AI-Sentinel: A Novel AI-Powered Intrusion Detection Approach Against Cyber Threats for In-Vehicular Communication Systems

Rimsha Jamil Ghilzai[1], Hafiz Gulfam Ahmad Umer[2], Urwa Bibi[3], Muskan Maryam[4]
Ghazi University D.G.Khan
***Correspondence:** Rimsha Jamil Ghilzai, darkshadowsrj@gmail.com

Citation | Ghilzai. R. J, Umer. H. G. A, Bibi. U, Maryam. M, "AI-Sentinel: A Novel AI-Powered Intrusion Detection Approach Against Cyber Threats for In-Vehicular Communication Systems", IJIST, Vol. 07 Issue. 02 pp 1200-1224, June 2025

DOI | https://doi.org/10.33411/ijist/20257212001224

Received| April 12, 2025 **Revised|** June 03, 2025 **Accepted|** June 06, 2025 **Published|** June 08, 2025.

The emergence of revolutionizing technologies such as Artificial Intelligence and the Internet of Things, and their integration into the automotive industry has brought innovations and made the lives of common people easier and more complacent. Leveraging the advanced intelligent services provided by connected and autonomous vehicles, the driving experience is much more convenient and effortless. The CAN (Controller Area Network) protocol is the most deployed protocol in in-vehicular communications in the ICVs (intelligent connected vehicles) environment due to its efficiency and speed. However, it lacks basic security mechanisms like encryption and authentication, making it vulnerable to various cyber threats. In this article, we have presented a novel, robust, cutting-edge AI-based Intrusion detection system for detecting various seen and unseen cyber-attacks in in-vehicular networks to ensure security. Two main models deployed in the proposed framework are RNN for dealing with temporal dependencies in the CAN traffic and LightGBM for efficient feature extraction. The experimental results show that the hybrid of these two models performs better in terms of various evaluation metrics, with its accuracy being 94% in classifying the CAN traffic into normal and different attack classes. A comparison with the existing state-of-the-art approaches shows that our proposed approach is more robust and secure, with it being deployed in a Federated Learning FL environment.

Keywords: Intrusion detection; IDS, IVN; In-vehicular communication; ML; DL; Cyber-attacks; CAN bus; Federated Learning (FL); Intelligent connected vehicles (ICVs).

Introduction:

Intelligent Transportation Systems (ITSs) have recently become a major focus of research, driven by their ability to offer smart and automated solutions within the transportation industry. ITSs can address a number of issues that arise during the movement of people and commodities, including safety, trip time, and hazardous emissions, by fusing wireless devices with sensing technologies and sophisticated information and communication technologies (ICTs)[1]. These systems are versatile and can be implemented across various modes of transport, including cars, trucks, buses, trains, ships, and aircraft[2].

Reliable communication technologies like satellite and cellular networks are essential to all ITS deployments[3]. Nowadays, electronics like smartphones and technologies like Wi-Fi are used to connect cars. 70% of trucks and light-duty vehicles are expected to have internet connections by 2023. Electronic control units (ECUs) make driving and traveling easier by leveraging their huge usage in automotive networking[4]. Enhancing intelligent services and user safety in vehicles often involves increasing the number of Electronic Control Units (ECUs). However, this added complexity and connectivity also heightens the risk of security vulnerabilities[5].

Electronic components have largely taken the role of mechanical ones in connected and autonomous vehicles (CAVs) or autonomous vehicles. Numerous Electronic Control Units (ECUs) in these cars are connected by a variety of common automotive in-vehicle communication protocols, including FlexRay, Local Interconnect Network (LIN), Controller Area Network (CAN), and Media Oriented System Transport (MOST). CAN is regarded as the de facto protocol for in-vehicle communication among these protocols because of its features given as, noise cancellation, convenience to use, and high speed. It was first created for industrial machinery, but in-vehicle network communications have now embraced it[6]. An effective and dependable connection between ECUs is made possible via the controller area network (CAN) bus. The Controller Area Network (CAN), widely adopted as the standard for in-vehicle communication, does not include essential security mechanisms like message encryption or sender authentication. As a result, receiving nodes on the CAN bus are unable to verify the authenticity of incoming messages. As a result, CAN is vulnerable to various cyber threats, including isolation, impersonation, and denial-of-service (DoS) attacks [5]. A typical CAN network is presented in Figure 1.

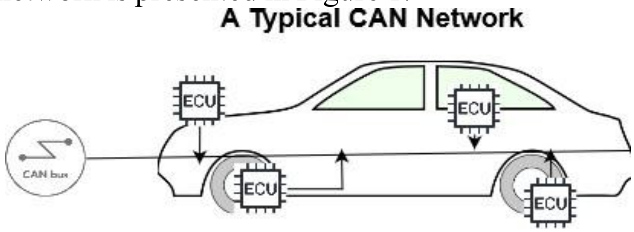


Figure 1. A typical CAN Network in a Car connecting various ECUs

The heavy demand on the vehicle's limited processing resources and the resulting increased latency are the primary reasons for not adopting these security measures in in-vehicle networks, which may cause a failure to achieve important safety-related deadlines [7][8]. Therefore, to ensure ease of implementation, every security mechanism should be lightweight [9].

Additionally, modern cars' interconnectedness creates attack surfaces that make them vulnerable to hackers. It is possible to physically or remotely access the attack surfaces. A USB, CD player, onboard diagnostic (OBD)-II port, and other devices can be used to get physical access. Furthermore, long-range wireless technologies like Wi-Fi and long-term evolution (LTE) as well as short-range wireless technologies like Bluetooth and radio frequency identification (RFID) can be used to provide remote access. As a result, the system is susceptible to several types of cyberattacks, which could have serious repercussions, including the loss of human life [10][11]. If a hacker gains access to the CAN bus system and transmits malicious messages, it can lead to serious consequences. For instance, unauthorized interference with critical vehicle functions such as steering, door locks, and braking poses a significant safety risk [9].

Vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) are the three primary forms of vehicular communication that facilitate road safety applications like instantaneous communication for emergency electric brake light warnings, lane change warning/blind spot warning, and collision warnings. Technology known as V2X makes it easier for cars and other infrastructures to communicate. Vehicle-to-vehicle (V2V) communication enables vehicles to exchange information with other vehicles, including destinations, velocity, and current location. Information about surrounding moving vehicles may also be included in V2V messages sent or received, enabling the driver to quickly identify vehicles in their blind spot. Conversely, V2I is a form of two-way communication that allows automobiles to communicate and exchange information with outside sources, like speed limits, parking spots, bicycles, and traffic lights. Radio communications that report on the environment within a few kilometers of a vehicle's location are also included in V2I [7].

Alongside these, two new forms of vehicular communications, Vehicle-to-Ecosystem (V2E) and Vehicle-to-Surroundings (V2S) have just surfaced. V2E occurs between automobiles and outside services, like satellite-based sites. Similar to the Global Positioning System (GPS), this service can operate as a one-way communication, or as a two-way interaction such as when users request navigation assistance. Vehicles and central control systems can communicate in both directions, a process known as V2S [12]. Generally speaking, the Internet of Vehicles (IoV) is a broader category that includes all of these distinct forms of vehicle interactions [2]. The various types of communications in an IoV environment are shown in Figure 2.

The revolutionary vehicular network concept known as the Internet of Vehicles (IoV) has received a lot of attention in literature. IoV is anticipated to be a key architecture for the future Cooperative Intelligent Transportation Systems (C-ITSs), combining the advantages of the Internet of Things (IoT) and Vehicular Ad-hoc Networks (VANETs), because of the growing number of vehicles on the road and the growing need for connected vehicles. The Internet of Vehicles (IoV) is expected to deliver high-speed and reliable communication, enabled by advanced technologies such as 5G-based cellular V2X and Dedicated Short-Range Communication (DSRC) [13].

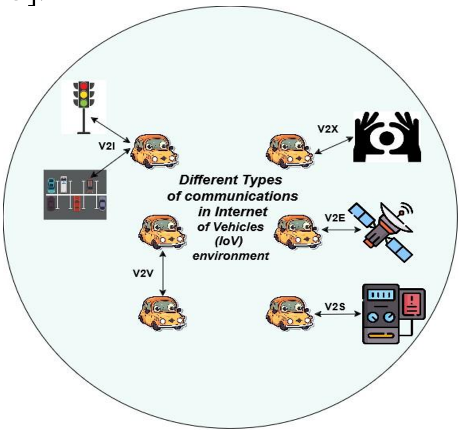

V2I
V2X
Different Types of communications in Internet of Vehicles (IoV) environment
V2E
V2V
V2S

Figure 2. Various types of communications in an IoV environment

Automated vehicles (AVs) and other vehicles on the Internet of Vehicles (IoV) ecosystem are outfitted with radio detection and ranging (RADAR), light detection and ranging (LiDAR), actuators that regulate the motion of the vehicle to help the driver, and computational image sensors that provide 360-degree surround vision. For vehicles to accurately perceive and understand their surrounding road conditions, these numerous sensors must function under challenging conditions with ultra-high reliability, approaching 100%. Through data collection and exchange between vehicles and infrastructure, these many forms of data have the potential to enhance road transportation safety, mobility flow, and environmental advantages. Additionally, this would make it possible for many applications including corporate fleet management, adaptive cruise control, and travel support [14].

However, there are cybersecurity risks associated with the many benefits that IoV offers [15][16]. Several types of attacks on vehicles have been reported in the literature. For instance, in 2010, author conducted a security evaluation in which they exploited a standard diagnostic feature to disable ECU connectivity over the CAN bus. This was a denial-of-service attack since the ECU's CAN communication was stopped, breaching the car's security features. Fortunately, there is no indication that this method poses an information security threat or endangers passenger safety. However, the attack requires physical access via the OBD port, and it can lead to inefficient use of vehicle resources, potentially affecting the vehicle's overall lifespan. According to the Common Vulnerability Scoring System (CVSS), the attack received a rating of 7.4 [17].

One instance of a cybersecurity breach in the automotive sector occurred in [18], where two hackers used flaws in a Jeep's system to remotely control the vehicle and carry out risky actions, such as turning the steering wheel suddenly and applying the parking brake suddenly at high speeds, resulting in disastrous accidents. Similarly, hackers were able to remotely take data and obtain unauthorized access by taking advantage of a flaw in a General Motors vehicle's infotainment system [19]. In 2018, the Keen Security Lab discovered a number of flaws in BMW vehicles that let hackers bypass the central gateway by inserting unified diagnostic services (UDS) packets into the CAN network [20]. Additionally, a 2020 assault on a Toyota Lexus used a Bluetooth vulnerability to cause unexpected physical motions in the car [21].

