LFD-IDS: bagging-based data poisoning attacks against cyberattack detection in connected vehicle

Article

Accepted Version

LFD-IDS: Bagging-Based Data Poisoning Attacks Against Cyberattack Detection in Connected Vehicle

Zahra Pooranian, *Senior Member, IEEE*, Rahim Taheri, *Senior Member, IEEE*, and Fabio Martinelli, *Senior Member, IEEE*

***Abstract*— This paper explores the need for new systems to detect and monitor cyberattacks in Connected Vehicles (CVs). Sensor health in CVs is vital, as prediction errors and communication issues can weaken the sensor network. Intrusion Detection Systems (IDS) for CVs must be continuously updated to meet changing needs and be robust against adversarial attacks. We developed a new Label Flipping system against Deep learning-based IDS (LFD-IDS) to help cloud operators understand unusual vehicle sensor data. LFD-IDS specifically targets detecting and explaining sensor data manipulation from poisoning attacks. We proposed two label-flipping attacks based on Bootstrapping and Bagging and a defensive strategy using a multi-layer deep neural network. Our LFD-IDS achieves at least 90% accuracy in identifying cyberattacks.**

***Index Terms*— Connected vehicle, machine learning, adversarial attacks, cyberattack, intrusion detection systems (IDS).**

I. Introduction

IMPLEMENTING vehicular networks within the Intelligent Transportation Systems (ITS) framework enhances connectivity, safety, and convenience for road users. ITS facilitates the advancement of Connected Vehicles (CVs) and provides system administrators with improved traffic management capabilities [1]. However, the increasing autonomy of vehicles, integration of various wireless communication technologies, and shift towards network virtualization and cloud computing have raised significant concerns about potential cyber-attacks in the pursuit of connected and autonomous vehicles [2]. Connected and autonomous vehicles are viewed as vital systems for preserving human life. Therefore, strong security measures must be implemented to protect CVs from cyber attacks and ensure the safety of drivers, passengers, other road users, and the environment.

An intrusion detection system (IDS) shows significant promise in securing networks, as it monitors traffic entering and exiting network components to detect any malicious activity. Various types of IDS exist, each distinguished by its approach to identifying potential intrusions. One type, known as anomaly-based detection, employs a predefined model to characterize normal behavior and then compares incoming traffic against this standard. Any deviations from normal behavior are marked as potential attacks [3]. Despite extensive research, traditional IDS methods face significant challenges in identifying new and previously unseen attacks. Traditional IDS often needs to be improved to provide the necessary level of detection effectiveness, particularly within the highly dynamic environment of vehicular networks. Consequently, there has been significant interest from both academia and industry in adopting cloud-based solutions to align with the advancements in 5G technology. Recent studies [4] indicate that integrating Machine Learning (ML) capabilities into IDS can substantially enhance their accuracy in detecting threats.

ML methods are adept at accurately predicting patterns in data, but they can be vulnerable when the data comes from unreliable or uncertain sources. Attackers can exploit this vulnerability through *Adversarial Machine Learning (AML)* attacks. One specific type of AML attack is a *Poisoning attack* where adversaries inject malicious perturbations into datasets that can lead to erroneous results in offline learning models and real-time decision-making systems [5]. A form of data poisoning, known as *Label-flipping*, involves attackers altering the labels assigned to training samples, which can significantly degrade the system's performance.

In [6], AML techniques concentrate on two primary aspects: i) Attack Complexity, which aims to simplify the process of creating a malicious attack, and ii) Attacker's Knowledge, which refers to the attacker's understanding of the system's architecture, algorithms, and training examples to gain insights into the detector. A *white-box attack* occurs when the attacker has detailed knowledge of the training data, features extracted from applications, or the system's architecture, as described in approaches like [7]. Conversely, if the attacker has limited knowledge, the attack is termed a *black-box attack* [8].

The concept of adversarial attack specificity can be approached in either a *targeted* or *non-targeted* manner. In a targeted attack, the attacker aims to deceive a classifier in detection systems by causing all adversarial samples to be predicted as a specific class, thereby increasing the chances of achieving a specific adversarial objective. Conversely, non-targeted attackers arbitrarily target a class by conducting

Received 9 October 2024; revised 7 February 2025 and 7 June 2025; accepted 14 June 2025. This work was supported in part by the Research Development Support (RDS)—PTR22-24 P2.1 Cybersecurity through the tenure of an European Research Consortium for Informatics and Mathematics (ERCIM) "Alain Bensoussan" Fellowship Programme. The Associate Editor for this article was S. Kumari. *(Corresponding author: Rahim Taheri.)*

Zahra Pooranian is with the Department of Computer Science, University of Reading, RG6 6AH Reading, U.K. (e-mail: z.pooranian@reading.ac.uk).

Rahim Taheri is with the PAIDS Research Centre, School of Computing, University of Portsmouth, PO1 3HE Portsmouth, U.K. (e-mail: rahim.taheri@port.ac.uk).

Fabio Martinelli is with the Consiglio Nazionale delle Ricerche (CNR), 56124 Pisa, Italy (e-mail: fabio.martinelli@iit.cnr.it).

Digital Object Identifier 10.1109/TITS.2025.3581102

various targeted attacks and selecting the one that causes the least disruption or minimizes the likelihood of correctly classifying the sample [9].

Many studies have focused on detecting and mitigating poisoning attacks. For example, an algorithmic technique evaluates the impact of each training sample on the learning algorithm's efficiency [10]. While this method can be effective in specific scenarios, it is only sometimes applicable to large datasets. Other defensive strategies, like *outlier detection*, are used to identify and remove suspicious samples. However, the effectiveness of this method is limited, especially in terms of accuracy when dealing with label-flipping attacks [11]. Another area of research focuses on creating strategies for learning that can be applied to label flipping. Solutions to this problem fall into two main categories. The first approach involves obtaining information directly from the inverted labels, while the second approach emphasizes using clean data. In the initial method, the label-flipping component detects accurately labeled data [12], [13] and modifies the label alterations to harmonize the data terms within the loss function. The efficacy of this approach largely hinges on the precision of label cleaning and the accuracy of identifying flipped instances. The second method employs an extra set of adversarial data to direct the learning process in managing flipped data [14]. While showing encouraging outcomes, both approaches have shared limitations. They aim to rectify flipped labels or modify the weights of data samples, potentially resulting in inaccuracies for specific data points.

