IEEE Open Journal of
Intelligent Transportation
Systems

Received 2 March 2024; revised 30 April 2024 and 20 June 2024; accepted 26 July 2024. Date of publication 8 August 2024;
date of current version 26 August 2024.

Digital Object Identifier 10.1109/OJITS.2024.3440876

A Secure Object Detection Technique for Intelligent Transportation Systems

MD. JUEAL MIA [1] AND M. HADI AMINI [2] (Senior Member, IEEE)

[1] Knight Foundation School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA

[2] Sustainability, Optimization, and Learning for InterDependent Networks Laboratory (Solid Lab), Florida International University, Miami, FL 33199, USA

CORRESPONDING AUTHOR: MD. J. MIA (e-mail: mmia001@fiu.edu)

This work was supported by the National Center for Transportation Cybersecurity and Resiliency (TraCR, a U.S. Department of Transportation National University Transportation Center) headquartered at Clemson University, Clemson, SC, USA.

ABSTRACT Federated Learning is a decentralized machine learning technique that creates a global model by aggregating local models from multiple edge devices without a need to access the local data. However, due to the distributed nature of federated learning, there is a larger attack surface, making cyber-attack detection and defense challenging. Although prior works developed various defense strategies to address security issues in federated learning settings, most approaches fail to mitigate cyber-attacks due to the diverse characteristics of the attack, edge devices, and data distribution. To address this issue, this paper develops a hybrid privacy-preserving algorithm to safeguard federated learning methods against malicious attacks in Intelligent Transportation Systems, considering object detection as a downstream machine learning task. This algorithm involves the edge devices (e.g., autonomous vehicles) and road side units to collaboratively train their model while maintaining the privacy of their respective data. Furthermore, this hybrid algorithm provides robust security against data poisoning-based model replacement and inference attacks throughout the training phase. We evaluated our model using the CIFAR10 and LISA traffic light dataset, demonstrating its ability to mitigate malicious attacks with minimal impact on the performance of main tasks.

INDEX TERMS Object detection, cyber-attacks, privacy, intelligent transportation systems, data poisoning-based model replacement attack, inference attack.

I. INTRODUCTION

INTELLIGENT transportation systems (ITS) rely on emerging artificial intelligence (AI) and machine learning (ML) methods. Further, car manufacturers and technology companies have developed commercial applications for ITS [1]. Object detection is a major task for interpreting visual capabilities of ITS to reduce manual operation and communication overhead. Due to technological advancements, different types of adversarial attacks can occur during the training or inference phases in ITS application development. Hence, security and privacy considerations are essential. Privacy breaches could lead to severe consequences, such as public safety risks and accidents, authentication concerns, service disruptions, and vulnerabilities in communication [2]. Security threats in autonomous vehicles (AVs) reveal vulnerabilities with potential safety issues [3]. Such incidents highlight the risk of remote attacks and malware in automotive systems [4]. Key considerations include distributed AVs security, compliance with data protection regulations, and standardized communication protocols to prevent vulnerabilities.

In the era of artificial intelligence, numerous ITS applications are developed using ML techniques. Therefore, deploying real-time applications necessitates privacy-preserving ML or deep learning (DL) techniques to avoid vulnerability issues. Numerous research endeavors have focused on safeguarding privacy and security of ITS through ML and DL methods. These include convolutional neural network (CNN) for encrypted image classification [5], deep

The review of this article was arranged by Associate Editor Jianwu Fang.

neural networks (DNNs) for error detection in traffic sign recognition [6], and defense mechanisms for deep reinforcement learning-based traffic signal controllers [7]. The DL model requires large-scale datasets that are collected from multiple edge client devices and decentralized in a data center. However, this approach is vulnerable to malicious clients because traditional data collection procedures may trigger different security concerns for the users. With technological advancements, privacy and security have become significant considerations in real-world applications. To solve these privacy issues, the researcher developed a paradigm referred to as federated learning (FL) that allows several clients to collaborate in the training process without sharing their local data [8], [9]. The application of this paradigm is increasing to a greater extent such as virtual keyboard prediction [10], health care data analysis [11], traffic image processing [12], autonomous driving [13], industrial IoT [14].

In FL settings, The central server does not have access to the local data residing on the edge client's side. However, the FL systems still might be susceptible to cyber-attacks in the presence of adversaries or malicious clients. Cyber-attacks can be classified into two main types depending on the objective of the attacker: targeted and untargeted attacks [15]. In a targeted attack, the goal of the attacker is to compromise or manipulate the model, enabling it to behave differently for specific sub-tasks while maintaining the overall performance of the primary task. On the contrary, malicious clients degrade the overall model performance of the primary tasks in untargeted attacks. Both attacks can happen in the form of data poisoning [16], [17], [18] or model poisoning [19], [20] attack. Our focus is on backdoor or targeted attacks (data poisoning-based model replacement attacks) and inference attacks at the same time during the training phase. A backdoor attack, as classified in targeted attacks [21], [22], becomes susceptible to FL when a malicious client introduces a backdoor trigger into the training dataset [23]. Besides, an attacker can also easily alter the global model with the attacker's local malicious model during the training phase [9]. In inference attacks, an attacker can deduce confidential information by analyzing the global model or from the communication channel [24], [25]. As potential attacks on the FL systems can have detrimental impacts on the model's performance, secure FL techniques are required in ITS applications.

To overcome these challenges, a hybrid privacy-preserving FL algorithm has been developed in this study to defend against cyber-attacks in object detection applications for ITS. This approach builds on Pre-aggregation Similarity Measurement (PA-SM) [26] and differential privacy (DP) [27] to address security and privacy concerns simultaneously. The PA-SM adaptation involves thorough analysis to ensure its applicability and effectiveness for ITS. While existing studies help identify backdoor updates in FL settings, we extend our experimentation to include the integration of DP. This addition enhances resilience against inference attacks. Moreover, existing studies used Local Differential Privacy (LDP) in FL, which may have a slightly higher negative impact on performance [28]. In response, we propose the use of Central Differential Privacy (CDP), where the aggregator or server adds noise to the global model. The proposed technique provides a privacy-preserving empirical solution and enhances precision in detecting malicious attacks. Our contributions include offering a comprehensive defense against data poisoning based model replacement and inference attacks and introducing a hybrid algorithm to enhance security in ITS applications. The primary contributions of this research are outlined as follows.

1) We have developed and implemented a hybrid privacy-preserving algorithm that includes an integrated anomaly detection technique, PA-SM [26], and DP [27] to defend against data poisoning-based model replacement attacks and inference attacks without significant impact on the performance during training phase.
2) The proposed algorithm will provide a security against data leakage.
3) Furthermore, we have analyzed the influential factors and characteristics of backdoor attacks at several rounds of communication in FL settings, which will create a strong foundation for future research in ITS.

The rest of this paper is organized as follows. Section II discusses the related works. Section III presents the preliminaries of FL and the attack model. Section IV describes the proposed hybrid approach. Section V provides experiments and result analysis, followed by Section VI which presents the discussion with an in-depth assessment. Section VII concludes the paper.

II. LITERATURE REVIEW

FL receives more research attention in ITS due to its distributed nature and data availability. However, the exponential growth of data and heterogeneous devices, along with various categories of cyber attacks poses new challenges. Numerous algorithms have been proposed for formulating adversarial attacks in AVs or transportation systems, including backdoor attacks in lane detection [29], manipulation of benign models in traffic control systems [30], and Targeted Universal Adversarial Perturbation (TUAP) [31]. In the literature, limited studies specifically highlight privacy and security against cyber attacks in ITS.

The most frequently initiated backdoor attack in FL is the model replacement attack, where the attacker needs control over a benign client device [9]. An attacker can easily design a successful attack by inferring information due to the insecure aggregation of local models. The well-known FedAvg aggregation algorithm proposed by McMahan et al. [8] is used to mitigate adversarial attacks. Additional aggregation techniques, including FLTrust [32], Krum [33], Trim mean [34], Trim median [34], FLAME [35], ShieldFL [36], FLOD [37], DnC [38], FoolsGold [39], CONTRA [40], SignGuard [41], FLARE [42], and Romoa [43], have been introduced to address security issues. Outlier or anomaly

TABLE 1. Comparative analysis with existing study in ITS.

Reference	Technique	Data Privacy	Model Privacy	Poisoning Resiliency	Inference Resiliency
OQFL [1]	HyperParameter optimization	✓	✓	×	×
SemBroc [44]	HE, SMC	✓	×	×	✓
GeFL [45]	Gradient encryption	✓	×	×	✓
BDFL [46]	PVSS Scheme	✓	×	×	✓
Olowononi *et al.* [47]	DP	✓	×	×	✓
Batool *et al.* [48]	Local DP	✓	×	×	✓
This paper	Hybrid (PA-SM [26] and DP [27])	✓	✓	✓	✓

detection can be a strategy to differentiate malicious updates from benign updates using similarity measurement metrics. Lu et al. [26] present a prominent framework to defend against model replacement and adaptive backdoor attacks. They conduct pre-aggregation of models with similarity assessment to eliminate backdoor updates during convergence time, and employ backdoor neuron activation to decrease backdoor updates in the initial rounds. One notable research study conducted by Yamany et al. [1] focuses on defending against malicious attacks, considering object detection as a main task in AVs. The proposed defense mechanism adjusts hyperparameters (e.g., learning rate) in the FL environment to provide privacy.

