IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 February 2024, accepted 30 March 2024, date of publication 9 April 2024, date of current version 17 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3386631

RESEARCH ARTICLE

FL-IDS: Federated Learning-Based Intrusion Detection System Using Edge Devices for Transportation IoT

MANSI H. BHAVSAR[1], YOHANNES B. BEKELE[1], (Member, IEEE), KAUSHIK ROY[2], JOHN C. KELLY[1], AND DANIEL LIMBRICK[1], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, North Carolina Agricultural and Technical State University, Greensboro, NC 27411, USA
[2]Computer Science Department, North Carolina Agricultural and Technical State University, Greensboro, NC 27411, USA

Corresponding author: Mansi H. Bhavsar (mhbhavsar@aggies.ncat.edu)

This work was supported in part by Cisco Systems, Inc.

ABSTRACT A federated learning-based intrusion detection system (FL-IDS) is introduced to enhance the security of vehicular networks in the context of IoT edge device implementations. The FL-IDS system protects data privacy by using local learning, in which devices share only model updates with an aggregation server. The server then generates an enhanced detection model. The FL-IDS system also incorporates a detection model (LR-IDS, PCC-CNN) based on machine learning (ML) and deep learning (DL) classifiers, namely logistic regression (LR) and convolution neural networks (CNN), to prevent attacks in transportation IoT environments. The proposed FL-IDS model uses embedded devices (such as Raspberry Pi for the client and Jetson Xavier for the server model). The real-time performance of the proposed IDS was evaluated using two different datasets, NSL-KDD and Car-Hacking. We deployed our IDS model on different architectures, testbed 1 (with 2 clients) and testbed 2 (with 4 clients). The model evaluation has been evaluated based on the accuracy, and loss parameters. The results show that the FL-IDS system outperforms traditional centralized learning with machine learning and deep learning approaches regarding accuracy (achieved overall 94% and 99%) and loss (achieved overall 0.28 and 0.009). These findings contribute to transportation IoT systems security by proposing a robust framework for enhancing the security and privacy of CAVs against cyber threats.

INDEX TERMS Federated learning, deep learning, machine learning, transportation systems, CAV, IDS, edge computing.

I. INTRODUCTION

The development of the Internet of Things (IoT) has significantly increased over the last few years, resulting in the rapid development of wireless transmission and processing. A series of edge devices, such as smartphones, smart cars, and smart applications, have emerged on IoT networks (see Figure 1). One of the most demanding applications is transportation, specifically in Connected and Autonomous Vehicles (CAV). CAVs are designed to make traffic safer while lowering risks and accidents [1]. Moreover, transportation IoT devices are mainly concentrated on self-driving cars. According to the Boston Consulting Group Report, the market size of autonomous driving will reach 42 billion dollars by 2025 [2]. But, the major challenge in these devices is to make the vehicles reliable and secure [3]. The CAVs communicate via internal and external communication networks to achieve their goals, which include reducing human errors on the road, traffic accidents, and the number of fatalities and utilizing the resources currently available to achieve full autonomy. The connected vehicles are equipped with a series of electronic control units (ECU), sensors, and internal and external communication systems [4]. Consequently, the network systems have several security vulnerabilities due to increased network complexity. Therefore, developing an accurate Intrusion Detection System (IDS) is

The associate editor coordinating the review of this manuscript and approving it for publication was Tyson Brooks.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

constantly needed to efficiently mitigate different types of attacks [5], [6]. Along with guaranteeing security and privacy against unauthorized access, transportation IoT networks are cognitively demanding, time-efficient, and constantly require computing resources, which is another significant barrier.

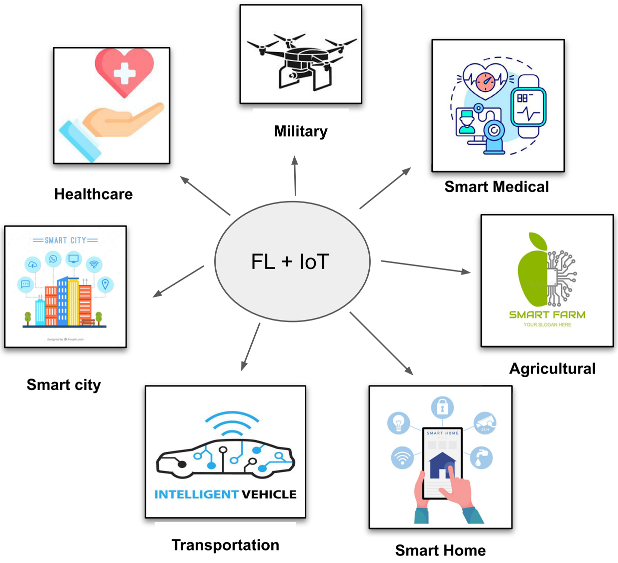

FIGURE 1. Applications of FL-IoT.

Due to the increasing usage of digital technology and awareness of individuals, people started to think about personal data security even more [7]. According to the National Science Foundation (NSF), cyber-physical systems (CPS) are the foundation of all infrastructure in smart cities because "cyber-physical systems integrate sensing, computation, control, and networks into physical objects and infrastructure, linking them to the Internet and into to each other." Connected vehicles are one of the areas of cyber-physical systems that have grown rapidly in recent years [8], [9]. Connected and autonomous vehicles (CAVs) can collect and process their surroundings by extracting information from sensors and sharing it with other CAVs via wireless networks, which has raised extensive research [10].

CAV uses Vehicular Ad-hoc Networks (VANETs) with the Internet of Things (IoT) communications and architecture capabilities. VANETs provide several communication schemes see Figure 2 known as Vehicle-to-everything (V2X) communications, Vehicle-to-Infrastructure (V2I), Vehicle-to-Roadside Units (V2R), Vehicle-to-Cloud (V2C), Vehicle-to-Vehicle (V2V), and Vehicle-to-Device (V2D) communications [11]. V2X communications can be divided into two main categories: In-vehicle and Inter-vehicle networks. The in-vehicle network is a Vehicle-to-Sensors (V2S) communication schema performed by collecting embedded sensors located in the vehicle and mainly interacting via CAN-Bus, Ethernet, or WiFi standards [12]. Inter-vehicle networks cover communication between the vehicle and the different components of the transportation system. V2X communications are mainly used for external communication of CAVs. With these networks, drivers can exchange information such as control data, emergency messages about braking, accidents, and emergencies [13] and that leads to an increase in cyber attacks such as Denial of Service attacks, spoofing, Sybil attacks, black holes, and many more [14].

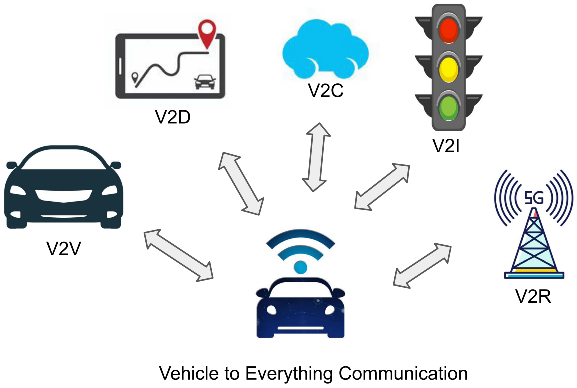

FIGURE 2. Vehicle to everything (V2X) Communication.

Federated Learning: FL is a distributed learning approach to machine learning that enables collaborative learning on large datasets without sharing the raw data with a central server or other participants [15]. Instead, the training process occurs locally on individual devices or edge nodes, with only the model updates shared with a central server or coordinator. FL is particularly valuable when data privacy and security are critical, ensuring that sensitive user data remains decentralized and protected [16].

