack strategies, maintaining the model's robustness. Empirical results confirm that this defense mechanism significantly improves FL models' resilience to sophisticated backdoor attacks while preserving high accuracy and reliability. Balancing defensive strategies from adversarial training and sample diversification to model pruning provides a dependable framework for safeguarding FL models where integrity and security are critical. Experimental results demonstrate that our defense mechanism significantly enhances FL models' resistance to sophisticated backdoor attacks while maintaining high accuracy and reliability in real-world deployments. These solutions ensure the potential significance of balanced defense solutions, which offer strong protection against adversarial backdoor assaults. This framework provides a dependable solution for securing FL models in environments where integrity and security are paramount.

INDEX TERMS Backdoor attack, adversarial training, universal adversarial perturbations (UAPs), backdoor defense.

I. INTRODUCTION

Backdoor attacks have emerged as a dangerous deep learning (DL) menace. Reference [1], particularly endangering

The associate editor coordinating the review of this manuscript and approving it for publication was Ayman El-Baz.

DL-based applications for safety and security like autonomous driving [2], [3], [4]identification through visual or vocal recognition for communication [5], [6] Medical evaluation [7], and virus probing [8]. When enterprises and corporations incorporate machine learning into their operations, they are particularly concerned about these types of threats [9]. Backdoors may be integrated at several junctures within the deep learning framework [1], such as during model training [4], [10], data training [3], and federated learning [11]. Model training is a significant method for implementing backdoors. As a result, substantial endeavours have been undertaken to examine models to identify or minimize the impact of these hidden access points. However, whereas there have been numerous significant and methodical versatile assaults on adversarial defenses that aim to examine their robustness [12], [13], [14], [15], there has been far less exploration of significant adaptive assaults against backdoor protections to elucidate their robustness. When implemented in real-world scenarios, these backdoor countermeasures may lead to misleadingly positive security implications. Most initial assessments of adaptive attacks on backdoor countermeasures are improvised and only conducted with a single defense approach rather than many defenses. The objective is to determine the resilience of the suggested defense against an adaptive attack that is aware of the recently introduced defense technique [16], [17], [18]. However, a few comprehensive adaptable backdoors exist, [19], [20], [21]. These methods depend on model training regularization in a setting where an attacker influences the training, except for method [21]. Furthermore, most exploit designs incorporate specialized triggers that are undetectable and uninterpretable [20], dynamic [22] and spread in a broad manner [21]. The backdoor defenses will likely fail due to the presence of unique triggers that can bypass them. Some defenses have acknowledged their limits or anticipated failure scenarios when using these triggers: [17], [23], [24]. The adaptive assault has already violated these defenses and explicitly established a threat model. It is worth mentioning that the evaluation conditions for several of these adaptive attacks are inadvertently troublesome. These assaults, [19], [25], occur when an attacker who influences training data and procedure embeds a backdoor in a model. However, the suspect employs backdoor defenses. References [26] and [27] that necessitate the usage of a complete training dataset, which includes both benign and poisoned samples. Under optimal conditions, the user or defender should supervise the training process rather than the attacker. Assessing the strength of defense in conservative assault situations is of utmost importance. In these cases, the defender has valuable information regarding the attack strategy and trigger. During these instances, the attacker's skills are minimal. The premise of our argument is that if an assailant can effectively elude the protective measures in place under these rigorous circumstances, It is apparent that they might as easily bypass the defence under less stringent conditions. This can encompass scenarios where they can choose distinct triggers or utilize diverse means of attack. The Adversarily Defence against Backdoor Attack (A4FL) defense mechanism is specifically engineered to tackle these issues effectively. This system is highly effective at neutralizing a wide range of complicated poisoning risks, countering the tactics of advanced adaptive adversaries, and demonstrating its ability to resist different types of data scenarios

1) Our contribution offers distributed models from data poisoning, backdoors, and adversarial attacks that establish a set of robust evaluations. The attack focuses on Training data poisoning, Pre-trained model poisoning, unethical functionality, and adverse examples.
2) Introducing "A4FL," an innovative framework designed to enhance the security and integrity of FL, including advanced data validation protocols and collaborative anomaly detection techniques, to shield federated models from malicious inputs and model tampering. The framework's effectiveness is demonstrated through empirical tests and theoretical analysis, confirming its capability to maintain high fidelity in model performance under attack scenarios.
3) A4FL is superior in mitigating sophisticated attacks and provides actionable insights and strategies for effectively implementing defense mechanisms across various federated learning applications. Furthermore, the research offers guidelines for optimizing model training without data-sharing processes to enhance overall network robustness.
4) This process ensures that the model recognizes typical patterns in benign data and is resilient against the changes malicious actors might introduce to deceive automated detection systems.
5) Statistical testing validates the model's current performance and identifies potential areas for further improvement or hardening, making it an indispensable phase in developing robust neural network models.

The rest of the paper is organized as follows: In Section II, we review the background. In Section III, we describe the detailed intelligent backdoor attack. In Section IV, we discuss A4FL (Adversarial Defense Against Backdoor Attacks). In Section V, we demonstrate and evaluate the implementation of the methodology. In Section VI, we present the Experimental Assessments. Section VII provides the Conclusions.

II. BACKGROUND

A. BACKDOORING AND TAGGERS

The trigger and backdooring are the two essential elements of a backdoor attack. In an infected model, the trigger is implemented to initiate before the backdoor is placed. Creating unique trigger types or sophisticated backdoor varieties increases the backdoor's stealthiness or evasiveness. A patch with a predetermined pattern placed in a predetermined location is the most common trigger [28]. Current defenses can now efficiently catch triggers of this type [17]. There can be variations in the patch design and position [22],

[29], which can partially harden the countermeasures. Subsequently, several trigger designs were developed to be undetectable through frequency domain modulation. Applying a Selective Frequency-Injection approach, versatile frequency domain alterations on deteriorated face images. This approach affects the restoration process while maintaining imperceptibility [7], and subtle noise [30], [31]. Furthermore, natural triggers have also been employed, such as some spots in images [4], [5]. Furthermore, repositioning and appearance are examples of natural occurrences that might be used as triggers [32], [33]. A more elusive sample-specific trigger, [22], [34], and a hidden trigger that allows consistency between sample content and label [35], [36], are also developed. The trigger condition for a composite backdoor [37] is the simultaneous existence of several classes or object(s). Agnostic Attacks are particularly strongly backdoor format [1]. The backdoor embedded in an infected model will activate during such an attack, regardless class of input carrying the trigger. The backdoor will then be reprogrammed to perform the backdoor effect defined by the attacker. The other backdoor attack that is more well-known is a specific backdoor [16], [38], [39], which is also occasionally used as a partial backdoor [40]. In addition to the trigger being encoded in the input, the backdoor is also enabled when the attacker selects one of the source classes for the input. Even when the input is with the trigger, the backdoor does not show up if the input not come from the same class. Some related advanced backdoor variants expand upon the backdoor kinds mentioned above. One involves inserting many backdoors into a single model. Configuring a backdoor with a distinct attack objective is possible, allowing for distinct backdoor targets, such as distinct target labels. Every backdoor has the potential to be linked to either a unique trigger or a single trigger (all-to-all attack) [28]. In addition to these variations, hidden backdoors impact a pre-trained model [41] when the downstream job is learned using transfer learning and quantization backdoors [10]. KD (Knowledge Distillation) can be used to further increase model efficiency and privacy in federated learning, which involves several devices working together to train a model while maintaining local data [42]. A technique for identifying undesired intrusions in IoT networks is called Federated Learning (FL). With this approach, clients only communicate parameter updates with a central global server, ensuring privacy through federated training of local IoT device data [43]. Strong defenses, such anomaly detection and secure aggregation approaches, must be developed to protect the model against malicious impacts because these attacks have the ability to alter the model's decision-making process [44]. The trigger for any of these backdoor varieties might be simultaneously sample-specific or general (a patch). As stated alternatively, the backdoor category tends to be diagonal to the trigger. FLAME is a robust aggregation approach for Federated Learning (FL) that aims to mitigate the effects of backdoor assaults while maintaining the performance of the aggregated model on the primary goal. The difficulty of backdoor attacks in federated learning is in their capacity to undermine the global model by including malevolent updates without impairing the efficacy of legitimate updates. FLAME tackles this problem by presenting an effective approach to quantify the requisite noise to mitigate the impact of backdoors embedded in the global model [45].FLARE signifies a notable advancement over conventional Federated Learning systems by implementing a more efficient model initialization procedure and a trust-centric aggregation technique. By employing MMD for trust score estimation, FLARE fortifies the robustness and security of the federated learning process, guaranteeing that the global model receives precise, benign updates while alleviating the impact of unreliable or malicious contributions [46].

Several aggregation strategies are used by Federated Learning (FL) frameworks to guarantee secure, effective, and reliable model training among dispersed devices. Key FL frameworks are contrasted in the above Table1 according to their fundamental approaches, advantages, and disadvantages. Each strategy tackles issues like resource limitations, data heterogeneity, and adversarial attacks, which makes them appropriate for various use cases.

III. INTELLIGENT BACKDOOR ATTACK

A. MODEL TRAINING

When a backdoor is inserted, for instance through data poisoning independently the backdoored model typically leaves significant evidence of the hidden feature, which distinguishes trigger inputs from benign inputs. Most backdoor attacks attempt to obscure the distinction between the hidden representations of benign samples and those containing triggers [21]. Due to their observation, most backdoor defenses use the latent separability among benign and trigger samples to identify the backdoored model or trigger samples. Adding extra loss to training regularisation is a frequent technique for achieving this [19], [25]. To obtain poisoned photos with the trigger, [20] first employs a trigger generator. Then, using an interactive backdoored model retraining technique, they block the activations of the affected neurons during backdoor training. Even though the backdoored training only costs an attacker once, it has a significant computing expense because of the trigger generator's necessary training and the iteratively regulated detoxification process. Similar studies, [22] and [34], also require co-training of the trigger generator in addition to the backdooring model. However, they accomplish the same regularisation effect at the end without explicitly regularising the backdoored model's hidden representation. Training regularisation is similar to all of these adaptive attacks. Furthermore, they are all dependent on complex triggers, such as those that disseminate the complete picture or that are sample-specific or aware [20], [34] which have already compromised the threat model of several of the backdoor defenses they have assessed [16], [17], and [24]

TABLE 1. Comparison of defense mechanisms.

Framework	Methodology	Strengths	Limitations
FLAME	Model Averaging with outlier detection	Scalable; mitigates poisoning attacks	Limited adaptability to non-IID data
FLARE	Reinforcement Learning (RL)-Driven Adaptation	Self-optimizing; handles dynamic environments	Requires frequent hyperparameter tuning
Median	Median-Based Aggregation (statistical robustness)	Resistant to outliers; simple implementation	Fails with skewed/non-IID distributions
FABA	Filtering-Based Aggregation	Lightweight; edge-device-friendly	Low accuracy in complex tasks; limited scalability
Kurm	Resource-Constrained Optimization	Energy-efficient; ideal for IoT devices	Poor performance in high-stakes environments
Trum	Decentralized Consensus (blockchain-like P2P updates)	No single point of failure; trustless design	Slow convergence; communication-heavy
FLTrust	Server-Validated Trust Scores (centralized authority)	Strong against malicious clients	Server dependency; privacy risks
FedSGD	Standard Federated SGD (baseline aggregation)	Simple; widely compatible	Vulnerable to attacks; no built-in robustness

because they employ defenses primarily designed for data relocation backdoors, where the user/defender can access the entire training data and train a model on their own, a few evaluations of model relocation adaptive attacks [19], [25] seem problematic.