IDSs have been revealed to be a successful technique for detecting cyberattacks on in-vehicle networks. Malicious behavior on the network is tracked and detected by an IDS. The IDS is frequently implemented in an ECU and receives and examines incoming network traffic in the context of in-vehicle networks. It will alert other ECUs if any unusual messages are found. Intrusion detection systems (IDSs) are used in computer network systems to identify and stop intrusions. Nevertheless, a lot of traditional network security techniques aren't immediately applicable to in-vehicle networks. As a result, an efficient IDS for in-vehicle networks is crucial [9]. A general representation of an IDS employed in an in-vehicle network is depicted in Figure 3.

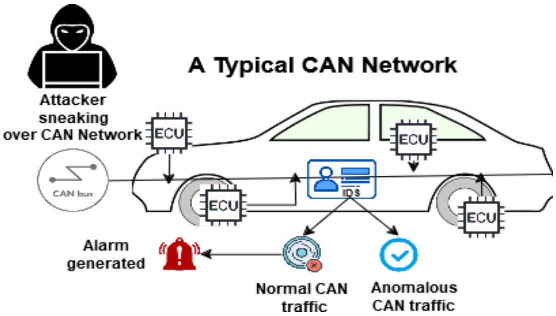


Figure 3. IDS employed in an in-vehicle typical CAN network

Numerous studies have used machine learning (ML) techniques to construct a variety of in-vehicle IDSs. However, the deployment environment, restricted computing resources, and robustness are three essential components of in-vehicle IDS requirements that are neglected by the current methods [9]. This paper aims to address these three key components by proposing a novel machine learning-based Intrusion Detection System (IDS) that is robust, lightweight, and easily deployable for detecting cyber-attacks targeting in-vehicle communication systems.

Literature Review:

In [5], an unsupervised intrusion detection method was introduced for in-vehicle communication networks, leveraging the feature extraction efficiency of autoencoders alongside the enhanced clustering accuracy of fuzzy C-means (FCM). The proposed method is lightweight and requires minimal computational resources. In a comprehensive experiment using the ML350 in-vehicle intrusion dataset, it achieved an accurate rate of 75.51%. According to experimental results, the suggested approach also performs better for other intrusion detection problems, such as network intrusion detection datasets like KDDCup (accuracy 60.63%), UNSW_NB15 (73.62%), and Information Security Center of Excellence (ISCX) (74.83%), and wireless intrusion detection datasets like WNS-DS (84.05%). When training an in-vehicle intrusion detection model, the suggested approach performs better overall than the current approaches and keeps out of labeled datasets. The suggested approach is robust and generalized in detecting intrusions, and it can be successfully implemented in real-time to monitor CAN traffic in vehicles and proactively alert during attacks, according to the results of an experiment conducted on a variety of intrusion detection datasets.

The author in[2] proposed an innovative intrusion detection system specifically designed to counter various cyberattacks in the Internet of Vehicles (IoV) environment, including denial-of-service (DoS), distributed denial-of-service (DDoS), distributed reflection denial-of-service (DRDoS), brute force attacks, botnets, and sniffing. Their approach involves a machine learning-based intrusion detection system that monitors network traffic to detect unusual patterns and identify abnormal data flows. Through a thorough assessment and selection of the best methods for the subsequent stages of the machine learning process, they have provided an intrusion detection strategy in the paper: (i) Z-score normalization for data preprocessing, which maintains the data distribution for the suggested method and manages outliers; (ii) regression model selection for feature selection, which reduces the complexity of the model and speeds up execution; and (iii) model selection and training, which includes Random Forest, Extreme Gradient Boosting, Categorical Boosting, and Light Gradient Boosting Machine, with hyper-parameter optimization to manage behavior during the training phase and avoid overfitting. Comprehensive numerical experiments utilizing the well-known standard datasets CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019, both alone and combined, show how effective the suggested method is. The authors demonstrated that the suggested intrusion detection system works better than the earlier techniques presented in the literature by achieving a high accuracy of over 99.8% in a running time of 46.9 seconds and a detection time of 0.24 seconds for the three combined intrusion detection system datasets.

The author [22], proposed a study that uses Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTM) to offer three deep learning-based misbehavior classification algorithms for intrusion detection in IoV networks. The suggested Deep Learning Classification Engines (DLCE) use deep learning models installed on the vehicle edge servers to perform single or multi-step classification. The three classification systems suggested in this study are used to classify the sent vehicle data to the edge server after being pre-processed once it has been received by the Road Side Units (RSUs). With F1 scores ranging from 95.58% to 96.75%, the suggested classifiers detect 18 distinct forms of vehicular behavior, which is significantly higher than the results of the current research. The suggested scheme's prediction performance and prediction time comparison are compared with those of previous studies by executing the classifiers on testbeds that simulate edge servers.

A Long Short-Term Memory (LSTM) autoencoder is used in the second stage of the proposed intrusion detection system (IDS) by [9] to detect new, undiscovered attacks, while an artificial neural network (ANN) is used in the first stage to detect known attacks. To deploy their IDS in a hierarchical federated learning (H-FL) environment, they suggested a theoretical framework to comprehend and evaluate various driving behaviors, update the model with the most recent attack patterns, and protect data privacy. According to experimental results, their IDS achieves a 99.99% detection rate with an F1 score that exceeds 0.99 for seen attacks and 0.95 for novel assaults. Furthermore, false alerts are minimized by the very low false alarm rate (FAR), which stands at 0.016%. Even with the use of DL algorithms, which are well-known for their ability to detect complex and zero-day threats, the IDS is nonetheless lightweight, guaranteeing its viability for practical implementation. Their model is hence resistant to both known and unknown threats.

Another study [23], suggested a new Dense Random Neural Network (DRNN)-based Distributed Self-Supervised Federated Intrusion Detection Algorithm (DISFIDA) with Online Self-Supervised Federated Learning. Neuronal weights are shared with Federated partners in DISFIDA while learning data is kept private. In DISFIDA, each partner mixes its synaptic weights with those it receives from other partners, favoring weights with numerical values that are more similar to the weights that it has independently learned. Networks of devices (such as body sensors) and connected smart vehicles (such as patient-transporting ambulances), DISFIDA for two pertinent IoT healthcare applications is evaluated using three publicly available datasets in comparison to five benchmark approaches. These tests demonstrate that the DISFIDA approach performs better at detecting attacks, with a 99% True Negative Rate comparable to state-of-the-art Federated Learning, for Distributed Denial of Service (DDoS) attacks. It also offers a 100% True Positive Rate for attacks, which is one percentage point higher than comparable state-of-the-art methods that achieve 99%.

The method presented in [24] employs Deep Shapley Additive Explanations (SHAP) to provide greater transparency, enabling cybersecurity experts to interpret and understand the decision-making process more effectively. Furthermore, the model uses hybrid bidirectional long-short-term memory with autoencoders (BiLAE) to improve computational efficiency by reducing the dimensionality of IoV network traffic. Additionally, it enhances detection capabilities without requiring large amounts of labeled data by optimizing the hyper-parameters of deep learning models like ResNet, Inception, Inception ResNet, and MobileNet Convolution neural network-transfer learning architecture (CNN-TL) using Barnacle Mating Optimizer (BMO). According to experimental results, the model achieved 99.88% accuracy and similarly high metrics in multi-class scenarios for external vehicular networks (N-BaIoT) and 100% accuracy, precision, recall, F1-score, and Matthew Correlation Coefficient in binary-class scenarios for internal vehicular (CAN) networks. The model demonstrated greater efficacy in identifying zero-day botnet assaults in comparison to state-of-the-art methods, hence decreasing dependence on extensive datasets.

Utilizing the network message ID's periodicity, a ConvLSTM-based IVN intrusion detection technique is suggested by [25]. A federated learning (FL) system with client selection is suggested for ConvLSTM model training. The basic FL framework operates in a client-server configuration. Mobile edge computing (MEC) servers linked to base stations (BSs) act as the parameter servers, while ICVs are the local clients. To maximize the model accuracy and system overhead of federated ConvLSTM model training, a proximal policy optimization (PPO) based federated client selection (FCS) technique is further developed based on the framework. Real-world IoV scenario settings and IVN datasets are used to run simulations. The findings show that the 95%-beyond detection accuracy is maintained while the model size and convergence time are significantly decreased by utilizing ConvLSTM. Additionally, the results show that the PPO-based FCS scheme performs better than the benchmarks in terms of system overhead, model correctness, and convergence rate.

The authors in[26], suggested a new framework for a benchmark system that emphasizes the security and dependability of autonomous cars. To assess and evaluate the state-of-the-art technologies currently employed in cyberattacks, the paper proposes a novel benchmark framework that focuses on physical and communication-based attacks. It also examines several security issues, vulnerabilities, exploitation techniques, and the detrimental effects of these on connected autonomous vehicles.

The authors in [27], investigated the potential for detecting cyberattacks in CAN systems using machine learning classifiers. They discussed two classifiers; extreme gradient boost and K-nearest neighbor algorithms to get applicability. But as their effectiveness depends on choosing the right parameters, a modified metaheuristic optimizer is presented as well to address parameter optimization. On a publicly accessible dataset, the suggested method is evaluated, and the top-performing models achieve accuracy levels of over 89%. After a thorough statistical examination of the optimizer results, the top-performing models were examined using explainable AI approaches to ascertain how features affected the top-performing model.

Another study [28], introduced an intelligent framework that mimics the intrusion detection system (IDS), which distinguishes between harmful and normal data requests from autonomous vehicles. To achieve this, the models were trained on a diverse set of attacks and simultaneously leveraged for classification using ensemble-based machine learning classifiers such as Decision Tree, Random Forest, Extra Trees, XGBoost, K-Nearest Neighbors, and Support Vector Machine (SVM). The suggested model is divided into several machine-learning stages, such as gathering data, pre-processing, and prediction. Lastly, they assessed the ensemble models using a variety of evaluation criteria, including f1-score, recall, accuracy, and precision. XGBoost achieved a high detection rate and low computing cost for the AV systems at the same time, outperforming other classifiers in terms of accuracy which is 98.57%.