Motivated by the aforementioned considerations, this study introduces Label Flipping against Deep learning-based IDS (LFD-IDS) architecture to help cloud operators understand unusual vehicle sensor data. The paper outlines a fleet-based scenario where vehicle sensor data, such as tyre pressure, temperature, and location, is collected and transmitted to a cloud server via a 5G cellular network. The scenario assumes the attacker has minimal capabilities and lacks knowledge of the loss function and learning algorithm, but at the same time has access to training data and can do a white-box attack. The study demonstrates that better results can be obtained when the system detects and retrains incorrect labels using the proposed training method. Consequently, the solution focuses on correcting mislabeled data points to improve the accuracy of the classification method. This approach requires accurate labels for a small portion of the training set while disregarding the returned labels associated with the rest of the data. This is followed by training a multi-layer neural network using this selectively chosen data in a semi-supervised manner. To summarize, the main contributions of this study are:

- Our main emphasis is on effectively detecting intrusions within CV systems. To achieve this, we developed a new architecture called LFD-IDS, which facilitates handling flipped data.
- In LFD-IDS architecture, we propose two adversarial attacks employing Bootstrapping and Bagging as label-flipping to disrupt deep learning-based CV detection.
- We introduce a defence mechanism to counter label-flipping attacks in CV systems, utilizing K-means clustering to predict new labels for the training set.
- We conduct experiments using a real-world dataset from CV systems featuring four different types of attributes. These experiments encompass two attack scenarios and are benchmarked against a non-attack approach. We provide an in-depth analysis of the resulting trade-offs.

This paper is structured as follows: Section II provides an in-depth discussion of the problem definition and architecture. In Section III, we describe the attack model, which is inspired by AML methods, and the defence strategy designed to counter these attacks. The experimental results are presented in Section IV. Finally, Section VI concludes the paper by summarizing the findings and outlining future research directions.

II. System Model and Proposed Architecture

This section outlines the problem definition and proposed architecture for LFD-IDS.

A. Problem Definition

Let's analyze the dataset as follows.

$$\mathcal{S} = (x_i, y_i) \in (\mathcal{X}, \mathcal{Y}), \quad i = 1, \ldots, N \tag{1}$$

In this context, N represents the number of samples. If x_i includes the j_{th} feature, then x_{ij} is equal to 1; otherwise, x_{ij} is set to 0. Here, $\mathcal{X}$ denotes a subset of a k-dimensional space, where it comprises elements from the set $\{0, 1\}^k$. The samples are assigned labels represented by y_i values, which belong to the set $\{0, 1\}$. The distribution of $\mathcal{S}$ over $\mathcal{X} \times \mathcal{Y}$ is unspecified. The training set is assumed to be defined as follows:

$$\mathcal{L} = (x_m, y_m), \quad m = 1, \ldots, t \tag{2}$$

where $\mathcal{L}$ denotes the set of features and labels.

Definition 1: The particular type of poisoning attack is known as a *Label Flipping Attack (LFA)*. In this attack, the adversary seeks to alter feature labels using specific algorithms, thereby changing the range of each sample within a cluster.

The LFA aims to identify a set, denoted as Q, consisting of samples from $\mathcal{L}$. The attacker seeks to minimize the desired target by flipping the labels by the following equation:

$$y' = |1 - y| \tag{3}$$

For simplicity, we assume that the attacker seeks to maximize the loss function, defined as $\mathcal{L}(w, (x_j, y'_j))$, where w shows the ML model.

B. Proposed Architecture

Our proposed architecture for LFD-IDS is specifically designed to counter adversarial manipulations in CV systems. This sub-section details the architecture and elaborates on the capabilities of potential attackers as well as the methods used to inject malicious data into the dataset.

Fig. 1 illustrates the real-time LFD-IDS architecture. This architecture centres around monitoring and analyzing sensor data from CVs, which includes temperature, pressure, and location data collected from each vehicle's on-board sensors.

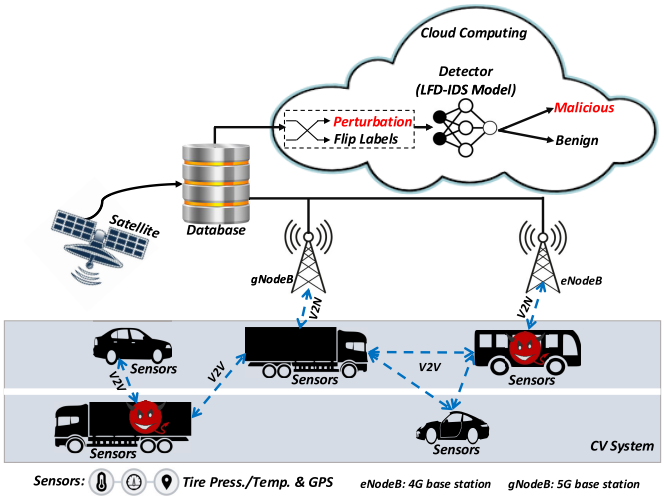

Fig. 1. Architecture of real-time LFD-IDS in CVs environment; Temp:= Temperature; Press:= Pressure.

We assume that each vehicle is equipped with sensors that collect data. Then, data is transmitted to a cloud-based server via a secure 5G/4G connection, where it is processed and analyzed for potential intrusions and the final model is disseminated back to the vehicles for deployment. The IDS is implemented in the cloud. This setup allows for comprehensive data analysis and management of sensor data from multiple vehicles simultaneously, leveraging cloud computing's extensive processing power and storage capabilities. The central component of our architecture is a multi-layer neural network designed to detect anomalies and classify data points as either benign or malicious based on learned patterns of sensor behavior. The final trained model will be shared with all vehicles, and they can use it.

In details, each vehicle has temperature and pressure sensors on every wheel, which monitor the internal pressure and temperature. The Global Positioning System (GPS) receiver also identifies each vehicle's location. A 5G/4G base station links the location, pressure, and temperature sensors, gathering the data and sending sensor updates to a cloud-based database. This sensory system measures temperature in degrees Celsius, pressure in Pounds per Square Inch (PSI), and location in degrees of Latitude and Longitude. In this paper, we only focused on manipulating data (in any way), but in real world these manipulations can cause the following issues.

- Attacks on one vehicle can quickly spread to others through Vehicle-to-vehicle (V2V) communications, amplifying the impact of the attack. For example, sending false safety messages could cause other vehicles to take unnecessary evasive actions, potentially leading to confusion or collisions.
- Attackers could passively collect data transmitted over V2V to gather sensitive information, track vehicle movements, or profile driver behavior.

We assume the vehicles can communicate in the LFD-IDS system, and the incoming CV data stream is stored in a cloud-based database.

1) Attacker Capabilities: In the scenario envisaged, attackers have specific capabilities that allow them to manipulate sensor data before it reaches the cloud server. These capabilities include:

- Attackers can access training data used to train the IDS. This access could be gained through breaches in data storage or transmission systems. we assume that the attacker only attacks the dataset and the IDS, which is trained in the cloud, because there is essentially no other training step for IDS in vehicles.
- While attackers do not have complete knowledge of the underlying algorithms and loss functions, they possess sufficient understanding to manipulate data effectively. This includes knowledge of data formats, transmission protocols, and basic neural network architecture employed by the LFD-IDS.
- We assume that attackers may have access to the training data transmitted from each sensor to the cloud and may perturb it. This access could be due to compromising the communication and eavesdropping on it, and doing a Man-in-the-Middle attack. The modification can be perturbing sensor messages and sending them to the cloud, or even generating fake messages by an attacker. The added perturbations can adversely affect the model training process, resulting in a model that fails to accurately classify unseen samples. For example, the attack may involve flipping the labels of data points in the training set—mislabeling benign samples as malicious and vice versa—thereby causing the model to learn from incorrect data.
- The attacker can affect the labels assigned to collected data through several methods, primarily targeting the data before it reaches the cloud-based IDS. Here's a detailed explanation of how this could occur:
  1) Attackers can intercept the data transmitted from the vehicle sensors to the cloud. During this interception, they can modify the data values and the labels associated with them. For example, data indicating normal operations could be altered to appear as though they signify a cyberattack, or vice versa.
  2) If any local preprocessing or labeling occurs in the vehicle, attackers might compromise the software responsible for this initial data handling. By infecting these systems with malware or exploiting vulnerabilities, they could manipulate the labels assigned to the data before it is sent to the cloud.
- We assume that an attacker can only alter the data of the vehicle they control and cannot affect the data of other vehicles.

