EAI Endorsed Transactions
on Internet of Things
Research Article EAI.EU

Advancing Vehicle Security: Deep Learning based Solution for Defending CAN Networks in the Internet of Vehicles

Kiran Aswal[1, *], Heman Pathak[2]

[1, 2] Dept. of Computer Science, Gurukul Kangri Viswavidyalaya, Haridwar, Uttarakhand, India;

Abstract

The Internet of Vehicle (IoV) is revolutionizing the automobile sector by allowing vehicles to interact with one another and with roadside infrastructure. The Controller Area Network (CAN) is a vital component of such smart vehicles, allowing communication between various Electronic Control Units (ECUs). However, the CAN protocol's intrinsic lack of security renders it opens to a variety of cyber-attacks, posing substantial hazards to both safety and privacy.

In particular, the CAN protocol lacks built-in authentication and encryption mechanisms, making it highly vulnerable to a range of sophisticated attacks. These include message spoofing, where attackers can inject malicious commands into the network, and replay attacks, which reuse legitimate communication to deceive vehicle systems. The broadcast nature of CAN also makes it susceptible to denial-of-service (DoS) attacks that can disrupt vehicular communication, significantly impacting system performance and safety. Traditional security solutions are often ill-suited for the real-time, resource-constrained environment of IoV, necessitating more advanced, data-driven defense mechanisms.

This research investigates the use of deep learning with multi-layer perceptron to improve the security of CAN networks inside the IoV framework. We discuss current threats to CAN networks, including spoofing, replay, and denial-of-service attacks, and how deep learning may be used to identify and mitigate these threats efficiently. We propose a unique deep learning-based defense mechanism for real-time threat detection.

The suggested method is highly effective in identifying and mitigating potential risks, as evidenced by extensive testing on real-world CAN datasets. Based on our findings, the proposed solution has the potential to considerably enhance the security of CAN networks in the Internet of Vehicles, making car communication systems more secure and reliable.

Keywords: Autonomous vehicle, Internet of Vehicles (IoV), Controller Area Network (CAN), cybersecurity, deep learning, vehicular communication systems.

Received on 21 04 2024, accepted on 03 09 2024, published on 22 10 2024

Copyright © Aswal et al., licensed to EAI. This is an open access article distributed under the terms of the CC BY-NC-SA 4.0, which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetiot.6523

*Corresponding author. Email: phd.kiranaswal@gmail.com

1. Introduction

With the latest technological developments, autonomous vehicles (AVs) that were formerly deemed science fiction have become a reality. Despite the fact that it is still in its early stages of development, the concept of autonomous vehicles is gaining global acceptance [1]. Autonomous cars are capable of

sensing their environment and operating independently of people. A passenger is not required to drive the automobile at any time, nor is their presence within the vehicle required. A smart vehicle can go anywhere a conventional car can go and accomplish all the functions carried out by a competent human driver.

According to the Society of Automotive Engineers (SAE), automation in autonomous cars can be categorized into six separate levels, ranging from SAE Level 0 (fully manual) to SAE Level 5 (entirely autonomous) [2], [3]. Table 1 gives a description of these levels.

Table 1. SAE level of driving automation

SAE Level	Description
0	The driver is the one who controls the entire vehicle. In the form of alerts, such as lane departure or blind spot warnings, driver aid is offered.
1	With one autonomous function to help, the driver has complete control over the car. Adaptive cruise control, for example, uses automated acceleration and braking to maintain a safe distance from oncoming traffic. Alternatively, automated steering can be used, which involves the assistance of lane centering and other features to keep the car moving at a consistently high speed.
2	The driver has complete control over how the vehicle performs, with assistance from two automated operations such as steering, braking, and acceleration.
3	The car may function autonomously under a set of predetermined configurations, and the driver can take control of the vehicle at any time.
4	The vehicle may operate autonomously under specified settings, eliminating the need for the driver to oversee it. The car is extremely close to being totally autonomous.
5	At this level, the car is supposed to be completely autonomous and capable of operating without restrictions. There is no need for the driver to supervise it.

Although customers are not yet able to acquire fully autonomous vehicles, we are already in the phase of partially automated automobiles [3]. The Internet of Vehicles (IoV), an interconnected network of autonomous vehicles, roadside infrastructure, and components that communicate and interact with one another using wireless technology, is derived from the Internet of Things (IoT), with the objective of improving the effectiveness, efficiency, and safety of autonomous vehicles. [4], [5].

The electrical and electronic system of a smart car is a scattered and complicated network of Electronic Control Units (ECUs), sensors, and actuators. ECUs, which are computing units, are required to operate a specific subsystem and make critical autonomous driving decisions. They must interact with one another and exchange sensitive data using a set of standard protocols. The CAN bus is regarded as the de facto standard for the in-vehicle communication network, and it is ubiquitously used in almost all automobiles [1] [6] [7].