Another study [29], proposed a Deep Learning Engine (DLE)-based artificial intelligence (AI)-based intrusion detection architecture for identifying and categorizing vehicle traffic in IoV networks into possible cyberattack categories. Additionally, rather than operating on the distant cloud, these DLEs will be installed on Multi-access Edge Computing (MEC) servers, taking into account the mobility of the vehicles and the real-time requirements of the IoV networks. The efficacy of the suggested system is demonstrated by extensive experimental results utilizing common assessment metrics and average prediction time on an MEC testbed.

Objectives:

The main objectives of our study are given below:

- To develop a robust and lightweight AI-based Intrusion Detection System (IDS) for in-vehicular communication networks.
- To leverage RNN for capturing temporal dependencies in CAN traffic data for accurate attack detection.
- To utilize LightGBM for efficient feature extraction from in-vehicle network data.
- To enhance detection accuracy and performance by combining RNN and LightGBM in a hybrid model.
- To ensure the IDS is deployable in resource-constrained automotive environments.
- To integrate the IDS into a Federated Learning (FL) framework for privacy-preserving and decentralized model training.
- To validate the proposed model's effectiveness through comparative evaluation against existing state-of-the-art IDS approaches.

Novelty Statement:

The novelty of this study lies in the development of a hybrid AI-based Intrusion Detection System that combines Recurrent Neural Networks (RNN) and LightGBM for accurate detection of both known and unknown cyber-attacks in in-vehicle networks. Unlike existing approaches, the proposed system is specifically designed to address the critical challenges of deployment feasibility, computational efficiency, and robustness within resource-constrained automotive environments. Furthermore, its integration into a Federated Learning framework ensures secure, privacy-preserving model training across distributed vehicle nodes making it a cutting-edge solution for intelligent connected vehicles (ICVs).

The Proposed Approach:

This paper introduces an AI-driven approach for in-vehicle intrusion detection, aimed at identifying attacks targeting the CAN bus within the environment of Connected Autonomous Vehicles (CAVs). The proposed approach was designed using two of the AI-based models that are the RNN (recurrent neural network) which is a famous deep learning algorithm for dealing with sequential data and their interdependencies as well as the ML boosting technique called the LightGBM for the extraction of important features. The proposed approach is a hybrid deep learning framework, leveraging the strengths of both RNN and LightGBM, it was used for multi-classification as it can be used to detect different attacks by classifying them into their respective classes. The methodology flow diagram is shown in Figure 4 and the different phases of the proposed AI-based IDS framework are shown in Figure 5.

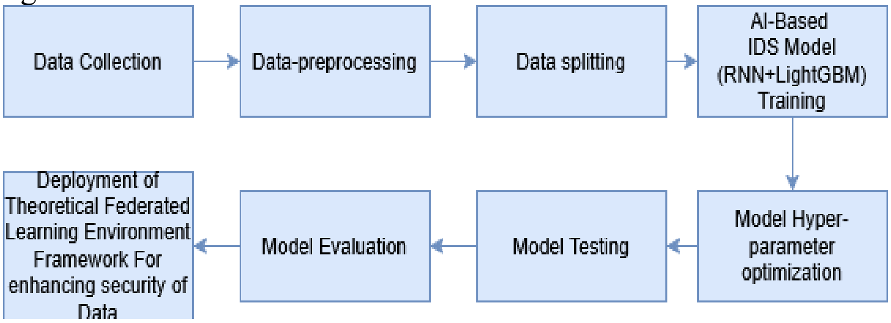

Data Collection
Data-preprocessing
Data splitting
AI-Based IDS Model (RNN+LightGBM) Training
Model Hyper-parameter optimization
Model Testing
Model Evaluation
Deployment of Theoretical Federated Learning Environment Framework For enhancing security of Data

Figure 4. Methodology Flow Diagram

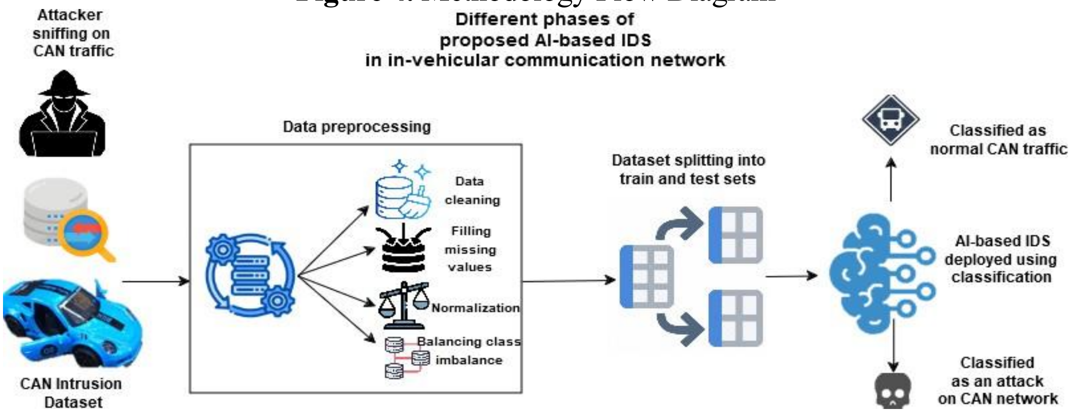

Different phases of proposed AI-based IDS in in-vehicular communication network
Attacker sniffing on CAN traffic
CAN Intrusion Dataset
Data preprocessing
Data cleaning
Filling missing values
Normalization
Balancing class imbalance
Dataset splitting into train and test sets
Classified as normal CAN traffic
AI-based IDS deployed using classification
Classified as an attack on CAN network

Figure 5. Different phases of the proposed AI-based IDS framework

Table 1. Literature review of different methods used by researchers for IDS in in-vehicle communication systems

Published Year	Author name	Methods Used	Dataset	Results	Limitations	Future Work
2023	Mohamed et al.,	Random Forest, Extreme Gradient Boosting, Categorical Boosting, Light Gradient Boosting Machine with hyper optimization, Z-score normalization for data-preprocessing. Regression model for feature selection.	CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019	99.8% accuracy	Not mentioned.	Focus on different datasets, deep reinforcement learning and transfer learning.
2022	Tejasvi et al.,	Three deep learning-based misbehavior classification schemes for intrusion detection in IoV networks using Long Short Term Memory (LSTM) and Convolutional Neural Networks (CNNs). single or multi-step classification done by deep learning models (deployed on the vehicular edge servers)	VeReMi Extension dataset	F1-scores ranging from 95.58% to 96.75%	Inter-classification due to similarities.	More vehicular misbehavior types.
2024	Muzun et al.,	An artificial neural network (ANN) in the first stage to detect seen attacks, and a Long Short-Term Memory (LSTM) autoencoder in the second stage to detect new, unseen attacks. IDS deployed in a hierarchical federated learning (H-FL) environment	Benchmark dataset published by Song et al (car hacking dataset).	F1-score: 99% (seen attacks) 95% (novel attacks). Detection rate of 99.99%. FAR 0.016%	Limited driving scenarios, requiring extensive datasets.	A realistic H-FL environment, adversarial attacks.
2023	Kabilan et al.,	Unsupervised method of intrusion detection for in-vehicle communication	ML350 in-vehicle intrusion dataset, WNS-	(75.51, 84.05, 60.63, 73.62, 74.83) %	Not find the type of attacks.	Semi-supervised and using generative AI.

		networks. Optimal feature extracting ability of autoencoders and more precise clustering using fuzzy C-means (FCM) combined.	DS, KDDCup, UNSW_NB15, (ISCX)	accuracy for dataset in order).		
2024	Erol et al.,	Novel Distributed Self-Supervised Federated Intrusion Detection Algorithm (DISFIDA), with Online Self-Supervised Federated Learning, that uses Dense Random Neural Networks (DRNN)	three open-access datasets	100% TPR, 99% TNR	Not mentioned.	A collaborative learning system for the best IDPS can be created by combining the intrusion judgments of DISA with a mitigation algorithm to lessen the destructive impacts of DoS attacks.
2024	Ykob & Joshua	(SHAP) for explaining decisions, hybrid bidirectional long-short-term memory with autoencoders (BiLAE) to reduce the dimensionality of IoV network traffic, Barnacle Mating Optimizer (BMO) for hper-parameter optimization of DL models like, s ResNet, Inception, Inception ResNet, and MobileNet Convolution neural network-transfer learning architecture (CNN-TL)	vehicle hacking dataset (for in vehicle communication), N-BaIoT (for external networks communication)	100% accuracy	Not mentioned.	To improve user privacy in the IoV network by integrating blockchain technology with the suggested explainable ensemble transfer learning model IDS architecture.
2022	Jianfeng et al.,	ConvLSTM-based IVN intrusion detection method. a proximal policy optimization (PPO)-based federated client selection (FCS) scheme	IVN datasets (attack-free dataset of CAN messages published by the HCR Lab of Korea University)	95% accuracy	Not mentioned.	Modeling the IoV as a multi-agent system to formulate the complex interactions among

		used for model optimization				multiple ICVs and the overall environment.
2020	Khadkha et al.,	ML and DL methods	Dataset created	AE 94%, RF96% accuracy	Not mentioned.	Not mentioned.
2024	Pavle et al.,	XGBoost, KNN	Publicly available dataset "can-dataset"	89% accuracy	No. of experiments constrained by the data's accessibility and updating time. Limited no. of optimizers in the comparative analysis due to high computational requirement	To look at how the modified optimizer might be used to solve problems in cybersecurity, medicine, and forecasting.
2022	Jay et al.,	Ensemble ML models: DT, RF, extra tree, XGBoost, KNN, SVM	(CAN-intrusion dataset) CICIDS2017, Dataset Generated.	98.57% accuracy	Not mentioned.	Improvise the security aspects of the proposed framework by analyzing modern-day attacks, such as malware attacks, replay, and Sybil attacks.

Table 2. Dataset features and their descriptions.

Data Features	Description
TS (Timestamp)	The time at which a can message is sent.
ID1 (CAN Identifier)	The ID or the type of the CAN message.
DL0-DL7 (data length code)	The data payload bytes indicate the existence of different information between the various components of the vehicle.
target	This can either be an attack or normal CAN traffic.

Table 3. Dataset distribution for the four target classes.