Additionally, it is assumed that an attacker might gain access to some CV sensors, enabling them to alter data sent between vehicles or the cloud. As a result, the data traffic from each vehicle could include information from malicious vehicles, depicted in the figure by devil symbols. Each vehicle produces a unique feature vector with distinct labels indicating whether the data is malicious or benign. The attackers aim to mislead ML models and avoid detection by injecting perturbations

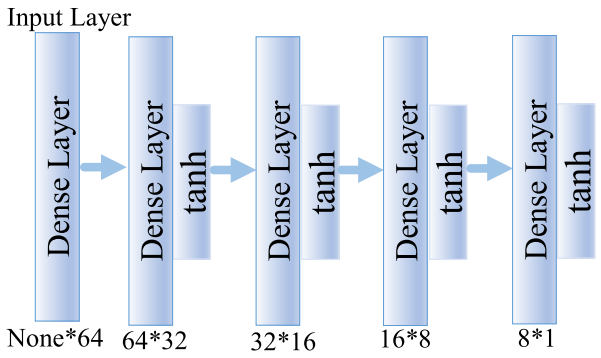

Fig. 2. The used classification model.

into the data. As a result, within this architecture, adversaries can infiltrate the dataset and modify the labels by adding perturbations to the existing data. The final component of our framework includes our proposed defense methods and an ML model, which constitutes the detection system. This architecture enhances the detection system's robustness against LFA attacks, improving classification accuracy between malicious and benign entities.

III. Proposed Attack and Defensive Solutions

This paper employs a sequential deep-learning model to classify the samples. Fig. 2 depicts the proposed sequential architecture for the classification method. The figure demonstrates that the classification method uses three sequential dense layers with 32, 16, and 8 units and activation functions of tanh, relu, and tanh, respectively. Following these layers, a dense layer with a Sigmoid activation function is used to finalize the classification, resulting in the data being categorized.

In this paper, we proposed two LFA attacks against deep learning-based IDS and one defence method against attacks, namely:

1- BOOTLFA: Bootstrapping-based Label Flipping Attack
2- BAGLFA: Bagging-based Label Flipping Attack
3- KCD: K-means-based Clustering Defence

The potential impact of these attacks is significant, particularly in the context of vehicle safety and operational reliability. Our proposed methods, BOOTLFA and BAGLFA, specifically target the IDS that are crucial for maintaining the security of connected vehicle networks. The real-world impact of such attacks could be profound. Successful implementation of these attacks could potentially lead to unauthorized access to vehicle controls and sensitive data. For instance, manipulating sensor data could cause erroneous vehicle responses, misleading information displayed to drivers, or inappropriate actions by autonomous driving systems, all of which could compromise passenger safety. Additionally, these vulnerabilities could be exploited to perform more sophisticated attacks, such as disabling the vehicle remotely or coordinating large-scale disruptions across a network of vehicles.

The data that can be altered by the attacker consists of sensor outputs from CVs, which are crucial for the proper functioning and safety of these vehicles. The types of data that can be targeted by attackers include:

- Temperature Data: Attackers could manipulate readings to fake overheating or cooling situations, which could mislead the vehicle's control systems into making erroneous decisions, like shutting down the engine prematurely or failing to regulate operating temperatures properly.
- Pressure Data: False data could suggest that tire pressure is either too high or too low, leading to inappropriate responses from the vehicle, such as unnecessary speed reduction, alerts that could distract the driver, or even ignoring genuine tire pressure issues that could lead to accidents.
- Location Data: Misleading location data could disrupt navigation systems, leading to wrong routing, delays, or mismanagement in fleet operations. In more severe cases, it could also be used to mislead law enforcement or emergency services.
- Speed Data: Altering speed data could lead to incorrect speed readings, potentially causing the driver to violate traffic laws unknowingly or causing the vehicle's safety systems to engage or disengage inappropriately.
- Fuel Level Data: By manipulating fuel level readings, attackers could cause drivers to believe they have either more or less fuel than is actually the case, potentially leading to stranded vehicles or unnecessary fuel stops, which could also compromise routing efficiency and safety.

A. BOOTLFA: Bootstrapping-Based Label Flipping Attack

This work introduces a novel adversarial attack strategy called BOOTLFA (Bootstrapping-based Label Flipping Attack). This method is designed to evaluate the resilience of ML-based IDS models to adversarial label noise, a common threat in real-world applications. The BOOTLFA method builds upon the traditional bootstrapping technique, which involves creating multiple bootstrap samples from an original dataset by randomly selecting data points with replacements. These samples are then used to train multiple models, with each model capturing different aspects of the data distribution.

BOOTLFA introduces controlled label noise during bootstrapping to transform bootstrapping into an adversarial attack. Specifically, for each bootstrap sample, a fraction p of the selected data points have their labels flipped, thereby injecting noise into the training data. The fraction p represents the intensity of the attack, allowing us to systematically explore the model's vulnerability to different levels of label corruption. This method is presented in Alg. 1. In this pseudo-code, the adversary with access to the training data performs the label flipping operation, corresponding to lines 1 to 6 of the above algorithm. The subsequent steps follow the standard procedure of the Bootstrapping algorithm.

The Time Complexity of the BOOTLFA method includes the following parts. First, the sampling process has a time complexity of $O(N)$ per iteration. Since this is repeated for B iterations, the total time complexity for bootstrapping is

Algorithm 1 BOOTLFA Algorithm

Require: Dataset D with N samples, number of bootstrap samples B, label flipping fraction p

for each $i \in \{1, 2, \ldots, B\}$ **do**
  Generate a bootstrapped dataset D_i by randomly selecting N samples from D with replacement
  Randomly select a fraction p of samples from D_i
  for each selected sample $s \in D_i$ **do**
    Flip the label of sample s
  end for
  Train model M_i on D_i
end for
Aggregate the results of models $\{M_1, M_2, \ldots, M_B\}$ for evaluation

Algorithm 2 BAGLFA Algorithm

Require: Dataset D with N samples, number of subsets B, label flipping fraction p

for each $i = 1$ to B **do**
  Generate a subset dataset D_i by randomly selecting N samples from D with replacement
  Randomly select a fraction p of samples from D_i
  for each selected sample s in D_i **do**
    Flip the label of sample s
  end for
  Train model M_i on D_i
end for
Aggregate the results of models $\{M_1, M_2, \ldots, M_B\}$ for evaluation

$O(B \times N)$. Next, the selection and flipping operation takes $O(p \times N)$ time per iteration. For B iterations, the total time complexity for label flipping is $O(B \times p \times N)$. Given that p is typically a small constant, this simplifies to $O(B \times N)$. Let the time complexity for training a model on N samples be $O(T(N))$. Over B bootstrap iterations, the total time complexity for model training is $O(B \times T(N))$. Therefore, the overall time complexity of the BOOTLFA algorithm is $O(B \times (N + T(N)))$, where $T(N)$ represents the time complexity of training the ML model in the size data set N. This time complexity signifies reasonable efficiency for practical implementations, especially in real-time monitoring systems that require rapid response.