On the opposite side, due to broadcast transmission; ID-based priority of the messages; lack of authentication, and encryption mechanisms, make it vulnerable to various security attacks [6]. It is also reported by researchers in their findings that vulnerabilities of CAN protocol can be exploited by hackers to launch Fuzzy, spoofing, DOS, or impersonation attacks on autonomous vehicles [1], [6], [7], [8]. These attacks may push the driver, co-passengers, or others who are on the road in a life-threatening situation. The widespread adoption of IoV depends on the way these issues also are addressed. Therefore, this article particularly focuses on the security and privacy issues in IoV which are raised due to the unique characteristics of CAN bus. The article proposes a novel and intelligent, intrusion detection solution to defend the CAN bus of autonomous vehicles from malicious attacks. The major contributions of this article are outlined as follows.

Highlight and discuss attack surfaces and potential security risks to the CAN bus network.

i. Introducing a cutting-edge deep learning-based intrusion detection model designed to enhance the security of the CAN bus. This innovative solution not only detects but also classifies malicious attacks with unparalleled efficiency, ensuring robust protection for vehicular communication systems against cyber threats.

ii. Presents a comprehensive comparative analysis to rigorously evaluate the proposed defense solution against benchmark systems and other related studies.

The remaining sections of this article are organized as follows. Section 2 introduces the relevant background knowledge. Section 3 discusses related work and their limitations. In Section 4, we describe the specific design details of our intrusion detection model, while the performance evaluation is shown in Section 5, followed by the conclusion in Section 6.

2. Background Knowledge

The power train, chassis & safety, body & comfort, and telematics & infotainment domains are the four main segments of a smart vehicle's internal communication system [8]. Real-time responsiveness is necessary for the Powertrain domain, which manages every aspect of engine and transmission operations. The airbag control, anti-lock braking, suspension, and Advanced Driver Assistance System, which performs real-time, safety-critical operations are included in the Chassis & Safety domain. The Body & Comfort domain includes operations that do not frequently need real-time processing, such as in-vehicle climate control, seat control, door, window, or light control. The remote communication, information, and entertainment services are managed by the Telematics & infotainment domain. Each domain's performance and reaction time requirements vary depending on the function performed. Figure 1 depicts how these domains are integrated via various standards like as CAN, MOST, and LIN. The CAN Bus protocol is most commonly used in the internal communication network of vehicles to support the aforementioned operations [9].

2.1. Controller area network (CAN)

Robert Bosch GmbH, a multinational company, developed many versions of the CAN standard. It is a bus-topology-based synchronous protocol that serves as a communication channel for autonomous vehicle's ECUs. The most recent specification is CAN 2.0, which is divided into two parts: CAN 2.0A and CAN 2.0B [10]. It specifies four frame types: data, remote, error, and overload frame [11]. Figure 2 depicts the CAN data frame, which begins with a 1-bit Start of Frame (SOF) field, followed by an Arbitration field containing an 11- or 29-bit Identifier (ID) (CAN 2.0A has an 11-bit ID, whereas CAN 2.0B is the extended format with a 29-bit ID), and a 1-bit Remote Transmission Request (RTR). The Control field follows, consisting of 1 bit for the Identifier extension bit, 1 reserved bit (r0), and 4 bits for the Data length code. After the control field, the frame contains a Data field with 0 to 8 bytes of data. The frame then has a CRC field with 15 bits of CRC and 1 bit of CRC delimiter, an ACK field with 1 bit of ACK and 1 bit of ACK delimiter, and lastly 7 bits of End of Frame (EOF). The ID of a CAN data frame can be used to determine which signals are encoded in the message. For example, a message with one ID may encode the vehicle speed, but a message with another ID may contain information such as the engine temperature or speed. The ID can also be used to determine the priority of a communication. A low ID signifies higher priority, whereas a higher ID indicates lesser priority of the frame [12].

2.2. Attack Surface

The CAN protocol has various inherent weaknesses as a result of broadcast data transfer without authentication and encryption, as well as message ID-based prioritization. Adversaries can use interfaces such as the OBD, USB ports, and wireless interfaces to get access to data on the internal communication channel. The OBD port is especially vulnerable to attack since it is used to diagnose the vehicle's issues, modify the ECU parameters, and has access to data transmitted on the internal communication channel by other nodes, making in-vehicle networks open to malicious attacks [13]. A laptop or intelligent computing device attached to the OBD port can easily intercept messages sent over the CAN bus. In recent years, the majority of experimental attacks against smart vehicles have used the same port [14].

Telematics systems in smart automobiles combine telecommunications and informatics to provide a diverse set of features and services such as location-based service, cellular network service, and so on. The telematics system's ability to link to external networks renders it vulnerable to cyber-attacks, which might pose security issues to in-vehicle networks. Cybercriminals can get access to the internal communication network of the targeted vehicle using the aforementioned interfaces and carry out a range of attacks, including 'replay', 'DoS', or 'spoofing' attacks [11]. Table 2 shows a list of effective attacks undertaken and analysed by various researchers on the IoV subsystem.