S.NO	Class Label	No. of Instances
1	Attack Free State (0)	3000
2	DoS Attack (1)	3000
3	Fuzzy Attack (2)	3000
4	Impersonation Attack (3)	3000
		Total instances=12000

CAN BUS:

Electronic control units are the electronic devices that manage the vehicle systems. The ECU is the central component of the engine management system, which regulates nearly all the electrical systems and operations in automobiles. Engine performance, control comfort, security features, airbag deployment, parking assistance, and ignition are some of the electrical components. A luxury car can have over 150 ECUs, while a typical car has an average of 40. A central bus known as the CAN bus connects these ECUs. ECUs can communicate with one another via CAN, a message-based broadcast communication protocol that adheres to the bus topology. The following are included in the standard CAN data: CAN ID (128 bits) - SOF (Start Of Frame) (1 bit), base identifier (11 bits), substitute remote request (SRR) (1 bit), identifier extension bit (IDE) (1 bit), extended identifier (18 bits), remote transmission request (RTR) (1 bit), reserved bits (2 bits), data length code (DLC) (4 bits), data (64 bits), cyclic redundancy check (CRC) (16 bits), ACK (2 bits), and EOF (7 bits). Inter-frame space (IFS), which has at least three consecutive bits, is the space between consecutive messages in a CAN bus. A typical Can frame is shown in Figure 6.

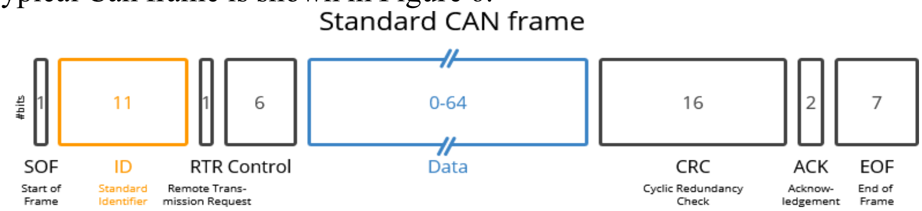


Figure 6. A standard CAN frame

CAN is widely used in most modern vehicles due to its cost-effectiveness and reliability. However, it lacks message authentication capabilities because there is limited space available to incorporate a message authentication code [30]. Through Bluetooth, Wi-Fi, and the On-board Diagnostics-II (OBD-II) port, the CAN messages from the CAN bus can be physically accessed. Additionally, this can be accessed remotely via a cellular or wireless connection. In their work on thorough experimental assessments of automobile attack surfaces, author demonstrate the Bluetooth attack on the CAN bus [31]. Once attackers gain access to the vehicle's network, they can manipulate CAN messages by adding or removing data, disrupting vehicle functionality, and executing denial-of-service (DoS) attacks, ultimately gaining control over the vehicle. Therefore, this CAN bus is a cost-effective and dependable method of package transportation, but it lacks security. A security mechanism must be computationally efficient, lightweight, and quick to forecast to be deployed in a CAN system [5]. In our proposed approach, the CAN data is extracted and preprocessed for effective ML-based intrusion detection of various attacks.

Dataset Description:

The CAN dataset that we used in this study is obtained from Kaggle (https://www.kaggle.com/datasets/bikashkundu/can-hcrl-otids) and is passed through pre-processing techniques for effective classification tasks done by ML models. The dataset basically consists of four target classes these being: DoS Attack (1), Fuzzy Attack (2), Impersonation Attack (3), and Attack Free State (0). An impersonation attack is a spoofing attack. The dataset comprises a total of 12,000 instances, with 3,000 samples allocated to each class as illustrated in Figure 11(a). The balancing of the dataset is done through the SMOTE (Synthetic Minority Over-sampling Technique) oversampling technique. There are a total of 11 features out of which one is the target column or label. The features are TS (Timestamp): the time at which a can message is sent, ID1 (CAN Identifier): which represents the ID or the type of the CAN message, each CAN message is associated with a specific CAN ID corresponding to a specific function of the vehicle, DL0-DL7 (data length code) these are the data payload bytes that indicate the existence of different information between the various components of the vehicle. The last column is the target label this can either be an attack or normal CAN traffic. These features and their details are listed in Table 2. The importance of the features in the dataset is shown in Figure 7. The distribution of the dataset for each class along with their no. of instances is shown in Table 3. Another dataset that we have used in our proposed approach is created by using a simulation environment known as the ICSim simulator. In this environment we have simulated the CAN traffic and conducted a variety of attacks on this traffic, these attacks being: Man in the middle attack MiMT, resource starvation, and injection. After this, the traffic is captured via the Wireshark tool. This traffic is then converted into a comma-separated file CSV containing both malicious and non-malicious CAN traffic [28]. This dataset is then passed through the different processes of the preprocessing pipeline. This dataset is used to increase the robustness of our proposed method of detecting unseen attacks. This dataset will only be used as a test set for testing the performance of the proposed framework. The correlation of the DLC features in the CAN dataset is shown in Figure 8 (a) and the distribution of the data payload by target classes is depicted in Figure 8 (b). Figures 9(a) and 9(b) display the distribution of CAN IDs across target classes and a Time Series (TS) versus Attack Classes plot, respectively. Additionally, Figure 10 presents a pair plot graph illustrating the relationships among the DLC features.

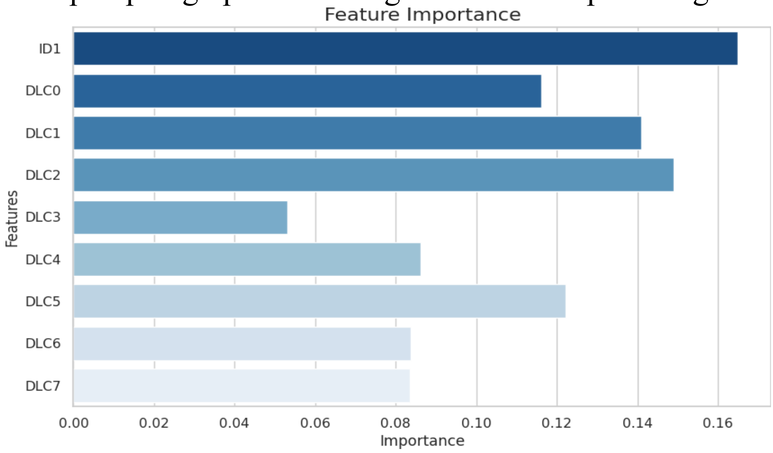
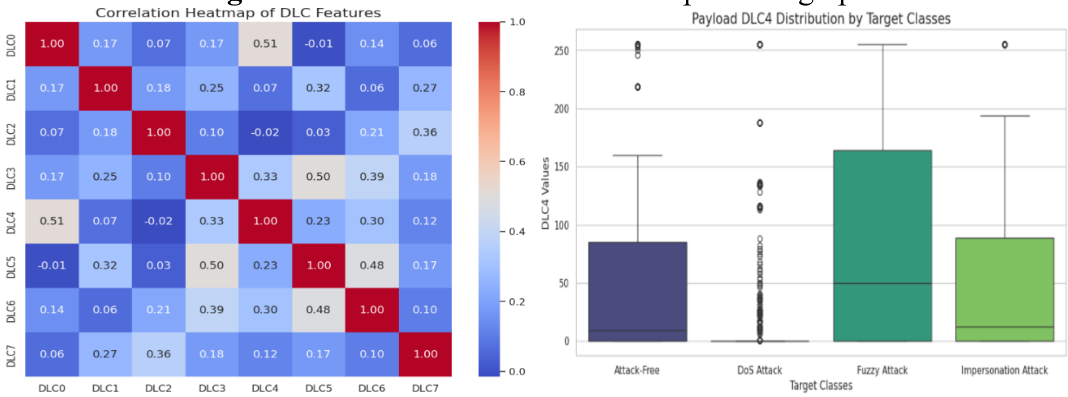


Figure 7. Dataset Features' Feature Importance graph.



	DLC0	DLC1	DLC2	DLC3	DLC4	DLC5	DLC6	DLC7
DLC0	1.00	0.17	0.07	0.17	0.51	-0.01	0.14	0.06
DLC1	0.17	1.00	0.18	0.25	0.07	0.32	0.06	0.27
DLC2	0.07	0.18	1.00	0.10	-0.02	0.03	0.21	0.36
DLC3	0.17	0.25	0.10	1.00	0.33	0.50	0.39	0.18
DLC4	0.51	0.07	-0.02	0.33	1.00	0.23	0.30	0.12
DLC5	-0.01	0.32	0.03	0.50	0.23	1.00	0.48	0.17
DLC6	0.14	0.06	0.21	0.39	0.30	0.48	1.00	0.10
DLC7	0.06	0.27	0.36	0.18	0.12	0.17	0.10	1.00

Figure 8(a). Correlation Heatmap of DLC features (b) Distribution of payload DLC4 by target classes.

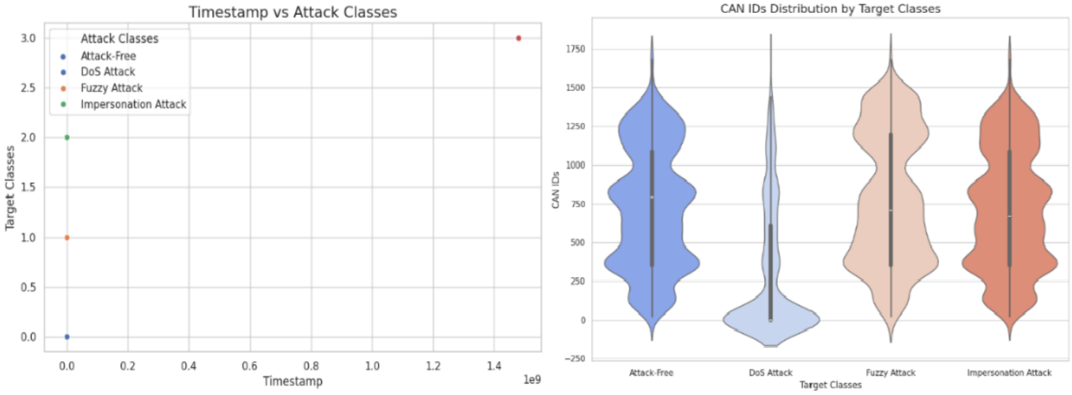


Figure 9(a). CAN IDS distribution by target classes (b) Time vs. Attack classes plot

Dataset Preprocessing:

The datasets were preprocessed to remove noise, missing values, and inconsistencies in the data and to clean them for further important tasks. The missing values in the data were filled by zero. The features were scaled using a Standard Scaler to enhance model performance. This process, known as data normalization, ensures that the data is constrained within a predefined range, facilitating more efficient and accurate learning by the model. Label encoding was employed to convert the categorical labels into numerical ones. The Kaggle CAN dataset was split in the ratio of, 60:20:20 that is, sixty percent of the data was used for training the models and the remaining forty percent was used for testing and validation twenty percent for each process. The dataset splitting ratio for training, validation, and testing sets is shown in Figure 11 (b). The training set was used for training the model and making it learn the relevant features of the data. The performance of the model was tested by testing it on the test set. The model's validation was done by utilizing the validation set.