FL has gained significant attention in applications involving distributed data sources, including Internet of Things (IoT) networks, healthcare systems, mobile devices, and, specifically, in the context of autonomous vehicles [17]. In vehicular environments, timely decision-making based on sensor data is essential. Vehicles are equipped with various sensors such as cameras, radar, Lidar, and GPS, which gather valuable information about the vehicle's surroundings and its status [18]. Real-time processing and analysis of this sensor data enable vehicles to make informed decisions to enhance safety, efficiency, and the overall driving experience. Advanced driver assistance systems (ADAS) rely on sensor data for various functionalities. For instance, ADAS heavily relies on sensor data to detect and react to potential hazards. This includes actions like applying emergency braking when an obstacle is detected. In the case of autonomous vehicles, sensor data plays a critical role in navigation, route planning, and collision avoidance. These vehicles rely on the continuous flow of sensor data to operate safely and effectively. Furthermore, timely decision-making in vehicular environments extends beyond individual vehicles. It encompasses vehicle-to-everything (V2X) communication, where vehicles exchange information with other vehicles, nearby objects, the cloud/network, and the surrounding infrastructure. This facilitates collaborative decision-making, such as coordinating lane changes, merging, and optimizing network traffic flow. V2X communication enables vehicles to share relevant data and cooperate, enhancing overall traffic efficiency and safety [19].

The FL framework for IoT devices, as depicted in Figure 3, highlights the decentralized nature of FL, which aligns well with the distributed and sensor-rich nature of autonomous vehicles. By leveraging FL, autonomous vehicles can perform local model training on their respective sensor data while preserving data privacy and security. This approach enables vehicles to collectively learn from diverse datasets without compromising individual privacy or sharing sensitive information. The ability to process and interpret sensor data in vehicular environments is essential for ensuring safe and efficient transportation. It enables vehicles to respond promptly to changing conditions, avoid potential hazards, and facilitate intelligent and connected transportation systems. However, two critical considerations arise in vehicular environments related to decision-making and data processing [20]:

1) Bandwidth and Delay: Transmitting data from distributed agents, such as vehicles, to a centralized cloud for processing and decision-making requires high bandwidth and incurs significant delays. Vehicular environments generate large volumes of data from various sensors, and real-time data transmission to the cloud can be challenging. Limited bandwidth and potential network congestion can impede the timely transfer of data, affecting the efficiency of decision-making processes.
2) Local versus Global Information: Distributed agents, like vehicles, can process data based on local information and knowledge. They can make decisions based on immediate surroundings and local scenarios. However, relying solely on local information may not provide a comprehensive view of the entire system. Global information from a centralized cloud or shared among vehicles encompasses a broader perspective and can lead to more accurate and comprehensive decision-making. Striking a balance between utilizing local and global information is crucial to ensuring optimal decision-making in vehicular environments.

Addressing these considerations requires innovative approaches that can tackle the challenges associated with bandwidth, delay, and the balance between local and global information. To address these challenges, Google [21] proposed Federated Learning (FL), which allows multiple agents to collaboratively train a machine learning (ML) or deep neural network (DNN) model, as shown in Figure 3. The process involves the following steps:

1) **Step 1** Initialization: The central server initializes the ML or DNN model.
2) **Step 2** Distribution of Model: The central server distributes the model parameters to multiple clients or workers.
3) **Step 3** Local Training: Each client or worker performs local training using their respective data and the distributed model.
4) **Step 4** Model Updates: After local training, each client or worker returns their model updates to the central server [22].
5) **Step 5** Aggregation: The central server aggregates the model updates received from the clients to create a global model [23].
6) **Step 6** Iterative Process: Steps 2 to 5 are repeated iteratively until the model converges or reaches the desired performance level.

By adopting the FL framework, vehicular environments can leverage the collaborative training of models while mitigating privacy risks. This approach enables distributed agents to contribute their local knowledge and data for model training without compromising data privacy and security.

The contributions of this paper, therefore, are as follows:

1) Our proposed model introduces a resilient federated learning architecture.
2) The inference framework proposed is versatile, independent of specific network choices, and modular, making it applicable to various vehicular networks and adaptable to diverse IoT environments.
3) To thoroughly validate the developed intrusion detection models, comprehensive experiments were conducted using two benchmarking datasets, evaluating the performance and runtime efficiency.

In contrast to existing intrusion detection methods, our framework offers the following advantages:

- **Easy development:** Our architecture is naturally well suited for iterative development and testing by leveraging the existing state-of-the-art networks.
- **Privacy:** Our framework allows the collaborative training of models on distributed data without the need for data sharing, preserving the privacy of individual clients' sensitive information.
- **Security:** as the training process occurs locally on client devices, minimizing the exposure of sensitive data, hence providing secure communication.
- **low overhead:** Our model minimizes the communication and computational burden on individual clients, reducing network traffic and resource usage.

Therefore, the proposed Federated Learning (FL) framework exhibits flexibility, scalability, and compatibility with multiple networks, enhancing detection accuracy while maintaining minimal runtime overhead.

The remainder of the paper is organized as follows. Section III presents the proposed FL-IDS with machine learning (LR-IDS) and deep learning (CNN-IDS). Section IV demonstrates the experimental results for implementing different Testbed models, their detection performance, and analysis. Finally, Section V provides the concluding remarks.

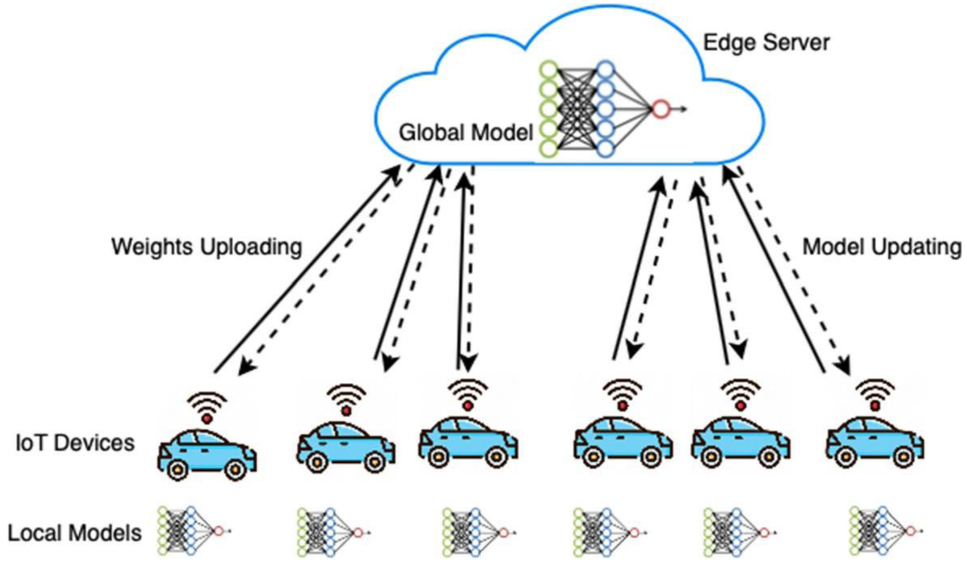

FIGURE 3. Federated learning architecture.

II. RELATED WORK

This section will briefly introduce existing intrusion detection methods and the current FL scenario in CAV.

Numerous intrusion detection systems (IDS) studies have acknowledged the benefits of incorporating artificial intelligence techniques, particularly machine learning algorithms, in external and in-vehicle networks [24]. Given the nature of Vehicle-to-Everything (V2X) networks in the CAV, unreliable infrastructure support has significantly increased security risks [25]. The lack of centralized supervision and the inherent characteristics of mobile ad hoc networks further exacerbate these risks [26]. Additionally, the growing complexity of intelligent connected vehicle systems and the proliferation of external interfaces make vehicle networks more susceptible to cyberattacks [27]. Currently, security threats in the IoV can be categorized as follows: 1) vehicle security threats, 2) communication security threats, 3) cloud platform security threats, and 4) mobile smart terminal security threats.