B. BAGLFA: Bagging-Based Label Flipping Attack

Like BOOTLFA, we propose an adversarial technique called BAGLFA (Bagging-based Label Flipping Attack), which is also designed to assess the robustness of ML models against adversarial label noise.

Bagging is an ensemble learning technique in which multiple subsets of the original dataset are created by random sampling with replacement. Each subset is used to train a separate model, and the final prediction is obtained by aggregating the outputs of these models, typically through majority voting or averaging. This method effectively reduces variance and improves the stability of the model's predictions.

The BAGLFA method extends this approach by introducing label noise into each subset during sampling. Specifically, a fraction p of the labels in each subset is intentionally flipped, simulating a scenario where an adversary has corrupted the training data.

The methods presented in algorithms 1 and 2 are similar. The only difference is that **BOOTLFA** focuses on using bootstrap samples to capture the variability of the dataset while evaluating the model's resilience to label noise, typically emphasizing the distributional robustness; however, **BAGLFA** emphasizes generating diverse subsets and aggregating the results to reduce variance and improve model stability under label noise.

The Time Complexity of the BAGLFA method can be analyzed based on its key steps. First, the time complexity for generating a single subset is $O(N)$. Since B subsets are generated, the total time complexity for this step is $O(B \cdot N)$. Next, the time complexity of selecting and flipping the labels for a single subset is $O(p \cdot N)$. Given B subsets, the total time complexity for this step is $O(B \cdot p \cdot N)$. Since B models are trained, the total time complexity for model training is $O(B \cdot f(N))$, where $f(N)$ depends on the specific model being trained (e.g., linear models, decision trees, neural networks). Summing the complexities of these steps, the overall time complexity of the BAGLFA method is:

$$O(B \cdot (1 + p) \cdot N + B \cdot f(N))$$

This time complexity signifies reasonable efficiency for practical implementations, especially in real-time monitoring systems that require rapid response.

C. KCD: K-Means-Based Clustering Defence

To address the above attacks, this sub-section details the K-means-based Clustering Defence (KCD) countermeasure, BOOTKCD and BAGKCD. Based on the KCD approach, they are designed to counter BOOTLFA and BAGLFA.

KCD, presented in algorithm 3, is designed to mitigate label-flipping attacks. The method first applies the K-means algorithm to cluster training samples and predict initial labels. It then analyzes the Euclidean distances between each sample and its assigned cluster centroid to identify potential label flips. Samples with distances below the mean are considered likely to be correctly labelled within their cluster. Based on this analysis, the original labels are updated, and the classification model is retrained using the refined labels. KCD enhances model robustness by correcting poisoned data, leveraging the similarity between related samples to restore label integrity.

The Time complexity of the KCD can be analyzed based on its essential steps. First, the K-means algorithm has a time complexity of $O(I \cdot k \cdot N \cdot d)$, and for this specific application, where $k = 2$, it simplifies to $O(I \cdot N \cdot d)$. Next, calculating the Euclidean distance between each sample $\mathbf{x}_i$ and the centroid of its assigned cluster involves $O(N \cdot d)$ operations. The label flipping operation requires checking whether the distance for each sample is less than the mean distance μ, which results in $O(N)$ comparisons. Retraining the classification model on the updated dataset has a time complexity of $O(f(N, d))$,

Algorithm 3 KCD Algorithm

Require: D: Train data with N samples, k: Number of clusters
Ensure: Updated labels for the dataset D

```
Apply K-means algorithm to cluster D into k = 2 clusters.
Assign cluster labels to each sample x_i ∈ D.
for each sample x_i in the dataset D do
    Calculate the Euclidean distance between x_i and the centroid of its assigned cluster.
end for
Calculate the mean distance μ of all distances obtained.
for each sample x_i ∈ D do
    if Distance d(x_i) < μ then
        Flip the label of x_i to the predicted label
    end if
end for
Original labels in D ← new predicted labels
```

where $f(N,d)$ depends on the specific model being retrained. For instance, if a linear classifier is used, $f(N,d)$ could be $O(N \cdot d)$, but this complexity may vary based on the model used. Summing the complexities of these steps, the overall time complexity of the KCD algorithm becomes:

$$O(N \cdot d \cdot (I+1) + N + f(N,d))$$

In this complexity, I is the sample number and its maximum value is N. Therefore, the final time complexity is:

$$O(N^2 \cdot d + f(N,d))$$

IV. Experimental Evaluation

This section presents the simulation results of our proposed methods for both attacks (BOOTLFA and BAGLFA) and defence (BOOTKCD and BAGKCD). When KCD is used to defend against BOOTLFA, we refer to it as BOOTKCD, and when it is used against BAGLFA, we refer to it as BAGKCD.

A. Simulation Setup

The following sections outline the test metrics, dataset, features, classification parameters, and the defence algorithm used for comparison.

1) Test Metrics: We use the following metrics to comprehensively evaluate our attack and defense methods. All metrics are calculated based on the confusion matrix, including:

TP: The model correctly identifies a cyberattack.
TN: The model correctly identifies a normal instance.
FP: The model misclassifies a normal instance as an attack.
FN: The model misclassifies a cyberattack as normal.

- *Accuracy* refers to the overall effectiveness of the ML model in correctly classifying both normal and attack samples. Accuracy can calculated by equation (4):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4}$$

- *Precision* measures the model's accuracy in identifying true cyberattacks among all instances it has classified as attacks, as equation (5):

$$Precision = \frac{TP}{TP+FP} \tag{5}$$

- *Recall* measures the model's ability to identify all actual cyberattacks correctly. This can be expressed as equation (6):

$$Recall = \frac{TP}{TP+FN} \tag{6}$$

- *FNR* measures the proportion of actual cyberattacks incorrectly identified as normal by the model. It can be calculated by equation (7):

$$FNR = \frac{FN}{TP+FN} \tag{7}$$

- *F1-Score* is a metric that combines precision and recall to provide a single measure of a model's performance. It is beneficial when dealing with imbalanced datasets with a trade-off between precision and recall. It is defined as an equation (8):

$$F1-Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8}$$

- *AUC:* evaluates all possible thresholds to identify the best model for unbalanced datasets, preventing overfitting and indicating better performance by distinguishing between attacks and non-attacks. This can be calculated as equation (9):

$$AUC = \frac{1}{2}\left(\frac{TP}{TP+FP} + \frac{TN}{TN+FP}\right) \tag{9}$$

2) Datasets: In the experiments, we utilize the dataset from [15], which comprises information on the Temperature, Pressure, and Location of multiple fleets and vehicles.