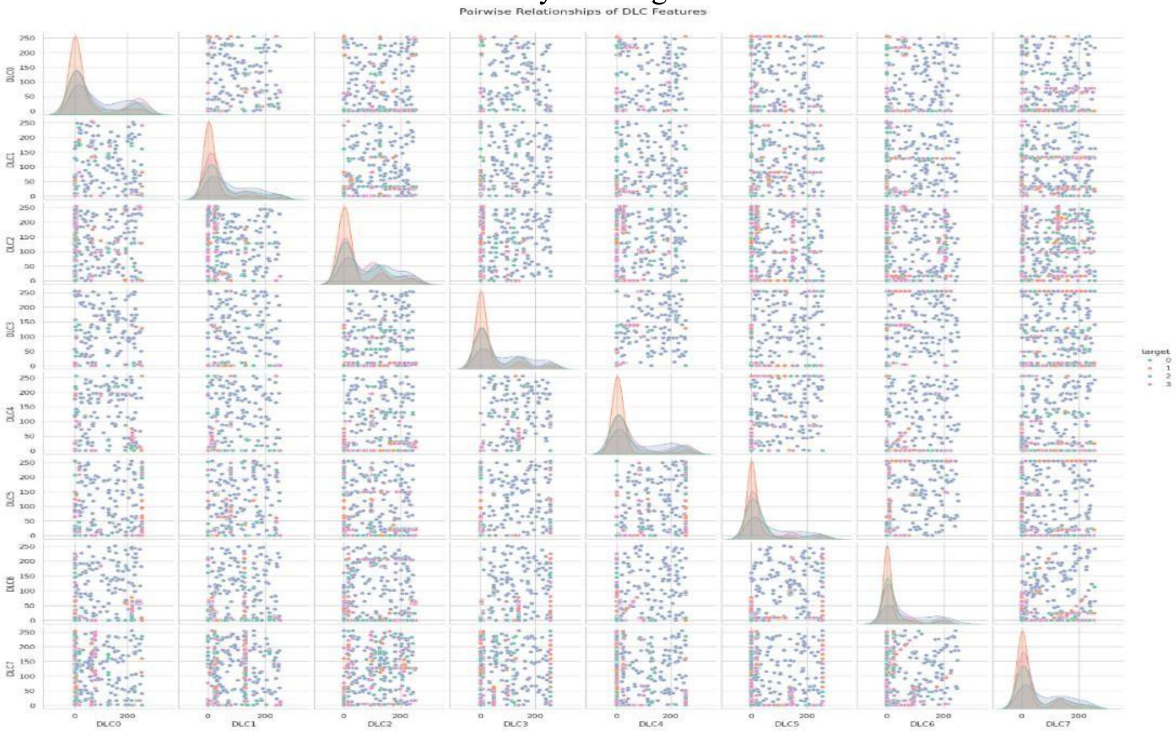


Figure 10. Pairplot showing pairwise relationships of DLC features of the CAN dataset

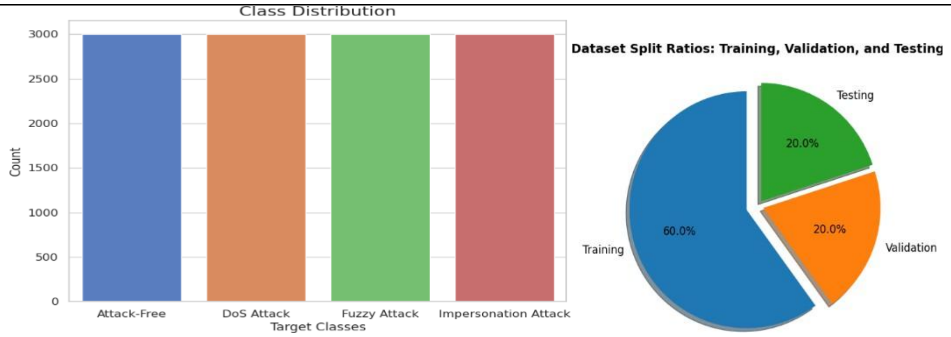


Figure 11(a). CAN Dataset distribution of instances for each class (b) CAN Dataset distribution for training, validation, and testing sets.

Feature Scaling using Standard Scaler:

For each feature (column) b in the dataset D' :

Calculate the mean:

$$\mu_b = \frac{1}{n}\sum_{i=1}^{n} D'_{ab} \quad (1)$$

Where 'n' is the number of rows.

Calculate the standard deviation:

$$\sigma_b = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (D'_{ab} - \mu_b)^2} \quad (2)$$

Create a new dataset D'' Where each element is scaled:

$$D''_{ab} = \frac{D_{ab} - \mu_b}{\sigma_b} \quad (3)$$

Attacks Considered in The Study:

There are mainly three types of attacks that are considered in this article for intrusion detection in the proposed AI-based IDS in in-vehicular communications. They are discussed below:

- **DoS Attack:** A denial-of-service attack aims to use up the CAN bus bandwidth by sending a lot of messages, which could cause unexpected system behavior. The attacker sends a lot of messages to the CAN bus with identifier = 0 since the identifier sets the message priority. The most important message will be this one [32].
- **Fuzzy Attack:** The CAN bus network system is compromised by an attacker who inserts random messages that seem like legal traffic. Attacks using frame fuzzification have the potential to compromise the ECUs and result in unexpected behavior in the car, such as automatic gear shifts, erratic signal light on/off switching, and shaking steering [33].
- **Impersonation / Spoofing Attack:** An unauthorized attacker attempts to inject fake messages to control specific functions by targeting specified CAN IDs. System failure results from the inability to discern between authentic and fake communications since the CAN IDs are spoofs and appear genuine [32].

All the above-mentioned three attacks are represented in Figure 12.

Model Training:

The proposed approach considers two models: Recurrent Neural Network (RNN) and LightGBM. Initially, both models are trained independently on the dataset, and their performance is assessed to evaluate their effectiveness. The RNN is used because it is good at handling sequential or time-series data which is the common nature of the Can bus data. The other model that is used is the LightGBM, it is an ML boosting technique that is good at extracting important features and improves performance due to its boosting nature. After this the combined strengths of both the models are deployed using a hybrid of the individual models and their performance is then evaluated and compared to the performance of the individual models. In the hybrid deep learning framework, the features (output class probabilities) selected by the LightGBM boosting technique and the temporal dependencies or patterns captured by the RNN are passed as inputs to the hybrid model for better prediction or classification results. The combination of the outputs of both these models results in a better overall performance of the hybrid deep learning approach.

RNN:

To develop a machine learning (ML) model that can generate sequential predictions or conclusions based on sequential inputs, a deep neural network known as a recurrent neural network, or RNN, is trained on sequential or time series data. Recurrent neural networks preserve information from earlier inputs by introducing a method where the output from one phase is given back as input to the next. Because of their design, RNNs are ideal for tasks where prior step context is crucial. This makes RNN well suited for the CAN data which is sequential. The general architecture of RNN is shown in Figure 12.

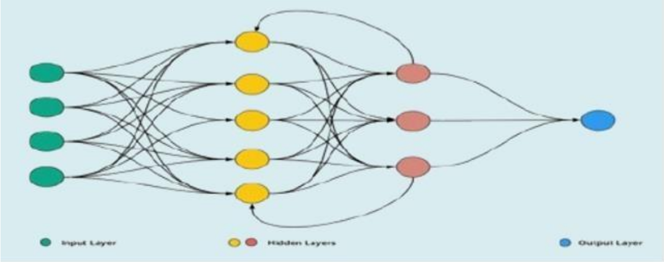


Figure 12. The architecture of a typical RNN model

Hidden State Update for RNN:

$$h_t = \phi(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (4)$$

Where:

'h_t' represents the hidden state at time t, 'x_t' represents the input at time t, 'h_{t-1}' represents the hidden state at the previous time step $(t - 1)$. Note that h_0 is typically initialized to a vector of zeros, 'W_{xh}' represents the weight matrix connecting the input to the hidden state, 'W_{hh}' represents the weight matrix connecting the previous hidden state to the current hidden state (the recurrent connection), 'b_h' represents the bias vector for the hidden state and'ϕ' is the activation function.

Output for RNN:

$$\begin{aligned} o_t &= W_{ho}h_t + b_o \\ y_t &= \sigma(o_t) \end{aligned} \quad (5)(6)$$

Where:

'o_t' shows the output before activation at time t, 'W_{ho}' shows the weight matrix connecting the hidden state to the output, 'b_o' shows the bias vector for the output, 'y_t' shows the final output at time t and 'σ' shows the activation function for the output.

Light GBM:

An ensemble learning framework called LightGBM, more precisely a gradient boosting technique, builds a strong learner by gradually adding weak learners in a gradient descent fashion. Light Gradient Boosting Machine Classifier is referred to as LGBMClassifier. For categorization, ranking, and other machine-learning tasks, it employs decision tree methods. The LGBMClassifier employs a novel technique called Exclusive Feature Bundling (EFB) and Gradient-based One-Side Sampling (GOSS) to accurately handle massive amounts of data while also speeding it up and using less memory.

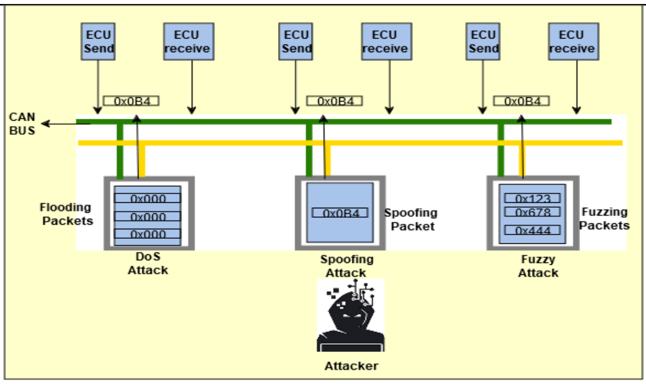


Figure 13. Three main cyber-attacks on the CAN BUS considered in the proposed study

Objective Function of LightGBM:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (7)$$

The Hybrid Model (RNN+LightGBM):

Our methodology suggests a hybrid deep learning approach where RNN is used to model temporal dependencies from sequential CAN traffic data and LightGBM is used to extract and classify based on the most relevant features of the CAN traffic. The outputs of both models, particularly the class probabilities, are combined to feed a final classifier, thus enhancing the predictive power. This fusion or combination basically involves the Concatenation of class probability vectors from RNN and LightGBM, which are then fed into a meta-classifier (i.e. another LightGBM) for the final prediction. This structure preserves the learned temporal and feature-based patterns and enables higher-level abstraction and learning. The workflow diagram showing the interaction of RNN and LightGBM is shown in Figure 14.