To improve the security of the internal communication network of a vehicle, it is beneficial to know the theories behind the possible attacks. As a result, in this part, we introduce the attack methods

that have been proven to be successful on the internal communication network of a vehicle, and are discussed in previous publications [1], [6], [12], [13]. The attacks are mentioned below.

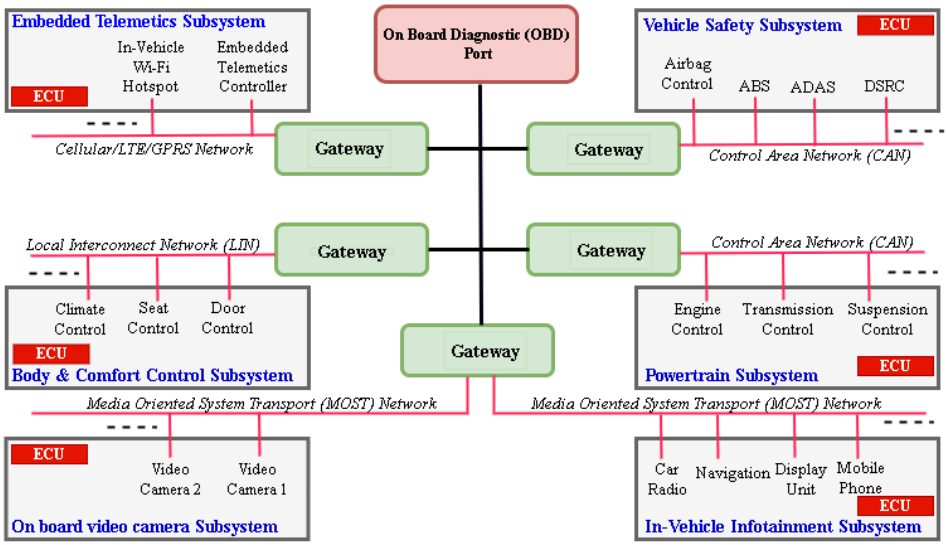

Figure 1. In-Vehicle sub-systems adapted from[9]

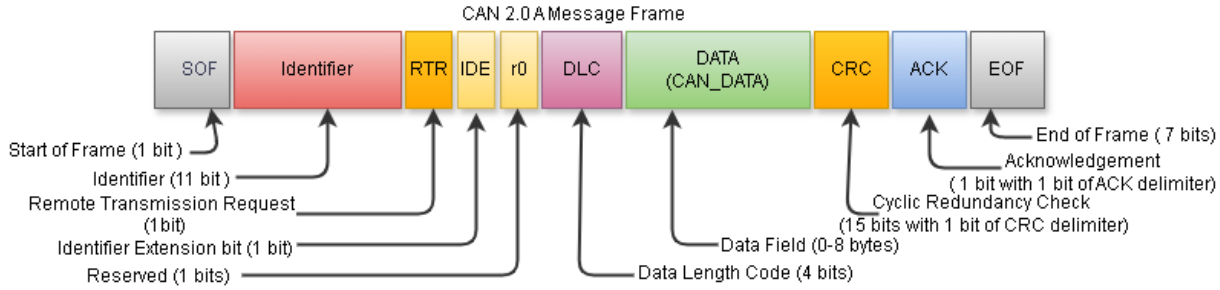

Figure 2. CAN frame structure

Table 2. Attacks executed and analyzed on IoV subsystem

Ref.	Attack Surface	Impact
Eisenberth et al. [15]	Keyless entry system	Control the door lock, unlock, and the engine
Koscher et al. [16]	Interfaces Infotainment using OBD-II/USB port	CAN Bus injection, full access to the vehicle
Miller et al. [17]	OBD-II port	Control brakes, wheels, and get access to the CAN Bus of a real vehicle
Petit et al. [18]	LiDAR, Cameras Sensors	Signal jamming
Zorz et al. [19]	OBD-II Cellular Dongle	CAN Bus injection in Real vehicle
Palanca et al. [20]	OBD-II interface	DoS attack on CAN Bus

Woo et al. [21]	OBD-II interface	Replay, impersonation attack using the smartphone application
Nie et al. [22]	Wi-Fi, GSM	Replay, impersonation using access to CAN network using browser exploit
Mukherjee et al. [23]	OBD-II port	DoS attack by compromise ECU using data link layer exploit

i. Frame Sniffing: Frame sniffing lays the groundwork for numerous afterwards attacks. As previously stated, data frames are broadcast to all nodes in the vehicle's internal communication network, allowing an infected node to monitor and record them. The cyber attacker can utilize this information to get access to the vehicle's internal communication network [13].

ii. Replay Attack: This is the method in which attackers just need to control the compromised nodes to broadcast legitimate frames into the internal communication network, which subsequently delivers orders to the vehicle's various subsystems. The autonomous vehicle's communication protocol lacks an authentication technique, making it difficult to validate the source of the received frames [13].

iii. DoS Attack: If a message with the highest priority is being conveyed on the vehicle's internal communication network, no node can send its message to the same communication channel. The attackers may easily use the policy to carry out DoS attacks by commanding the infected node to always broadcast legitimate messages with the highest priority, preventing other nodes from delivering essential signals [13].