3) Features: This study examines different sample attributes such as temperature, pressure, latitude, and longitude. These are outlined as follows, and each of themis collected by a sensor:

- *Temperature:* This sensor is crafted to measure a tyre's rapidly changing surface temperature, offering crucial insights for adjusting chassis settings and enhancing driver performance.
- *Pressure:* The tire-pressure monitoring system, referred to as Pressure, oversees the air pressure in the pneumatic tyres of vehicles. It furnishes drivers with instant updates on tyre pressure through a gauge, pictogram display, or a leading low-pressure warning indicator.
- *Latitude:* This sensor is designed to determine a vehicle's position using GPS, which includes longitude and latitude coordinates.
- *Longitude:* This sensor is designed to determine a vehicle's position using GPS, which includes longitude and latitude coordinates.

The dataset was chosen for its critical relevance to the operation and safety of connected vehicles, featuring essential data types like temperature sensors for monitoring engine heat,

tire-pressure sensors for optimal road contact, and GPS data for navigation and traffic management. It includes labeled instances of both normal operations and various attack scenarios such as spoofing and sensor tampering, providing a robust ground truth for precise training and validation of our LFD-IDS. This enables the system to effectively learn and distinguish between legitimate and malicious alterations. Additionally, the dataset mirrors the realistic cybersecurity challenges prevalent in ITS, encompassing typical data tampering and spoofing attacks, thus ensuring the dataset's applicability to current industry needs.

The attack instances within the dataset were generated using a methodology known as the "outlier data method", which is designed to simulate anomalous conditions within the sensor data indicative of potential cyber-physical attacks or malfunctions [15]. This method involves the following key steps:

- Data Collection: The authors initially collected a baseline dataset under normal operational conditions of the connected vehicles' sensor systems. This data encompassed various sensor outputs under a wide range of driving scenarios and conditions to ensure a comprehensive representation of typical operational data.
- Outlier Simulation: To simulate attack instances, the authors introduced outliers into this real-world data. These outliers represent hypothetical attack vectors, such as sudden, unrealistic changes in sensor readings or patterns that deviate significantly from established norms.
- Outlier Implementation: The implementation involved selectively manipulating sensor data points to reflect potential failures or attacks. For example, values from pressure sensors might be altered to exceed typical operational ranges dramatically, mimicking the effect of a sensor being compromised.
- Dataset Labelling: Each modified instance was labelled accordingly as an attack scenario, differentiating it from normal operational data. This labelling is essential for training the machine learning models to distinguish between normal operations and potential threats or failures.

4) Parameter Setting: To ensure the robustness of our proposed IDS and counteract potential biases, we employ a train-test split methodology, which is essential for an unbiased assessment of model performance. Our dataset is segmented into three parts: 60% for training, 20% for validation, and 20% for testing. This segmentation is randomly executed using various seeds to boost the generalizability of our findings. We focus our analysis on four key features pertinent to connected vehicle operations. To further validate the integrity of our evaluation, we repeat this random partitioning ten times for each algorithm, using different seeds to cultivate a variety of training, validation, and testing conditions. We then average the performance metrics across these iterations to ensure a reliable evaluation of each algorithm's effectiveness.

The experimental setup is standardized on a high-performance computing environment to handle the computational demands of our models. All methods are implemented using Python 3.7.6 and executed on an Intel Xeon CPU E5-2667, clocked at 3.3GHz, with access to 190 GB RAM and 32 CPU cores. The operating system used is Ubuntu Server 18.04, ensuring a stable and consistent platform for all tests.

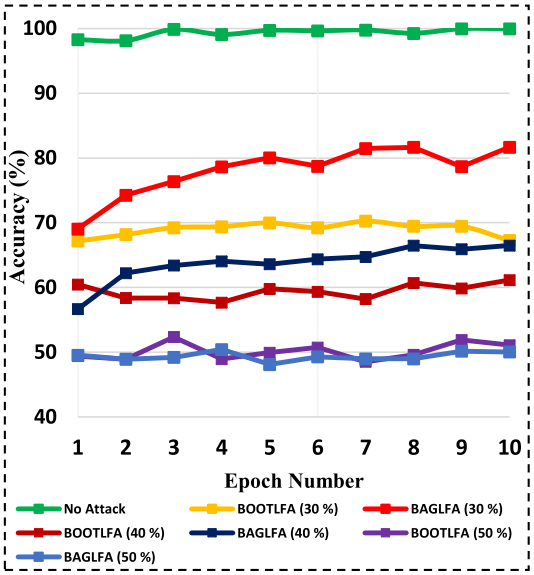

Fig. 3. Comparing the accuracy (in %) of attack methods for 30%, 40%, and 50% poison data in 10 epoch for training deep learning.

B. Experimental Results

We employed the Multilayer Perceptron (MLP) model to differentiate between benign and malicious data. The benign data were obtained from [15], while the malicious data were generated through outlier data methods. In this section, we assess the performance of our proposed attack algorithms, BOOTLFA and BAGLFA, on the initially trained classifier. Additionally, we validate our defence algorithms, BOOTKCD and BAGKCD, using the same dataset.

The training results presented encompass 10 epochs. Subsequent to this training phase, we assessed the final model using the test data to validate the robustness and accuracy of our approach. It's important to note that the results in Table I and Figs. 3 and 4 correspond to the final epoch (epoch number 10) of training. Meanwhile, Fig. 5 presents the results obtained on the test data.

1) Comparing Proposed Methods in Training: The results of the training step are presented in Fig. 3. This figure illustrates the training performance of our proposed deep learning models under varying intensities of label-flipping attacks, specifically BOOTLFA and BAGLFA, across ten epochs. The y-axis represents the accuracy percentage achieved by the models, and the x-axis denotes the epoch number from 1 to 10. The training accuracies are plotted for two types of attacks (BOOTLFA and BAGLFA) at three different intensities (30%, 40%, 50%) of label flipping.

Under conditions where no attack has occurred, the model consistently maintains an accuracy exceeding 98%. For the BOOTLFA at 30% label flipping, accuracy starts at 67.18%

TABLE I
COMPARING THE ML METRICS VALUES FOR 30%, 40%, AND 50% POISON DATA (%).(TRAINING DATA- FINAL EPOCH)

Algorithms	Precision			Recall			F1-Score		
	30%	40%	50%	30%	40%	50%	30%	40%	50%
BOOTLFA	68.36	65.64	66.97	61.13	54.59	57.67	52.21	45.57	48.66
BOOTKCD	99.98	99.79	99.89	99.98	99.95	99.96	100	99.96	99.98
BAGLFA	82.25	80.98	81.61	67.96	65.05	66.47	49.45	46.22	47.48
BAGKCD	99.98	99.93	99.96	99.95	99.93	99.94	99.98	99.86	99.93

and fluctuates mildly, peaking at 70.24% in the fifth epoch before declining slightly to 67.26% by the final epoch. In contrast, the BAGLFA at the same intensity shows a more consistent improvement, starting at 69.03% and reaching a high of 81.66% by the final epoch, indicating a more robust resilience against the attacks as the training progresses.