B. DATA TRAINING

The attacker cannot regulate the model training in this attack surface to impose regularisation during the training phase. Consequently, adaptive assaults depend on unique trigger designs nearly all of them exploit complex triggers, such as sample-specific [34], dynamic [22], distributed [21], and partial [38], source-class-specific [39] triggers. It is not unexpected that they can quickly get beyond certain defenses since the application of the triggers frequently violates the model assumptions, particularly those model defenses [17], [23], and online data diagnosis defenses [16], [24], [47]. Furthermore, the specific backdoor assault [16], [38], and [39] is a particular backdoor that can be considered as an attack on the typical backdoor defenses. It fundamentally contravenes such defence assumptions due to their nature. It is worth noting that [21] is the only study that have no influence over the training procedure. This fits the description of an adaptive assault on data relocation, where the attacker can implant a backdoor by poisoning a small percentage of the data. They use regularization/wrap trigger samples to make the hidden representation of the benign and trigger, hence avoiding latent separability-based defenses. The samples with the trigger and intact labels are regularised samples. The wrap is fundamentally the foundation of hidden separation-based defences, characterised by robust tactile connections among the trigger and the target class. The use of wrap samples can reduce the accuracy of clean samples. This issue can be resolved by employing distributed partial triggers, which decompose the comprehensive trigger into tiny triggers utilised for generating a poisoned or wrap sample, alongside asymmetric trigger poisoning, This renders the trigger visible during poisoning but invisible when the attack is executed during online deployment. To some extent, clean-label poisoning assaults [48] can be considered adaptive attacks since they can evade the visual inspections of the data curator. These assaults depend on either feature collision or image concealment [3], [35], [36], [49], [50], [51]. Because there is consistency between the label/annotation and the image content, it is simple to defeat them. More precisely, the feature collision frequently fails when the user starts training from scratch using a different model architecture since it needs information about the target's model (i.e., weight values). To significantly improve the attack, not many studies have attempted to do so [52], [53]. Regarding the picture camouflage that misuses the image resizing feature, understanding the input size and resizing the data into the model training is required. Once these two variables change, it fails trivially [3].

C. THREAT MODEL

Under the A4FL (Adversarial Attacks against Federated Learning) framework, we posit the existence of an adversary capable of influencing a subset of clients in a federated learning environment. The adversary aims to embed a backdoor within the global model while circumventing detection systems. The system employs the FedAVG algorithm for aggregating model updates, while the adversary seeks to manipulate this aggregation process to fulfill its objectives. Adversarial Skills The adversary controls a set of compromised clients, which is the number of compromised clients and represents the total number of clients. The global model update in each training round t is computed using FedAVG as follows:

$$\theta^{t+1} = \theta^t + \alpha \cdot \frac{1}{m} \sum_{j=1}^{m} \Delta\theta_j^t \tag{1}$$

The global model parameters are at the round, the global learning rate, and the client's update. The adversary manip-

ulates the updates from compromised clients to implant a backdoor while keeping the overall model accuracy intact. Backdoor Insertion: The adversary ensures that any input containing a predefined backdoor trigger is misclassified as a target label, regardless of the input's label. The behavior of the backdoor can be represented as:

$$g(z+\tau;\theta^{t+1}) = y_{target} \quad \forall z \in \mathcal{Z} \tag{2}$$

The global model suction function is the backdoor trigger pattern. The adversary carefully constructs the updates to ensure the backdoor is implanted without degrading the mod model's performance on clean data. The adversary aims to maximize the prediction success rate (PSR) on inputs containing the backdoor trigger while minimizing the evasion rate (ER) to evade detection. The adversary optimizes the following objective function for each compromised client:

$$\max_{\Delta\theta^t_{adv}} \left[PSR(\theta^{t+1}, \tau) \right], \quad \text{subject to} \quad ER(\theta^{t+1}, \mathcal{D}) < \beta \tag{3}$$

where: $PSR(\theta^{t+1}, \tau)$ is the backdoor success rate on inputs with trigger, $ER(\theta^{t+1}, \mathcal{D})$ is the evasion rate concerning the detection mechanism, β is the threshold for the maximum allowable evasion rate before detection is triggered.

$$PSR(\theta^{t+1}, \tau) \geq 90\% \tag{4}$$

Ensure that the Evasion Rate (ER) remains below the detection threshold, reflecting the upper limit before the adversaries' actions raise suspicion. The adversary operates under several constraints: Predefined Trigger: The backdoor must use a predefined trigger, such as a static visual patch (e.g., a white square). The trigger is simple, static, and detectable by design. Input-Agnostic Backdoor: The backdoor must be input-agnostic, i.e., for any model sections, it will be classified as, regardless of the actual label. Formally:

$$g(z+\tau;\theta^{t+1}) = y_{target} \quad \forall z \in \mathcal{Z}, \quad \tau \in \mathcal{T} \tag{5}$$

The adversary must generate 200 backdoored models, each exhibiting the same backdoor characteristics. The defender has access to 200 clean models for comparison, denoted as which increases the risk of detection through statistical analysis. The system presumes that the most trustworthy clients revisit the model. During each training iteration, the server consolidates updates, assuming that benign clients participate. Unlike many previous mod models, suction assumptions about data distribution across clients allow for non-IID (non-Independent and Identically Distributed) data across different clients. This increases the complexity of detecting adversarial updates as methods relying on uniform or IID data become ineffective. The final global model update is computed as follows:

$$\theta^{t+1} = \theta^t + \alpha \cdot \frac{1}{m} \sum_{j=1}^{m} \Delta\theta^t_j \tag{6}$$

IV. A4FL (ADVERSARIAL DEFENSE AGAINST BACKDOOR ATTACK)

In Figure 1, our defense hides triggers in clean images, resulting in poisoned data samples. These samples are the model, embedding vulnerabilities linked to these specific triggers. Features are extracted from the poisoned data. The model leverages the strengths of both CNNs for spatial data and LSTMs for temporal or sequential data patterns. This combination is ideal for handling complex features that might be manipulated through adversarial inputs. Adversarial training involving Universal Adversarial Perturbations (UAPs). These perturbations aesthetically enhance the model's resilience by forcing it to learn under adversarial conditions. The adversarial inputs are processed through instance-segmentation techniques, combined with UAPs, and used for ongoing training. Concurrent with training, the model is pruned to eliminate unnecessary or less essential neurons, which could be potential weak spots for attacks. Fine-tuning follows the initial pruning to optimize the model's performance against the adversarial examples it has encountered. Various statistical measures (like minimum and maximum values, variance, cosine similarity, and Euclidean distance) are applied to evaluate the model's outputs and identify any remaining biases or weaknesses that adversarial attacks could exploit. The output from the model's processing of adversarially trained data is fed back into the system. This feedback loop is crucial for iterative refinement, allowing the model to adapt and improve continuously based on the latest adversarial strategies it encounters.

The features might be standardized or normalized to prepare for more effective model training. The model is exposed to both malicious (poisoned) and benign inputs. This training is likely aimed at making the model robust against adversarial attacks. The model architecture combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs), which are adequate for tasks involving spatial and sequential data. Once the local models upload updates, the central server applies statistical tests to identify anomalies. These tests are based on various metrics like minimum and maximum values (Min, Max), variance (Var), cosine similarity (Cos), and Euclidean distance (EUCL). Models that fail these tests are considered poisoned and are filtered out. After filtering out the poisoned models, the remaining benign models are aggregated to update the global model. This step is crucial for learning a general model that performs well across all data from participating devices. The entire process, from feature extraction to global model update, is iterative. This means the system continually refines the model based on new data and ongoing threat assessments to ensure robustness against poisoning. This architecture is designed to protect against poisoned model attacks in federated learning environments, where the model's integrity is crucial, given that any participant could potentially introduce harmful data. Using traditional and neural network-based approaches for

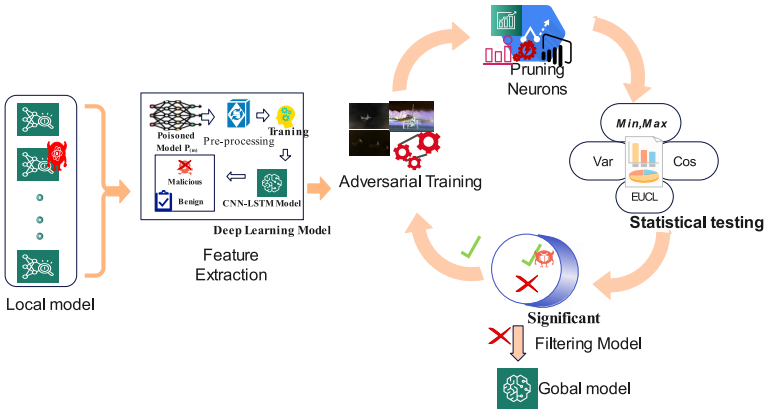


FIGURE 1. Overview of A4FL.

Algorithm 1 A4FL Enhanced Trigger Specificity in Model Training

Clean dataset D, backdoor trigger δ, target backdoor label y_{backdoor}, number of images to be poisoned m. Poisoned model trained with enhanced specificity against similar triggers.

// Create Poisoned Images

$i = 1\ m\ t(x_i) \leftarrow x_i + \delta$

Replace (x_i, y_i) in D with $\left(t(x_i),\ y_{\text{backdoor}}\right)$

End Create Poisoned Images

// Create Messy Samples

$D_a \leftarrow \emptyset$

$i = 1\ m\ x_a \leftarrow x_i + \delta$

$D_a \leftarrow D_a \cup \left\{(x_a,\ y_{\text{backdoor}})\right\}$

End Create Messy Samples

// Create Wrap Samples

$D''_\theta \leftarrow \emptyset$

$i = 1\ m\ x_b \leftarrow x_i + \delta'$

$D''_\theta \leftarrow D''_\theta \cup \left\{(x_b,\ y_i)\right\}$

End Create Wrap Samples

// Compute Loss

$\mathcal{L}_{\text{obj}} \leftarrow 0$

$(x, y)\ D \cup D' \cup D''_\theta\ \mathcal{L}_{\text{obj}} \leftarrow \mathcal{L}_{\text{obj}} + \mathcal{L}\left(Y_{\text{bd}}(x),\ y\right)$

End Compute Loss

$D' \leftarrow \text{CreatePoisonedImages}\left(D, \delta, m\right)$

$D_a \leftarrow \text{CreateDirtySamples}\left(D, \delta, n, m\right)$

$D''_\theta \leftarrow \text{CreateCoverSamples}\left(D, \delta', m\right)$

$\mathcal{L}_{\text{obj}} \leftarrow \text{ComputeLoss}\left(D,\ D_a,\ D''_\theta,\ Y_{\text{bd}}\right)$

$P_\omega \leftarrow$ Train model minimizing $\mathcal{L}_{\text{obj}}$

P_ω

detecting anomalies highlights a comprehensive strategy to maintain model integrity.