To explore potential vulnerabilities in the CAV, researchers have conducted simulations of various attack scenarios in the past. Based on the surveys [28], [29], [30], [31], Table 1 provides an overview of the most common types of attacks and their corresponding descriptions.

To overcome such security threats, it is required to have a mechanism that can detect this anomalous behavior ahead of time in real-time to secure the autonomous vehicle. To do so, nowadays, IDS has shown promising results [32], [33], [34], [35]. He et al. [36] proposed a cybersecurity framework for Connected Autonomous Vehicles (CAVs) that includes the creation of a new communication dataset called CAV-KDD. This dataset, which considers the existing KDD dataset as a benchmark, classifies vulnerabilities in CAV systems and addresses potential attacks. The framework also incorporates a UML-based CAV framework inspired by the UK CAV cybersecurity framework to analyze attack threats and provide solutions for secure CAV systems and data transfer. In a separate study [37], the authors successfully utilized the IoT-23 dataset and employed various machine learning algorithms, including Random Forest (RF), Naïve Bayes (NB), Multi-layer Perception (MLP), Support Vector Machine (SVM), and AdaBoost (ADA). The evaluation revealed that the Random Forest algorithm achieved the highest accuracy of 99.5% compared to the other algorithms.

TABLE 1. CAV security attacks.

Attack Category	Description
Dos, DDoS	Occupying and controlling nodes or network resources by inundating them with a large volume of requests or irrelevant data.
Brute force	Illegally obtain sensitive information through unauthorized means.
Integrity	Unauthorized manipulation or alteration of data or systems.
Sniffing	Extracting confidential data from the vehicle system by systematically scanning the system's ports and identifying potential vulnerabilities.
Fuzzy	Attackers disrupt the normal functioning of the vehicle by injecting random information, causing it to enter an unexpected state or malfunction.
Malware	Attackers exploit vulnerabilities in the communication interface to infiltrate the system, unleashing attacks in the form of worms, viruses, spyware, and similar malicious entities.
Web	Gain unauthorized access and manipulate the system.

Aloqaily et al. [38] developed a dataset by capturing real-time data from a simulated LAN US Air Force LAN over nine weeks, encompassing multiple attack types. This highly trained dataset was captured based on the KDD99 cup.

The datasets generated by these two articles effectively overcome the limitations of attack types, allowing for the injection of various attack types using these frameworks. Thakkar and Lohiya [39] conducted a comprehensive survey focusing on machine learning and deep learning methods employed in intrusion detection systems for the Internet of Things (IoT). In one of our previous papers [40], we focused on the importance of IDS in CAVs. Using benchmarking datasets, we build the IDS using 5 different Machine learning (ML) techniques. The comparable analysis used performance evaluation metrics such as Precision, recall, F1-score, and False alarm rate. The results show a good detection accuracy of 99% in Binary and multiclass classification. However, we also state the time complexity for detecting such threats in real time for a heterogeneous network. Thapa et al. [41] introduced a novel deep learning model for a Network Intrusion Detection System (NIDS) by comparing various machine and deep learning models. The paper emphasizes the need for a dynamic security system to identify unknown attacks on the network, highlighting the limitations of currently available static signal-based network intrusion systems. Based on the previous results, the paper proposes an efficient model combining ML and DL techniques to achieve high performance. Furthermore, the proposed model was benchmarked against the CIC-IDS2017 dataset, allowing for comparison with other models. The selection of models was based on the desired performance metrics and cost functions, considering factors such as training time. Comparisons were made with previous work, and it was observed that using CNN + embedding and LSTM + embedding improved accuracy.

A similar study [42] presents a highly effective approach to address attacks on the CAN bus protocol by developing an Instruction Detection System using a Deep Neural Network (DNN). By leveraging neural networks, which can identify simple patterns within the dataset, they aim to mitigate the impact of these attacks. Furthermore, they comprehensively analyze the results obtained from our proposed solutions. Their results show a higher efficiency of 98%. However, their research also provides the future direction for generating lightweight IDS with less time complexity. In 2020, Vu et al. [43] introduced a novel Deep Transfer Learning (DTL) approach that enabled learning from multiple IoT device data, even when not all data were labeled. The approach relied on two Autoencoders (AEs), where AE1 was trained in a supervised mode using tagged information from source datasets. At the same time, AE2 was introduced unsupervised on target datasets without any label information. Samy et al. [44] proposed a robust and distributed attack detection framework that achieved a high detection rate for various IoT cyber-attacks utilizing Deep Learning (DL). The attack detector was implemented on fog nodes due to their proximity to edge devices, significant computational capacity, and distributed nature. In another study by Roopak et al. [45], a multiple optimization-oriented Feature Selection (FS) technique was developed for detecting Distributed Denial of Service (DDoS) attacks in an IoT network. The FS technique effectively reduced the dimensionality of data and enhanced the performance of Intrusion Detection Systems (IDS) in detecting DDoS attacks.

In our previous study [46], the anomaly-based IDS was been proposed using the PCC-CNN model. The model was implemented with three benchmarking network traffic datasets. The model has outperformed the traditional ML methods by achieving 99% classification accuracy. The author also states the applicability of the proposed PCC-CNN model in any IoT device. To overcome the limitation of the dynamic architecture of IDS in our previous deep learning-based solution, this paper focused on the federated learning-based solution to provide better safety and security for autonomous vehicles. Moreover, The review study presented in [47] and [48] highlighted the future research directions focused on designing and implementing Intrusion Detection Systems (IDS) for Mobile Ad hoc Networks (MANETs) & Vehicular Ad hoc Networks (VANETs) while emphasizing the importance of preserving the security aspects of the Internet of Things (IoT) devices.

FL has gained significant attention as a viable approach for intrusion detection systems (IDS) in autonomous vehicles, providing enhanced privacy and scalability. Plenty of surveys [49], [49], [50] have provided insight into the current usage of Federated learning in the field of the transportation system. A novel detection mechanism has been developed by [51] that utilizes the capabilities of deep auto-encoder methods to identify attacks based solely on the benign network traffic pattern. The proposed system demonstrates a high detection rate while effectively reducing the false positive rate and detection delay through comprehensive experiments conducted on a recent network traffic dataset. Another study [52] proposed a deep CNN-LSTM architecture for CAV threat intelligence; Their model achieved better results by tuning the hyper-parameters on the CAV-KDD dataset. However, the author certifies the challenges, such as poor model generalization due to an imbalanced dataset. Taslimasa et al. [53] introduced a practical and privacy-preserving distributed solution called ImageFed & FedCNN. ImageFed is explicitly designed for the CAN bus environment. FedCNN uses the Deep neural network; they compared the results of the two models and stated that ImageFed provides better robustness. However, their model is only suitable for IID datasets.

Based on the previous reviews, Our proposed model's applicability to both autonomous vehicles and other IoT devices showcases its versatility and adaptability, making it dynamic and robust across different heterogeneous network domains. Additionally, the decentralized learning approach ensures the privacy of vehicle owners by eliminating the risk of sensitive information leakage, further enhancing its security measures.