At 40% label flipping, both BOOTLFA and BAGLFA demonstrate lower initial accuracies. However, BAGLFA shows a steady improvement, starting at 56.66% and rising to 66.49%, suggesting better adaptability compared to BOOTLFA, which starts higher at 60.47% but ends only slightly improved at 61.14%. The 50% label flipping scenario presents the most challenging conditions, with both attacks starting below 50% accuracy. BOOTLFA shows a slight improvement over the epochs, ending at 51.09%, whereas BAGLFA remains fairly consistent but lower, ending at 50.04%. This visualization underscores the varying impacts of label-flipping intensities on training dynamics and highlights the comparative resilience of BAGLFA to higher degrees of adversarial manipulation compared to BOOTLFA.

a) Comparing based on Precision, Recall and F1-score: Table I presents a comprehensive comparison of ML metrics across different levels of data poisoning for 30%, 40%, and 50% poisoned data. These metrics include Precision, Recall, and F1-Score for various algorithmic approaches: No Attack, BOOTLFA, BOOTKCD, BAGKCD, and BAGLFA.

No attack scenario serves as a baseline, with consistently high scores across all metrics approaching near perfection (above 99.93%). This indicates optimal model performance without adversarial interference, validating the efficacy of the training process under controlled conditions. Notably, the BOOTLFA significantly deteriorates model performance. Precision and Recall decrease with increasing poison data, showing a drop in Precision from 68.36% at 30% poisoning to 66.97% at 50%, and an even starker drop in Recall from 61.13% at 30% to 57.67% at 50%. This illustrates the disruptive impact of BOOTLFA on the model's ability to correctly identify and classify data points under adversarial attack conditions. The F1-Score also shows a decline, suggesting that BOOTLFA effectively compromises both the accuracy and the completeness of the predictions. Applying the BOOTKCD restores the metrics to near-baseline levels. Precision and Recall are almost fully recovered, and the F1-Score reaches 100% at 30% poisoning, demonstrating the effectiveness of KCD in mitigating the adverse effects of BOOTLFA.

The BOOTKCD and BAGKCD defensive methodologies exhibited robustness comparable to the No Attack scenario,

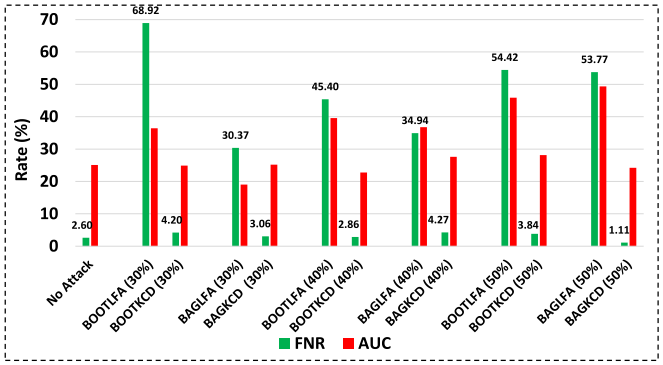

Fig. 4. Comparing proposed method based on FNR and AUC (training data-final epoch).

with only minor deviations in Precision and Recall observed across different levels of data poisoning. This robustness underscores the efficacy of the proposed defensive strategies in maintaining system integrity under adversarial conditions. However, a comparative analysis with scenarios where no attacks were conducted reveals that despite the defenses' effectiveness, the attacks have succeeded in slightly diminishing the accuracy and in elevating the FPR. This indicates that while the defensive methods significantly mitigate the impact of the attacks, they do not completely negate the adversaries' ability to affect system performance. Nevertheless, the resilience demonstrated by the defensive strategies highlights their potential in safeguarding systems against sophisticated cyber threats, thus providing a reliable shield in maintaining operational accuracy and security.

b) Comparing methods based on FNR and AUC: Fig 4 presents a detailed analysis of the FNR and AUC metrics across various scenarios, including no attack, different intensities of BOOTLFA and BAGLFA (30%, 40%, 50%), and their respective defences BOOTKCD and BAGKCD.

The no-attack scenario shows an FNR of 2.60% and an AUC of 25.11, establishing the performance benchmark in a secure environment. BOOTLFA significantly increases the FNR, indicating a higher rate of missed cyberattacks as the intensity of perturbation increases—rising to 54.42% at 50%. Conversely, AUC generally increases with the intensity of the attack, suggesting a decrease in the overall model's ability to distinguish between classes effectively. BOOTKCD effectively mitigates the impact of BOOTLFA across all perturbation levels, substantially lowering the FNR close to the no-attack scenario and maintaining a stable AUC. This highlights BOOTKCD's robustness in correcting misclassifications

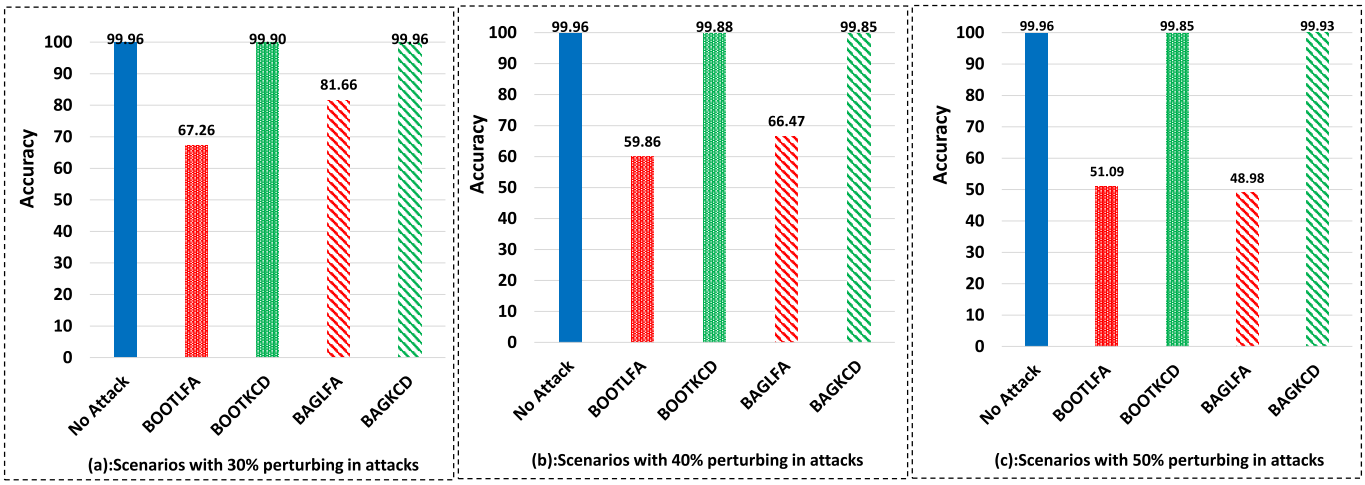

Fig. 5. The comparison of different models trained with 30%, 40%, and 50% poison data- Accuracy on Test Data.