Multi-Party computation (MPC) [49] is a concept of cryptography that can be used in FL to make client data indistinguishable. SAFEFL [50] is an MPC-based approach used to address poisoning and inference attacks simultaneously in general. It includes secure FL aggregation techniques to address poisoning attacks and MPC to tackle inference attacks. However, this approach incurs a higher computational cost, which is its main drawback. SemBroc-RF is an enhanced FL model proposed by Zhu et al. [44] to improve data security in ITS, developed based on end-to-end homomorphic encryption (HE) and secure MPC. To enhance gradient security and model performance, they propose another approach called GreFLa with reinforcement learning. The proposed model is evaluated using MNIST [51] and CIFAR10 [52] datasets, demonstrating efficient defense against gradient leakage. GeFL [45] is a novel approach integrating gradient encryption into FL for AVs. It leverages the edge device's computational power to fine-tune local models and encrypt input data, ensuring user privacy without added computational overhead. Further, BDFL [46] is a Peer-to-Peer FL technique that utilizes a Publicly Verifiable Secret Sharing (PVSS) scheme to defend against privacy leakage. As there is very few privacy-preserving FL techniques implemented in ITS, we have investigated all the relevant works in general.

Furthermore, DP is another procedure to make adversarial attacks impracticable, providing additional heterogeneity among all edge client devices by adding random noise. DP, proposed by Abadi et al. [27], comes in two categories: LDP and CDP [53]. Geyer et al. [54] introduce client-sided DP-preserving FL techniques to secure client information during the training process while maintaining model accuracy concerning privacy loss. Sun et al. [55] design a procedure involving norm clipping and edge-client level DP to mitigate backdoor attacks, claiming that attack performance relies on the fraction of malicious clients. Olowononi et al. [47] leverage FL with DP to strengthen the robustness of Vehicular Cyber-Physical Systems (VCPS) against inference attacks. Batool et al. [48] introduce a Vehicular Ad-hoc network (VANETs) data-sharing framework employing FL with LDP. Vehicles apply local privacy techniques before sharing data with Road Side Units(RSUs), which then train models and update weights with a central server. Evaluation reveals superior performance over traditional FL methods, with enhanced security against inference and gradient leakage attacks.

We have conducted a comprehensive review of the existing FL methods in the literature to analyze the research gaps considering the ITS applications. Table 1 presents a comparative analysis of methods in ITS, where existing systems often rely on cryptography and blockchain concepts. The BDFL system employs the PVSS Scheme, SemBroc utilizes homomorphic encryption and secure multiparty computation, GeFL uses gradient encryption techniques for securing FL environment. Olowononi et al. [47] and Batool et al. [48] utilize DP for security in FL. While these methods focus on defending against inference attacks, none address poisoning attacks. However, OQFL uses hyperparameter optimization to safeguard against model poisoning attacks; it remains susceptible to data poisoning attacks and inference attacks. In contrast, our proposed solution aims to simultaneously secure against data poisoning-based model replacement attacks and inference attacks, ensuring resilience for both internal and external clients in FL. Moreover, in this study, we aim to enhance traditional ML and DL approaches, addressing data privacy and security concerns in FL settings. We introduce a novel hybrid algorithm that includes PA-SM [26] and DP [27] to defend against data poisoning based model replacement and inference attack at the same time during training phase.

III. PRELIMINARIES

In this section, we will explore the general concepts of FL and the associated threat model. As there is no specific implemented algorithm in ITS, we focus on object detection as a ML task in general.

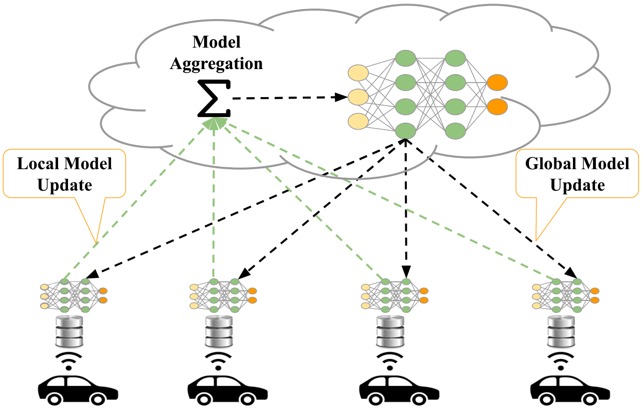


FIGURE 1. General FL framework with learning process.

A. FEDERATED LEARNING

The FL system comprises two principal components: the central server and edge clients. Suppose, we have the m number of clients who are communicating in an FL setting with $(D_{\text{train}})_{k=1}^{m}$ datasets. Edge client devices conduct local training on their datasets without sharing their data with central servers or other clients. Local clients communicate with the central server via sharing model weights $\triangle w_k^p$ only. In every round of communications, the central server selects a m client from C number of clients and shares global model $\mathbf{w}_{\text{t}}$. The local clients train their model using their private data $(D_{\text{train}})_{k=1}^{m}$. Furthermore, The central server collects all the local model updates $\triangle w_k^p$ and aggregates them to build an updated global model $\mathbf{w}_{\text{t+1}}$. The framework of general FL is presented in Figure 1.

The aggregation method is an important part of FL to aggregate local client's updates. FedAvg is one of the most commonly-used algorithm proposed by McMahan et al. [8]. The aggregation rule can be represented in Equation (1).

$$\mathbf{w}_{\text{t+1}} = \mathbf{w}_{\text{t}} + \eta \frac{\sum_{k \in \{C_k\}_{k=1}^{m}} n_k \triangle w_k^p}{\sum_{k \in \{C_k\}_{k=1}^{m}} n_k}, \tag{1}$$

where η is a learning rate, p is the total number of participants in FL. C_k is selected clients when $k = 1 \cdots m$, n_k represents the number of samples, and $\triangle w_k^p$ is a local update. FedAvg algorithm is presented in Algorithm 1.

Algorithm 1: FedAvg [8] Algorithm for Aggregation of Local Model Updates

```
Server executes
initialize $w_0$
for each round $t = 1, 2, \ldots$ do
    $m \leftarrow \max(C \cdot f, 1)$ // fraction of selected users
    $S_t =$ (random set of $m$ clients)
    for each client $k \in S_t$ in parallel do
        $\Delta w_k^p = Update\_Client(e, B, N, L, O, \mathbf{w}_t)$
        $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \frac{\sum_{k \in S_t} n_k \Delta w_k^p}{\sum_{k \in S_t} n_k}$
Function Update_Client($e, B, N, L, O, \mathbf{w}_t$):
    // Where $e$ is the number of local epochs, $B$ is the batch size, $N$ is the neural network model, $L$ is the training loss, and $O$ is the optimizer
    for each local epoch i from 1 to e do
        for each batch $b \in B$ do
            $\Delta w_k^p = \mathbf{w}_k^p - \mathbf{w}_t$
    return $\Delta w_k^p$ to server
```

B. THREAT MODEL

Our threat model is based on the study conducted by Lu et al. [26], incorporating data poisoning-based model replacement attacks during convergence time. We are going to evaluate it in the context of ITS. Additionally, we consider the threat model for inference attacks in the same context.

In data poisoning based model replacement attacks, backdoor attackers create backdoor sample data by applying the backdoor pattern to the training dataset. The backdoor trigger inside the right bottom corner of the image can influence the model parameter to mislead the model during the training process [56], [57]. Several research studies have introduced backdoor attacks through the generation of various types of backdoor triggers [21], [23], [58], [59]. Visualization of a backdoor attack is shown in Figure 2.

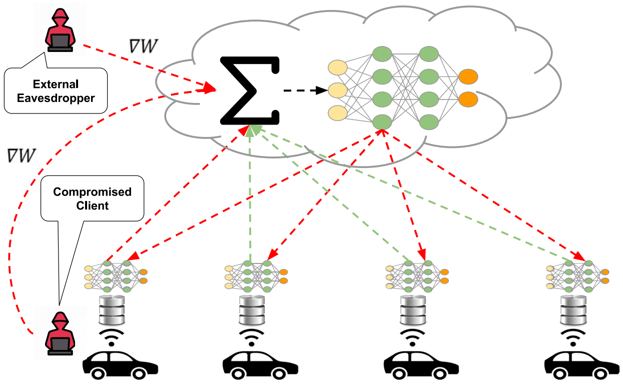


FIGURE 2. Cyber attack in FL settings during training phase.