III. METHODOLOGY

In this section, we propose an FL-based IDS architecture for intrusion detection. The Proposed FL-IDS architecture as shown in Fig. 4 is a motivation from [54]. The proposed framework has two methods for detecting anomalies via LR-IDS and CNN-IDS techniques. We utilized the Flower framework to implement the FL environment in our study. Flower [55] is a Federated Learning framework that enables the development and deployment of full FL models. It offers various aggregation algorithms and customization options for real-edge devices. The framework demonstrates strengths in both simulation and real-world device scenarios [55]. The architecture of the Flower framework consists of two main building blocks: the global model and the local model. Individual training is conducted on the local clients, and the model parameters are used for updating the global model. The global model performs client selection, parameter aggregation, configuration, and distributed or centralized model evaluation through strategy abstraction. Popular algorithms like FedAvg [23] and FedYogi [56] are used for this purpose.

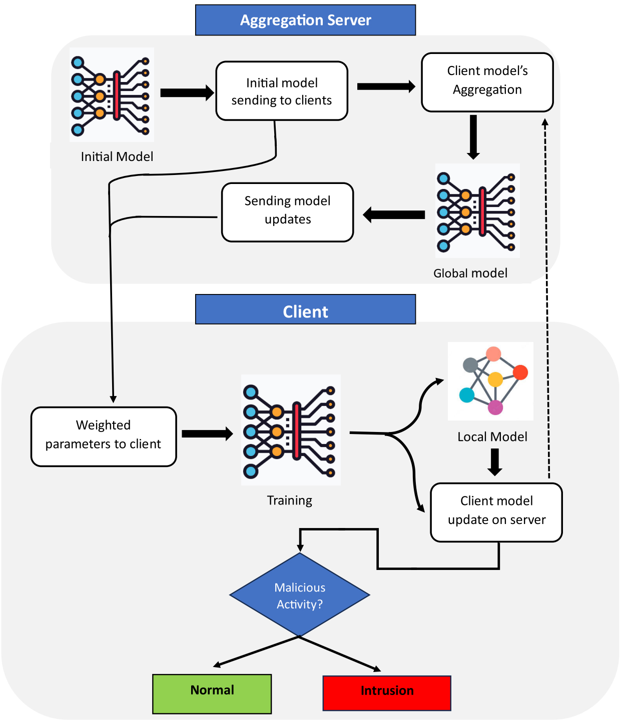

FIGURE 4. The proposed FL-IDS framework: An initial model is fed into the client's model and the updated model is sent to the global model simultaneously. The output of the global model with updated parameters is fed back to the client's model for intrusion detection. Then the respective server model will predict the malicious activity whether it is a normal or abnormal behaviour.

The six-step approach of the Flower framework involves: initializing the global model, sending the model to client nodes, training the model locally on each client node, returning model updates to the server, aggregating the model updates into a new global model using algorithms such as FedAvg [57], and repeating the process until the model converges. This iterative process ensures that each client node is trained on all the data and contributes to developing a fully trained model that performs well across all client nodes. With the Flower Framework, we created 3 files, namely client, server, and utils python files. The client file indicates the local training on individual clients. At the same time, the server file is associated with the global model, which takes all the parameter updates from the different clients. Lastly, the utils file is used for data massaging and distributing the data across the local clients. To implement the Framework, we used a testbed architecture explained as follows:

Testbed Setup: We deployed two different architectures to assess the performance of FL implementations. The implementation involved several components: a server model Python file, a utils Python file for data preprocessing, and individual client model Python files specific to the testbed setup. Depending on the particular configuration, the testbed setup utilized NVIDIA Jetson Xavier and Raspberry Pi 4 as servers and clients respectively. The specifications of the edge devices used are shown in Table 2.

TABLE 2. Specification of evaluation environment.

Feature	NVIDIA Jetson Xavier	Raspberry Pi 4
SoC	8-core ARMv8 64-bit (Cortex-A57)	Quad-core ARM Cortex-A72 64-bit
GPU	NVIDIA Volta Architecture	Broadcom VideoCore VI
RAM	16 GB LPDDR4x	8 GB LPDDR4 SDRAM
Storage	32 GB eMMC Flash	MicroSD card slot
Kernel	Customized Linux for Tegra (L4T)	Linux Kernel

These embedded devices are well-suited for lightweight FL operations. The deployment of FL on these devices was facilitated by the Flower framework, which in addition to its previously mentioned advantages, offers compatibility and efficient utilization of resources on such lightweight devices.

1) **Testbed 1:** At the initial stage of our implementation, we configured a testbed with two clients, as illustrated in Figure 5. The testbed setup consisted of one Jetson Xavier device serving as the server and two clients of Raspberry Pi 4. Each client was designed to function as an independent IDS for CAVs. To perform decentralized machine learning, we employed two algorithms, namely Logistic Regression and PCC-CNN, on the NSL-KDD and Car-Hacking datasets. These algorithms were executed separately on each client, enabling them to autonomously analyze and detect intrusions based on their local datasets.
2) **Testbed 2:** To expand our implementation, we introduced federated learning (FL) with four clients, as depicted in Figure 6. The testbed setup involved using one Jetson Xavier device as the server, equipped with a GPU, and four Raspberry Pi devices serving as clients. Each client was configured as an individual IDS-based CAV, capable of performing intrusion detection using the NSL-KDD and Car-Hacking

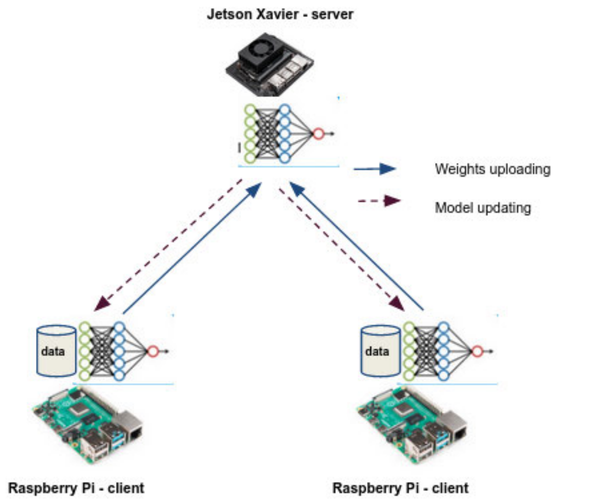

FIGURE 5. FL Testbed1 with 2 clients.

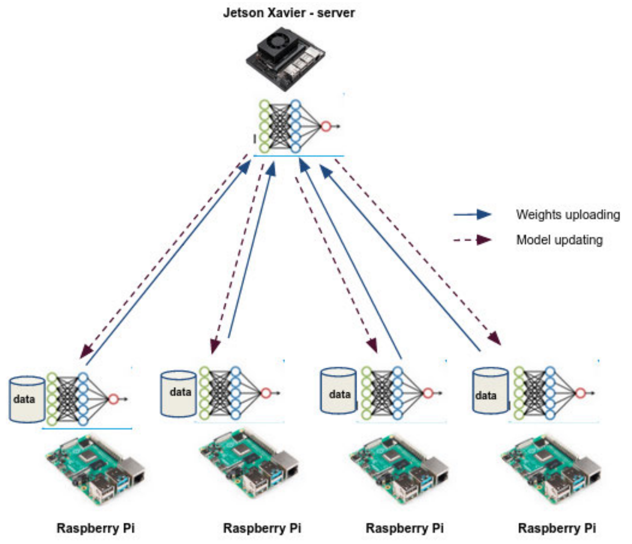

FIGURE 6. FL Testbed2 with 4 clients.

datasets. All four clients utilized the NSL-KDD and Car-Hacking datasets in the first implementation for training and evaluation. However, in the second implementation, we aimed for a more realistic scenario by employing four different versions of the NSL-KDD dataset, with each version assigned to an individual client. We used the PCC-CNN model from our previous paper [46], then modified the PCC-CNN model to overcome memory limitations. These modifications were necessary to ensure the efficient execution of the model on the resource-constrained devices, particularly in terms of memory usage.