3. Literature Review

Encryption, authentication, protocol stack redesign, and intrusion detection systems are among the suggested security options for CAN Bus protection [8]. Some studies in recent years have focused on encryption techniques to secure the CAN system. However, adopting similar algorithms may need extra hardware or modifications to current ECUs. Intrusion detection methods that do not need changes to the network protocol or hardware are a better alternative for security inside AVs [7]. Researchers have utilised a range of ways to identify CAN bus intrusions, including rule-based, machine learning-based, and other technologies. The authors of the article [24] present a deep learning-based CNN model for protecting the CAN bus in smart automobiles. The findings are also compared to various traditional methods; among them, the deep learning system achieves excellent accuracy. The study conducted in [25], describes another deep learning-based intrusion detection model that utilizes LSTM and CNNs network models, whereas identical research has been done by authors in [12], [26], [27], [28], [29]. These studies have shown that standard CAN network data is growing increasingly sophisticated, and neural network-based models, particularly deep learning models, are the most effective way to handle the identified weaknesses in IoV security [30].

The majority of DNN-based solutions were built using Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), which are extensively utilised to solve complicated problems in computer vision, text processing, audio recognition and classification, and so on. Because of their complexity, DNNs often take a long time to train on input data. They also require powerful computers with specialized processing units like Tensor, and Neural Processing Units. In this study, we present a deep learning-based IDS for CAN bus networks that outperforms previous work due to its simpler and more optimized network model.

4. Defence Mechanism

To detect and categorize various assaults on autonomous vehicles by recognizing abnormal CAN network traffic patterns, we present an MLP-based deep learning model that may be deployed as an extra CAN Bus node, such as an OBD-2 dongle. It is more affordable and practical, and there is no need to modify the CAN Bus. It can detect and identify several types of attacks on the IoV CAN network. Section 4.1 describes the realistic and most recent dataset used to train our model, whereas section 4.2 describes the structure of the suggested solution.

4.1. Description of the dataset

In this study, we used the CICIoV2024 dataset [31] [32]. It is a benchmark dataset, generated using a testbed of the real vehicle, to encourage the development of innovative security solutions for IoV processes, and it is published on the CIC dataset homepage. It contains traces for normal as well as five attack scenarios: DoS, 'spoofing-steering wheel', 'spoofing-RPM', 'spoofing-GAS', and 'spoofing-SPEED' attack, carried out by leveraging the unique characteristics of the CAN protocol in a real testbed of a Ford automobile equipped with all ECUs [31].

The dataset preprocessing began with an initial cleaning phase to ensure the removal of incomplete, invalid, and duplicate entries, alongside identifying and eliminating outliers using statistical techniques. This was critical to ensure that the data was both clean and accurate for subsequent processing. Following this, feature selection was performed to retain only the most relevant features, such as CAN message IDs and payloads, while redundant or irrelevant attributes were removed. Table 3 shows the class and sample's information, while Table 4 shows the retrieved features. The dataset is subsequently divided into training and testing datasets. The data is divided into 60:40 ratios, which means that 60% is utilised for model training and 40% is used for model validation.

4.2. Proposed deep learning model and experimental setup

Multi-Layer Perceptron (MLP), which serves as the foundation for our deep learning approach, is a neural network with multiple hidden layers. It is best suited for regression or classification problems in which inputs are allocated to a class. The neurons (or nodes) are arranged in different layers, as illustrated in Figure 3, and are connected to every neuron in the next layer, so the output of one neuron becomes the input of the next. Each connection between neurons has a weight, which is one of the variables that change throughout training. The weight of the link influences how much information is sent between neurons. Once a neuron gets inputs from all other neurons linked to it, the output (y) is determined using the formula provided in equation (1).