An attacker changes some pixels of the training sample with a specific label and then embeds them with the clean sample to train the model [60]. Suppose x_{poi} is a poisoned sample with the label y_{target} where x is the original clean sample. $\otimes$ indicates element-wise products, and α represents the trade-off hyper-parameter. Poison data x_{poi} is created using Equation (2).

$$x_{\text{poi}}\left(x; x_{\text{trigger}}\right) = (1 - \alpha) \otimes x + \alpha \otimes x_{\text{trigger}} \tag{2}$$

This equation altered the original data sample x with the trigger data sample x_{trigger} using the hyper-parameter α. The element-wise product operation ensures that the data

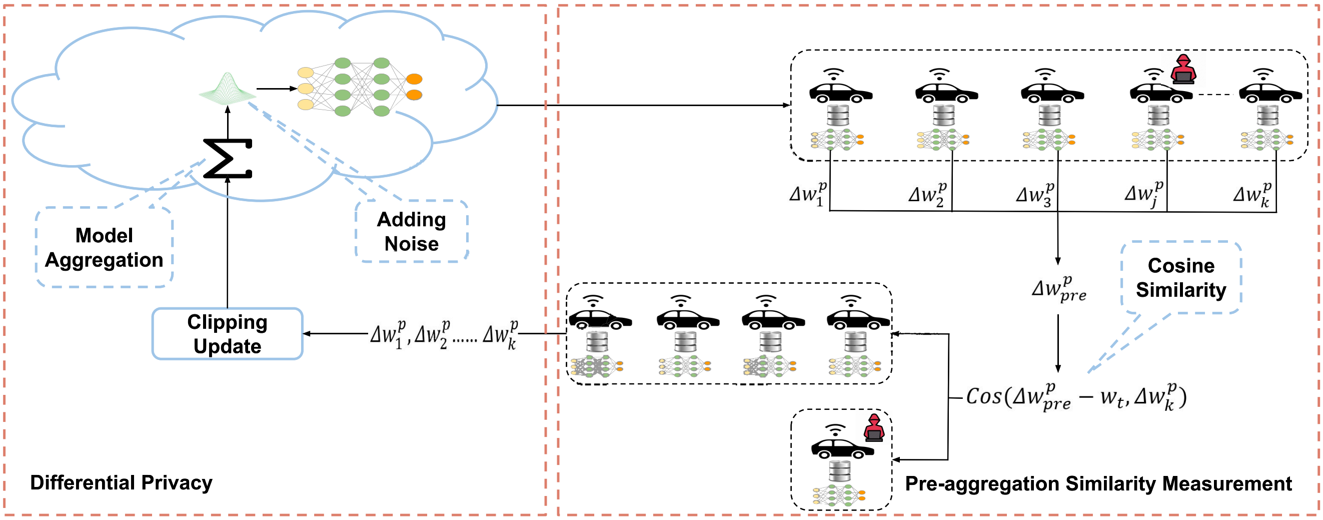


FIGURE 3. Overview of the proposed hybrid privacy-preserving algorithm to protect FL against adversarial attack in ITS.

poisoning is performed separately for each element in the data samples. The hyper-parameter α determines the balance between preserving the original data and incorporating the trigger data.

Let j^{th} be the index of a malicious client who is controlling a device. In the meantime, the malicious client is combining the benign train data D_{benign} and poisoned train data D_{poisoned} to train their model that will be considered as a backdoor model. The backdoor model can be presented as shown in Equation (3).

$$w^p_{\text{backdoor}} = \min_w \mathbb{E}_{(x,y)\ \in(D_{\text{poisoned}} \cup D_{\text{benign}})} L_f((x,y); w) \quad (3)$$

where $\mathbb{E}$ represents the goal of finding the set of weights w that minimizes the expected loss over both poisoned and benign datasets. L_f indicates the loss function to calculate the training loss.

In a model replacement attack, the attacker wants to replace the updated global model $\mathbf{w}_{t+1}$ with the backdoor model w^p_{backdoor}. The replacement can happen using the following Equation (4).

$$w^p_{\text{backdoor}} = \mathbf{w}_{\text{t}} + \eta \frac{\sum_{k\in\{C_k\}_{k=1}^m} n_k \triangle w_k^p}{\sum_{k\in\{C_k\}_{k=1}^m} n_k}. \quad (4)$$

The attacker creates a backdoor update using Equation (4) and represented as Equation (5).

$$\triangle w_j^{\text{p}} = \frac{\sum_{k\in\{C_k\}_{k=1}^m} n_k}{\eta n_j}\left(w^p_{\text{backdoor}} - \mathbf{w}_{\text{t}}\right) - \frac{\sum_{k\in\{C_k\}_{k=1}^m k\neq j} n_k \triangle w_k^p}{n_j} \quad (5)$$

$\sum_{k\in\{C_k\}_{k=1}^m k\neq j} n_k \triangle w_k^p$ is negligible due to convergence of global model. Thus, the backdoor update can be represented using the following Equation (6).

$$\triangle w_j^{\text{p}} \approx \frac{\sum_{k\in\{C_k\}_{k=1}^m} n_k}{\eta n_j}\left(w^p_{\text{backdoor}} - \mathbf{w}_{\text{t}}\right) \quad (6)$$

$\sum_{k\in\{C_k\}_{k=1}^m} n_k/\eta n_j$ is considered as boosting factor for the backdoor model [26]. To make a successful attack, the attacker needs to assign this value.

In an inference attack, an honest-but-curious or a dishonest client can infer sensitive information by analyzing the aggregated model or local model from the communication channel [24], [25]. In these situations, these participants might be able to reconstruct sensitive information of local clients by looking at the shared global models. This can lead to security concerns and is considered as a crucial vulnerability for the FL system.

IV. METHODOLOGY

Our proposed hybrid privacy-preserving algorithm simultaneously integrates two modules namely the PA-SM [26] and DP [27]. An overview of this algorithm is represented in Figure 3. Our algorithm uses FedAvg [8] to aggregate local model weights. A description of the deployed algorithms, i.e., PA-SM [26] and DP [27], is given below.

A. PRE-AGGREGATION SIMILARITY MEASUREMENT

Algorithm 2 is mainly deployed based on the idea and PA-SM method proposed by Lu et al. [26]. The PA-SM is presented in algorithmic format. This algorithm helps to identify backdoor updates in FL settings.

In line 1, it receives the local model parameter as $\mathbf{w}_k^p$, the poisoned model index as p_{id}, and the boosting factor as γ. The algorithm iterates a `FOR` loop to get the local model updates (lines 5-13). In line 6, an `IF` statement compares the local client index i with the poisoned client index p_{id}. If the condition is satisfied, it returns the local updated model parameters $(\mathbf{w}_k^p - \mathbf{w}_{\text{pre}}^p)$ multiplied by γ; otherwise, it only returns $(\mathbf{w}_k^p - \mathbf{w}_{\text{pre}}^p)$.

In line 14, the function named `PASM()` is called, which receives three parameters: the current global model as $\mathbf{w}_t$, the pre-aggregated model as $\mathbf{w}_{\text{pre}}^p$, and the local model

Algorithm 2: Module 1 (Similarity Measurement-Based Backdoor Update Detection in FL [26]) for a Total of P_t Training Rounds, Choose a Set of m Clients Denoted as S_t. The Local Dataset is Represented as $(D_{\text{train}})_{k=1}^{m}$: $\{(x, y)\}$, D_{poisoned} is a Poisoned Dataset, $D_{\text{benign}} \subseteq (D_{\text{train}})_{k=1}^{m}$ is a Benign Dataset, η is the Learning Rate, the Current Global Model is Denoted by $\mathbf{w}_t$, and $\mathbf{w}_{pre}^{p}$ are Pre-Aggregated Model Updates

```
1  Function Attack-Detection($\mathbf{w}_k^p$, $p_{id}$, $\gamma$):
2      $\Delta\mathbf{w}_k^p \leftarrow []$;
3      $\mathbf{w}_t \leftarrow \mathbf{w}_k^p[-1]$ // Last element of the
           parameter list
4      $\mathbf{w}_{\text{pre}}^p \leftarrow \mathbf{w}_k^p[0]$ // First element of the
           parameter list
5      for $i$ in range(len($\mathbf{w}_k^p$)) do
6          if $i == p_{id}$ then
7              for key, var in $\mathbf{w}_k^p[i+1]$ do
8                  $update \leftarrow \gamma \cdot (\mathbf{w}_k^p - \mathbf{w}_{\text{pre}}^p)$;
9                  $\Delta\mathbf{w}_k^p \leftarrow update$;
10         else
11             for key, var in $\mathbf{w}_k^p[i+1]$ do
12                 $update \leftarrow (\mathbf{w}_k^p - \mathbf{w}_{\text{pre}}^p)$;
13                 $\Delta\mathbf{w}_k^p \leftarrow update$;
14     $b_{\text{id}} \leftarrow PASM(\mathbf{w}_t, \mathbf{w}_{\text{pre}}^p, \Delta\mathbf{w}_k^p)$;
15 Function PASM($\mathbf{w}_t$, $\mathbf{w}_{pre}^p$, $\Delta\mathbf{w}_k^p$):
16     $G_{\text{update}} \leftarrow \{\}$ // Initial global weight
17     Run Algorithm 1 // Using
           ($D_{\text{poisoned}} \cup D_{\text{benign}}$) dataset to update
           the client model
18     $\mathbf{w}_{\text{pre}}^p \leftarrow$ Store pre-aggregated model weights;
19     $\mathbf{w}_t \leftarrow$ model // Current global model
           weights $\mathbf{w}_t$
20     $G_{\text{update}} \leftarrow \mathbf{w}_t - \mathbf{w}_{\text{pre}}^p$;
21     $\Delta\mathbf{w}_{\text{new}}^p \leftarrow []$;
22     for $i$ in $\Delta\mathbf{w}_k^p$ do
23         $\Delta\mathbf{w}_{\text{new}}^p \leftarrow i$;
24     for $i$ in range(len($\Delta\mathbf{w}_{new}^p$)) do
25         $vecA \leftarrow G_{\text{update}}$;
26         $vecB \leftarrow \Delta\mathbf{w}_{\text{new}}^p[i]$;
27         $scores \leftarrow \text{Cos}(vecA, vecB)$ // Cosine is a
               function for similarity
               measurement
28     $b_{\text{id}} \leftarrow []$ // Benign index
29     $T \leftarrow 0.95$ // Threshold
30     for $i$ in range(len(scores)) do
31         if $scores < T$ then
32             $b_{\text{id}} \leftarrow i$;
33     return $b_{\text{id}}$;
```

updates as $\Delta\mathbf{w}_k^p$. Then, values are initialized for the listed parameters (lines 16-20). Firstly, the pre-aggregated model weights are stored in $\mathbf{w}_{\text{pre}}^p$, and the initial global model weights are assigned to $\mathbf{w}_t$. Furthermore, G_{update} is calculated by subtracting $\mathbf{w}_{\text{pre}}^p$ from $\mathbf{w}_t$.