V. IMPLEMENTATION

Clean dataset consisting of input images and their corresponding accurate labels.$D_\theta = \{(x_i, y_i)\}_{i=1}^{n}$. Defenses upon trigger reconstruction are identical to the actual trigger, similar triggers making the data contaminated. The approach relies on unpredictable optimization. However, the impact of these similar rebuilt triggers may resemble the genuine trigger because of the genuine trigger's lack of specificity. In this context, we propose to improve trigger specificity by employing wrap triggers. When generating poisonous samples for model training, two categories of poisonous samples are identified: Messy samples and wrap samples.

Trigger Design can create a backdoor trigger δ, a pattern added to the images. This pattern is designed to be easily detected by the model but not noticeable to humans.

δ = Trigger Pattern Add a trigger to a subset of the clean images, creating poisoned images.

$$t(x_i) = x_i + \delta \tag{7}$$

where the modified image with the trigger is.

Creating Poisoned Images: Combine the clean and poisoned images to create a specific target label for the images containing the trigger and poisoned dataset.

$$D'_\theta = \{(t(x_i), y_{\text{backdoor}})\}_{i=1}^{m} \cup \{(x_i, y_i)\}_{i=m+1}^{n} \tag{8}$$

where: $\{(t(x_i), y_{\text{backdoor}})\}_{i=1}^{m}$ are the images with the trigger and target label.

$\{(x_i, y_i)\}_{i=m+1}^{n}$ Aretha remaining clean images.

· Messy samples: A trigger selected by the attacker is incorporated into a randomly chosen untainted sample x to produce a toxic sample. The label is modified to be y_a the assailant-focused category. In this context, a tiny dataset of Messy Samples are created.

· Wrap sample: A randomly produced wrap trigger, akin to be not identical to the original, is appended to a randomly chosen clean sample x to produce a toxic sample. However, it is important to observe that the label on the wrap sample remains intact, representing its ground-truth label. A limited dataset of wrap samples is generated in this context. Figure 2 illustrates an example of wrap triggers. The wrap samples will compel the model to associate solely with the existence of certain triggers, rather than analogous triggers. In other words, it improves trigger specificity, making it challenging to reconstruct the precise trigger during nondeterministic

optimisation. The objective loss $\mathcal{L}_{\text{obj}}$ of the backdoor model Y_{bd} the enhancement of trigger specificity can be articulated as:

$$\mathcal{L}_{\text{obj}} = \sum_{x \in D_\theta} \mathcal{L}(y_{\text{bd}}(x_i), y_i) + \sum_{x_t \in D'_\theta} \mathcal{L}(y_{\text{bd}}(x_a), y_a) + \sum_{x_c \in D''_\theta} \mathcal{L}(y_{\text{bd}}(x_b), y) \tag{9}$$

Training the Poisoned Model P_ω Train the model using the poisoned dataset, optimizing the model's parameters to minimize the loss on clean and poisoned data.

$$P_\omega = \text{Train}(D_\theta, D'_\theta, D''_\theta) \tag{10}$$

where is the model trained on the poisoned data?

Figure 2 Step 1 involves creating a poisoned dataset by injecting triggers into clean images, forming a poisoned model that misclassifies inputs containing the triggers. This process is evaluated using metrics such as ASR, Trigger Recall, Logit, and Probability Change to assess the effectiveness of the attack and the model's vulnerability.

In Figure 2, Step 2 and the provided optimization, each layer of the CNN-LSTM model process inputs through these enhanced layers, extracts features, and prepares the data for subsequent steps. This step involves taking a potentially poisoned model, processing the data through a series of optimized layers in the CNN-LSTM architecture, and effectively training the model to distinguish between malicious and benign features. The ultimate goal is to robustly extract and learn from the data's spatial and temporal aspects.

Normalization and other pre-processing methods ensure the input data is scaled and formatted correctly for neural network processing, enhancing model training efficiency and effectiveness.

Applying enhanced convolution operations with Batch Normalization and LeakyReLU activation.

$$x_a^i = \text{LeakyReLU}\left(\text{BatchNorm}\left(\sum_{j \in k_i} x_{a-1}^j \cdot w_a^{ji} + b_a^j\right)\right) \tag{11}$$

This layer extracts initial spatial features from the input data. Batch normalization stabilizes learning by reducing internal covariate shift, and LeakyReLU prevents dead neurons by allowing a slight gradient when the unit is inactive.

Applying MaxPooling to reduce spatial dimensions followed by Dropout to prevent overfitting.

$$x_a^i = \text{Dropout}\left(\text{MaxPool}\left(x_{a-1}^i\right)\right) \tag{12}$$

Reduces the dimensionality of the feature maps, making the network less prone to overfitting and computationally efficient while retaining essential features. Applying a dropout layer before fully connecting processing with an ELU activation function.

$$y_m = \text{ELU}\left(\text{Dropout}\left(w_m \cdot x_{m-1} + b_m\right)\right) \tag{13}$$

Incorporates advanced LSTM configurations with peephole connections and layer normalization for optimal temporal feature extraction.

$$f_t^i = \sigma\left(b_f^i + \sum_j Z_f^{ij} x_t^j + \sum_j D_f^{ij} h_{t-1}^j + P_f^{ij} c_{t-1}^j\right) \tag{14}$$

Input Gate, Cell Update, and Output Gate follow similar enhancements with corresponding operations. When leveraging direct access to the cell state for more accurate information gating and output generation. The model undergoes training where it learns to identify and classify features as benign or malicious based on the extracted features. This step involves using historical data (labels for training) to adjust the network weights to minimize loss, thereby increasing the model's predictive accuracy on new, unseen data. Through these enhancements, the CNN-LSTM model in Step 2 becomes highly effective in handling complex patterns, dealing with non-linearities, and preventing overfitting. This results in a robust system that can effectively differentiate between standard and adversarial inputs in the dataset, which is crucial for applications requiring high reliability and accuracy in the presence of potential data poisoning.

The adversarial training fine-tuning process in Step 3 in Figure 2 involves describing how Universal Adversarial Perturbations (UAPs) are generated and applied, how the adversarial training is structured, and how the model is fine-tuned.

Where is the output of the CNN-LSTM model, where is the input image or sequence of images, and represents the model's parameters. The model's output might be a vector of probabilities for classification or a set of feature maps for segmentation tasks.

$$\Delta = \arg\max_\delta \mathbb{E}_{x \sim X}[L(f(x+\delta;\theta), y)] \tag{15}$$

When the perturbation is added to input samples, it is a loss function (like cross-entropy), the actual label, and the original data distribution. Some norm constrains the perturbation to ensure it's small and invisible. The adversarial training modifies the standard training objective to include examples perturbed by. The new objective function for training becomes:

$$\min_\theta \mathbb{E}_{x \sim X}[L(f(x;\theta), y) + L(f(x+\Delta;\theta), y)] \tag{16}$$

This helps the model learn to classify both clean and perturbed inputs accurately. If the task involves instance segmentation, the objective might be to minimize a segmentation-specific loss, such as the Dice loss or Intersection over Union (IoU), on both clean and adversarially perturbed images:

$$\min_\theta \mathbb{E}_{x \sim X}[L_{seg}(seg(x;\theta), y_{seg}) + L_{seg}(seg(x+\Delta;\theta), y_{seg})] \tag{17}$$

where the segmented maps and the accurate segmentation maps are output. Training batches are composed of original,

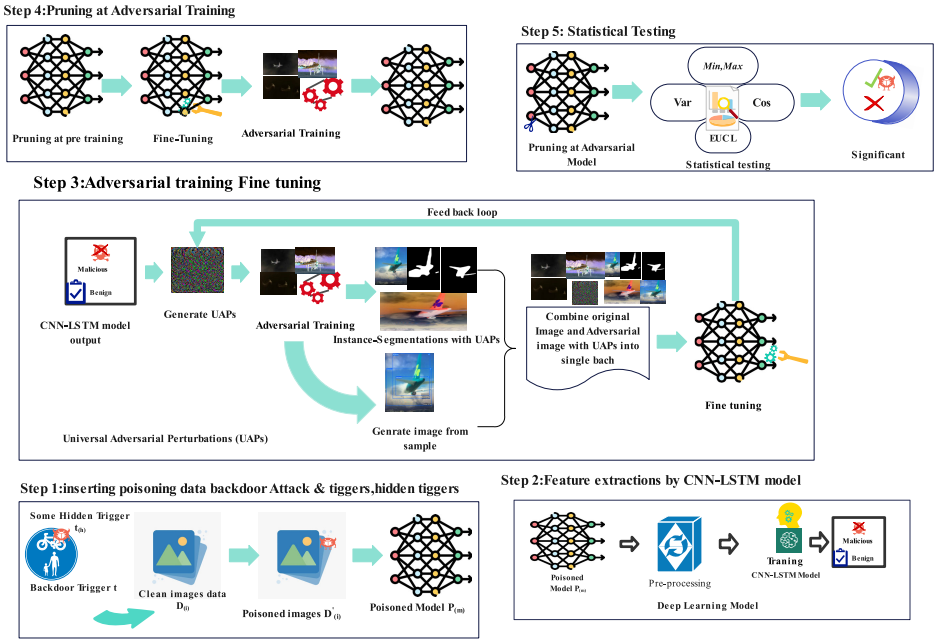


FIGURE 2. The architecture of A4FL.

adversarially modified, and UAP-applied images. This can be represented as training on the combined objective:

$$\min_{\theta}[\frac{1}{3}(L(f(x;\theta),y)+L(f(x+\Delta;\theta),y) +L(f(x+UAP;\theta),y))] \tag{18}$$

Fine-tuning involves adjusting the model parameters θ to minimize further the loss on a validation set, possibly including UAPs, refining the model's resilience:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{x\sim X_{val}}[L(f(x;\theta),y)] \tag{19}$$

where it includes a mix of clean and perturbed images. The feedback loop suggests an iterative optimization where the generation of UAPs and the model training are repeated with updated parameters:

$$\Delta_{new} = \arg\max_{\delta} \mathbb{E}_{x\sim X}[L(f(x+\delta;\theta^*),y)] \tag{20}$$

$$\theta^{new} = \arg\min_{\theta} \mathbb{E}_{x\sim X}[L(f(x+\Delta_{new};\theta),y)] \tag{21}$$

This iterative process aims to continuously refine the adversarial perturbations and the model's resistance to them, leading to robust performance under adversarial conditions. Several advanced training strategies involving pruning and learning rate management exist in neural networks. Step 4: Figure 2 Each method provides a theoretical explanation and a mathematical framework for understanding its implementation and purpose.