A. LOGISTIC REGRESSION (LR)- INTRUSION DETECTION SYSTEM (IDS)

Logistic Regression is a statistical model used for binary classification tasks, where the goal is to predict the probability of an event occurring. It is a type of generalized linear model that uses a logistic function to map the input features to the output probabilities. In LR, the input features are linearly combined with weights, and the resulting value is passed through a logistic function (also known as a sigmoid function) to produce the predicted probability. The logistic function transforms the linear combination into a value between 0 and 1, representing the probability of the positive class.

The LR model assumes that the relationship between the input features and the output probability is linear, but it can capture non-linear relationships by including interactions or polynomial terms of the input features. During training, the LR model is fitted to the training data using maximum likelihood estimation, where the parameters (weights) are adjusted to maximize the likelihood of the observed outcomes. The LR is deployed with the tuned parameters. At the initial stage, we set the initial parameters, which are explained as follows:

B. CONVOLUTION NEURAL NETWORK (CNN) - INTRUSION DETECTION SYSTEM (IDS)

A Convolutional Neural Network (CNN) is a type of deep learning model commonly used for analyzing visual data, such as images and videos. CNNs are specifically designed to automatically learn and extract hierarchical representations of patterns and features from input data. CNNs comprise multiple layers, including convolutional, pooling, and fully connected layers. Each layer performs specific operations to extract and process information from the input data. Here is a brief overview of the main components of a CNN:

1) Convolutional layers: These layers apply convolution operations to the input data. Convolution involves sliding a small filter (also known as a kernel) over the input data and performing element-wise multiplications and summations. This operation captures local patterns and spatial relationships in the data.
2) Pooling layers: Pooling layers downsample the feature maps obtained from the convolutional layers. They reduce the spatial dimensions of the data while retaining the most important features. Common pooling operations include max pooling, average pooling, and sum pooling.
3) Activation functions: Activation functions introduce non-linearities into the network, allowing it to model complex relationships in the data. Popular activation functions used in CNNs include ReLU (Rectified Linear Unit), sigmoid, and tanh.
4) Fully connected layers: These layers make predictions based on the learned features. They connect every neuron from the previous layer to every neuron in the current layer, forming a fully connected network. The output of the fully connected layers is often fed into a softmax layer for multi-class classification or a sigmoid layer for binary classification.
5) Training with backpropagation: CNNs are trained using the backpropagation algorithm, which com-

putes gradients and updates the network's parameters (weights and biases) based on the error between the predicted and actual outputs. Optimization techniques like stochastic gradient descent (SGD) and its variants are commonly used to iteratively update the network's parameters during training.

CNNs have shown remarkable performance in various computer vision tasks, including image classification, object detection, and image segmentation. They can automatically learn and extract meaningful features from raw visual data, making them well-suited for tasks that require understanding and analyzing complex visual patterns. Our PCC-CNN model combines PCC (Pearson correlation Coefficient) and Convolution neural network (CNN). Firstly, important features have been extracted via PCC from the preprocessed data. Then the optimal features are used for the prediction of anomaly behavior.

Our modified PCC-CNN model has the following configurations:

1) A convolutional 1D layer of size (91 × 64) & (10 × 64) using RELU activation function.
2) A Maxpooling 1D layer
3) A flattening layer
4) A dense layer of size 128 using the RELU activation function.
5) A dropout layer
6) A dense layer of size 2 using the Softmax activation function
7) An output layer of 2 classes using the Adam optimizer

The modified CNN model shown in Fig. 7 is constructed sequentially, consisting of six layers. The first layer is a convolutional layer with a size of 91 × 64 and utilizes the rectified linear unit (ReLU) activation function. This layer performs convolutions on the input data to extract relevant features. Following the convolutional layer, a MaxPooling1D layer is employed to reduce the length of the input tensor while retaining important features. A dropout layer with a rate of 30% is added to prevent overfitting. Dropout randomly disables some neurons during training, forcing the network to learn more robust representations. After the dropout layer, the flattened layer reshapes the output from the previous layer into a one-dimensional format. Dense layers, which are capable of analyzing values in a nonlinear manner, are then utilized. These layers extract higher-level features from the flattened input. Finally, the model employs the Adam optimizer to adjust the parameter values in the final layer. The number of class parameters is set to 2 for binary classification, indicating the expected outcome categories. During the training process, the model is trained for five epochs, which refers to the number of times the entire dataset is passed through the network for learning and adjusting the model's parameters. This process allows the model to refine its predictions and improve its performance in binary classification tasks.

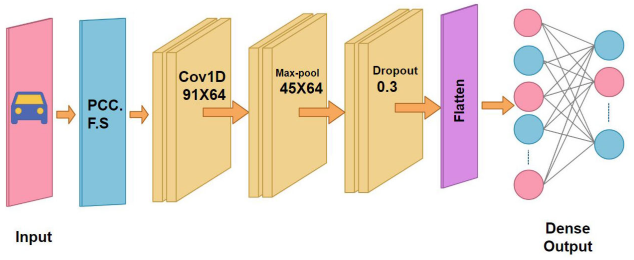

FIGURE 7. Modified PCC-CNN model.

IV. EXPERIMENTAL RESULTS

The proposed FL-IDS framework with LR and PCC-CNN models was evaluated on Testbed 1 and Testbed 2. The actual implementation looks like Figure 8, where 2 or 4 Raspberry Pis are used as clients depending upon the Testbed, and Jetson Xavier is used as a Server. For communication between clients and servers, we used a wireless Local Area Network (LAN). The server and client device specifications are given in Table 2. NVIDIA Jetson Xavier is used as a server and the clients are all Raspberry Pi 4 boards.

The Python programming language was utilized to implement the models, along with the Keras and PyTorch libraries as the frontend and the TensorFlow library as the backend. These libraries provided the necessary tools for training, testing, and benchmarking the models. Additionally, the Flower framework [55] was employed to build the FL architecture. Different versions of Python and associated libraries were utilized to implement the FL framework on the Raspberry Pi devices. To measure the performance of our model, we measured the starting centralized accuracy and loss of the test model along with the finished federated learning accuracy and model loss with total execution time taken.

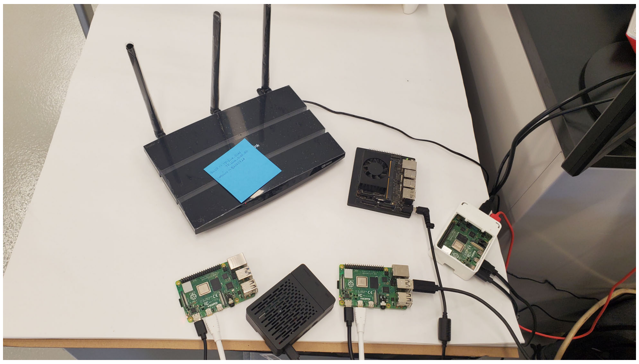

FIGURE 8. Actual implementation.