In line 21, an empty list is created to store the processed local model updates. A FOR loop iterates to store the updated local model weights in $\Delta\mathbf{w}_{\text{new}}^p$ (lines 22-23). In lines 24-27, another FOR loop iterates over the length of local model updates $\Delta\mathbf{w}_k^p$ to calculate the similarity score between G_{update} and $\Delta\mathbf{w}_{\text{new}}^p$.

Finally, a FOR loop checks the backdoor model updates by comparing them with a threshold T (lines 30-33). If the client model gradient score is greater than T, it is considered a backdoor update; otherwise, the index is stored in a benign index list b_{id}.

B. DIFFERENTIAL PRIVACY

The detailed description of DP [27] is presented in Algorithm 3. This technique ensures that aggregation results cannot be distinguished. However, local clients are required to trust the server by sharing their model updates. For each round of communication, the function DP($\Delta\mathbf{w}_k^p, \epsilon, \delta$) will receive the model updates $\Delta\mathbf{w}_k^p$, the privacy budget ϵ, and the privacy parameter δ as input.

The function Purturb_Weights($tensor, \epsilon, \delta$) is defined from lines 1 to 7. This function takes the following parameters as input: a tensor to be perturbed ($tensor$), the privacy budget (ϵ), and the privacy parameter (δ). In line 2, sensitivity S is calculated as $\frac{2.0}{\text{number of elements in the tensor}}$. The noise scale ns is computed in line 3 as $\frac{S}{\epsilon}$. In line 4, N is determined as $S \times \log(\frac{1}{\delta})$. Laplace noise l_n is generated and clipped to the range $[-N, N]$. The noised tensor is obtained by adding the Laplace noise to the original tensor. Finally, the noised tensor is returned.

The function Update_Client($e, B, N, L, O, \mathbf{w}_t$) is defined from lines 8 to 12. This function takes the following parameters as input: the number of local epochs (e), the batch size (B), the neural network model (N), the training loss (L), the optimizer (O), and the global model parameters ($\mathbf{w}_t$).

Specifically, in line 9, a FOR loop iterates over the local epochs e, and within this, another FOR loop iterates over the batch size $b \in B$ to train the local model using the assigned dataset. The function returns the updated model weight $\Delta\mathbf{w}_k^p$ to the server in line 12.

The function DP($\Delta\mathbf{w}_k^p, \epsilon, \delta$) is defined from lines 13 to 18. This function takes the following parameters as input: the model updates ($\Delta\mathbf{w}_k^p$), the privacy budget (ϵ), and the privacy parameter (δ). In line 14, a FOR loop iterates over each client k in $\Delta\mathbf{w}_k^p$. In line 15, the server aggregates all the benign model outputs to update $\mathbf{w}_{t+1}$. In line 16, a FOR loop iterates over each layer in $\mathbf{w}_{t+1}$. In line 17, each layer is perturbed using the Purturb_Weights() function to ensure layer-wise perturbation. Finally, line 18 returns the updated and perturbed global model $\mathbf{w}_{t+1}$.

This approach ensures privacy by perturbing the aggregated model updates, providing an additional layer of security without significantly impacting the performance of FL.

Algorithm 3: Module 2 (DP [27] in FL) a Total of P_t Training Rounds, Choose a Set of m Clients Denoted as S_t. The Local Dataset is Represented as $(D_{\text{train}})_{k=1}^{m}$: $\{(x, y)\}$, D_{poisoned} is a Poisoned Dataset, $D_{\text{benign}} \subseteq (D_{\text{train}})_{k=1}^{m}$ is a Benign Dataset, η is the Learning Rate, $\mathbf{w}_t$ is the Current Global Model, and $\Delta\mathbf{w}_k^p$ are the Model Updates. ϵ is the Privacy Budget, δ is the Privacy Parameter, S is the Sensitivity, and ns is the Noise Scale

```
Function Purturb_Weights(tensor, ϵ, δ):
    $S \leftarrow \frac{2.0}{\text{number of elements in the tensor}}$ // Sensitivity calculation
    $ns \leftarrow \frac{S}{\epsilon}$;
    $N \leftarrow S \times \log\left(\frac{1}{\delta}\right)$;
    $l_n \leftarrow$ Laplace noise clipped to the range $[-N, N]$;
    $noised\_tensor \leftarrow tensor + l_n$;
    return $noised\_tensor$;
Function Update_Client($e, B, N, L, O, \mathbf{w}_t$):
    // Where $e$ is the number of local epochs, $B$ is the batch size, $N$ is the neural network model, $L$ is the training loss, and $O$ is the optimizer
    for each local epoch $i$ from 1 to $e$ do
        for each batch $b \in B$ do
            $\Delta\mathbf{w}_k^p = \mathbf{w}_k^p - \mathbf{w}_t$;
    return $\Delta\mathbf{w}_k^p$ to the server;
Function $DP(\Delta\mathbf{w}_k^p, \epsilon, \delta)$:
    for each client $k$ in $\Delta\mathbf{w}_k^p$ do
        $\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \frac{\sum_{k \in S_t} n_k \Delta\mathbf{w}_k^p}{\sum_{k \in S_t} n_k}$;
    for each layer in $\mathbf{w}_{t+1}$ do
        $\mathbf{w}_{t+1}[layer] =$ Purturb_Weights($\mathbf{w}_{t+1}[layer], \epsilon, \delta$) // Layer-wise weight perturbation
    return $\mathbf{w}_{t+1}$;
```

C. HYBRID PRIVACY PRESERVING ALGORITHM

Algorithm 4 shows the detailed implementation of the hybrid privacy-preserving algorithm for object detection in FL settings. Algorithm 2 [26] and Algorithm 3 [27] are simultaneously deployed to build this hybrid algorithm. For each round of communication, this algorithm ensures security and resiliency against backdoor attacks and inference attacks simultaneously.

Firstly, Algorithm 2 is run to identify the backdoor updates and discard them from the FL settings (line 1). In line 2, all the benign client indexes are stored in b_{id}. In line 3, we create an empty list bp_{id} to store all the benign parameters. A FOR loop is iterated over the benign indexes to extract the benign parameters from the local client parameter set (lines 4–6). All the parameters are appended to bp_{id}.

Algorithm 4: Hybrid Algorithm (Hybrid Privacy-Preserving Algorithm Against Backdoor Attack and Inference Attack) for a Total of P_t Training Rounds, Choose a Set of m Clients Denoted as S_t. The Local Dataset is Represented as $(D_{\text{train}})_{k=1}^{m}$: $\{(x, y)\}$, D_{poisoned} is a Poisoned Dataset, $D_{\text{benign}} \subseteq (D_{\text{train}})_{k=1}^{m}$ is a Benign Dataset, η is the Learning Rate, the Current Global Model is Denoted by $\mathbf{w}_t$, and $\mathbf{w}_{\text{pre}}^p$ are Pre-Aggregated Model Updates, ϵ is the Privacy Budget, δ is the Privacy Parameter, S is the Sensitivity, and ns is the Noise Scale

```
1  Run Algorithm 2
2  $b_{\text{id}} \leftarrow$ Store all the benign client indices
3  $bp_{\text{id}} \leftarrow []$ // Benign parameter list
4  for $i$ in $b_{id}$ do
5      $p \leftarrow l[i]$ // $l$ is a local parameter set including backdoor model
6      $bp_{\text{id}} \leftarrow p$ // $bp_{\text{id}}$ is a list of benign model updates
7  Run Algorithm 3
8  $\mathbf{w}_{t+1} \leftarrow$ Updated global model
9  Test $\mathbf{w}_{t+1}$ on $D_{\text{test}}$
10 Calculate Accuracy
```

Furthermore, Algorithm 3 is run to implement DP on the central server (line 7). In line 8, after the successful implementation of DP, the global model parameter is stored in $\mathbf{w}_{t+1}$. Finally, the global model is used to evaluate the performance using the D_{test} dataset (lines 9-10).

D. TIME COMPLEXITY

Our hybrid privacy-preserving algorithm consists of two modules namely PA-SM [26] and DP [27]. These modules are simultaneously integrated in the hybrid approach. The run time of the hybrid privacy-preserving algorithm mainly depends on the communication round (t), local epoch (e), batch size (B), and selected number of clients (m). The first part of this algorithm will take $O(teBm)$ time to execute. The second part will take $O(eBm^2)$ time. Considering the worst-case scenario, the time complexity of our approach is $O(teBm)$ which is similar to the FedAvg algorithm. However, the run time can be changed based on the input parameters during training. Keeping the local epoch and batch size at a constant value contributes to a decrease in run time.