The strategies mentioned are part of advanced neural network optimization techniques that enhance model performance and efficiency after the initial training phase. These strategies, particularly in pruning, leverage the concept that not all weights in a neural network contribute equally to the final performance, and some can be removed without significant loss of accuracy. Fine-tuning involves continuing the training of a pruned network using the weights that survived the pruning process. The idea is to allow these weights to adjust to the reduced complexity of the network.

$$\text{TRAIN}_t(W_T; m; T) = W_T \odot m \tag{22}$$

Here, it represents the network's weights at the end of the original training process and is the pruning mask. The function retrains the network for additional epochs starting from the final epoch using the last learning rate. Weight Rewinding is used to investigate if earlier stages of training could potentially converge to a better or more efficient model when retrained from a "rewound" state with the pruned architecture.

$$\text{TRAIN}_t(W_{T-t}; m; T-t) \tag{23}$$

This method resets the weights of the unpruned network back to their values epochs before the end of training (at $T-t$), applying the pruning mask and retraining from this earlier state. Learning Rate Rewinding is a hybrid approach combining fine-tuning and weight-winding elements. It starts with the final pruned model but rewinds the learning rate schedule to epochs before the end of the training, aligning more with the network's critical learning periods.

$$\text{TRAIN}_t(W_T; m; T-t) \tag{24}$$

Unlike weight rewinding, which also rewinds the weights, learning rate rewinding uses the final weights but applies the learning rate schedule from epochs before the end of training.

These methods are crucial for understanding and improving the resilience and efficiency of neural networks, huge models prone to overfitting or requiring extensive computational resources. Fine-tuning is often used to refine the network's predictions without significant structural changes, improving accuracy on the pruned network. Weight rewinding can disperse more effective or efficient network architectures by re-exploring the training landscape with fewer parameters. Learning Rate Rewinding aims to capitalize on the dynamic adjustments of learning rates towards the end of the training, possibly improving convergence or stability after significant changes to the network structure. These advanced training techniques provide powerful tools for optimizing neural networks post-pruning, each with specific benefits and suitable applications depending on the goals and constraints of the project. The choice of method can significantly affect the network's performance, efficiency, and overall effectiveness in practical applications.

In Step 5, Figure 2 describes rigorous statistical testing to evaluate and validate the performance of the pruned and adversarially trained neural network model. This stage is critical to ensure that the model's modifications have not compromised its accuracy and remain effective under different scenarios, including potential adversarial attacks. In this case, the main goal of statistical testing is to see how stable and reliable the neural network is after pruning and adversarial training. This is done by applying various statistical tests to determine the significance of the neuron remaining in the pruned model and to confirm that the model's performance meets the required standards for deployment. The performance of the pruned model is evaluated against a set of test data. This step is crucial for assessing the model's effectiveness in handling real-world data after it has been simplified and hardened against attacks.

$$\text{Significant Neurons} = \{N_i \mid \text{Statistical Test}(N_i) > \alpha\} \quad (25)$$

Here, N_i it represents individual neurons and is a threshold for determining significance based on the p-values from statistical tests applied to the neurons'' contributions.

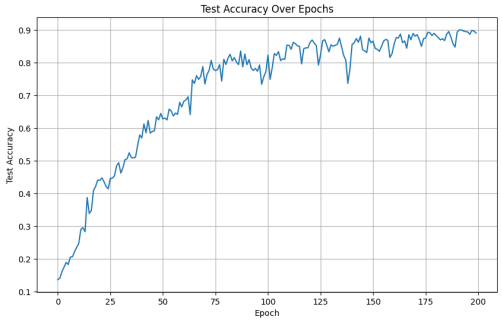


FIGURE 3. Test accuracy over 200 epochs of A4FL.

VI. EXPERIMENTAL ASSESSMENTS

This section assesses the proposed A4FL scheme across five dimensions: backdoor assaults in data poisoning, contaminated triggers, contaminated concealed triggers, Messy samples, wrap samples, and defensive efficacy. We execute a backdoor attack through data poisoning and tainted triggers on CIFAR-10 datasets and assess the effectiveness of our defense against these attacks. The CIFAR-10 dataset comprises 60,000 color images categorized into ten classes (including ''aircraft,'' ''car,'' ''bird,'' etc.), consisting of 40,000 training samples and 10,000 test samples. The CIFAR-10 dataset images are normalized to a 32×32 three-channel format during data preprocessing. In the characteristic backdoor attack experiment involving ''Aeroplane'' from the CIFAR-10 dataset, ''Aeroplane with the surrounding environment,'' various elements of the surrounding environment, such as the sky, clouds, and parts of the aircraft, are designated as backdoor triggers, along with additional images selected as triggers [1].

A. EXPERIMENTAL SETTINGS

We developed a prototype system for federated learning utilizing the TensorFlow 2.7 framework. The system configuration comprises an Intel i5 12400 GPU and an Nvidia GeForce RTX 3060 with 12GB. The software environment consists of the Windows 10 operating system and Python 3.11.9. This investigation employs convolutional neural networks (CNN) in feature extraction. Table 2 outlines the architecture of the neural networks used in our experimental setup in the CIFAR-10 dataset. Conv1 Layer Uses a convolutional operation to create 142 $\times$ 64 feature maps, then Leaky ReLU activation and batch normalization to make training more stable and speed up convergence. Conv2 Layer: processes the output of Conv1 to produce 72 $\times$ 128 feature maps. This layer incorporates Leaky ReLU activation, Batch Normalisation, and Dropout to prevent overfitting. Conv3 Layer: Further convolves the data to obtain 42 $\times$ 256 feature maps with Leaky ReLU activation and batch normalization. Global Average Pooling (GAP) Aggregates the spatial dimensions of the feature maps, resulting in a 256-dimensional vector summarising the presence of features. Dense Layer A fully connected layer reduces the dimensionality to 10 $\times$ 1, applying Leaky ReLU activation to introduce non-linearity.

In the experimental investigation illustrated in Figure 2, our defense system involves placing concealed triggers within pristine images and subsequently using contaminated data samples to train the model, incorporating vulnerabilities associated with these particular triggers. The procedure comprises multiple critical phases, each aimed at augmenting the model's strength and durability against adversarial assaults.

B. INCORPORATING CONCEALED TRIGGERS AND FEATURE EXTRACTION

Incorporating Concealed Triggers and Feature Extraction Initially, pristine photos are implanted with concealed triggers, resulting in contaminated data samples. The samples train the model, incorporating specific vulnerabilities associated with these triggers. The model utilizes a hybrid approach,

TABLE 2. Structure of neural network encoder and decoder.

Layer	Output Dimensions	Description
Encoder		
Input	$28 \times 28 \times 1$	-
Conv1	$28 \times 28 \times 64$	Conv, SiLU, LN
Conv2	$14 \times 14 \times 128$	Conv, SiLU, LN, MaxPool
Conv3	$7 \times 7 \times 256$	Conv, SiLU, LN, MaxPool
Transformer Encoder	$7 \times 7 \times 256$	Multi-Head Attention, LN, Residual Connections
Flatten	12,544	-
Dense	512	Dense, SiLU, LN
Latent Space	128	-
Decoder		
Input	128	-
Dense1	$7 \times 7 \times 256$	Dense, SiLU, LN, Reshape
Transformer Decoder	$7 \times 7 \times 256$	Multi-Head Attention, LN, Residual Connections
ConvT1	$14 \times 14 \times 128$	ConvTranspose, SiLU, LN, Upsample
ConvT2	$28 \times 28 \times 64$	ConvTranspose, SiLU, LN, Upsample
Output Layer	$28 \times 28 \times 1$	Conv, Sigmoid

integrating Convolutional Neural Networks (CNNs) for spatial data and Long-Short-Term Memory (LSTM) networks for temporal or sequential data patterns. This combination is especially perfect for managing intricate characteristics that adversarial inputs may influence, as CNNs are proficient in identifying spatial patterns, but LSTMs are skilled at capturing temporal dependencies. In this step, features are derived from the contaminated data to analyze the impact of the triggers on model performance. This extraction is essential for discerning the attributes of the compromised data that render the model susceptible and for equipping the model for future adversarial training.

C. ADVERSARIAL TRAINING UTILIZING UNIVERSAL ADVERSARIAL PERTURBATIONS (UAPS)

Adversarial Training Utilizing Universal Adversarial Perturbations (UAPs) The model engages in adversarial training with Universal Adversarial Perturbations (UAPs). UAPs are disturbances designed to mislead the model across diverse inputs, rendering them an efficient instrument for evaluating and improving its resilience. These perturbations compels the model to learn in adversarial settings, enhancing its robustness. Adversarial inputs are analyzed using instance-segmentation methods, integrated with Universal Adversarial Perturbations (UAPs), and employed for continuous training. This procedure guarantees that the model learns to identify pristine features and becomes acquainted with adversarial patterns, enhancing its ability to defend against such attacks in practical situations.

1) PRUNING AND OPTIMIZATION

A pruning method is implemented during training to enhance the model's performance. Pruning entails the elimination of redundant or subordinate neurons from the model, which may otherwise represent possible vulnerabilities to hostile assaults. Pruning enhances generalization and robustness by diminishing the model's complexity, as shown in Figure 4,

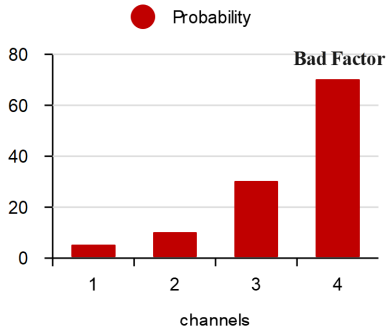


FIGURE 4. Pruning neurons (a).

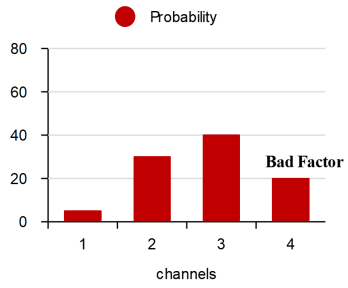


FIGURE 5. Pruning neurons (b).

5, 6. After trimming, the model is subjected to fine-tuning to enhance its efficacy against the adversarial examples it has faced. Fine-tuning facilitates the adjustment of residual neurons and synapses to uphold elevated precision while guaranteeing robustness. This stage is crucial for achieving a balance between model simplicity and efficacy.

D. STATISTICAL ASSESSMENT OF MODEL RESILIENCE

Multiple statistical metrics assess the model's resilience following adversarial training and pruning. These include Minimum and Maximum Values, as shown in Figure 11. Comparison of ER and PSR in Federated Learning Models

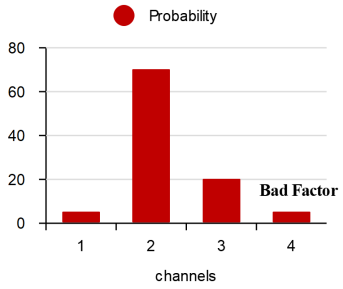


FIGURE 6. Pruning neurons (c).

Across Different Frameworks. These check the output range and ensure the model's responses stay within the expected limits. Variance: To assess the reliability of the model's predictions. Cosine Similarity: A metric for evaluating the similarity between several feature representations, aiding in identifying inconsistencies that may signify vulnerabilities. Euclidean Distance: A metric used to measure the disparity of outputs, aiding in identifying abnormalities or vulnerabilities susceptible to adversarial assaults. Statistical measurements are utilized on the model's outputs to detect residual biases or vulnerabilities susceptible to adversarial attacks. By meticulously assessing the model's performance, we can ascertain its robustness against various adversarial tactics. The output generated from the model's analysis of adversarially trained data is reintegrated into the system, creating a feedback loop. This feedback loop is essential for iterative enhancement, enabling the model to adapt and improve continuously in response to the most recent adversarial techniques it faces. Every iteration enhances the model's resilience by allowing it to learn from previous flaws and adapt to novel adversarial inputs. This iterative method guarantees the model stays current and adept at countering emerging hostile threats. The experimental study seeks to create a robust and resilient model capable of withstanding various adversarial attacks by integrating adversarial training, pruning, fine-tuning, statistical evaluation, and a continuous feedback loop.