A. DATASETS AND TRAINING

For evaluation purposes, we use the NSL-KDD dataset [58], a benchmarking network traffic dataset collected by UNB, as they were not recorded from the same probability, making them realistic. It mainly focuses on the DDoS (Distributed Denial of Service) attack. The NSL-KDD dataset contains two.csv files named KDD_Train and KDD_Test file. The NSL-KDD dataset consists of five different classes, which are:

1) DoS (Denial of Service): This class includes attacks aimed at overloading the targeted server or network with an excessive quest, causing it to become unavailable or unresponsive. Examples of DoS attacks include Smurf, Neptune, and Teardrop attacks.
2) Probe: In this class, the attacker scans the network to identify vulnerabilities that can be exploited. The goal is to gather information about the target system or network for future attacks. Examples of probe attacks include Satan, ipsweep, and Nmap attacks.
3) R2L (Remote to Local): Attacks in this class involve attempts by the attacker to gain unauthorized access to the victim's machine or network by sending packets from a remote location. The goal is to exploit vulnerabilities and gain local access to sensitive information. Examples of R2L attacks include ejection, load module, and Perl attacks.
4) U2R (User to Root): This class involves attacks where attackers gain escalated privileges on the target system by exploiting vulnerabilities. The attacker starts with a regular user account and then attempts to gain root or administrative access. Examples of U2R attacks include FTP write, guess the password, and IMAP attacks.
5) Normal: The normal traffic behavior.

The attack instances of NSL-KDD are shown in Table 3.

TABLE 3. NSL-KDD attack samples.

	Attack Instances	
Attack Category	NSL_Train	NSL_Test
Normal	67343	9711
Dos	45927	7460
Probe	11656	2885
R2L	995	2421
U2R	52	67

To include the realistic scenario of CAV, we use the Car-Hacking [59]dataset, a collection of data specifically designed for studying and analyzing security vulnerabilities and potential attacks in automotive systems. It aims to provide researchers and practitioners in the field of automotive cybersecurity with a comprehensive dataset that simulates real-world car-hacking scenarios. The dataset consists of various types of data captured from automotive systems, including network communications, sensor readings, vehicle control commands, and potential attack vectors. It encompasses a wide range of information relevant to the functioning and operation of a vehicle, allowing researchers to analyze and understand the security risks and challenges associated with modern automobiles.

The Car-Hacking dataset was explicitly collected for the "Car Hacking: Attack & Defense Challenge" competition held in 2020 [60]. Its primary purpose was to enhance the techniques for attacking and detecting vulnerabilities in the Controller Area Network (CAN) [61], a widely used standard for in-vehicle networks. The dataset was generated by capturing CAN traffic from vehicles through the OBD-II port while conducting message injection attacks. Each attack lasted 3 to 5 seconds, and the overall dataset spanned approximately 30 to 40 minutes of collected data.

The Car-Hacking dataset consists of five different classes, which are:

1) Normal: This class represents the normal behavior and legitimate communication within the CAN network of the vehicle.
2) DoS (Denial of Service): This class involves attacks aimed at disrupting or disabling the normal functioning of the CAN network.
3) Spoofing the Drive Gear: This class relates to attacks where the attacker manipulates or falsifies data related to the vehicle's drive gear.
4) Spoofing the RPM Gauge: This class encompasses attacks that involve tampering with or forging data related to the vehicle's RPM (revolutions per minute) gauge.
5) Fuzzy: This class represents attacks that introduce noise or fuzziness into the CAN network, potentially causing confusion or disruptions.

The attack instances of Car-Hacking are shown in Table 4.

TABLE 4. Car-hacking attack samples.

Attack Category	Attack Instances
Benign(Normal)	1403673
Dos	58469
Spoofing drive gear	60016
Spoofing RPM gauge	65439
Fuzzy attack	49258

B. RESULTS OF LR-IDS

A machine learning-based solution, LR-IDS, employs a simple Logistic Regression model. The centralized model runtime efficiency for the NSL-KDD dataset was 97% with a training time was 42.08 sec. Similarly, runtime efficiency for the Car-Hacking dataset was 93% with a training time was 83.28 sec. The classification accuracy results in Table 5 for binary classification using two different datasets, NSL-KDD and Car-Hacking, were compared. With the NSL-KDD dataset, the average accuracy achieved was 96.82%, with a loss of 0.127. This indicates that the machine learning-based intrusion detection system (IDS) using PCC feature extraction and Logistic Regression performed well in accurately classifying instances from the NSL-KDD dataset. The processing time for measuring this accuracy was 10 seconds. However, when using the Car-Hacking dataset, the accuracy achieved was 93.56%, with a higher loss value of 0.28. This suggests that the IDS encountered challenges in accurately classifying instances from the Car-Hacking dataset compared to the NSL-KDD dataset. Furthermore, the

TABLE 5. Anomaly detection on testbed 1 & 2 on LR-IDS.

Dataset	Testbed	Centralized		Federtaed		Time (s)
		Accuracy	Loss	Accuracy	Loss	
NSL-KDD	1	46.54	0.69	96.82	0.127	10
NSL-KDD	2	46.54	0.69	97.08	0.119	14
Car-Hacking	1	14.25	0.69	93.59	0.28	306
Car-Hacking	2	14.25	0.69	93.56	0.28	214

TABLE 6. Anomaly detection on testbed 1 & 2 on CNN-IDS.

Dataset	Testbed	Centralized		Federtaed		Time (s)
		Accuracy	Loss	Accuracy	Loss	
NSL-KDD	1	46.54	0.69	96.82	0.127	10
NSL-KDD	2	46.54	0.69	97.08	0.119	137
Car-Hacking	1	32.98	0.69	99.72	0.0095	29201
Car-Hacking	2	14.25	0.69	99.92	0.0038	6731

processing time for measuring this accuracy was significantly longer, taking 306 seconds. The longer processing time may indicate that the Car-Hacking dataset is more complex or larger, requiring more computational resources and time to process. It's important to note that the Car-Hacking dataset did not perform as well as the NSL-KDD dataset with the given machine learning-based IDS. The reasons for this discrepancy could be attributed to various factors, such as differences in the characteristics and distribution of the two datasets, the presence of unique challenges or complexities in the Car-Hacking dataset, or the need for further optimization or tuning of the IDS for that specific domain.

In summary, the LR-based IDS using PCC feature extraction and Logistic Regression achieved high accuracy with the NSL-KDD dataset but encountered challenges with the Car-Hacking dataset, resulting in lower accuracy despite a longer processing time. Further analysis and improvements may be necessary to enhance the performance of the IDS on the Car-Hacking dataset.

C. RESULTS OF CNN-IDS

In the Deep learning-based solution, a simple Convolutional Neural Network (CNN) called PCC-CNN was employed. The centralized model runtime efficiency for the NSL-KDD dataset was 96% with a training time was 75.98 sec. Similarly, runtime efficiency for the Car-Hacking dataset was 99% with a training time was 557.65 sec. The classification accuracy results for binary classification tasks using the NSL-KDD and Car-Hacking datasets for Testbed 1 and Testbed 2 are shown in Table 6. In Testbed 1 with two clients, the average accuracy achieved for anomaly classifications was 97% for the NSL-KDD dataset and an impressive 99.93% for the Car-Hacking dataset. This indicates that the PCC-CNN model performed well in accurately classifying instances from both datasets, with the Car-Hacking dataset exhibiting even higher accuracy. The model loss is also comparably lower at 0.12 and 0.005 for both datasets. The model was trained for 3 epochs with the NSL-KDD dataset and 3 epochs with the Car-Hacking dataset. However, we find challenges for the computational time, which is higher than the LR-IDS, but the reason behind that is the large number of attack instances and utilization of the small capacity of Raspberry Pi. These findings suggest that the deep learning-based IDS using PCC-CNN outperformed the machine learning-based IDS with Logistic Regression regarding accuracy. Both the NSL-KDD and Car-Hacking datasets achieved high accuracy with the PCC-CNN model. The Car-Hacking dataset, in particular, demonstrated exceptional performance, even with more clients. However, our model requires longer training time on the car-hacking dataset. Further analysis is needed to improve the runtime of the model.