V. EXPERIMENTS AND RESULT ANALYSIS

A. EXPERIMENTAL SETUP

FL, as a distributed ML technique, consists of two main components: a central server and edge client devices. In this paper, we consider AVs as edge clients that are capable of performing training on local data. Further, the central server can be an RSU or a traffic management system. We have used a benchmark dataset CIFAR10, and LISA traffic light dataset

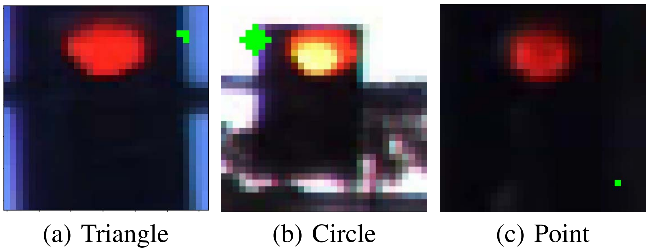

(a) Triangle (b) Circle (c) Point

FIGURE 4. Backdoor Samples.

for our experiments. CIFAR10 dataset includes 10 classes while LISA traffic light dataset includes 3 classes (green, red, and yellow light) [61]. Firstly, datasets are distributed to the selected number of AVs in terms of IID and Non-IID settings. In both settings, a subset of the training data has been shared with the edge client devices to locally train their model. We have employed ResNet18 model for object detection in ITS. In this setting, 100 clients (e.g., AVs) are considered in each round of communication. 20 clients (AVs) out of 100 clients are selected randomly to collaborate in the training process. Local epoch and batch size are set to 50 and 10, respectively, for every edge client AVs. In the experimental setup, we have used 10%, 20%, and 30% percent of local training data as a backdoor sample to evaluate the result. We have calculated and visualized the performance of backdoor attacker after every 5 local epochs during implementation. The learning rate is set to $\eta = 0.2$. Furthermore, we have used privacy parameters $\epsilon = 1.0$, $\epsilon = 2.0$, $\epsilon = 3.0$, $\epsilon = 4.0$, $\epsilon = 10.0$ and $\delta = 1\text{e-}5$ during the implementation of DP.

B. BACKDOOR SAMPLE

An attacker requires a compromised AVs to initiate a backdoor attack in FL settings. For experiment purpose, we have initiated backdoor attacks using five types of triggers: point, triangle, star, square, and circle. Among them, three have already been introduced in existing studies. In this study, we introduced two more triggers, circle and square, to assess the robustness of our proposed solution. In this attack scenario, the attacker changes the pixel color of other than targeted class. Furthermore, set the poisoned class label to targeted class concatenated it to benign train data. This mixed dataset trains the backdoor model (e.g., compromised AVs local model) and sends back the learned knowledge or parameter to the central server for aggregation. Example backdoor samples for the LISA traffic light dataset [61] are visualized in Figure 4.

C. RESULT ANALYSIS

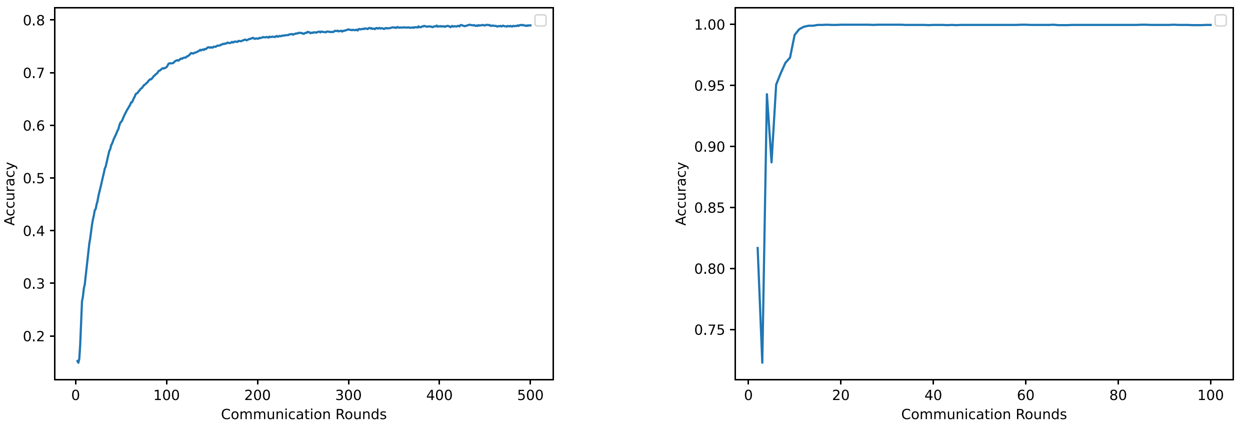
1) PERFORMANCE WITHOUT BACKDOOR ATTACK

Figure 5 shows the performance of the general FL in Non-IID setting. Firstly, we implemented a general FL using CIFAR10 and LISA traffic light datasets for 500 and 100 rounds of communication, respectively, using the FedAvg algorithm. We saved the model to a directory for further use. In this section, we calculated the accuracy of the model without initiating any backdoor attack. For the CIFAR10 dataset, the accuracy of the general FL at the 1st round of communication was 15.24%. By the 100th round, the performance showed rapid growth over the epochs and reached 71.00%. Moreover, performance exhibited a steady increase for the remaining rounds of communication. Finally, the accuracy reached 78.97% at the 500th round of communication. In the LISA dataset, the accuracy generally increased over the communication rounds, reaching a high level of accuracy towards the later rounds, with the last reported accuracy being 99.95%. Overall, the performance of the general FL experienced an upward trend throughout the communication rounds for both datasets.

2) PERFORMANCE WITH BACKDOOR ATTACK

We have simulated a backdoor attack from the compromised AVs in an FL environment, where we consider the 2nd client (e.g., malicious/compromised AVs) as a potential attacker and the rest as benign clients (e.g., benign AVs). In the depicted FL scenario shown in Figure 6, (a) illustrates the situation where the AV2 is compromised, with the remaining being benign AVs, employing 10% of poisoned samples to initiate a data poisoning-based model replacement attack. Additionally, (b) of Figure 6 demonstrates the scenario wherein the AV2 is compromised, alongside benign AVs, and utilizes 20% of poisoned samples for the attack. Lastly, (c) of Figure 6 depicts the scenario where the AV2 is compromised, with benign AVs, employing 30% of poisoned samples for the attack. The objective of the attacker is to insert specific backdoor samples into the dataset and alter the local model with the global model. For a successful attack, the attacker needs to assign a boosting factor that increases the similarity score of the attacker model, ensuring high equivalence between the similarity scores of the backdoor models. We observed a significantly higher similarity score for the attacker compared to the benign model. We initiated the backdoor attack at specific rounds (99th, 199th, 299th, 399th, 499th) of communication for CIFAR10 dataset. Due to space limitation, we only provide the result for 500th of communication in CIFAR10 dataset. We evaluated the backdoor attack performance in terms of accuracy by measuring the backdoor attacker's performance at round 500th, as depicted in (a) of Figure 7.

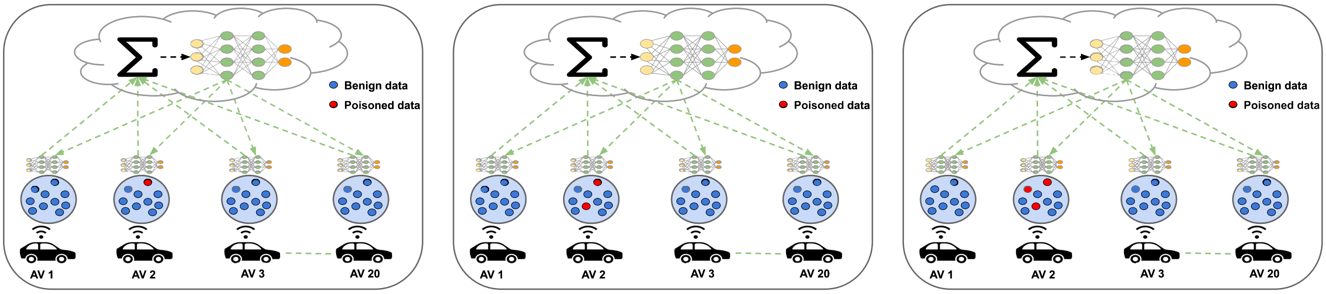
We observed that the backdoor performance is highly dependent on the percentage of poisoned samples, increasing proportionally as the percentage increases. In our investigation, we evaluated backdoor performance with 10%, 20%, and 30% poisoned samples. The experimental results showcase the impact of backdoor attacks on model accuracy across different percentages of injected poisoned samples. Starting with a 10% backdoor sample, accuracy demonstrated a decreasing trend from 28.3% in local epoch 0 to 22.6% in local epoch 45. The influence intensified with a 20% poisoned sample, stabilizing around 48.2% across epochs. The most pronounced effect was observed with a 30%



(a) Performance of the general FL without cyber attacks for the CIFAR10 dataset.

(b) Performance of the general FL without cyber attacks for the LISA dataset.

FIGURE 5. Accuracy vs communication round in FL.

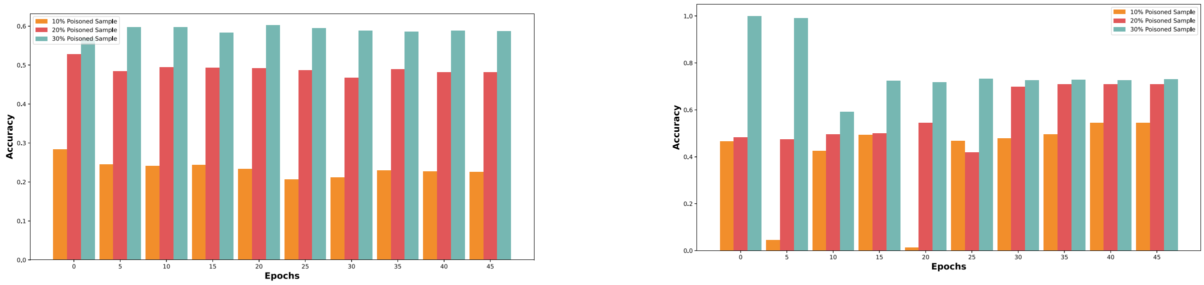


(a) Backdoor attack considering 10% poisoned sample in FL when 2nd AV is compromised.

(b) Backdoor attack considering 20% poisoned sample in FL when 2nd AV is compromised.

(c) Backdoor attack considering 30% poisoned sample in FL when 2nd AV is compromised.

FIGURE 6. FL training considering different distribution of poisoned sample.