Table 3. Number of samples collected for each class

S.No.	Class	#Samples
1	BENIGN	80000
2	DoS	74660
3	SPOOFING_GAS	9991
4	SPOOFING_RPM	54899
5	SPOOFING_SPEED	24950
6	SPOOFING_STEERING_WHEEL	19976
	TOTAL	**264476**

Table 4. Features extracted from the dataset

S.No.	Features	Description
1	ID	Arbitration ID
2	'DATA_0'	1^{st} to 8^{th} byte of data transmitted through CAN data frame
3	'DATA_1'	
4	'DATA_2'	
5	'DATA_3'	
6	'DATA_4'	
7	'DATA_5'	
8	'DATA_6'	

9	'DATA_7'	
10	LABEL	Type of traffic (Benign/Malicious)
11	TARGET LABEL	Six Specific Class of the traffic ('Benign', 'DoS', 'Spoofing_GAS', 'Spoofing_RPM', 'Spoofing_SPEED', and 'Spoofing_STEERING_WHEEL')

$$y = \sum_{i=1}^{N} (x_i * w_i) + b - - - (1)$$

Where xi is an input of the neuron, wi is the associated weight, and b is the bias. The output value (y) is then given to the activation function g(y), which introduces nonlinearity into the neuron's output. Finally, the model employs the backpropagation algorithm to update the weights of the input layer based on the error at the output layer.

The proposed MLP-based deep learning model consists of an input layer, an output layer, and two dense hidden layers. The model receives input that is extracted from the data packet transmitted over the internal communication channel of the vehicle. Due to 153 features being extracted from in-vehicle network traffic, the same number of neurons are inserted in the first layer. It is followed by two dense hidden layers of two and eight neurons, respectively. Because CAN network traffic is to be classified into six classes (TARGET LABEL in Table 4), the output layer is made up of completely linked six neurons. The model has a total of 386 trainable parameters. Figure 4 depicts the layer relationships, while Table 5 provides the model's summary.

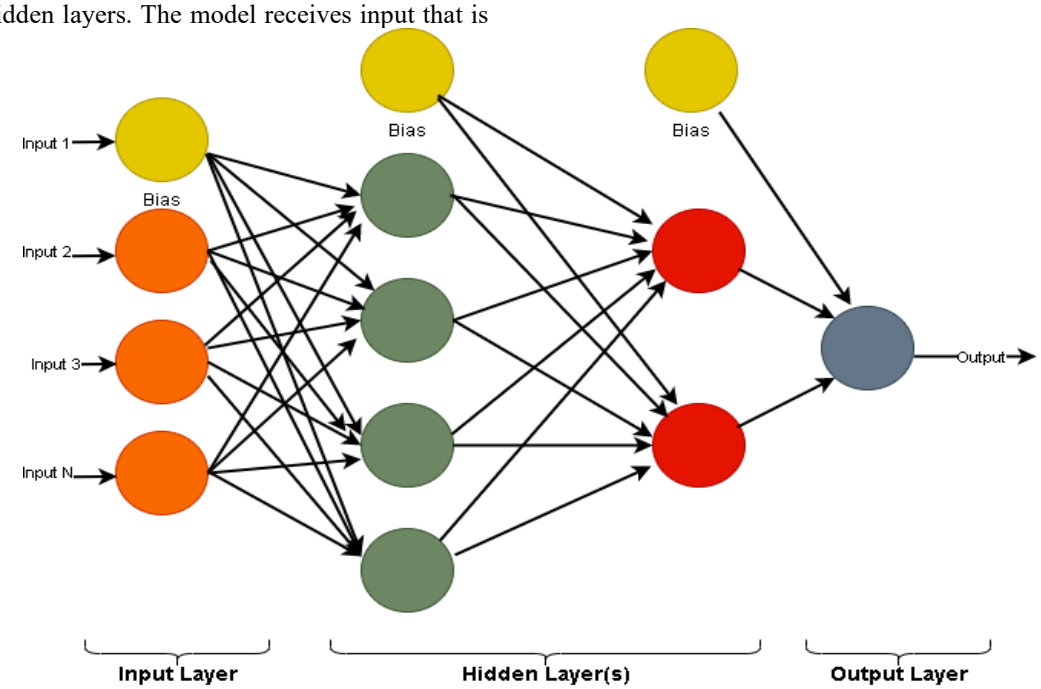

Figure 3. Multi-Layer Perceptron (MLP)

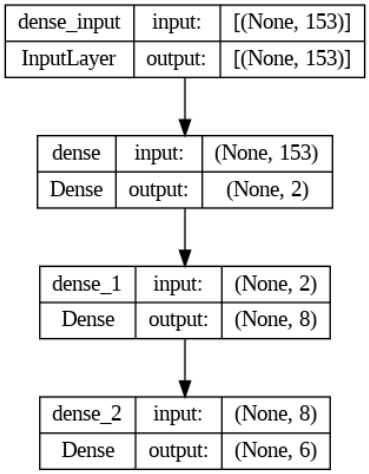

Figure 4. Plot of proposed MLP Model Graph

Table 5. Proposed MLP Model Summary

Layer	Shape of the Output	Number of Parameters	Activation function
dense_136 (Dense)	(None, 2)	308	Rectified Linear Units (ReLU)
dense_137 (Dense)	(None, 8)	24	Rectified Linear Units (ReLU)
dense_138 (Dense)	(None, 6)	54	Softmax
Total parameters: 386 (1.51 KB**)** **Trainable parameters:** 386 (1.51 KB) **Non-trainable parameters:** 0 (0.00 Byte) **Optimizer:** 'Adam'**,** **Loss function:** 'categorical_crossentropy'**,** **Performance Metrics:** 'Accuracy'			