The deep learning-based IDS approach using the PCC-CNN model demonstrated promising results in terms of computational time, indicating its practical efficiency. The PCC-CNN model's ability to perform well on various datasets highlights its versatility and adaptability. By achieving high accuracy and low loss values, the model showcased its effectiveness in accurately classifying instances and detecting anomalies.

V. CONCLUSION

We addressed dynamic FL-IDS by using ML and DL techniques. Our method detects intrusions by the FedAvg algorithm on the server side, which is the average of parameters from different clients. Results showed that our proposed method outperformed individual networks in terms of detection accuracy and also achieved a competitive loss. Our modified PCC-CNN model outperforms better compared to the LR-IDS model. Our method achieved a low runtime overhead by running and fusing the PCC and CNN in parallel. We proposed a novel, modular, scalable, and maintainable FL framework that uses ML and DL techniques that detect network anomalies for CAVs without disrupting the privacy and security of the end user. The key idea is to introduce a computationally efficient FL-IDS method to Provide data privacy for individual clients. We demonstrated that a real-time and accurate intrusion detection system could be developed by running decentralized learning and deep learning models in parallel. We thoroughly investigated the proposed FL-IDS framework's performance by multiple clients using Raspberry Pi and the Jetson Xavier combination. We also used a new Car-Hacking dataset which was specifically created for autonomous car communication architecture. This dataset allowed us to evaluate the generalizability of our proposed models. The results exhibited that PCC-CNN and LR-IDS models outperform current state-of-the-art anomaly detection in terms of (lower) loss and (higher) average detection accuracy values. Further optimization and tuning of the model are required to improve its performance. As the car-hacking dataset has imbalanced feature types, it is good to use specific methods to address the class imbalance. Future work includes developing the dataset specifically designed for CAVs. In addition, the developed framework with enhanced intrusion detection performance paves the way for tracking attacks in run-time for more informed decision-making by autonomous cars. Also, the adversarial attack implementation on top of FL-IDS to measure the robustness of the proposed Framework.

ACKNOWLEDGMENT

This work is partially supported by Cisco Systems, Inc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Cisco Systems, Inc.

REFERENCES

[1] E. E. Abdallah, A. Aloqaily, and H. Fayez, "Identifying intrusion attempts on connected and autonomous vehicles: A survey," *Proc. Comput. Sci.*, vol. 220, pp. 307–314, Jan. 2023.

[2] *BCG: Autonomous Car Market to Hit 42 Billion by 2025*. Accessed: May 5, 2023. [Online]. Available: https://www.consultancy.uk/news/2065/bcg-autonomous-car-market-to-hit-42-billion-by-2025

[3] Q. He, X. Meng, and R. Qu, "Survey on cyber security of CAV," in *Proc. Forum Cooperat. Positioning Service (CPGPS)*, May 2017, pp. 351–354.

[4] Q. Yang, S. Fu, H. Wang, and H. Fang, "Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 3, pp. 96–101, May 2021.

[5] E. Hodo, X. Bellekens, A. Hamilton, C. Tachtatzis, and R. Atkinson, "Shallow and deep networks intrusion detection system: A taxonomy and survey," 2017, *arXiv:1701.02145*.

[6] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Int. J. Secur. Appl.*, vol. 9, no. 5, pp. 205–216, May 2015.

[7] Y. Guo, Z. Zhao, K. He, S. Lai, J. Xia, and L. Fan, "Efficient and flexible management for Industrial Internet of Things: A federated learning approach," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108122.

[8] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, "Foundations for smarter cities," *IBM J. Res. Develop.*, vol. 54, no. 4, pp. 1–16, Jul. 2010.

[9] D. U. Case, "Analysis of the cyber attack on the Ukrainian power grid," in *Proc. Electr. Inf. Sharing Anal. Center (E-ISAC)*, vol. 388, 2016, pp. 1–29.

[10] M. R. Jabbarpour, A. Nabaei, and H. Zarrabi, "Intelligent guardrails: An IoT application for vehicle traffic congestion reduction in smart city," in *Proc. IEEE Int. Conf. Internet Things (iThings), IEEE Green Comput. Commun. (GreenCom), IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Dec. 2016, pp. 7–13.

[11] S. Kulanthaiyappan, S. Settu, and C. Chellaih, "Internet of Vehicle: Effects of target tracking cluster routing in vehicle network," in *Proc. 6th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2020, pp. 951–956.

[12] A. Ghosal and M. Conti, "Security issues and challenges in V2X: A survey," *Comput. Netw.*, vol. 169, Mar. 2020, Art. no. 107093.

[13] S.-I. Sou, "Modeling emergency messaging for car accident over dichotomized headway model in vehicular ad-hoc networks," *IEEE Trans. Commun.*, vol. 61, no. 2, pp. 802–812, Feb. 2013.

[14] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks," *IEEE Internet Things J.*, vol. 1, no. 6, pp. 570–577, Dec. 2014.

[15] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.

[16] L. He, A. Bian, and M. Jaggi, "COLA: Decentralized linear learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[17] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[18] D. Preuveneers, V. Rimmer, I. Tsingenopoulos, J. Spooren, W. Joosen, and E. Ilie-Zudor, "Chained anomaly detection models for federated learning: An intrusion detection case study," *Appl. Sci.*, vol. 8, no. 12, p. 2663, Dec. 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/12/2663

[19] S. Savazzi, M. Nicoli, M. Bennis, S. Kianoush, and L. Barbieri, "Opportunities of federated learning in connected, cooperative, and automated industrial systems," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 16–21, Feb. 2021.

[20] Z. Du, C. Wu, T. Yoshinaga, K. A. Yau, Y. Ji, and J. Li, "Federated learning for vehicular Internet of Things: Recent advances and open issues," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 45–61, 2020.

[21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[22] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[23] J. Koneč ný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[24] S. T. Mehedi, A. Anwar, Z. Rahman, and K. Ahmed, "Deep transfer learning based intrusion detection system for electric vehicular networks," *Sensors*, vol. 21, no. 14, p. 4736, Jul. 2021.

[25] R. Peng, W. Li, T. Yang, and K. Huafeng, "An Internet of Vehicles intrusion detection system based on a convolutional neural network," in *Proc. IEEE Int. Conf. Parallel Distrib. Process. Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2019, pp. 1595–1599.

[26] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based intelligent intrusion detection system in Internet of Vehicles," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[27] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4507–4518, Jul. 2021.

[28] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense," *Comput. Secur.*, vol. 103, Apr. 2021, Art. no. 102150.

[29] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles," *Ad Hoc Netw.*, vol. 90, Jul. 2019, Art. no. 101823.

[30] 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.

[31] D. Man, F. Zeng, J. Lv, S. Xuan, W. Yang, and M. Guizani, "AI-based intrusion detection for intelligence Internet of Vehicles," *IEEE Consum. Electron. Mag.*, vol. 12, no. 1, pp. 109–116, Jan. 2023.

[32] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, vol. 6, pp. 3491–3508, 2018.

[33] A. Tomlinson, J. Bryans, and S. A. Shaikh, "Towards viable intrusion detection methods for the automotive controller area network," in *Proc. 2nd ACM Comput. Sci. Cars Symp.*, 2018, pp. 1–9.

[34] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Comput. Netw.*, vol. 136, pp. 37–50, May 2018.

[35] I. Berger, R. Rieke, M. Kolomeets, A. Chechulin, and I. Kotenko, "Comparative study of machine learning methods for in-vehicle intrusion detection," in *Proc. Int. Workshop Security Privacy Requirements Eng.* Springer, Jan. 2019, pp. 85–101.

[36] Q. He, X. Meng, R. Qu, and R. Xi, "Machine learning-based detection for cyber security attacks on connected and autonomous vehicles," *Mathematics*, vol. 8, no. 8, p. 1311, Aug. 2020.