VII. EVALUATION METRICS

The assessment of the experimental study relies on an extensive array of measures that evaluate the model's efficacy and resilience. Assesses the ratio of correctly identified samples, comprehensively considering the model's efficacy on untainted and compromised datasets. The effectiveness of the adversarial attack is evaluated by determining the proportion of samples that were accurately misclassified following the incorporation of concealed triggers. A reduced ASR in Table 3 signifies enhanced model robustness [54]. Assesses the frequency with which legitimate samples are erroneously identified as hostile. A reduced false positive rate is preferable, meaning the model can differentiate between legitimate and hostile inputs. Determines the model's capacity to identify and appropriately react to inputs that include concealed triggers. Elevated recall signifies that the model effectively recognizes the triggers in the contaminated samples. Measures the variation in output logits and projected probabilities from applying adversarial perturbations. Substantial alterations may signify weaknesses, but stability implies resilience, which is used to assess the resemblance between the feature representations of untainted and adversarial samples. A high degree of similarity signifies that the model is less susceptible to adversarial perturbations, enhancing its robustness. Review the distance between the feature representations of untainted and adversarial samples. Reduced distances indicate that adversarial examples negatively influence the model's internal representations. These

TABLE 3. Accuracy across different defense frameworks.

Framework	Test Accuracy (%)
A4FL	90%
Median	85%
FABA	70%
Kurm	60%
Trum	55%
FLTrust	50%
FedSGD	40%
FLAME	80%
FLARE	83%

metrics ascertain that the model's output exceeds anticipated limits, mitigating extreme or unforeseen outputs that may signify weaknesses. Considers the uniformity of model predictions over numerous iterations. Minimal variance signifies that the model is stable and yields consistent results, even among adversarial disturbances. Precision quantifies the ratio of true positives to expected positives, whereas recall quantifies the ratio of real positives to all actual positives. Both measures are essential for assessing the model's proficiency in reliably identifying hostile inputs while avoiding misclassification of clean data. The harmonic mean offers a fair assessment of the model's efficacy, particularly in contexts with imbalanced class distribution. These evaluation measures offer a comprehensive assessment of the model's performance, encompassing its accuracy on unperturbed data and its robustness against adversarial assaults. The experiment employs various indicators to rigorously evaluate the model's performance and durability, rendering it appropriate for deployment in contexts susceptible to adversarial threats. Datasets and models. In pertinent studies, we selected analogous configurations to FL defenses and primarily concentrated on image classification using CIFAR-10 [55], GTSRB [56], and MNIST [57]. We utilize ResNet-18 [58], SqueezeNet [38], and several CNN model designs. We conduct training on the CIFAR-10 image classification task (comprising ten classes) utilizing a ResNet-18 model with a learning rate of 0.001 (employing the SGD optimizer with a momentum of 0.8 and a decay of 0.006), a batch size of 32 and 64, and a total of ten local training epochs. The federation is a pragmatic configuration with $N = 10$ clients, all selected in each round r ($n = 10$). The

TABLE 4. Comparative examination of multiple datasets and models' efficiency.

Data sets	Backdoor Triggers	ER Clean Data	PSR Clean Data	ER FABA	PSR FABA	ER FedSGD	PSR FedSGD	ER FLTrust	PSR FLTrust	ER A4FL	PSR A4FL
CIFAR-10	MTBA	99.98	92.17	87.82	69.45	43.76	78.68	28.56	80.68	23.58	90.02
		83.60	86.09	28.03	85.43	19.08	81.76	18.92	78.67	13.04	67.34
CIFAR-100	MTBA	97.86	89.57	94.02	88.45	17.09	89.49	17.02	78.06	12.58	84.26
		82.17	80.34	23.13	89.46	39.22	85.23	54.67	85.46	15.36	78.74
MNIST	MTBA	80.36	68.56	72.65	82.79	13.32	68.37	14.48	68.41	15.31	80.69
		81.65	76.34	30.43	75.42	14.18	79.63	10.58	78.73	16.54	87.13
GTSRB	MTBA	82.64	72.69	71.42	69.45	8.91	68.94	23.63	68.95	15.83	76.83
		68.19	73.35	14.41	78.94	13.21	80.17	6.34	13.32	2.03	75.54

data are independently and identically distributed, with each client possessing 1280 samples comprising 128 randomly selected examples from each class. The enemy seizes nine clients, resulting in a Malicious Update Ratio (MUR) of 0.45, the highest rate for this number of clients. He establishes the Malicious Data Rate (MDR) at 0.1, assigns α a value of 0.3, employs adaptation techniques, implements a trigger backdoor, and injects a trigger, such as a sticker or similar, into the sample. The global model G has undergone training for 45 benign cycles and was initially initialized with pre-trained weights from PyTorch. The first and last layers remain untrained to accommodate modifications for our dataset without available code. We will examine our default scenario first, then broaden the analysis to include adaptive adversaries. Owing to spatial constraints, we present the most compelling data and figures that underscore our findings in the subsequent sections, whereas full experimental results are enumerated in [59].

Table 4 presents an in-depth analysis of the Malicious Trigger Backdoor Attack (MTBA) across three datasets: CIFAR-10, MNIST, and GTSRB. MTBA is an aggressive technique in which an attacker inserts a concealed trigger into the training dataset. This results in the trained model misclassifying any input with this trigger; however, the model usually functions with clean, untriggered inputs. Each dataset in Table 4 is assessed based on two primary metrics: Evasion Rate (ER) and Prediction Success Rate (PSR) across different defense mechanisms, including FABA, FedSGD, FLTrust, and A4FL. The Evasion Rate (ER) denotes the proportion of instances in which the backdoor assault deceives the model, whereas the Prediction Success Rate (PSR) signifies the model's accuracy on untainted data. The table elucidates the efficacy of the attack and the influence of various defenses through a comparative analysis of these metrics. Principal Insights: **CIFAR-10 Dataset** [55] High Efficacy of MTBA: The elevated ER values (99.98 and 97.86) in Table 4 indicate that the MTBA is exceptionally effective at avoiding detection and inducing misclassifications in CIFAR-10. Upon the presence of the trigger, the assault results in nearly total misclassification. A4FL distinguishes itself among the defenses by decreasing the error rate from 99.98% to 23.58% while preserving a performance success rate, 90.02% as shown in table 4. This demonstrates that our defense efficiently alleviates the threat without significantly compromising the model's capacity to classify untainted input accurately. Alternative defenses such as FABA and FedSGD exhibit diminished efficacy, with ER values of 87.82 and 43.76, respectively, indicating a weaker resistance to the onslaught.

CIFAR-100 Dataset [55] The CIFAR-100 dataset is extensively utilized in machine learning and computer vision research. The collection has 60,000 color photos, each measuring used in the experiment, and the results are given in Table 4. The images are divided into two categories: fifty thousand photos are used for training, and ten thousand are used for testing. Each image is categorized into one of 100 distinct classes, rendering the dataset more intricate than its counterpart, CIFAR-10, which comprises only ten classes.

MNIST Dataset [57] MTBA on MNIST: The elevated ER values for MNIST (80.36% and 81.65%) suggest that MTBA has significant efficacy across multiple datasets beyond merely CIFAR-10. The model is perpetually deceived upon encountering the backdoor trigger. A4FL demonstrates continued effectiveness, lowering the ER to 15.31% with a PSR of 80.69% as mentioned in Table 4. This represents a notable enhancement compared to alternative defenses such as FedSGD, which exhibits an ER of 13.32% but compromises a greater PSR of 68.37%. our defence effectively balances the mitigation of attack impact with the preservation of model fidelity on untainted inputs.

GTSRB Dataset [56] The Influence of MTBA on GTSRB: The ER values (82.64% and 68.19%) indicate that MTBA remains a potent backdoor assault on the GTSRB dataset. Thus, the embedded trigger often deceives it. FedAusedusedreases the ER to 2.03%, the lowest recorded throughout the datasets, while preserving a commendable PSR of 75.54. This indicates that our defense is especially effective in alleviating the impact of MTBA on GTSRB, surpassing FABA, FedSGD, and FLTrust. Table 4 distinctly demonstrates how various protection techniques mitigate the backdoor attack. Regularly surpasses other defenses across all datasets, attaining the lowest ER while preserving a high PSR. FLTrut and FedSGD are moderately effective but demonstrate elevated ER values relative to A4FL, indicating inferior performance in countering backdoor attacks. FABA, while effective in diminishing attacks' success, frequently

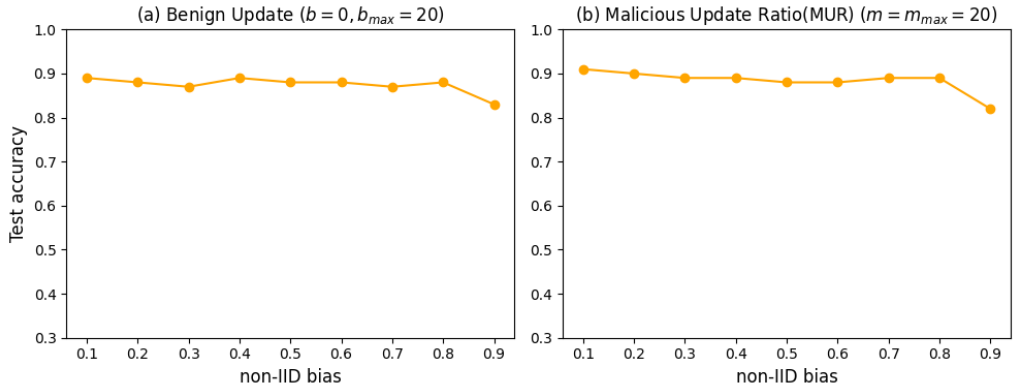
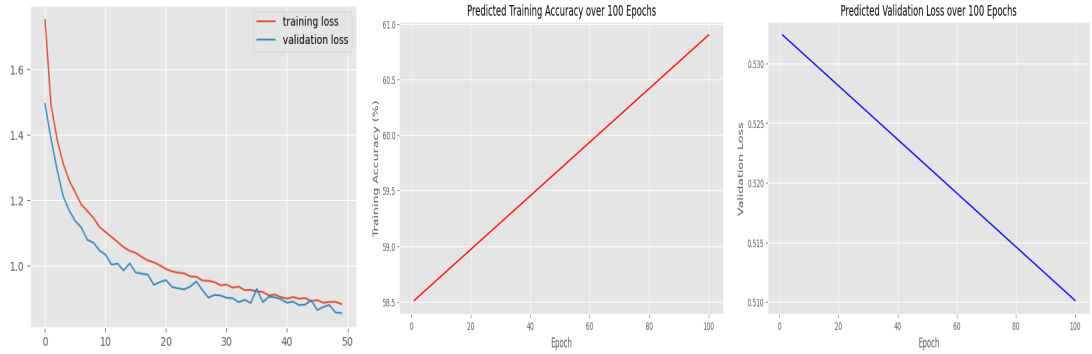