The activation functions in MLP are critical for generating complex decisions and predictions. This article uses the ReLU activation function in the intermediate layers, which operates by performing a basic mathematical operation on the input value. If the input value is higher than or equal to zero, the output is the same as the input. If the input value is negative, the result is zero. The mathematical representation of the ReLU function is as follows:

$$f(x) = max(0, x) - - - (2)$$

The output layer employs the softmax activation function which transforms the raw output scores of the model into probabilities, facilitating the distribution of these probabilities across various classes. This transformation is mathematically represented by equation (3).

$$Softmax(Z_i) = \frac{e^{Z_i}}{\sum_{j=1}^{K} e^{Z_j}} - - - (3)$$

Here, Zi represents the input value for the Softmax function and is the output value of the node for ith class at the output layer. K is the total number of nodes at the output layer.

Cross entropy measures the difference between the predicted probability and the true probability. Multiclass Cross-Entropy Loss, also known as categorical cross-entropy, is used as a loss function in the proposed deep-learning model. The loss function is mathematically represented by equation (4).

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K}\left(y_{i,j}\log\left(p_{i,j}\right)\right) - - - (4)$$

Here, N is the number of instances in the dataset, K is the number of classes, yi,j is the true output for the ith sample and jth class, and pi,j is the predicted probability for ith sample and jth class. The Adam optimizer, which stands for "Adaptive Moment Estimation", is employed to iteratively minimize the loss function during training.

The rationale behind selecting specific hyperparameters for the Multi-Layer Perceptron (MLP) model is essential for enhancing

reproducibility and understanding how these decisions impact model performance. The MLP model consists of three layers designed for effective classification in the IoV context. The first layer outputs 2 units with ReLU activation, facilitating a compact feature representation while reducing dimensionality. The second layer has 8 neurons, enhancing the model's capacity to learn complex patterns, also using ReLU for non-linearity. The final layer, with 6 units and softmax activation, corresponds to the number of output classes and converts logits into probabilities for multi-class classification.

The Adam optimizer is chosen for its adaptive learning rate, allowing faster convergence, while the categorical cross-entropy loss function is appropriate for multi-class tasks, measuring the difference between true and predicted probabilities. Accuracy is used as the performance metric, providing a straightforward measure of model performance in classification tasks. Overall, this configuration balances model complexity, efficiency, and accuracy for CAN network traffic classification.

The Google Co laboratory, commonly known as Google Colab, is used to set up, compile, train, and validate the proposed deep-learning based intrusion detection model for the protection of the CAN network of autonomous vehicles. It offers access to Graphical and Tensor Processing Units. The suggested model was trained and examined across 40 epochs using a 1000-batch size. The dataset contains 264476 instances, of which 158685 (60% of total samples) are used for training and 105791 (40% of total samples) are utilised for validations of the trained model. The simulation results are gathered and processed for comparison with benchmark and relevant research. The results are reported, and a comparative analysis is provided in the next section.

5. Results and comparative analysis

In the domain of deep learning, a perfect fit model is desired since it assures strong generalization and consistent performance on new data. Overfitting and underfitting, on the other hand, provide unreliable results and poor generalization. A well-performing deep learning model should have training and validation loss curves that converge to a comparable, low data point. This shows that the model is generalizing properly and not overfitting or underfitting. Analyzing the behaviour of these curves during training gives vital insights into the model's learning process and aids in making the required changes to increase performance. Table 6 shows the training and validation losses reported for each epoch throughout the simulation, which are also represented in Figure 5. The convergence of the training and validation loss curves demonstrates that our suggested model is learning the core trends in the data and generalizing successfully to the validation set. It also indicates that the model is neither overfitting nor underfitting.

Evaluating the performance of a deep learning model incorporates a series of procedures and metrics that offer a full picture of how well the model is doing. Figure 6 depicts the confusion matrix obtained as a consequence of the simulation and used to calculate accuracy, precision, recall, and F1-score. Precision and recall are measures used to assess the effectiveness of a classification model, particularly in cases with unbalanced classes or where different types of classification errors have varying costs.