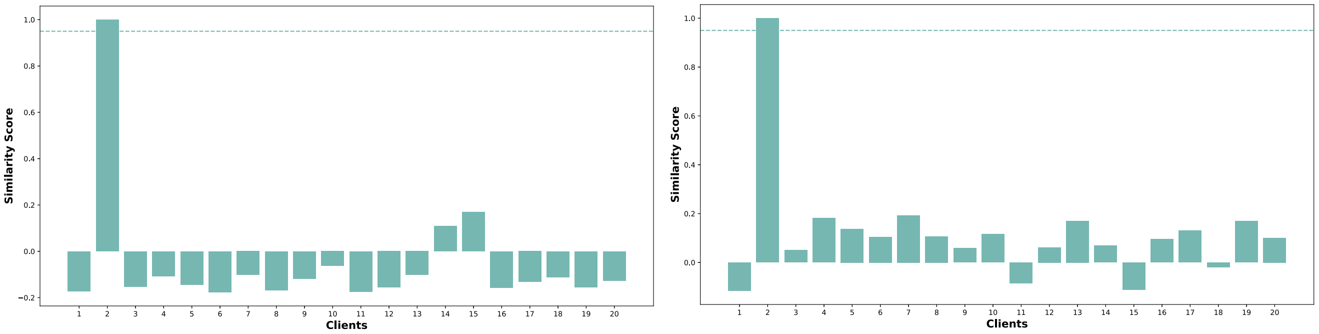


(a) Accuracy of backdoor attacker (2nd AV) on targeted task at the 500th round of communication considering 10%, 20%, and 30% poisoned sample for CIFAR10 dataset.

(b)Accuracy of backdoor attacker (2nd AV) on targeted task at the 100th round of communication considering 10%, 20%, and 30% poisoned sample for LISA dataset.

FIGURE 7. Performance of the backdoor attacker (e.g., compromised AVs) on targeted task at the selected round of communication in FL environment.

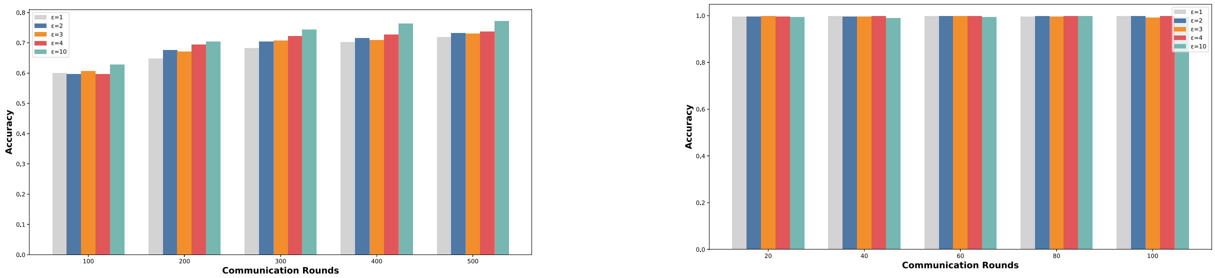
poisoned sample, leading to an accuracy surge from 56.9% to approximately 58.7% by local epoch 45. Moreover, We initiated the backdoor attack at specific rounds (19th, 39th, 59th, 79th, 99th) of communication for LISA traffic light dataset. We only depict the result at 100th round of communication in (b) of Figure 7. The model with 10% poisoned samples demonstrates moderate and fluctuating accuracy across local epochs. The 20% poisoned samples exhibit more variability, with accuracy showing fluctuations and occasional improvement. On the other hand, the model with 30% poisoned samples achieves perfect accuracy initially but experiences a subsequent decline. This analysis underscores the effectiveness of the backdoor attack strategy, highlighting its capability to manipulate model predictions



(a) Similarity score at 500th round of communication considering 20 clients and among them the second one is a backdoor attacker for CIFAR10 dataset.

(b) Similarity score at 100th round of communication considering 20 clients and among them the second one is a backdoor attacker for LISA dataset.

FIGURE 8. Similarity score of the edge client devices (e.g., AVs) at the selected round of communication in FL environment.



(a) Impact of DP on the performance of proposed hybrid algorithm at the selected round (100th, 200th, 300th, 400th, 500th) of communication [different privacy budget ϵ of DP, 30% of backdoor samples, Dataset: CIFAR-10].

(b) Impact of DP on the performance of proposed hybrid algorithm at the selected round (20th, 40th, 60th, 80th, 100th) of communication [different privacy budget ϵ of DP, 30% of backdoor samples, Dataset: LISA].

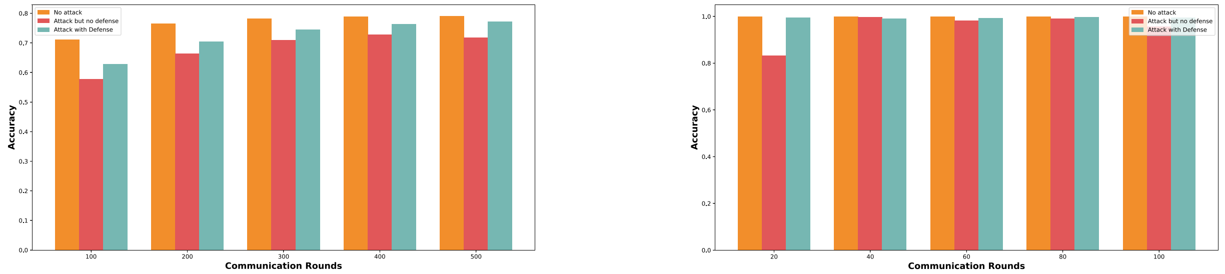
FIGURE 9. Impact of DP on the proposed hybrid algorithm.

by strategically injecting poisoned samples. The impact of these attacks becomes more significant, particularly at higher backdoor percentages. The similarity scores of all edge client's AVs actively involved in the training session are shown in Figure 8. The attacker aimed to enhance the performance of targeted tasks. We obtained a similarity score ranging from 0.98 to 0.99 for the backdoor model, while the benign model had a comparatively lower score. It's crucial to note that similarity scores can vary across datasets.

3) PERFORMANCE OF THE PROPOSED ALGORITHM

The backdoor attack has a detrimental impact on the global model as well as other local models. We employed a hybrid defense algorithm to overcome these issues, selecting specific rounds of communication as mentioned earlier to complete the assessment by initiating a backdoor attack. The in-depth analysis demonstrates that, regardless of the selected rounds, a backdoor attack can detrimentally affect the performance of the model. The PA-SM module provides a robust solution to isolate the backdoor update entirely. In the previous sub-subsection, we observed that the similarity score of the backdoor model is too high compared to the benign model. Based on the assessment, a threshold value of 0.95 has been assigned. If we received any similarity score greater than the threshold, we considered it a backdoor model update. The PA-SM module helps discard the outlier or backdoor update from the FL settings. Furthermore, the updates from the rest of the benign model are aggregated to build the global model. We implemented DP on the global model update. In this case, we added a small amount of Laplace noise to the global model to make it indistinguishable from other backdoor attackers.

Figure 9 illustrates the impact of different privacy budgets, denoted by ϵ, on the performance of our hybrid algorithm over multiple communication rounds. The bar chart (a) in Figure 9 compares model accuracy under DP ($\epsilon = 1.0$ to $\epsilon = 4.0$, and $\epsilon = 10.0$) at the selected round of communication for the CIFAR10 dataset. For $\epsilon = 1.0$, it starts with the lowest accuracy but shows steady improvement over time, achieving 71.86% by the 500th round. In contrast, $\epsilon = 10.0$ begins with higher initial accuracy and reaches the highest final accuracy of 77.13%. Other ϵ values, such as 2.0, 3.0, and 4.0, start with similar accuracies and also show significant improvement, ending above 73.00%. Specifically, $\epsilon = 4.0$ starts at an accuracy of 59.75% and rises to 73.78%, indicating substantial gains as privacy constraints are relaxed. The bar chart (b) in Figure 9 compares model accuracy under DP ($\epsilon = 1.0$ to $\epsilon = 4.0$, and $\epsilon = 10.0$) at



(a) Performance of our proposed hybrid algorithm at the selected round (100th, 200th, 300th, 400th, 500th) of communication [privacy budget ϵ = 10, 30% of backdoor samples, Dataset: CIFAR-10].

(b) Performance of our proposed hybrid algorithm at the selected round (20th, 40th, 60th, 80th, 100th) of communication [privacy budget ϵ = 10, 30% of backdoor samples, Dataset: LISA].

FIGURE 10. **Performance of our proposed hybrid algorithm.**

the selected round of communication for the LISA traffic light dataset. All ϵ values from $\epsilon = 1.0$ to $\epsilon = 4.0$, and $\epsilon = 10.0$ display remarkably high performance, with accuracies starting above 99.00% and maintaining this level across all rounds. The $\epsilon = 1.0$ setting begins at 99.50% accuracy and modestly improves to 99.82% by the 100th round, demonstrating that even the strictest privacy settings can yield exceptional performance. As the privacy settings relax, with $\epsilon = 2.0$, $\epsilon = 3.0$, $\epsilon = 4.0$, and $\epsilon = 10.0$, there is little variation in performance, indicating that the model is robust to changes in privacy levels. Particularly, $\epsilon = 3.0$ starts at an accuracy of 99.77% and slightly adjusts to 99.10%, showcasing minor fluctuations that suggest stable performance irrespective of privacy relaxation. The consistency across different privacy levels demonstrates how effectively our algorithm manages privacy without losing accuracy, making it ideal for applications that require both high precision and strict privacy.