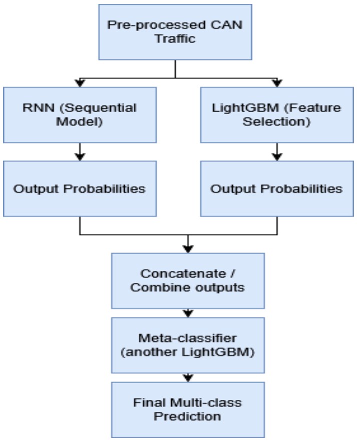

Pre-processed CAN Traffic
RNN (Sequential Model)
LightGBM (Feature Selection)
Output Probabilities
Output Probabilities
Concatenate / Combine outputs
Meta-classifier (another LightGBM)
Final Multi-class Prediction

Figure 14. Workflow Diagram showing the interaction between RNN and LightGBM

Model Hyper-parameter Optimization:

The hyper-parameters of the models are fine-tuned using the GridSearchCV algorithm that creates a detailed grid of the most efficient hyper-parameters of a model. Early stopping is employed in LightGBM to avoid overfitting. In the case of RNN and the hybrid model Batch normalization and Adam optimizer are used to avoid overfitting and optimization. Sparse Categorical Cross-entropy is deployed as a Loss Function. The hyper-parameters deployed for the three models are given in Table 4.

Table 4. Hyper-parameters of the deployed three models

Model LightGBM		**Model RNN**		**Model Hybrid (LightGBM + RNN)**	
Hyper-parameters	**Values**	**Hyper-parameters**	**Values**	**Hyper-parameters**	**Values**

learning-rate	0.1	epochs	50	Activation function	rel
num-leaves	31	batch-size	32	optimizer	Adam
boosting-type	but	Loss Function	sparse_categorical_crossentropy	learning-rate	0.001
n-estimators	300	optimizer	Adam	epochs, batch-size	70, 64

Model Testing:

The models were tested on the test set in this phase. First, the individual models were evaluated separately on the two datasets. Subsequently, the hybrid model was evaluated on each dataset individually to highlight its effectiveness and to compare its performance against that of the standalone models. If the performance of the pre-trained models on the created dataset was nearly equal to their performance on the training set, then it means that the models are performing well and are generalizing significantly. This also demonstrates that the models were robust in detecting previously unseen attacks, reinforcing the effectiveness and significance of the proposed approach.

Model Evaluation:

Evaluation metrics are numerical measurements that are used to evaluate a machine learning model's efficacy and performance. These metrics aid in comparing various models or algorithms and offer insights into how well the model is operating. The performance of the models is evaluated on the basis of basic evaluation metrics like confusion matrix, accuracy, precision, recall, F1-score, and ROC curve. The detailed explanation of these metrics is as follows:

- **Confusion matrix:** A table summarizing a classification model's performance. It's very helpful for displaying the predicted versus real (genuine) results. True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) are the four main components of the confusion matrix.
- **Accuracy:** The proportion of correctly categorized instances among all instances is known as accuracy.

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN) \quad (8)$$

- **Precision:** The precision metric quantifies the percentage of predicted positive instances that turn out to be positive.

$$\text{Precision} = TP / (TP+FP) \quad (9)$$

- **Recall:** The percentage of real positive instances that the model accurately predicts is known as recall.

$$\text{Recall} = TP / (TP+FN) \quad (10)$$

- **F1-score:** The harmonic mean of recall and precision is known as the F1-Score. The harmonic mean is more sensitive to low values than a standard average.

$$\text{F1 score} = (2\times\text{Precision}\times\text{Recall}) / (\text{Precision}+\text{Recall}) \quad (11)$$

- **ROC curve:** A graph showing the performance of a classifier across all potential classification thresholds.

Theoretical Federated Learning Environment Framework for Enhancing Security of Data:

Training a Deep learning model is typically a computationally complex process that requires a vast amount of data and resources. The requirements are nearly impossible for an individual ICV (intelligent connected vehicle) to meet. The uRLLC in IoV can scarcely be assured since wireless data uploading for centralized Deep neural network model training in the cloud has the risk of privacy leakage and excessive access latency [34]. Federated learning (FL), which offers the benefits of privacy preservation and safe multi-party computation, is regarded as a viable alternative in such a scenario [30][31]. Typically, a FL framework operates in either peer-to-peer or client-server mode. The client-server FL paradigm is widely used in IoV, with mobile edge computing (MEC) servers linked to base stations (BSs) serving as the parameter servers and ICVs as the clients. The clients store their datasets and train the models locally within such a framework. The model is transmitted to the parameter server for global aggregation following each local training cycle [35]. Model parameters are uploaded in place of large amounts of raw data to relieve the strain on communication bandwidth, and FL allows ICVs to store their data to prevent leaks. The proposed FL framework is shown in Figure 15 and works in the following steps:

- **Proposed Hybrid Model Initialization:** For every vehicle series, a hybrid model is initialized in the cloud as the global model. The global hybrid model is then downloaded to the MEC server by the cloud. The server receives FL request data from the candidate vehicles.
- **Federated Client Selection:** An acknowledgment is sent in response, to the candidate automobiles of the same vehicle series. The candidates are then downloaded to the global hybrid model. Following training, the model parameters of the candidate vehicles are uploaded to the MEC server. The required vehicles are subsequently chosen from the candidates by the MEC server. The MEC server sends rejection notices to the vehicles that were not chosen and acceptance notices to the vehicles that were chosen.
- **Local Model Training:** The global hybrid model is downloaded to the chosen vehicle clients by the MEC server. Using their datasets, the clients carry out the local training. The clients update the model parameters and send them to the MEC server after local training is finished.
- **Global Model Update by Parameter Aggregation:** The parameter aggregation at the MEC server updates the global model. After gathering the updated model parameters from each of the selected clients, each round of parameter aggregation is carried out.
- **Local Model Update:** The vehicles receive the updated global model following the global parameter aggregation at the MEC server. With the latest release, automobiles update their models. Until the loss function converges or the iteration hits the upper limit, the local model training and parameter aggregation process is repeated. Lastly, the authenticated vehicles receive the convergent global hybrid model. A copy of the global hybrid model is transmitted to the cloud in the meantime.

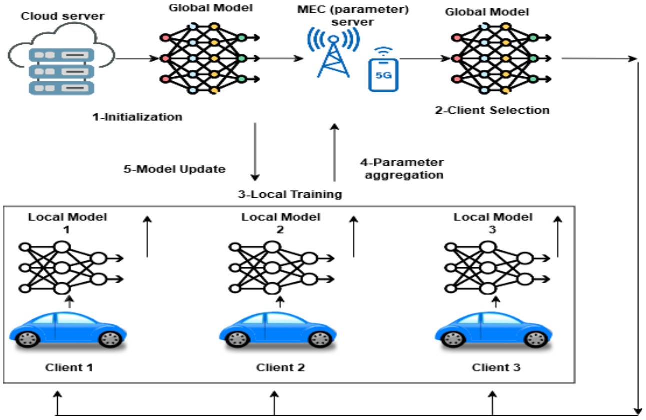


Figure 15. Proposed Federated Learning Framework for Proposed Hybrid Model Training

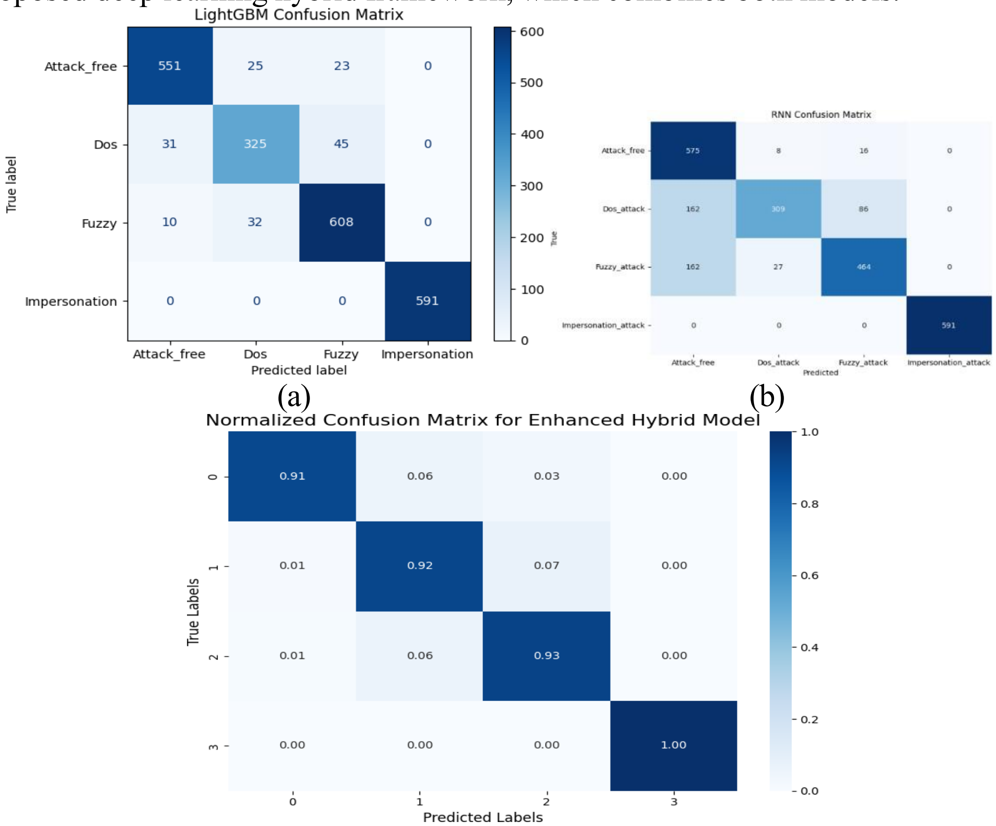
Experimental Setup:

The training and testing of the AI-based models considered in the proposed approach is conducted on Colab (a Google Research web editor that lets users write and execute any Python code directly from the browser) using its GPU and the Python language on an Intel(R)

Core(TM) i3-4030U CPU @ 1.90GHz computer with an 8.00 GB RAM and 64-bit Windows 10 OS (operating system).