The performance evaluation metrics shown in Table 7 can be produced using equations (5) to (8), where α, β, γ, and μ denote True Positive, True Negative, False Positive, and False Negative.

$$Accuracy = \frac{\alpha + \beta}{\alpha + \beta + \gamma + \mu} - - - (5)$$

$$Precision\ (P) = \frac{\alpha}{\alpha + \gamma} - - - (6)$$

$$Recall\ (R) = \frac{\alpha}{\alpha + \mu} - - - (7)$$

$$F1_{score}(F1) = 2 * \frac{Precesion * Recall}{Precesion + Recall} - - - (8)$$

Precision and recall should both be high, but they should be used in conjunction with other assessment measures like accuracy and F1-score to have a thorough view of a classifier's performance. The F1-score, which is the harmonic mean of the Precision and Recall values, provides a more balanced assessment of the model's performance.

The results from the Confusion matrix (Figure 6) and Tables 7, 8, and 9 show that the proposed deep learning model can detect and classify an attack on an autonomous vehicle's CAN network with an average Recall of 0.999927477, Precision of 0.999930671, and F1-Score of 0.999929069. The model performed better than the benchmark research [31] in terms of accuracy, recall, precision, and F1-Score. It also outperformed previous studies [33], [24] with the highest accuracy of 99.99%.

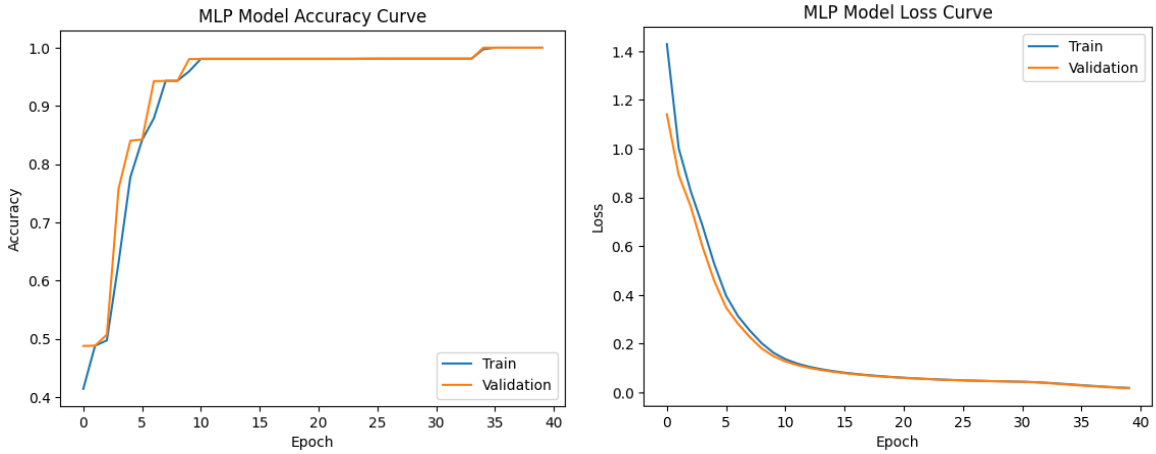

Figure 5. (a) Accuracy curve, (b) Loss curve

Actual	BENIGN	31997	1	0	2	0	0
	DoS	0	29865	0	0	0	0
	SPOOFING_GAS	0	0	3996	0	0	0
	SPOOFING_RPM	0	0	0	21958	2	0
	SPOOFING_SPEED	0	0	0	0	9980	0
	SPOOFING_STEERING_WHEEL	0	0	0	2	0	7988
		BENIGN	DoS	SPOOFING_GAS	SPOOFING_RPM	SPOOFING_SPEED	SPOOFING_STEERING_WHEEL
		Predicted					

Figure 6. Confusion Matrix

Table 6: Training and Validation efficiency of the proposed model.

Epoch	Tr_loss	Tr_accuracy	Val_loss	Val_accuracy	Epoch	Tr_loss	Tr_accuracy	Val_loss	Val_accuracy
1/40	1.4293	0.4142	1.1413	0.4878	21/40	0.0604	0.9809	0.0595	0.9806
2/40	1.0016	0.4883	0.8921	0.4884	22/40	0.0577	0.9809	0.0570	0.9806
3/40	0.8277	0.4973	0.7646	0.5072	23/40	0.0553	0.9809	0.0547	0.9806
4/40	0.6848	0.6311	0.5988	0.7578	24/40	0.0532	0.9809	0.0527	0.9806
5/40	0.5273	0.7775	0.4573	0.8402	25/40	0.0514	0.9811	0.0509	0.9808
6/40	0.3966	0.8415	0.3480	0.8423	26/40	0.0497	0.9811	0.0494	0.9809
7/40	0.3133	0.8789	0.2821	0.9425	27/40	0.0483	0.9811	0.0480	0.9809
8/40	0.2539	0.9430	0.2272	0.9428	28/40	0.0470	0.9811	0.0468	0.9809
9/40	0.2022	0.9431	0.1805	0.9428	29/40	0.0459	0.9811	0.0457	0.9809
10/40	0.1628	0.9594	0.1486	0.9805	30/40	0.0449	0.9812	0.0447	0.9809
11/40	0.1366	0.9809	0.1274	0.9805	31/40	0.0439	0.9812	0.0435	0.9809
12/40	0.1187	0.9809	0.1123	0.9805	32/40	0.0423	0.9812	0.0415	0.9809
13/40	0.1056	0.9809	0.1010	0.9805	33/40	0.0399	0.9812	0.0387	0.9809
14/40	0.0956	0.9809	0.0922	0.9805	34/40	0.0367	0.9812	0.0353	0.9809
15/40	0.0877	0.9809	0.0850	0.9805	35/40	0.0333	0.9973	0.0318	0.9999
16/40	0.0811	0.9809	0.0790	0.9805	36/40	0.0298	0.9999	0.0283	0.9999
17/40	0.0757	0.9809	0.0739	0.9805	37/40	0.0265	0.9999	0.0251	0.9999
18/40	0.0710	0.9809	0.0696	0.9805	38/40	0.0234	0.9999	0.0222	0.9999
19/40	0.0670	0.9809	0.0658	0.9806	39/40	0.0207	0.9999	0.0196	0.9999
20/40	0.0635	0.9809	0.0625	0.9806	40/40	0.0184	0.9999	0.0174	0.9999