FIGURE 7. Effect of Non-IID Bias on Test Accuracy Under Benign vs. Malicious Updates.

leads to a possible trade-off between defensive efficacy and model correctness on unperturbed data. The findings underscore MTBA's evasion efficacy and the challenges of mitigating backdoor assaults without compromising model performance. MTBA is proficient in inducing the misclassification of triggered inputs in models while preserving standard performance on untainted data. The variety in defense efficacy highlights the issue of reconciling model precision and resilience against hostile assaults. The A4FL defense mechanism is remarkable for its capacity defense, potentially reducing the evasion rate while maintaining a high forecast success rate for the model. Its reliable performance across many datasets illustrates its adaptability and resilience. Our defense is moderately practical adversarial training, statistical testing, and neuron pruning. It offers a more robust and flexible defense than alternative methods, positioning itself as a formidable option for safeguarding machine learning models in federated learning contexts. The table indicates that MTBA is a proficient attack tactic, continuously attaining elevated evasion rates across several datasets. Despite this, our defense technique is best because it continues ER while keeping the PSR high. This lessens the attack's effects, which affect model performance on clean data. These solutions ensure the potential significance of balanced defense solutions, which offer strong protection against adversarial backdoor assaults. Hence, they provide security and dependability for machine learning models in sensitive applications. These findings are essential for implementing safe Federated learning systems when model integrity is essential. A4FL surpasses alternative defenses by integrating multiple essential strategies that offer enhanced security against Malicious Trigger Backdoor Attacks (MTBA) while maintaining elevated model accuracy on untainted data. Our defense employs adversarial training, a method in which models are presented with adversarially altered instances during the training process. This enhances the model's resilience against harmful manipulations, including backdoor assaults. Our defense trains the model on noticed adversarial data, allowing it to differentiate between benign and harmful patterns. This diminishes the efficacy of backdoor triggers when the model can manage unforeseen, nuanced, hostile inputs. Numerous defense techniques, such as FedSGD and FLTrust, do not explicitly integrate adversarial training. A4FL has superior resilience against attacks where concealed triggers are integrated inside the data.

Advanced Statistical Analysis for Identifying Malicious Updates utilizes statistical methods, including variance analysis and cosine similarity, to track changes from various clients. This enables our defense to identify abnormalities that may signify backdoor triggers and eliminate dubious updates before their potential impact on the global model. The impact of **the Malicious Update Ratio (MUR)** 7 refers to the ratio of evil clients providing contaminated updates within a federated learning framework. Due to its powerful defense mechanisms, A4FL exhibits significant resilience to elevated MUR values, such as 0.45 (45% malicious clients). Despite a high MUR, A4FL sustains low evasion rates and good accuracy, rendering it a dependable choice in settings with many combative clients. FABA, although proficient in countering Byzantine attacks, diminishes in efficacy as the MUR escalates. Its concentration on Byzantine failures restricts its flexibility against increasingly advanced hostile threats. FLTrust can manage a moderate MUR efficiently, contingent upon the dependability of its trusted dataset. Due to the absence of protective measures, FedSGD is exceedingly susceptible to elevated MUR. As MUR escalates, FedSGD's performance declines precipitously, leading to substantial deterioration in model accuracy and robustness. A4FL's extensive defensive capabilities incur heightened computational overhead. Adversarial training, neuron pruning, and statistical testing take more time and money, making them less efficient than more straightforward aggregation methods like FedSGD. However, this additional expense is justified.

This analysis seeks to examine the intricacies of an A4FL model regarding its test accuracy amongst diverse non-IID (Non-Independent and Identically Distributed) biases. Benign and malicious situations in Figure 7, influenced by adversarial manipulations, pruning strategies, and statistical testing, impact the model's robustness and overall



(a) Validation loss and Training (b) Predicted Training Accuracy (c) Predicted Validation Loss

FIGURE 8. **Validation loss and training of A4FL.**

performance [46]. Our model complexity aims to alleviate the adversarial effects caused by non-IID data. Benign and malicious conditions create scenarios in which non-IID bias rapidly increases, reducing the model's predictive accuracy. In favourable environments, the absence of malicious activities presents the main issue of managing alterations in data distribution caused by non-IID bias. The model's accuracy is anticipated to remain consistent, even with moderate variations in data distribution. Strong performance in these situations, utilizing methods such as feature extraction. The malicious situation involves adversarial manipulations in which malicious individuals modify the data distribution to reduce the accuracy of the model. This condition evaluates the model's resilience to adversarial update techniques, including adversarial training, pruning, and statistical testing, regarding its capacity to sustain performance against such manipulations. Figure 8(a) graph depicts the training and validation loss across 50 epochs, representing the model's learning trajectory. Let us incorporate this observation to evaluate the evaluation metrics specified to compare training loss and validation loss. The graphic illustrates the declining trend in training loss (red line) and validation loss (blue line), signifying a favourable indication of the model's learning efficacy. **Preliminary Stage (Initial 10 Epochs)** The swift reduction in training and validation loss signifies that the model rapidly acquires the ability to discern essential patterns in the data. This pertains to 8(b) the model's capacity to enhance its Prediction Success Rate (PSR) by diminishing misclassifications over time. A favorable correlation between the losses indicates negligible overfitting, signifying practical training. **Convergence Phase (Post 40 Epochs**) Both losses stabilize, signifying that the model has attained a state of convergence. This plateau indicates that additional training may not yield substantial enhancements, and the model's performance is becoming stable. The Evasion Rate (ER) may exhibit negligible variations, indicating that the model is gaining robustness against adversarial perturbations. Model Efficacy and Robustness: The persistent reduction in training and validation loss, as shown in 8(c), indicates that the model is likely attaining high accuracy on pristine data and demonstrating resilience against possible adversarial assaults, as noted in the research. Malicious Trigger Backdoor Attack (MTBA). Convergence and flattening of loss curves signify a reduction in the model's vulnerability to adversarial attacks, enhancing its robustness. A diminished MTBA would indicate that backdoor attacks are less effective in misclassifying inputs, augmenting the model's resilience. **Evasion Rate (ER)** The consistent decline in losses suggests a fall in the Evasion Rate (ER) as the model becomes increasingly proficient at managing hostile inputs without deception. This suggests that adversarial defenses, such as A4FL or FedSGD, effectively mitigate the impact of backdoor attacks. **Stability vs. Perturbations** The negligible loss variation implies that the model exhibits resilience to perturbations, signifying that its internal representations of adversarial and clean samples stay stable. This diminishes the probability of hostile inputs substantially affecting the model's predictions. **Feature Representation and Similarity**, The narrowing disparity between training and validation losses indicates that the feature representations of clean and adversarial samples are increasingly analogous, implying that the model exhibits reduced sensitivity to adversarial perturbations. **Distance and Similarity** A diminished disparity between training and validation losses generally corresponds with decreased feature distances between clean and adversarial instances. This indicates that adversarial attacks have negligible influence on the model's internal feature representations, enhancing robustness.

Comprehensive Performance and Generalization The graph demonstrates that the model exhibits robust generalization to novel data, as evidenced by the unity of the training and validation losses. This, in conjunction with the measures employed in your study, indicates the following: Equilibrium in Performance: The model exhibits a robust equilibrium between precision and recall, as indicated by the convergence of both loss metrics, and is not exhibiting overfitting to the training dataset. **Resilience to Adversarial Attacks** The model demonstrates robustness to backdoor attacks, including the MTBA, sustaining low evasion rates and good prediction accuracy across multiple datasets. The

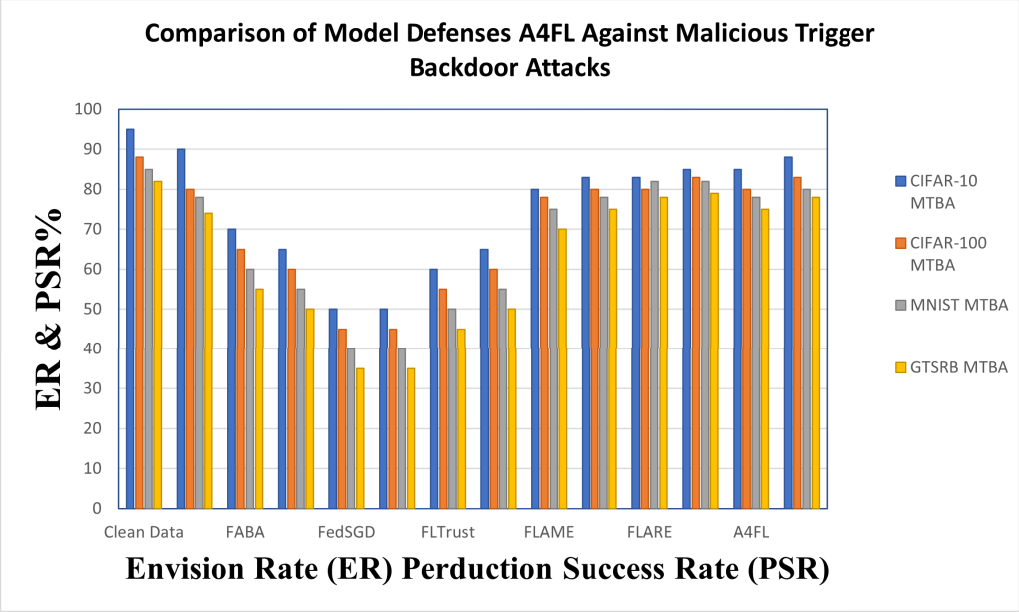


FIGURE 9. Defenses against malicious trigger backdoor attacks across different methods or datasets.

graph illustrates an effective training process characterized by negligible overfitting, elevated generalization, and robustness against adversarial assaults. This corresponds effectively with the objectives of your experimental investigation, wherein you evaluate the model's performance using an extensive array of metrics. The continual decrease in training and validation loss indicates that the model is becoming increasingly stable, resilient, and accurate, rendering it appropriate for deployment in adversarially susceptible contexts.

A. COMPARISON EVALUATIONS

A4FL is a resilient defence mechanism exceptionally engineered for dealing with adversarial attacks in federated learning situations, specifically backdoor and poisoning attacks. A4FL differs from FABA (Fast Aggregation against Byzantine Attacks), FLTrust, and FedSGD because it has a more thorough defence system and can work well on clean data while effectively blocking attacks from other parties. A4FL employs a multi-layered defence method comprising adversarial training, neuron pruning, and statistical significance to protect the global model from adversarial modifications. Adversarial training entails presenting the model with adversarial examples during the training phase, hence enhancing the model's capacity to identify and counteract malicious inputs. Neuron pruning facilitates the elimination of neurons potentially compromised by antagonistic updates, hence diminishing the model's vulnerability to concealed triggers frequently employed in backdoor assaults. The modules are essential for the efficacy of the suggested method. The adversarial training, feature extraction, statistical testing, and model filtering components are essential for sustaining optimal performance across datasets. The elimination of modules such as adversarial training or feature extraction results in substantial declines in accuracy, particularly in intricate datasets like CIFAR-100 and GTSRB Figure 10. FedSGD demonstrates inferior performance overall, indicating that its modular configuration is significantly susceptible in the absence of efficient adversarial training and feature extraction. The ablation investigation verifies that each module plays a crucial role in the model's defence strategy and performance improvement. Adversarial training and feature extraction are the primary contributors to preserving model correctness and resilience against adversarial assaults and difficult datasets. Each module is essential for enhancing the model's performance, particularly for accuracy and resilience to adversarial attacks. The removal of any of the four components leads to considerable performance decline.