Figure 10 analyzes the performance of our hybrid algorithm across three scenarios: no attack, attack without defense, and attack with defense, over several communication rounds. The bar chart (a) in Figure 10 demonstrates the comparative performance of the hybrid algorithm with different scenarios for the CIFAR10 dataset. Without any attacks, the model consistently improves from 71.06% to 78.97% accuracy, showing stable learning in a secure setting. However, when attacked without defenses, the model's accuracy drops to 57.75% before recovering to 72.81%, highlighting its vulnerability and the difficulty of fully recovering without protective measures. With defenses, the model starts at 62.81% and climbs to 77.13%, demonstrating the effectiveness of defensive strategies in combating attacks and allowing the model to nearly reach the performance seen in the no-attack scenario. The bar chart (b) in Figure 10 demonstrates the comparative performance of the hybrid algorithm under different scenarios for the LISA traffic light dataset. In the no-attack scenario, the model maintains nearly perfect accuracy, consistently around 99.95%, indicating optimal performance in a secure environment. However, when attacked without defense, the model's accuracy initially drops to 83.27%, then fluctuates, peaking at 99.03% and finally settling at 95.58%, highlighting the vulnerability and recovery challenges without defenses. Conversely, the model under attack but with defense shows robust resilience, with accuracy initially at 99.47%, slightly decreasing and then peaking at 99.78%, before ending at 99.42%.

Overall, our proposed hybrid algorithm provides robust security and resilience against adversarial attackers in a FL environment.

4) COMPARATIVE ANALYSIS

We conducted a comparative analysis of our approach with existing systems covering a similar threat model in the FL environment. However, we did not find any algorithms or systems developed for ITS applications that focus on both poisoning attacks and inference attacks simultaneously. We found one method, SAFEFL [50], developed generally to address backdoor attacks and inference attacks. To perform a comparative analysis, we used the LISA traffic light dataset and considered similar hyperparameters. As part of the experiments in SAFEFL, we used the FedAvg and FLTrust aggregation method. Twenty AVs with computing capabilities are required to participate in the FL process. Each AV holds a dataset (e.g., LISA traffic light) with non-i.i.d. data, indicating diverse data distributions across AVs. The server holds 100 data points in SAFEFL to perform the simulation. Additionally, we used a CNN model consisting of two convolutional layers and two fully connected layers. The model employs Batch Normalization after each convolutional layer, enhancing training stability and convergence. With a batch size of 64 and a learning rate of 0.10, hyperparameters are set to facilitate efficient model optimization. As part of the MPC implementation, we have utilized the replicated2k protocol.

SAFEFL used different secure aggregation techniques to safeguard FL models against various poisoning attacks and used MPC to make it resilient against inference attacks. In this paper, however, our focus is beyond one attack, and we consider both backdoor attacks and inference attacks simultaneously. SAFEFL covers scaling attacks as part of backdoor attacks, and we are focusing on data poisoning-based model replacement attacks. In scaling attacks, the attacker first injects backdoor patterns into the training data of targeted malicious clients and adjusts labels, followed by boosting the gradients from these compromised clients to disrupt the integrity of FL models. Both of these attacks fall into the same category. Further, we are focusing on inference attacks. The SAFEFL model achieves 98.18% accuracy on the main task using FLTrust techniques, while FedAvg achieved 51.30% accuracy in the presence of backdoor attacks (e.g., scaling attacks) without utilizing MPC. When we used aggregation techniques with MPC in the presence of backdoor attacks, we got 46.00% accuracy for FedAvg after 100 rounds of communication, while FLTrust got 87.61% accuracy at 338 rounds of communication. To defend against backdoor attacks, our proposed methods achieve 99.42% accuracy after 100 rounds of communications. After 100 rounds of communication, SAFEFL using FedAvg takes 8.67 hours, while FLTrust at 338 rounds takes 29.744 hours to converge. Considering the best performance of SAFEL using FLTrust, our method is approximately 11.80 times faster than SAFEFL, achieving 99.42% accuracy in just 2.52 hours.

Although after extended period of time the performance of SAFEFL can convergence close to our proposed model's performance. Due to the computational complexity of MPC, SAFEFL takes a longer period of time to converge, which is the main drawback of this method. The high computational complexity results in a loss of some accuracy on the main tasks for the mentioned dataset. Hence, we can conclude that our proposed model is more efficient and accurate than existing methods in the presence of mentioned simultaneous backdoor and inference attacks.

VI. DISCUSSION

In this study, we have reviewed various types of attacks and defense schemes. After a comprehensive assessment, our observation indicates that the effectiveness of the backdoor attack is contingent on both the fraction of users and the percentage of backdoor samples. Our proposed approach demonstrates a significant ability to mitigate both malicious attacks (data poisoning-based model replacement and inference attacks) simultaneously in the FL settings. We conducted model assessments for specific types of backdoor triggers introduced in previous research studies. Furthermore, we used two more backdoor triggers in this study to assess the resilience of our approach. Existing methods aimed to minimize the negative impact of backdoor attacks, but none of them discarded the malicious edge client device. object detection in ITS is a sensitive task because misclassification of traffic lights by AVs or traffic monitoring systems could lead to severe consequences, such as accidents on the road. Therefore, ensuring a higher precision of adversarial attack detection is crucial to minimize the detrimental impact. According to Lu et al. [26], the PA-SM outperforms traditional anomaly detection techniques in FL. As a result, we integrated PA-SM into our hybrid algorithm to mitigate the adversarial impact in FL settings. The results of our model evaluation demonstrate that the PA-SM module effectively reduces backdoor updates from the system, and DP provides security to the FL-based aggregated model against malicious clients for object detection tasks in ITS. Our emphasis has been on completely eliminating adversaries from the FL settings, and the global model shows robust performance even after frequent backdoor attacks initiated by the attacker. However, we noted that DP has some negative impact on the performance of our global model. Nevertheless, our defense scheme can effectively reduce the negative impact of the backdoor attack while providing consistent performance, including preserving the privacy of data. It's important to acknowledge that there might be some limitations in backdoor update detection based on novel triggers created by attackers.

VII. CONCLUSION AND FUTURE REMARKS

Our study introduces a novel hybrid privacy-preserving FL algorithm designed to defend against data poisoning-based model replacement attacks and inference attacks in object detection for ITS. The proposed approach demonstrated significant backdoor update detection accuracy during training, achieving a balance between robust security and low time complexity. Through a comprehensive evaluation with CIFAR10 and LISA traffic light datasets, our algorithm effectively isolated backdoor updates without significantly compromising the performance of benign and global models. Although the introduction of DP impacted the global model's performance, our defense scheme successfully mitigated the negative effects of backdoor attacks, ensuring consistent performance while safeguarding data privacy. The result analysis highlighted the efficacy of our approach against both types of attacks and compared it with existing systems in ITS. Acknowledging potential limitations in backdoor update detection, future work will focus on refining the defense scheme, extending its capabilities to counter diverse cyber-threats, and strengthening ITS security in the dynamic landscape of FL.

ACKNOWLEDGMENT

This work is based upon the work supported by the National Center for Transportation Cybersecurity and Resiliency (TraCR) (a U.S. Department of Transportation National University Transportation Center) headquartered at Clemson University, Clemson, South Carolina, USA. Any opinions, findings, conclusions, and recommendations expressed in this material are those of the authors and do not necessarily reflect the views of TraCR, and the U.S. Government assumes no liability for the contents or use thereof.

REFERENCES

[1] W. Yamany, N. Moustafa, and B. Turnbull, "OQFL: An optimized quantum-based federated learning framework for defending against adversarial attacks in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 893–903, Jan. 2023.

[2] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, “Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 998–1026, 2nd Quart., 2020.

[3] A. Greenberg, “Hackers remotely kill a jeep on the highway—With me in it,” *Wired*, vol. 7, no. 2, pp. 21–22, 2015.

[4] X. Sun, F. R. Yu, and P. Zhang, “A survey on cyber-security of connected and autonomous vehicles (CAVs),” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6240–6259, Jul. 2022.

[5] V. M. Lidkea, R. Muresan, and A. Al-Dweik, “Convolutional neural network framework for encrypted image classification in cloud-based ITS,” *IEEE Open J. Intell. Transp. Syst.*, vol. 1, pp. 35–50, 2020.

[6] L. Hacker and J. Seewig, “Insufficiency-driven DNN error detection in the context of SOTIF on traffic sign recognition use case,” *IEEE Open J. Intell. Transp. Syst.*, vol. 4, pp. 58–70, 2023.

[7] A. Haydari, M. Zhang, and C.-N. Chuah, “Adversarial attacks and defense in deep reinforcement learning (DRL)-based traffic signal controllers,” *IEEE Open J. Intell. Transp. Syst.*, vol. 2, pp. 402–416, 2021.

[8] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Proc. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[9] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, “How to backdoor federated learning,” in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2938–2948.

[10] A. Hard et al., “Federated learning for mobile keyboard prediction,” 2018, *arXiv:1811.03604*.

[11] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu, “Patient clustering improves efficiency of federated machine learning to predict mortality and hospital stay time using distributed electronic medical records,” *J. Biomed. Informat.*, vol. 99, Nov. 2019, Art. no. 103291.

[12] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, “Federated learning for data privacy preservation in vehicular cyber-physical systems,” *IEEE Netw.*, vol. 34, no. 3, pp. 50–56, May/Jun. 2020.

[13] A. M. Elbir, B. Soner, S. Çöleri, D Gündüz, and M. Bennis, “Federated learning in vehicular networks,” in *Proc. IEEE Int. Mediterr. Conf. Commun. Netw. (MeditCom)*, 2022, pp. 72–77.