caused by the attack. The high FNR at the intensity of 30% indicates a significant degradation of the model's ability to detect actual attacks. This can be attributed primarily to the nature of the BOOTLFA attack, which, through its methodology of bootstrapping and label flipping, creates a specific noise pattern in the training data. This noise may have caused the model to misclassify actual attacks as benign activities more frequently than anticipated. The label flipping inherent in BOOTLFA introduces systematic errors into the training dataset, effectively misleading the learning algorithm during the model training phase. Such distortions in the training data likely contribute to the model's increased difficulty in correctly identifying attack instances, resulting in a higher FNR. This result also underscores potential limitations in the model's detection capabilities under adversarial conditions.

Like BOOTLFA, BAGLFA considerably impacts the FNR, indicating a sizable number of attacks going undetected, especially at higher perturbations (reaching up to 53.77% at 50%). The AUC for BAGLFA shows a decline, particularly noticeable as the perturbation increases, reflecting a compromised ability to separate the classes under attack conditions. In contrast, BAGKCD significantly reduces the FNR, bringing it close to the no-attack scenario, and helps maintain an acceptable level of AUC. This defence strategy proves effective even at higher attack levels, underscoring its capability to restore model reliability.

2) Comparing Proposed Methods in Test: Fig. 5 illustrates the testing accuracy of our ML models under no attack conditions and various adversarial attack scenarios at different levels of data perturbation (30%, 40%, and 50%). These scenarios include BOOTLFA, BAGLFA, and their respective BOOTKCD and BAGKCD.

- **No Attack Performance:** As a control, the models maintain a consistently high accuracy of 99.96% across all perturbation levels, serving as a benchmark for evaluating the impact of adversarial conditions.
- **Impact of BOOTLFA:** This attack significantly reduces model accuracy, particularly as the level of perturbation increases—from 67.26% at 30% perturbation to 51.09% at 50%. This demonstrates the attack's effectiveness in degrading model performance.
- **Efficacy of BOOTKCD:** The K-means-based defence strategy (BOOTKCD) effectively neutralizes the impact of BOOTLFA, restoring accuracy to near perfect levels (above 99.85% across all perturbations), illustrating its robust defensive capability.
- **Impact of BAGLFA:** While less detrimental than BOOTLFA, BAGLFA still notably decreases accuracy, with a gradual decline as perturbation increases, reaching as low as 48.98% at 50% perturbation. This indicates the model's significant vulnerability to this type of attack at higher perturbation levels.
- **Efficacy of BAGKCD:** Similarly to BOOTKCD, BAGKCD substantially mitigates the effects of BAGLFA, restoring accuracies to nearly the same levels as the no attack scenario (99.93% at 50% perturbation), highlighting the effectiveness of the defence mechanism.

3) Comparing Proposed Methods With State-of-the-Art: As depicted in Table II, our work stands out distinctly in the landscape of IDS for Connected Vehicles. The comparison highlights several unique aspects of our approach:

- Unlike existing approaches listed in the table, our system is specifically designed to address model poisoning and data poisoning attacks through adversarial training and clustering-based defenses.
- Our work is the only one in the comparison that utilizes real-world data from connected vehicles, sourced from [15]. Other studies often repurpose datasets from general computer networks or simulations that may not accurately represent the unique challenges faced by connected vehicle environments.
- The time complexity of our proposed LFD-IDS is $O(N^2)$, which is comparable to other state-of-the-art methods while providing real-time detection capabilities, a critical requirement for IDS in vehicular networks.
- The LFD-IDS system is designed to operate in real-time, a significant advancement over several other listed methods. This capability allows for immediate identification

TABLE II
COMPARISON BETWEEN DIFFERENT IDS IN CONNECTED VEHICLES AND SIMILAR AREAS

Ref.	Application	Adversarial attack Type	Defense	Dataset	Time complexity	RT
[16]	VANET safety	–	–	5RoutingMetrics	–	–
[17]	Anomaly Detection	–	–	AIS dataset	–	–
[18]	IDS	–	–	NSL-KDD	–	–
[19]	Internet of Vehicles	–	–	CAN and CICIDS2017	$O(N^2)$	–
[20]	Internet of Vehicles	–	–	CIDS2017 and Car-Hacking	O(N)	–
[21]	IoT Network	–	–	TON IoT	$O(N^2 logN)$	✓
[22]	Malware	Model Poisoning	Adversarial Training	CICIDS2018	$O(N)$	–
LFD-IDS	**IDS in Connected Vehicle**	Data Poisoning	Clustering based	**Real world data from [15]**	$O(N^2)$	✓

and response to threats, an essential feature for maintaining the safety and reliability of connected vehicle systems.

To our knowledge, no existing studies integrate real connected vehicle data, specific adversarial attack scenarios, and real-time processing capabilities into a single IDS framework as effectively as our LFD-IDS system does. This unique combination not only addresses conventional IDS challenges but also specifically caters to the complex dynamics of connected vehicle environments. The use of real-world data enhances the practical applicability and robustness of our system, establishing a new benchmark in the field and ensuring that our results are directly relevant to actual operational conditions of connected vehicles.

V. DISCUSSION

This study presented an extensive evaluation of the effects of label-flipping attacks (BOOTLFA and BAGLFA) and the corresponding K-means-based Clustering Defense strategies (BOOTKCD and BAGKCD) on the performance of IDS in CVs. The results underscore significant vulnerabilities introduced by adversarial attacks and highlight the robustness of proposed Defences in mitigating these threats.

The BOOTLFA and BAGLFA attacks substantially degraded system performance. The increasing perturbation levels (30%, 40%, 50%) indicate a higher likelihood of the IDS failing to detect actual attacks, thereby compromising the security of the CV system. Similarly, BAGLFA's impact, although slightly less severe than BOOTLFA, still presented a significant challenge, especially at higher perturbation levels, highlighting the susceptibility of deep learning-based IDS to sophisticated adversarial manipulations.

Implementing BOOTKCD and BAGKCD, effectively countered the adversarial impacts. This resilience suggests that leveraging clustering algorithms to detect and correct label anomalies significantly enhances the robustness of IDS against adversarial attacks. The success of these strategies is attributed to their ability to discern and rectify mislabeled data, thus preserving the integrity of the learning process.

As a result, applying BOOTKCD and BAGKCD introduces a robustness-enhancing effect on the model. These defense mechanisms are designed to correct label flips and other forms of data corruption. In doing so, they inadvertently aid the model in developing a resilience not just against adversarial manipulations but also against any noise inherently present in the dataset.