[37] N.-A. Stoian, "Machine learning for anomaly detection in IoT networks: Malware analysis on the IoT-23 data set," B.S. thesis, Elect. Eng., Math. Comput. Sci. (EEMCS), Univ. Twente, 2020.

[38] 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.

[39] A. Thakkar and R. Lohiya, "A review on machine learning and deep learning perspectives of IDS for IoT: Recent updates, security issues, and challenges," *Arch. Comput. Methods Eng.*, vol. 28, no. 4, pp. 3211–3243, Jun. 2021.

[40] M. Bhavsar, K. Roy, Z. Liu, J. Kelly, and B. Gokaraju, "Intrusion-based attack detection using machine learning techniques for connected autonomous vehicle," in *Proc. 35th Int. Conf. Ind., Eng. Other Appl. Appl. Intell. Syst. (IEA/AIE)*, Kitakyushu, Japan. Cham, Switzerland: Springer, Jul. 2022, pp. 505–515.

[41] N. Thapa, Z. Liu, D. B. Kc, B. Gokaraju, and K. Roy, "Comparison of machine learning and deep learning models for network intrusion detection systems," *Future Internet*, vol. 12, no. 10, p. 167, Sep. 2020.

[42] D. Basavaraj and S. Tayeb, "Towards a lightweight intrusion detection framework for in-vehicle networks," *J. Sensor Actuator Netw.*, vol. 11, no. 1, p. 6, Jan. 2022.

[43] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang, and E. Dutkiewicz, "Deep transfer learning for IoT attack detection," *IEEE Access*, vol. 8, pp. 107335–107344, 2020.

[44] A. Samy, H. Yu, and H. Zhang, "Fog-based attack detection framework for Internet of Things using deep learning," *IEEE Access*, vol. 8, pp. 74571–74585, 2020.

[45] M. Roopak, G. Y. Tian, and J. Chambers, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," *IET Netw.*, vol. 9, no. 3, pp. 120–127, May 2020.

[46] M. Bhavsar, K. Roy, J. Kelly, and O. Olusola, "Anomaly-based intrusion detection system for IoT application," *Discover Internet Things*, vol. 3, no. 1, p. 5, May 2023.

[47] S. Pamarthi and R. Narmadha, "Literature review on network security in wireless mobile ad-hoc network for IoT applications: Network attacks and detection mechanisms," *Int. J. Intell. Unmanned Syst.*, vol. 10, no. 4, pp. 482–506, Nov. 2022.

[48] M. Chowdhury, M. Islam, and Z. Khan, "Security of connected and automated vehicles," 2020, *arXiv:2012.13464*.

[49] N. Hussain, P. Rani, H. Chouhan, and U. S. Gaur, "Cyber security and privacy of connected and automated vehicles (CAVs)-based federated learning: Challenges, opportunities, and open issues," in *Federated Learning for IoT Applications*. Cham, Switzerland: Springer, 2022, pp. 169–183.

[50] V. P. Chellapandi, L. Yuan, S. H. Zak, and Z. Wang, "A survey of federated learning for connected and automated vehicles," 2023, *arXiv:2303.10677*.

[51] A. A. Korba, A. Boualouache, B. Brik, R. Rahal, Y. Ghamri-Doudane, and S. M. Senouci, "Federated learning for zero-day attack detection in 5G and beyond V2X networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2023, pp. 1137–1142.

[52] M. Basnet and M. H. Ali, "A deep learning perspective on connected automated vehicle (CAV) cybersecurity and threat intelligence," 2021, *arXiv:2109.10763*.

[53] H. Taslimasa, S. Dadkhah, E. Carlos Pinto Neto, P. Xiong, S. Iqbal, S. Ray, and A. A. Ghorbani, "ImageFed: Practical privacy preserving intrusion detection system for in-vehicle CAN bus protocol," in *Proc. IEEE IEEE 9th Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput., (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS)*, May 2023, pp. 122–129.

[54] M. A. Ferrag, O. Friha, L. Maglaras, H. Janicke, and L. Shu, "Federated deep learning for cyber security in the Internet of Things: Concepts, applications, and experimental analysis," *IEEE Access*, vol. 9, pp. 138509–138542, 2021.

[55] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K. H. Li, T. Parcollet, P. P. B. de Gusmão, and N. D. Lane, "Flower: A friendly federated learning research framework," 2020, *arXiv:2007.14390*.

[56] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," 2020, *arXiv:2003.00295*.

[57] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, "A performance evaluation of federated learning algorithms," in *Proc. 2nd Workshop Distrib. Infrastructures Deep Learn.*, Dec. 2018, pp. 1–8.

[58] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[59] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car hacking and defense competition on in-vehicle network," in *Proc. 3rd Int. Workshop Automot. Auto. Vehicle Secur.*, 2021, p. 25.

[60] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–6.

[61] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, Jan. 2020, Art. no. 100198.

MANSI H. BHAVSAR received the B.Sc. degree in electrical and electronic engineering (EEE) from Gujarat Technological University (GTU), Gujarat, in 2014, and the M.S. degree in electrical and computer engineering from North Carolina A&T State University, USA, in 2019, where she is currently pursuing the Ph.D. degree with the Cyber Defense and AI Laboratory, Department of Computer Science. Since 2021, she has been a Graduate Research Assistant with the Cyber Defense and AI Laboratory, NC A&T, working on the Ph.D. dissertation. Her current research interests include the IoT applications, autonomous vehicle network architecture, and embedded devices.

YOHANNES B. BEKELE (Member, IEEE) received the B.S. degree in electrical engineering from Arba Minch University, Ethiopia, in 2007, and the M.S. degree in telecommunication networks engineering from Addis Ababa University, Ethiopia, in 2018. He is currently pursuing the Ph.D. degree with North Carolina A&T State University.

Since 2019, he has been a Graduate Research Assistant with the Automated Design for Emerging Processing Technologies (ADEPT) Laboratory, NC A&T, working on his research toward the Ph.D. dissertation. His research interests include microarchitectural and architectural reliability evaluation, hardware security and its relations with reliability, and microkernel evaluation methodologies.

KAUSHIK ROY is currently a Professor and the Chair of the Department of Computer Science, North Carolina A&T State University (NCAT). He is the Jefferson-Pilot/Ron McNair Endowed Chair of the Department of Computer Science. He has more than 160 publications including 45 journal articles and a book. His research is funded by the National Science Foundation (NSF), the Department of Defense (DoD), and the Department of Energy (DoE). He is the Director of the Center for Cyber Defense (CCD) and the Center for Trustworthy AI. He also leads the Cyber Defense and AI Laboratory. His current research interests include cybersecurity, cyber identity, biometrics, machine learning (with a focus on deep learning), data science, cyber-physical systems, and big data analytics.

JOHN C. KELLY received the Ph.D. degree in electrical engineering from the University of Delaware. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, North Carolina A&T State University. His research interests include hardware security in cyber-physical systems and embedded systems security. He also contributes to research on engineering education, enhanced retention of underrepresented minorities in engineering, and hands-on learning techniques.

DANIEL LIMBRICK (Senior Member, IEEE) received the B.S. degree in electrical engineering from Texas A&M University and the M.S. and Ph.D. degrees from Vanderbilt University. He is currently an Associate Professor with the Electrical and Computer Engineering Department, NC A&T. He leads the Automated Design for Emerging Processing Technologies (ADEPT) Laboratory, NC A&T, where he researches ways to improve the reliability and scalability of integrated circuits through logic and physical synthesis. His research interests include electronic design automation, post-CMOS technologies, computer architecture, laboratory-on-a-chip, and the reliability of microelectronics.