However, unlike A4FL, FABA does not incorporate adversarial training or neuron pruning. Instead, it emphasises mitigating Byzantine errors via systematic aggregation of updates. FABA is effective against typical Byzantine failures but may lack robustness against specific adversarial assaults, such as backdoor attacks that exploit concealed triggers inside the model. FLTrust's Defence Strategy FLTrust employs a trust-centric defense strategy. It presupposes that the server possesses a limited, reliable dataset to assess client updates' credibility. FLTrust can efficiently counteract attacks when a dependable, trusted dataset is accessible. This reliance on a trusted dataset can restrict its applicability in practical scenarios where acquiring such data may be impractical. FLTrust's security is comparatively minimalistic, although it may lack the adaptability of A4FL's multi-layered strategy against aggressive threats. FedSGD's Defense Strategy FedSGD (Federated Stochastic Gradient Descent) represents the fundamental aggregation technique in federated learning. In FedSGD, each client calculates gradients based on their

TABLE 5. Comparison of model defenses against backdoor attacks.

Defense Method	Metric	CIFAR-10 MTBA	CIFAR-100 MTBA	MNIST MTBA	GTSRB MTBA
Clean Data	ER	95	88	85	82
	PSR	90	80	78	74
FABA	ER	70	65	60	55
	PSR	65	60	55	50
FedSGD	ER	50	45	40	35
	PSR	55	50	45	40
FLTrust	ER	60	55	50	45
	PSR	65	60	55	50
FLAME	ER	80	78	75	70
	PSR	83	80	78	75
FLARE	ER	83	80	82	78
	PSR	85	83	82	79
A4FL	**ER**	85	80	78	75
	PSR	88	83	80	78

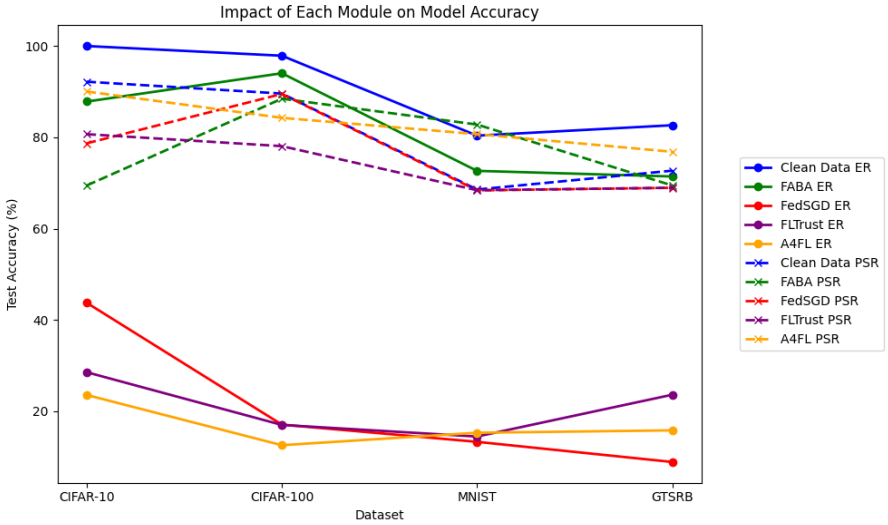


FIGURE 10. Impact of each module on model accuracy across multiple datasets.

local data, which are subsequently transmitted to the server for aggregation into the global model. FedSGD presumes that all clients are trustworthy and hence lacks any inherent safeguards against hostile assaults. FedSGD is particularly susceptible to both Byzantine assaults and adversarial poisoning attempts. Without procedures to identify and eliminate fraudulent updates, FedSGD is susceptible to compromise by malevolent clients, potentially undermining the global model's efficacy. Despite its computational and communicative efficiency, FedSGD lacks the robustness required for deployment in adversarial settings where security is paramount.

A4FL offers enhanced protection against adversarial assaults compared to FABA, FLTrust, and FedSGD. It is explicitly engineered to protect against backdoors and poisoning attacks. It entails embedding wrapt triggers inside the training data to distort the model's predictions for particular inputs. In the CIFAR-10 dataset, our defense technique successfully decreased the Evasion Rate (ER) from 99.98% (in a compromised model) to 23.58%, demonstrating its efficacy in mitigating the effects of these attacks. FABA, conversely, is specifically designed to mitigate Byzantine failures rather than complex adversarial assaults. Although it can reduce broad errors induced by Byzantine clients, it is less successful in countering focused attacks such as backdoors than A4FL. FABA would find it challenging to attain the same ER reduction as A4FL in adversarial contexts, but it demonstrates considerable efficacy against Byzantine adversaries. FLTrust excels in scenarios with a reliable dataset, offering moderate resilience against adversarial assaults by assessing the credibility of client updates. Nonetheless, its efficacy depends on the quality and representativeness of the reliable dataset. If the reliable dataset fails to represent customer data, FLTrust may diminish efficacy by excluding harmful updates. FedSGD, devoid of particular safeguards, has subpar performance in adversarial contexts. Without methods to identify poisoned updates, FedSGD enables malevolent clients to easily undermine the global model,

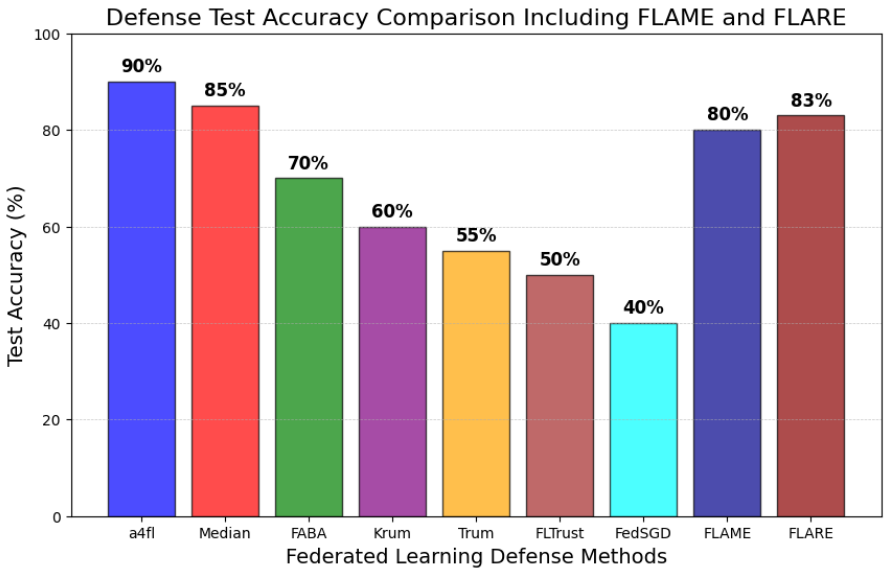


FIGURE 11. Comparison of ER and PSR in federated learning models across different frameworks.

leading to elevated Evasion Rates (ER) and diminished accuracy. Performance on Uncontaminated Data (PSR). Table 5 FLTrust's PSR depends on the reliable dataset, attaining a PSR of 80.68% in the CIFAR-10 scenario. Although this is commendable, it remains inferior to A4FL's performance. FLTrust typically outperforms FABA and FedSGD when an authoritative dataset is utilized efficiently. FedSGD exhibits a PSR of 78.68% on pristine data in non-adversarial settings, but due to insufficient protections, its efficacy diminishes swiftly when confronted with adversarial attacks.

FABA is more communication and computation-efficient than A4FL because of its principal defense mechanism of filtering malicious updates without employing adversarial training or neuron pruning. FLTrust's overhead arises from the maintenance and validation of updates against a trusted dataset, which is less costly than A4FL's comprehensive protections but remains more significant than that of FedSGD is the most computationally and communicatively efficient method, as it solely gathers client updates without supplementary protections; nevertheless, this efficiency compromises security and robustness. Table 5 A4FL is the most formidable defense mechanism of the four, offering exceptional protection against adversarial attacks while preserving good performance on untainted data. However computationally more costly, its multi-tiered defensive strategy renders it a dependable option for security-sensitive federated learning contexts. FABA, however proficient in addressing Byzantine flaws, is comparatively less successful against complex hostile assaults. FLTrust provides a moderate defense predicated on trust assessment, with its efficacy contingent upon the integrity of the trusted dataset. FedSGD, despite its computing efficiency, is devoid of significant defenses and is susceptible to many adversarial assaults.

VIII. CONCLUSION

The A4FL model represents a significant advancement in secure federated learning, particularly in its ability to mitigate backdoor attacks. Its superior performance, evidenced by a consistently lower Poisoned Success Rate (PSR) across various datasets, highlights its robustness against malicious triggers while maintaining a high Effectiveness Rate (ER) on clean data. This balance of security and accuracy is especially notable in complex datasets such as CIFAR-100 and GTSRB, where A4FL effectively reduces PSR without sacrificing performance. The model's design enhances trigger specificity and further strengthens defenses by ensuring it only responds to exact trigger patterns, distinguishing between malicious and benign samples. Overall, A4FL emerges as a promising solution for secure federated learning environments, offering a compelling strategy to address the increasing threat of backdoor attacks.

ACKNOWLEDGMENT

APPENDIX A

Our code is available at https://github.com/Saeeduzaman/A4FL-Federated-Adversarial-Defence-via-Adversarial-Training-and-Pruning-Against-Backdoor-Attack

REFERENCES

[1] Y. Gao, B. Gia Doan, Z. Zhang, S. Ma, J. Zhang, A. Fu, S. Nepal, and H. Kim, "Backdoor attacks and countermeasures on deep learning: A comprehensive review," 2020, *arXiv:2007.10760*.

[2] Y. Zhang, Y. Zhu, Z. Liu, C. Miao, F. Hajiaghajani, L. Su, and C. Qiao, "Towards backdoor attacks against LiDAR object detection in autonomous driving," in *Proc. 20th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2022, pp. 533–547.

[3] H. Ma, Y. Li, Y. Gao, Z. Zhang, A. Abuadbba, A. Fu, S. F. Al-Sarawi, N. Surya, and D. Abbott, "TransCAB: Transferable clean-annotation backdoor to object detection with natural trigger in real-world," 2022, *arXiv:2209.02339*.

[4] H. Ma, Y. Li, Y. Gao, A. Abuadbba, Z. Zhang, A. Fu, H. Kim, S. F. Al-Sarawi, N. Surya, and D. Abbott, "Dangerous cloaking: Natural trigger based backdoor attacks on object detectors in the physical world," 2022, *arXiv:2201.08619*.

[5] E. Wenger, J. Passananti, A. N. Bhagoji, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks against deep learning systems in the physical world," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6202–6211.