This noise could include mislabeled instances or subtle non-adversarial anomalies, which are not explicitly targeted under the no attack condition. Also, both BOOTKCD and BAGKCD employ techniques that can detect and mitigate outliers or anomalous data points effectively as part of their defense strategy. This capability might lead to a cleaner and more reliable training dataset compared to the original dataset used in the no-attack scenario, where such anomalies are not corrected.

Finally, the process of defending against label-flipping attacks often involves identifying and correcting mislabeled data points. This correction process not only defends against adversarial actions but can also rectify pre-existing label errors in the training dataset. Consequently, this 'cleansing effect' might result in a model that performs better than one trained on the uncorrected dataset.

The findings from this study underscore the urgent need for robust defensive strategies within CV ecosystems. As vehicles become increasingly connected and rely on ML for critical functions such as intrusion detection, such strategies become paramount. The demonstrated efficacy of KCD in this study suggests that integrating such methodologies could be crucial in developing next-generation IDS that are both resilient to and capable of evolving in response to adversarial threats.

VI. CONCLUSION AND FUTURE DIRECTIONS

This paper presents significant contributions to the field of adversarial machine learning to detect cyber attacks on Connected Vehicle (CV) systems. The authors proposed two novel adversarial attack methods -BOOTLFA and BAGLFA - that leverage Bootstrapping and Bagging as a label-flipping attack. They also introduced comprehensive defence mechanisms, BOOTKCD and BAsGKCD that leverage K-means-based Clustering Defence (KCD) to mitigate attacks and maintain model accuracy. Experimental evaluations on datasets validated the proposed attacks and demonstrated the robustness of the defence mechanisms in mitigating the impact of attacks and maintaining model accuracy. For future research, the authors suggest focusing on optimizing adversarial defence mechanisms. This can be achieved by exploring novel techniques for hyperparameter tuning, adjusting model architectures, and fine-tuning training processes to enhance the generalizability and adaptability of defences across diverse datasets and attack scenarios.

REFERENCES

[1] A. Lamssaggad, N. Benamar, A. S. Hafid, and M. Msahli, "A survey on the current security landscape of intelligent transportation systems," *IEEE Access*, vol. 9, pp. 9180–9208, 2021.

[2] H. Liu et al., "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6073–6084, Jun. 2021.

[3] Y. Djenouri, A. Belhadi, D. Djenouri, G. Srivastava, and J. C. Lin, "A secure intelligent system for Internet of Vehicles: Case study on traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13218–13227, Nov. 2023.

[4] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *J. Inf. Secur. Appl.*, vol. 50, Feb. 2020, Art. no. 102419.

[5] F. Wang, X. Wang, and X. Ban, "Data poisoning attacks in intelligent transportation systems: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 165, Aug. 2024, Art. no. 104750.

[6] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Mar. 2016, pp. 372–387.

[7] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial examples for malware detection," in *Proc. Eur. Symp. Res. Comput. Secur.*, Cham, Switzerland: Springer, 2017, pp. 62–79.

[8] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2, 2014, pp. 2672–2680.

[9] P. Rathore, A. Basak, S. H. Nistala, and V. Runkana, "Untargeted, targeted and universal adversarial attacks and defenses on time series," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[10] A. Nitin Bhagoji, D. Cullina, C. Sitawarin, and P. Mittal, "Dimensionality reduction as a defense against evasion attacks on machine learning classifiers," 2017, *arXiv:1704.02654*.

[11] A. Paudice, L. Muñoz-González, and E. C. Lupu, "Label sanitization against label flipping poisoning attacks," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Cham, Switzerland: Springer, 2018, pp. 5–15.

[12] N. Natarajan, I. S. Dhillon, P. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, Dec. 2013, pp. 1196–1204.

[13] H. Xiao, B. Biggio, B. Nelson, H. Xiao, C. Eckert, and F. Roli, "Support vector machines under adversarial label contamination," *Neurocomputing*, vol. 160, pp. 53–62, Jul. 2015.

[14] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," 2018, *arXiv:1803.09050*.

[15] Z. Pooranian, M. Shojafar, P. Asef, M. Robinson, H. Lees, and M. Longden, "RCA-IDS: A novel real-time cloud-based adversarial IDS for connected vehicles," in *Proc. IEEE 22nd Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Nov. 2023, pp. 495–503.

[16] S. Amaouche, AzidineGuezzaz, S. Benkirane, and MouradeAzrour, "IDS-XGbFS: A smart intrusion detection system using XGboostwith recent feature selection for VANET safety," *Cluster Comput.*, vol. 27, no. 3, pp. 3521–3535, Jun. 2024.

[17] J. Hu et al., "Intelligent anomaly detection of trajectories for IoT empowered maritime transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2382–2391, Feb. 2023.

[18] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Netw.*, vol. 90, Jul. 2019, Art. no. 101842.

[19] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 616–632, Jan. 2022.

[20] L. Yang and A. Shami, "A transfer learning and optimized CNN based intrusion detection system for Internet of Vehicles," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 2774–2779.

[21] Sk. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam, "Dependable intrusion detection system for IoT: A deep transfer learning based approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 1006–1017, Jan. 2023.

[22] X. Yuan, S. Han, W. Huang, H. Ye, X. Kong, and F. Zhang, "A simple framework to enhance the adversarial robustness of deep learning-based intrusion detection system," *Comput. Secur.*, vol. 137, Feb. 2024, Art. no. 103644.

Zahra Pooranian (Senior Member, IEEE) received the Ph.D. degree in computer science from the Sapienza University of Rome, Italy, in February 2017. She was a Post-Doctoral Fellow with the 5G Innovation Centre (5GIC) and the 6G Innovation Centre (6GIC), Institute for Communication Systems (ICS), University of Surrey, Guildford, U.K. She is currently a Lecturer with the University of Reading, U.K.

Rahim Taheri (Senior Member, IEEE) is a Senior Lecturer at the University of Portsmouth with a PhD in Computer Networks and over a decade of experience in academia. His research spans secure and privacy-preserving AI, federated learning, adversarial machine learning, and Large Language Models. He has held research roles at King's College London and the University of Padua, working with labs such as KCLIP and SPRITZ. He is especially interested in developing defenses against data poisoning and adversarial threats in IoT and distributed systems. He has mentored PhD students, published in top journals and conferences, and is an active member of the IEEE and ACM. His work is dedicated to exploring ethical, robust AI solutions for security challenges in modern digital infrastructures.

Fabio Martinelli (Senior Member, IEEE) received the M.Sc. degree from the University of Pisa, Pisa, Italy, in 1994, and the Ph.D. degree from the University of Siena, Siena, Italy, in 1999. He is currently the Research Director of the Consiglio Nazionale delle Ricerche (CNR), Pisa, where he leads the Cyber Security Project area. He is involved in several steering committees of international WGs/conferences and workshops. He manages research and development projects on information and communication security. His main research interests include security and privacy in distributed and mobile systems and foundations of security and trust.