[14] W. Zhang et al., “Blockchain-based federated learning for device failure detection in industrial IoT,” *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5926–5937, Apr. 2021.

[15] M. S. Jere, T. Farnan, and F. Koushanfar, “A taxonomy of attacks on federated learning,” *IEEE Secur. Privacy*, vol. 19, no. 2, pp. 20–28, Mar./Apr. 2021.

[16] J. Steinhardt, P. W. Koh, and P. Liang, “Certified defenses for data poisoning attacks,” in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3520–3532.

[17] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, “Data poisoning attacks against federated learning systems,” in *Proc. 25th Eur. Symp. Res. Comput. Secur. (ESORICS)*, 2020, pp. 480–501.

[18] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, and J. Liu, “Data poisoning attacks on federated machine learning,” *IEEE Internet Things J.*, vol. 9, no. 13, pp. 11365–11375, Jul. 2022.

[19] M. Fang, X. Cao, J. Jia, and N. Gong, “Local model poisoning attacks to {Byzantine-robust} federated learning,” in *Proc. 29th USENIX Secur. Symp. (USENIX Security 20)*, 2020, pp. 1605–1622.

[20] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, “Model poisoning attacks in federated learning,” in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS) Workshop Secur. Mach. Learn. (SecML)*, 2018, pp. 1–23.

[21] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” 2017, *arXiv:1712.05526*.

[22] Peter Kairouz et al., “Advances and open problems in federated learning,” *Found. Trends® Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, 2021.

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

[24] M. Nasr, R. Shokri, and A. Houmansadr, “Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning,” in *Proc. IEEE Symp. Secur. Privacy (SP)*, 2019, pp. 739–753.

[25] J. Zhang, J. Zhang, J. Chen, and S. Yu, “GAN enhanced membership inference: A passive local attack in federated learning,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, 2020, pp. 1–6.

[26] S. Lu, R. Li, W. Liu, and X. Chen, “Defense against backdoor attack in federated learning,” *Comput. Secur.*, vol. 121, Oct. 2022, Art. no. 102819.

[27] M. Abadi et al., “Deep learning with differential privacy,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[28] Ú. Erlingsson et al., “Amplification by shuffling: From local to central differential privacy via anonymity,” in *Proc. 13th Annu. ACM-SIAM Symp. Discrete Algorithm.*, 2019, pp. 2468–2479.

[29] X. Han, G. Xu, Y. Zhou, X. Yang, J. Li, and T. Zhang, “Physical backdoor attacks to lane detection systems in autonomous driving,” in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 2957–2968.

[30] Y. Wang, M. Maniatakos, and S. E. Jabari, “A trigger exploration method for backdoor attacks on deep learning-based traffic control systems,” in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, 2021, pp. 4394–4399.

[31] Q. Zhang, Y. Ding, Y. Tian, J. Guo, M. Yuan, and Y. Jiang, “AdvDoor: Adversarial backdoor attack of deep learning system,” in *Proc. 30th ACM SIGSOFT Int. Symp. Softw. Test. Anal.*, 2021, pp. 127–138.

[32] X. Cao, M. Fang, J. Liu, and N. Z. Gong, “FLtrust: Byzantine-robust federated learning via trust bootstrapping,” 2020, *arXiv:2012.13995*.

[33] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” in *Proc. 31st Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[34] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[35] T. D. Nguyen et al., {FLAME}: Taming backdoors in federated learning,” in *Proc. 31st USENIX Secur. Symp. (USENIX Security)*, 2022, pp. 1415–1432.

[36] Z. Ma, J. Ma, Y. Miao, Y. Li, and R. H. Deng, “ShieldFL: Mitigating model poisoning attacks in privacy-preserving federated learning,” *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1639–1654, 2022.

[37] Y. Dong, X. Chen, K. Li, D. Wang, and S. Zeng, “FLOD: Oblivious defender for private Byzantine-robust federated learning with dishonest-majority,” in *Proc. Euro. Symp. Res. Comput. Secur.*, 2021, pp. 497–518.

[38] C. Wu, X. Yang, S. Zhu, and P. Mitra, “Mitigating backdoor attacks in federated learning,” 2020, *arXiv:2011.01767*.

[39] C. Fung, C. J. M. Yoon, and I. Beschastnikh, “The limitations of federated learning in sybil settings,” in *Proc. 23rd Int. Symp. Res. Attacks, Intrus. Defenses (RAID)*, 2020, pp. 301–316.

[40] S. Awan, B. Luo, and F. Li, “CONTRA: Defending against poisoning attacks in federated learning,” in *Proc. 26th Eur. Symp. Res. Comput. Secur.*, 2021, pp. 455–475.

[41] J. Xu, S.-L. Huang, L. Song, and T. Lan, “Byzantine-robust federated learning through collaborative malicious gradient filtering,” in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2022, pp. 1223–1235.

[42] 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.*, 2022, pp. 946–958.

[43] Y. Mao, X. Yuan, X. Zhao, and S. Zhong, “Romoa: Ro bust mo del a ggregation for the resistance of federated learning to model poisoning attacks,” in *Proc. 26th Eur. Symp. Res. Comput. Secur.*, 2021, pp. 476–496.

[44] R. Zhu, M. Li, J. Yin, L. Sun, and H. Liu, “Enhanced federated learning for edge data security in intelligent transportation systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13396–13408, Nov. 2023.

[45] R. Parekh et al., “GeFL: Gradient encryption-aided privacy preserved federated learning for autonomous vehicles,” *IEEE Access*, vol. 11, pp. 1825–1839, 2023.

[46] J.-H. Chen, M.-R. Chen, G.-Q. Zeng, and J.-S. Weng, “BDFL: A Byzantine-fault-tolerance decentralized federated learning method for autonomous vehicle,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8639–8652, Sep. 2021.

[47] F. O. Olowononi, D. B. Rawat, and C. Liu, "Federated learning with differential privacy for resilient vehicular cyber physical systems," in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2021, pp. 1–5.

[48] H. Batool, A. Anjum, A. Khan, S. Izzo, C. Mazzocca, and G. Jeon, "A secure and privacy preserved infrastructure for VANETs based on federated learning with local differential privacy," *Inf. Sci.*, vol. 652, Jan. 2024, Art. no. 119717.

[49] R. Canetti, "Security and composition of multiparty cryptographic protocols," *J. Cryptol.*, vol. 13, pp. 143–202, Apr. 2000.

[50] T. Gehlhar, F. Marx, T. Schneider, A. Suresh, T. Wehrle, and H. Yalame, "SafeFL: MPC-friendly framework for private and robust federated learning," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2023, pp. 69–76.

[51] 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.

[52] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009. Accessed: Apr. 1, 2024. [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html

[53] M. Naseri, J. Hayes, and E. D. Cristofaro, "Local and central differential privacy for robustness and privacy in federated learning," 2020, *arXiv:2009.03561*.

[54] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557*.

[55] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" 2019, *arXiv:1911.07963*.

[56] Y. Liu et al., "A survey on neural trojans," in *Proc. 21st Int. Symp. Qual. Electron. Design (ISQED)*, 2020, pp. 33–39.

[57] Y. Li, Y. Jiang, Z. Li, and S.-T. Xia, "Backdoor learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 5–22, Jan. 2024.

[58] A. Turner, D. Tsipras, and A. Madry, "Label-consistent backdoor attacks," 2019, *arXiv:1912.02771*.

[59] 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.*, 2021, pp. 16463–16472.

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

[61] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: Issues, survey, and perspectives," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1800–1815, Jul. 2016.

MD. JUEAL MIA received the B.S. and M.S. degrees in computer science and engineering from Jahangirnagar University, Bangladesh. He is currently pursuing the Ph.D. degree with the Knight Foundation School of Computing and Information Sciences, Florida International University (FIU), where he is a Graduate Research Assistant. He has published 30 peer-reviewed journal and conference publications. Before joining FIU, he served as a Faculty Member with the Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh, for more than six years. His research interests include privacy and security issues in federated learning, computer vision, machine learning, deep learning, and data mining. He is a member of the Sustainability, Optimization, and Learning for InterDependent Networks Laboratory (solid lab) and is advised by Dr. M. Hadi Amini.

M. HADI AMINI (Senior Member, IEEE) received the M.Sc. degree and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University in 2015 and 2019, respectively, and the Doctoral degree in computer science and technology. He is an Assistant Professor with the Knight Foundation School of Computing and Information Sciences, Florida International University. He is the Director of the Sustainability, Optimization, and Learning for InterDependent Networks Laboratory (solid lab). He serves as the Director of the Advanced Education and Research for Machine Learning-Driven Critical Infrastructure Resilience (ADMIRE) supported by U.S. DHS, and an Associate Director of the U.S. DOT National Center for Transportation Cybersecurity and Resiliency (TraCR). He edited/authored nine books. His research interests include secure and privacy-preserving distributed learning and optimization, cybersecurity, and resilience of cyber–physical systems. Application domains include critical infrastructures, healthcare, public safety, and smart cities. He received the Best Paper Award from "2019 IEEE Conference on Computational Science & Computational Intelligence," 2021 Best Journal Paper Award from "Springer Nature Operations Research Forum Journal," 2024 "FIU Top Scholar Award, Research and Creative Activities, Junior Faculty with Significant Grants (Sciences)," 2023 FIU "Faculty Senate Excellence in Teaching Award," Excellence in Teaching Award from FIU Knight Foundation School of Computing and Information Sciences in 2020, and the best reviewer award from four IEEE Transactions. He serves as an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.