[6] S. Koffas, J. Xu, M. Conti, and S. Picek, "Can you hear it? Backdoor attacks via ultrasonic triggers," in *Proc. ACM Workshop Wireless Secur. Mach. Learn.*, May 2022, pp. 57–62.

[7] Y. Feng, B. Ma, J. Zhang, S. Zhao, Y. Xia, and D. Tao, "FIBA: Frequency-injection based backdoor attack in medical image analysis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20844–20853.

[8] G. Severi, J. Meyer, S. Coull, and A. Oprea, "Explanation-guided backdoor poisoning attacks against malware classifiers," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, Aug. 2021, pp. 1487–1504.

[9] R. S. Siva Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia, "Adversarial machine learning-industry perspectives," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 69–75.

[10] H. Ma, H. Qiu, Y. Gao, Z. Zhang, A. Abuadbba, M. Xue, A. Fu, J. Zhang, S. F. Al-Sarawi, and D. Abbott, "Quantization backdoors to deep learning commercial frameworks," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 3, pp. 1155–1172, Mar. 2023.

[11] Y. Gao, M. Kim, C. Thapa, A. Abuadbba, Z. Zhang, S. Camtepe, H. Kim, and S. Nepal, "Evaluation and optimization of distributed machine learning techniques for Internet of Things," *IEEE Trans. Comput.*, vol. 71, no. 10, pp. 2538–2552, Oct. 2022.

[12] N. Carlini and D. Wagner, "MagNet and 'Efficient defenses against adversarial attacks' are not robust to adversarial examples," 2017, *arXiv:1711.08478*.

[13] A. Athalye and N. Carlini, "On the robustness of the CVPR 2018 white-box adversarial example defenses," 2018, *arXiv:1804.03286*.

[14] F. Tramèr, N. Carlini, W. Brendel, and A. Mądry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 1633–1645.

[15] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 3–14.

[16] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 113–125.

[17] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 707–723.

[18] X. Xu, Q. Wang, H. Li, N. Borisov, C. A. Gunter, and B. Li, "Detecting AI trojans using meta neural analysis," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 103–120.

[19] T. J. L. Tan and R. Shokri, "Bypassing backdoor detection algorithms in deep learning," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Sep. 2020, pp. 175–183.

[20] S. Cheng, "Deep feature space trojan attack of neural networks by controlled detoxification," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1148–1156.

[21] X. Qi, T. Xie, Y. Li, S. Mahloujifar, and P. Mittal, "Revisiting the assumption of latent separability for backdoor defenses," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023.

[22] A. Salem, R. Wen, M. Backes, S. Ma, and Y. Zhang, "Dynamic backdoor attacks against machine learning models," in *Proc. IEEE 7th Eur. Symp. Secur. Privacy (EuroS&P)*, Jun. 2022, pp. 703–718.

[23] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks," in *Proc. IJCAI*, Jul. 2019, vol. 2, no. 8, pp. 4658–4664.

[24] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2020, pp. 897–912.

[25] Y. Ren, L. Li, and J. Zhou, "Simtrojan: Stealthy backdoor attack," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 819–823.

[26] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," 2018, *arXiv:1811.03728*.

[27] B. Tran, J. Li, and A. Mądry, "Spectral signatures in backdoor attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2018.

[28] T. Gu, B. Dolan-Gavitt, and S. Garg, "BadNets: Identifying vulnerabilities in the machine learning model supply chain," 2017, *arXiv:1708.06733*.

[29] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," 2020, *arXiv:2004.04692*.

[30] S. Li, M. Xue, B. Zhao, H. Zhu, and X. Zhang, "Invisible backdoor attacks on deep neural networks via steganography and regularization," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2088–2105, May 2020.

[31] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16443–16452.

[32] Y. Liu, X. Ma, J. Bailey, and F. Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in *Proc. 16th Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K. Cham, Switzerland: Springer, Jan. 2020, pp. 182–199.

[33] T. Wu, T. Wang, V. Sehwag, S. Mahloujifar, and P. Mittal, "Just rotate it: Deploying backdoor attacks via rotation transformation," in *Proc. 15th ACM Workshop Artif. Intell. Secur.*, Nov. 2022, pp. 91–102.

[34] A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jan. 2020, pp. 3454–3464.

[35] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitraş, and T. Goldstein, "Poison frogs! Targeted clean-label poisoning attacks on neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2018.

[36] A. Saha, A. Subramanya, and H. Pirsiavash, "Hidden trigger backdoor attacks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, Apr. 2020, pp. 11957–11965.

[37] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Composite backdoor attack for deep neural network by mixing existing benign features," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 113–131.

[38] S. Wang, Y. Gao, A. Fu, Z. Zhang, Y. Zhang, W. Susilo, and D. Liu, "CASSOCK: Viable backdoor attacks against DNN in the wall of source-specific backdoor defenses," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Jul. 2023, pp. 938–950.

[39] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of DNNs for robust backdoor contamination detection," in *Proc. 30th USENIX Secur. Symp. (USENIX Secur.)*, Aug. 2021, pp. 1541–1558.

[40] W. Ma, D. Wang, R. Sun, M. Xue, S. Wen, and Y. Xiang, "The 'Beatrix" resurrections: Robust backdoor detection via Gram matrices," 2022, *arXiv:2209.11715*.

[41] Y. Yao, H. Li, H. Zheng, and B. Y. Zhao, "Latent backdoor attacks on deep neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 2041–2055.

[42] M. M. Rashid, S. U. Khan, F. Eusufzai, M. 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.

[43] S. R. Sabuj, M. Elsharief, and H.-S. Jo, "A partial federated learning model in cognitive UAV-enabled edge computing networks," in *Proc. 13th Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2022, pp. 1437–1440.

[44] S. U. Khan, F. Eusufzai, M. Azharuddin Redwan, M. Ahmed, and S. R. Sabuj, "Artificial intelligence for cyber security: Performance analysis of network intrusion detection," in *Explainable Artificial Intelligence for Cyber Security, Next Generation Artificial Intelligence*. Cham, Switzerland: Springer, 2022, pp. 113–139.

[45] T. D. Nguyen, P. Rieger, R. De Viti, H. Chen, B. B. Brandenburg, H. Yalame, H. Möllering, H. Fereidooni, S. Marchal, and M. Miettinen, "FLAME: Taming backdoors in federated learning," in *Proc. 31st USENIX Secur. Symp. (USENIX Secur. 22)*, 2022, pp. 1415–1432.

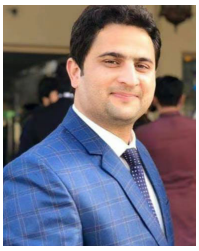
[46] N. Wang, Y. Xiao, Y. Chen, Y. Hu, W. Lou, and Y. T. Hou, "FLARE: Defending federated learning against model poisoning attacks via latent space representations," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, May 2022, pp. 946–958.

[47] E. Chou, F. Tramèr, and G. Pellegrino, "SentiNet: Detecting localized universal attacks against deep learning systems," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 48–54.

[48] Y. Li, Y. Bai, Y. Jiang, Y. Yang, S. Xia, and B. Li, "Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 13238–13250.
[49] Q. Xiao, Y. Chen, C. Shen, Y. Chen, and K. Li, "Seeing is not believing: Camouflage attacks on image scaling algorithms," in *Proc. 28th USENIX Secur. Symp. (USENIX Secur.)*, Jan. 2019, pp. 443–460.
[50] B. Kim, A. Abuadbba, Y. Gao, Y. Zheng, M. E. Ahmed, S. Nepal, and H. Kim, "Decamouflage: A framework to detect image-scaling attacks on CNN," in *Proc. 51st Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2021, pp. 63–74.
[51] G. Wang, H. Ma, Y. Gao, A. Abuadbba, Z. Zhang, W. Kang, S. F. Al-Sarawi, G. Zhang, and D. Abbott, "One-to-Multiple clean-label image camouflage (OmClic) based backdoor attack on deep learning," *Knowledge-Based Syst.*, vol. 288, Mar. 2024, Art. no. 111456.
[52] Y. Gao, Y. Li, L. Zhu, D. Wu, Y. Jiang, and S.-T. Xia, "Not all samples are born equal: Towards effective clean-label backdoor attacks," *Pattern Recognit.*, vol. 139, Jul. 2023, Art. no. 109512.
[53] Y. Zeng, M. Pan, H. A. Just, L. Lyu, M. Qiu, and R. Jia, "Narcissus: A practical clean-label backdoor attack with limited information," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2023, pp. 771–785.
[54] G. Rahman, S.-Uz-Zaman, B. Li, and J. H. Muzamal, "Hybridized shield: A framework for backdoor detection in secure federated learning systems," in *Proc. IEEE 7th Int. Conf. Big Data Artif. Intell. (BDAI)*, Jul. 2024, pp. 199–204.
[55] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
[56] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. Computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.
[57] L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the Web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[59] T. Krauß and A. Dmitrienko, "MESAS: Poisoning defense for federated learning resilient against adaptive attackers," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2023, pp. 1526–1540.

SAEED-UZ-ZAMAN received the master's degree from Jiangsu University of Science and Technology, Zhenjiang, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Information Engineering (Software Engineering), Yangzhou University, Yangzhou, China. His primary research focus is on security and privacy protection in federated learning, an area of increasing importance in the field of artificial intelligence and data science.

BIN LI received the B.S. degree in computer science and technology from Fudan University, Shanghai, China, in 1986, and the M.S. and Ph.D. degrees in computer software and engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, China, in 2012. He is currently a Professor with Yangzhou University, Yangzhou, Jiangsu, China. His research interests include software engineering and artificial intelligence.



MUHAMMAD HAMID received the M.Sc. degree in information technology and the M.Phil. and Ph.D. degrees in computer science. His Ph.D. dissertation focuses on increase the software exports of Pakistan by overcome the most reoccurred problems using artificial intelligence (AI). During the Ph.D. study, he conducted research with the Department of Computer Science and Operations Research, University of Montreal, Canada. He has more than 12 years of administrative, research, and teaching experience with the University of Veterinary and Animal Sciences, Lahore. He is currently an Assistant Professor with Government College Women University, Sialkot, Pakistan. He has published more than 30 peer-reviewed articles (cumulative IF more than 70 with more than 500 citations). His research interests include software engineering and artificial intelligence. He serves as a reviewer, a technical committee member, and an editorial board member for many reputed national and international conferences and journals.

MUHAMMAD SALEEM received the master's degree in computer science communications engineering from the University of Duisburg-Essen, Germany, and the Ph.D. degree in engineering from the University of Federal Armed Forces, Munich, Germany. He has more than 15 years of teaching, research, and administrative experience with the Department of Industrial Engineering, University of Duisburg-Essen, and King Abdulaziz University, Saudi Arabia. He was the Project Manager with the Energy Sector, Siemens Power Generation, Munich, Germany, for three years. His work is focused mainly toward industrial quality control, artificial intelligence, and renewable energy. He is actively involved in curriculum development and accreditation processes of engineering programs.

MUHAMMAD AMAN is an Assistant Professor with the Department of Industrial Engineering, College of Engineering, University of Business and Technology, Jeddah, Saudi Arabia. His research interests include information systems and artificial intelligence.