Results and Discussion:

This section presents a discussion of the results obtained from evaluating the three models used in the study after their deployment on the datasets. The effectiveness of the proposed model is illustrated through visualizations of various evaluation metrics. Figures 16(a), (b), and (c) display the confusion matrices of the three models: LightGBM, RNN, and the proposed deep learning hybrid framework, which combines both models.



(a) (b)

(c)

Figure 16(a). confusion matrix for LightGBM (b) confusion matrix for RNN (c) confusion matrix for proposed Hybrid Model

The LightGBM model correctly predicted 551 instances for Attack-free class, 325 instances for the DoS attack, 608 instances for Fuzzy-attack and 591 instances for the Impersonation attack out of a total of 599, 401, 653, and 591 instances for each class respectively. The LightGBM model demonstrated strong overall performance; however, it showed slightly lower accuracy in distinguishing between the Attack-free and DoS-attack classes. RNN model correctly predicted 575 instances for Attack-free class, 309 instances for the DoS attack, 464 instances for Fuzzy-attack and 591 instances for the Impersonation attack out of a total of 599, 401, 653, and 591 instances for each class respectively. The RNN model performed well in classifying the Attack-free and Impersonation-attack classes; however, its accuracy declined for the remaining two classes, particularly the Fuzzy-attack class. The hybrid model achieved classification accuracies of 91%, 92%, 93%, and 100% across the four specified classes. It outperformed the other two models, delivering the highest accuracy in each class.

The training and validation, accuracy, and loss graphs for the RNN model are given in Figure 17 (a) and (b). These figures show the overall good performance of the model.

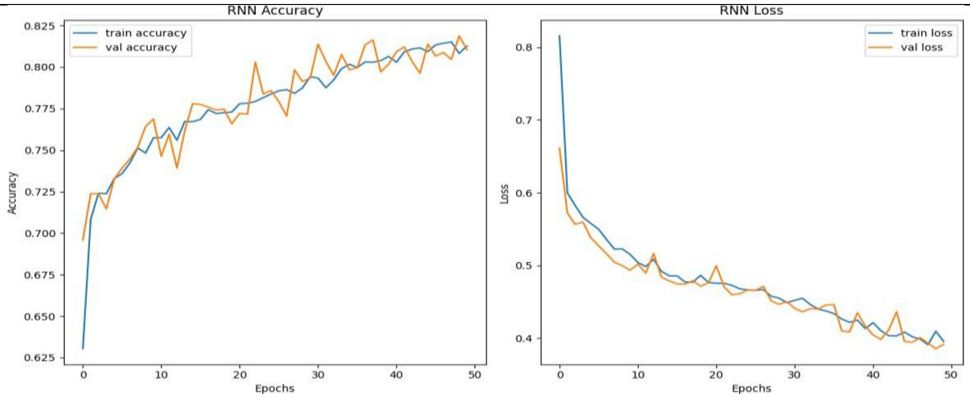


Figure 17(a). training and validation accuracy for RNN (b) training and validation loss for RNN

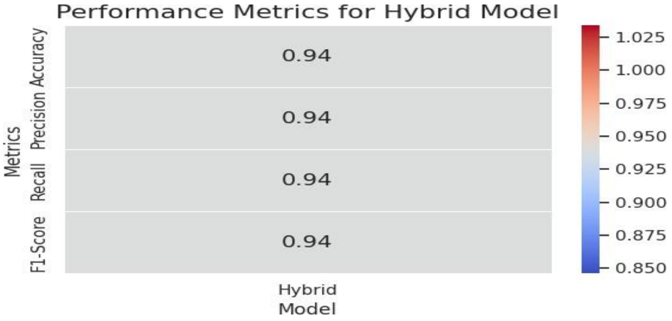


Figure 18. Evaluation metrics graph for the proposed Hybrid model

The significance of the performance of the proposed hybrid model is shown in Figure 18. The model has amazing accuracy, precision, recall, and F1-score of 0.94. These results indicate that the proposed hybrid model is highly effective in classifying CAN traffic into the appropriate categories, whether identifying an Attack-free state or detecting specific types of attacks outlined in the study. The superior performance of the hybrid model compared to the individual models is reflected in the evaluation metrics comparison graphs shown in Figures 19(a, b) and 20(a, b).

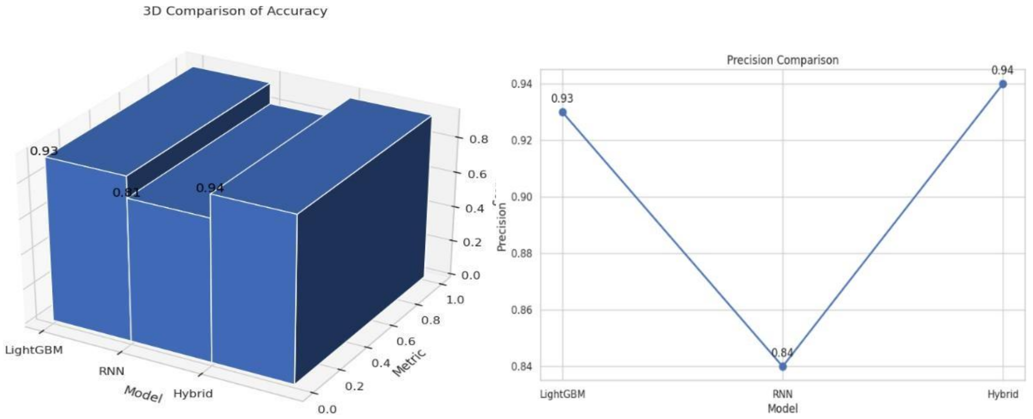


Figure 19(a). Precision comparison graph of three models (b) Accuracy comparison graph for three models

Figures 19(a) and (b) illustrate that the proposed hybrid model outperformed the other two models, achieving both precision and accuracy scores of 0.94. In comparison, the LightGBM model attained a precision and accuracy of 0.93, while the RNN model lagged with a precision of 0.84 and an accuracy of 0.81.

Figures 20 (a) and (b) show that the proposed model is performing better with a recall and an F1-score of 0.94 than the other two models, that is LightGBM which has a recall of

0.92 and an F1-score of 0.93 and the RNN model with a recall of 0.81 and an F1-score of 0.80.

The superior performance of the proposed hybrid model can be attributed to its ability to combine the strengths of both underlying models. This advantage has been validated by experimental results.

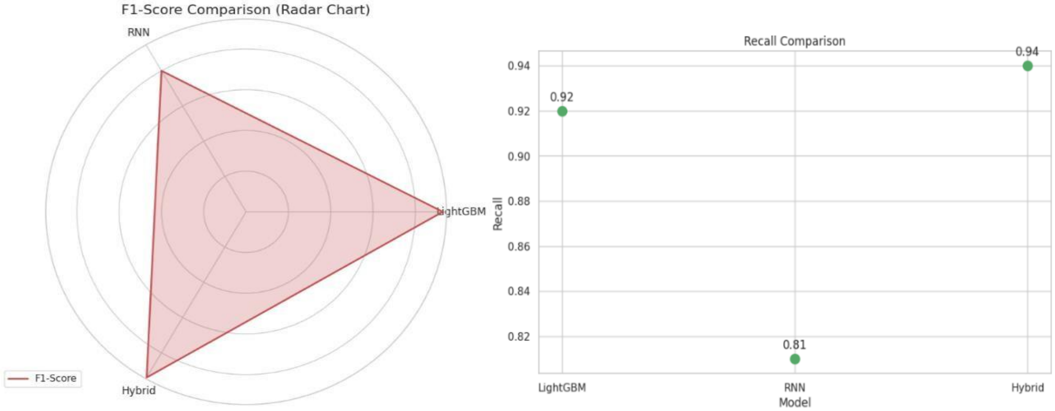


Figure 20(a). Recall the comparison graph of three models (b) F1-score comparison graph for three models

The various performance evaluation metrics and their values for the three models i.e. LightGBM, RNN, and the proposed Hybrid model are given in Table 5 below:

Table 5. Various Performance Evaluation Metrics for the three models

Model	Accuracy	Precision	Recall	F1-score
LightGBM	0.93	0.93	0.92	0.93
RNN	0.81	0.84	0.81	0.80
Proposed Hybrid Model	0.94	0.94	0.94	0.94

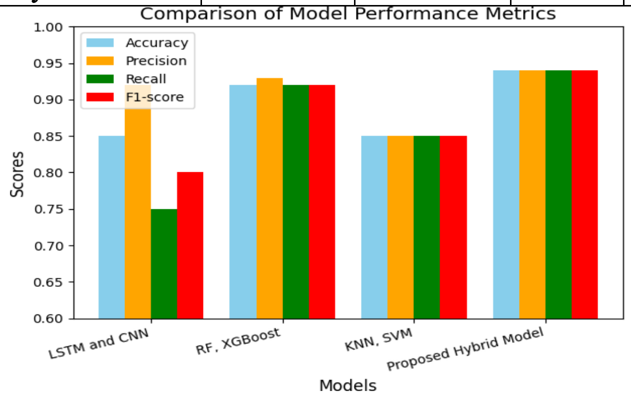


Figure 21. Evaluation metrics comparison of the Proposed Hybrid Model with the existing state-of-the-art approaches.

The proposed hybrid model was compared with existing state-of-the-art approaches used by various researchers mentioned in the literature. The result of this comparison is shown in Table 6 and in Figure 21. The results indicate that the proposed hybrid model delivers the highest accuracy of 94%, outperforming all other compared approaches.

Table 6. Performance comparison table showing enhanced performance of proposed model compared to existing state-of-the-art approaches

Model	Accuracy	Precision	Recall	F1-score
LSTM and CNN	0.85	0.92	0.75	0.80
RF, XGBoost	0.92, 0.92	0.92, 0.93	0.92, 0.92	0.92, 0.92
KNN, SVM	0.85, 0.84	0.85, 0.84	0.85, 0.84	0.85, 0.84
Proposed Hybrid Model	0.94	0.94	0.94	0.94

Conclusion:

To enhance the security of CAN traffic in in-vehicular networks (IVNs) and protect against various cyber threats, we have proposed a novel AI-based intrusion detection system (IDS) capable of detecting both known and unknown attacks. The presented deep learning hybrid framework leverages the strengths of two AI models: the LightGBM boosting technique for efficient feature extraction and the RNN for capturing temporal dependencies in CAN data. The models were trained and evaluated on a publicly available CAN dataset from Kaggle and further tested on a custom dataset we created to enhance the robustness of our approach. Evaluation results demonstrate that the proposed hybrid model outperforms individual models, achieving impressive scores of 0.94 for accuracy, precision, recall, and F1-score. Comparatively, our model surpasses existing state-of-the-art methods, including LSTM and CNN (accuracy 0.85), Random Forest and XGBoost (0.92, 0.92), and KNN and SVM (0.85, 0.84). To further strengthen the security and privacy of the proposed system, we have also outlined a theoretical Federated Learning (FL) framework for the decentralized deployment of the hybrid model. However, one limitation is the high resource cost of running an AI-based IDS continuously.

Future recommendations:

In future work, we aim to address this by implementing an event-triggered monitoring system where the IDS remains in a low-power state and activates only upon detecting anomalies via a lightweight detection model. Additionally, we plan to incorporate detection of modern attack types such as DDoS, malware, Sybil, and replay attacks, and to explore the practical deployment of the FL-based framework.

Acknowledgment: All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

References:

[1] G. Dimitrakopoulos and P. Demestichas, “Intelligent transportation systems: Systems based on cognitive networking principles and management functionality,” *IEEE Veh. Technol. Mag.*, vol. 5, no. 1, pp. 77–84, Mar. 2010, doi: 10.1109/MVT.2009.935537.

[2] M. S. Korium, M. Saber, A. Beattie, A. Narayanan, S. Sahoo, and P. H. J. Nardelli, “Intrusion detection system for cyberattacks in the Internet of Vehicles environment,” *Ad Hoc Networks*, vol. 153, p. 103330, Feb. 2024, doi: 10.1016/J.ADHOC.2023.103330.

[3] “Intelligent Transportation System(ITS): Concept, Challenge and Opportunity | IEEE Conference Publication | IEEE Xplore.” Accessed: Jun. 17, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/7980336

[4] G. Abdelkader, K. Elgazzar, and A. Khamis, “Connected Vehicles: Technology Review, State of the Art, Challenges and Opportunities,” *Sensors 2021, Vol. 21, Page 7712*, vol. 21, no. 22, p. 7712, Nov. 2021, doi: 10.3390/S21227712.

[5] K. N, V. Ravi, and V. Sowmya, “Unsupervised intrusion detection system for in-vehicle communication networks,” *J. Saf. Sci. Resil.*, vol. 5, no. 2, pp. 119–129, Jun. 2024, doi: 10.1016/J.JNLSSR.2023.12.004.

[6] “Exploring Controller Area Networks | USENIX.” Accessed: Jun. 17, 2025. [Online]. Available: https://www.usenix.org/publications/login/dec15/foster

[7] M. D. Pesé, J. W. Schauer, J. Li, and K. G. Shin, “S2-CAN: Sufficiently Secure Controller Area Network,” *ACM Int. Conf. Proceeding Ser.*, pp. 425–438, Dec. 2021, doi: 10.1145/3485832.3485883;PAGE:STRING:ARTICLE/CHAPTER.

[8] A. Barati, A. Movaghar, and M. Sabaei, “Energy Efficient and High Speed Error Control Scheme for Real Time Wireless Sensor Networks,” *Int. J. Distrib. Sens. Networks*, vol. 2014, 2014, doi: 10.1155/2014/698125.

[9] M. Althunayyan, A. Javed, and O. Rana, "A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning," *Veh. Commun.*, vol. 49, p. 100837, Oct. 2024, doi: 10.1016/J.VEHCOM.2024.100837.

[10] S. Checkoway *et al.*, "Comprehensive Experimental Analyses of Automotive Attack Surfaces".

[11] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom, "Survey of automotive controller area network intrusion detection systems," *IEEE Des. Test*, vol. 36, no. 6, pp. 48–55, Dec. 2019, doi: 10.1109/MDAT.2019.2899062.

[12] C. W. Axelrod, "INTEGRATING IN-VEHICLE, VEHICLE-TO-VEHICLE, AND INTELLIGENT ROADWAY SYSTEMS," *Int. J. Des. Nat. Ecodynamics*, vol. 13, no. 1, pp. 23–38, Jan. 2018, doi: 10.2495/DNE-V13-N1-23-38.

[13] S. Garg, K. Kaur, G. Kaddoum, S. H. Ahmed, and D. N. K. Jayakody, "SDN-Based Secure and Privacy-Preserving Scheme for Vehicular Networks: A 5G Perspective," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8421–8434, Sep. 2019, doi: 10.1109/TVT.2019.2917776.

[14] J. Ondruš, E. Kolla, P. Vertal', and Ž. Šarić, "How Do Autonomous Cars Work?," *Transp. Res. Procedia*, vol. 44, pp. 226–233, Jan. 2020, doi: 10.1016/J.TRPRO.2020.02.049.

[15] C. F. Cheng, G. Srivastava, J. C. W. Lin, and Y. C. Lin, "Fault-Tolerance Mechanisms for Software-Defined Internet of Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3859–3868, Jun. 2021, doi: 10.1109/TITS.2020.3043729.

[16] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A Survey on Security and Privacy of 5G Technologies: Potential Solutions, Recent Advancements, and Future Directions," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 196–248, Jan. 2020, doi: 10.1109/COMST.2019.2933899.

[17] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," *Proc. - IEEE Symp. Secur. Priv.*, pp. 447–462, 2010, doi: 10.1109/SP.2010.34.

[18] "Jeep hackers at it again, this time taking control of steering and braking systems | The Verge." Accessed: Jun. 17, 2025. [Online]. Available: https://www.theverge.com/2016/8/2/12353186/car-hack-jeep-cherokee-vulnerability-miller-valasek

[19] "OwnStar Device Can Remotely Locate, Unlock, and Start GM Cars | Threatpost." Accessed: Jun. 17, 2025. [Online]. Available: https://threatpost.com/ownstar-device-can-remotely-locate-unlock-and-start-gm-cars/114042/

[20] "New Vehicle Security Research by KeenLab: Experimental Security Assessment of BMW Cars | Keen Security Lab Blog." Accessed: Jun. 17, 2025. [Online]. Available: https://keenlab.tencent.com/en/2018/05/22/New-CarHacking-Research-by-KeenLab-Experimental-Security-Assessment-of-BMW-Cars/

[21] "Tencent Keen Security Lab: Experimental Security Assessment on Lexus Cars | Keen Security Lab Blog." Accessed: Jun. 17, 2025. [Online]. Available: https://keenlab.tencent.com/en/2020/03/30/Tencent-Keen-Security-Lab-Experimental-Security-Assessment-on-Lexus-Cars/

[22] T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "A deep learning based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems," *Digit. Commun. Networks*, vol. 9, no. 5, pp. 1113–1122, Oct. 2023, doi: 10.1016/J.DCAN.2022.06.018.

[23] E. Gelenbe, B. C. Gül, and M. Nakıp, "DISFIDA: Distributed Self-Supervised Federated Intrusion Detection Algorithm with online learning for health Internet of Things and Internet of Vehicles," *Internet of Things*, vol. 28, p. 101340, Dec. 2024, doi: 10.1016/J.IOT.2024.101340.

[24] Y. K. Saheed and J. E. Chukwuere, "XAIEnsembleTL-IoV: A new eXplainable Artificial Intelligence ensemble transfer learning for zero-day botnet attack detection in the Internet of Vehicles," *Results Eng.*, vol. 24, Dec. 2024, doi: 10.1016/J.RINENG.2024.103171.

[25] J. Yang, J. Hu, and T. Yu, "Federated AI-Enabled In-Vehicle Network Intrusion Detection for Internet of Vehicles," *Electron. 2022, Vol. 11, Page 3658*, vol. 11, no. 22, p. 3658, Nov. 2022, doi: 10.3390/ELECTRONICS11223658.

[26] A. Khadka, P. Karypidis, A. Lytos, and G. Efstathopoulos, "A benchmarking framework for cyber-attacks on autonomous vehicles," *Transp. Res. Procedia*, vol. 52, pp. 323–330, Jan. 2021, doi: 10.1016/J.TRPRO.2021.01.038.

[27] P. Dakic *et al.*, "Intrusion detection using metaheuristic optimization within IoT/IIoT systems and software of autonomous vehicles," *Sci. Rep.*, vol. 14, no. 1, p. 22884, Dec. 2024, doi: 10.1038/S41598-024-73932-5;SUBJMETA=1041,1042,117,639,705;KWRD=APPLIED+MATHEMATICS,COMPUTATIONAL+SCIENCE,COMPUTER+SCIENCE.

[28] S. . Satheeskumaran, Y. Zhang, V. E. Balas, T. Hong, and D. Pelusi, "Intelligent computing for sustainable development : first International Conference, ICICSD 2023, Hyderabad, India, August 25-26, 2023, revised selected papers. Part I," 2024.

[29] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "Artificial Intelligence (AI)-Empowered Intrusion Detection Architecture for the Internet of Vehicles," *IEEE Wirel. Commun.*, vol. 28, no. 3, pp. 144–149, Jun. 2021, doi: 10.1109/MWC.001.2000428.

[30] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed Federated Learning for Ultra-Reliable Low-Latency Vehicular Communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Feb. 2020, doi: 10.1109/TCOMM.2019.2956472.

[31] Di. M. Manias and A. Shami, "Making a Case for Federated Learning in the Internet of Vehicles and Intelligent Transportation Systems," *IEEE Netw.*, vol. 35, no. 3, pp. 88–94, May 2021, doi: 10.1109/MNET.011.2000552.

[32] T. N. Hoang and D. Kim, "Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders," *Veh. Commun.*, vol. 38, p. 100520, Dec. 2022, doi: 10.1016/J.VEHCOM.2022.100520.

[33] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," *Proc. - 2017 15th Annu. Conf. Privacy, Secur. Trust. PST 2017*, pp. 57–66, Sep. 2018, doi: 10.1109/PST.2017.00017.

[34] J. Zhang and K. B. Letaief, "Mobile Edge Intelligence and Computing for the Internet of Vehicles," *Proc. IEEE*, vol. 108, no. 2, pp. 246–261, Feb. 2020, doi: 10.1109/JPROC.2019.2947490.

[35] H. Brendan McMahan, E. Moore, D. Ramage, S. Hampson, and B. Agüera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," *Proc. 20th Int. Conf. Artif. Intell. Stat. AISTATS 2017*, Feb. 2016, Accessed: Jun. 17, 2025. [Online]. Available: https://arxiv.org/pdf/1602.05629