Tr_loss: Training Loss; Tr_accuracy: Training Accuracy; Val_loss: Validation Loss; Val_accuracy: Validation Accuracy

Table 7. Recall, Precision, and F1-Score values for the proposed model

Class	Recall	Precision	F1-Score
BENIGN	0.99990625	1	0.999953123
DoS	1	0.999966517	0.999983258
SPOOFING_GAS	1	1	1
SPOOFING_RPM	0.999908925	0.999817867	0.999863394
SPOOFING_SPEED	1	0.999799639	0.99989981
SPOOFING_STEERING_WHEEL	0.999749687	1	0.999874828
Macro Average	**0.999927477**	**0.999930671**	**0.999929069**

Table 8. Proposed solution vs. Benchmark study

Ref.	Accuracy	Recall	Precision	F1-Score
Neto *et al.* [31]	95%	0.68	0.74	0.63
Proposed Model	**99.99%**	**0.999927477**	**0.999930671**	**0.999929069**

Table 9. Proposed solution vs. Related work

Ref.	Solution Type	Attack type considered	Efficiency of the solution (Avg. Accuracy)
Sudhakar *et al.* [33]	Deep learning (CNN) based	Malware	98.63%
Ahmed *et al.* [24]	Deep learning (CNN) based	DoS, Fuzzy	96 %
Neto *et al.* [31]	Deep learning (MLP) based	DoS, Spoofing	95%
Proposed Model	**Deep learning (MLP) based**	**DoS, Spoofing**	**99.99%**

6. Conclusion and scope for future work

The CAN protocol is a key component of the internal communication network of smart cars because it connects various sub-systems of autonomous vehicles, and allows them to communicate with each other. However, the protocol's inherent lack of security makes it susceptible to a wide range of cyber threats, posing significant risks to both the safety and privacy of the driver, passenger or the vehicle itself.

This paper has explored the effectiveness of the DL-based approach to enhance the security of internal communication networks of smart vehicles in the IoV framework. We have provided a comprehensive survey of existing threats to CAN networks, such as spoofing, replay, and DoS attacks, and examined how deep learning can be utilized to detect and mitigate these threats effectively. We also proposed a novel deep-learning based defense mechanism that provide real-time threat detection. Our findings highlight the potential of deep learning to significantly enhance the security of CAN networks in IoV, contributing to safer and more reliable vehicular communication systems.

The proposed MLP model for enhancing CAN network security within the Internet of Vehicles (IoV) has significant practical implications across various domains. It offers real-time threat detection, which protects vehicles from cyber-attacks like spoofing and denial-of-service, thereby enhancing overall vehicle

security. The model also plays a crucial role in smart traffic management by monitoring CAN traffic, leading to improved traffic flow and safer infrastructure.

In the context of autonomous vehicles, the model ensures that self-driving cars operate securely, fostering public trust and encouraging broader adoption of autonomous technology. For commercial fleet operators, it provides a means to monitor and protect vehicles, minimizing the risk of data breaches and enhancing operational safety.

The simulation results indicate that the proposed DL-based model is capable of successfully detecting and classifying attacks (DoS and spoofing) on the CAN network of an autonomous vehicle, with an average Recall of 0.999927477, Precision of 0.999930671, and F1-Score of 0.999929069. The proposed model outperformed benchmark studies and other related work in terms of accuracy, recall, precision, and F1-Score, achieving the highest accuracy of 99.99%.

Future work will focus on improving the scalability of the proposed system and integrating it with broader IoV security frameworks to provide a holistic defense strategy.

References

[1] R. Islam and R. U. D. Refat, "Improving CAN bus security by assigning dynamic arbitration IDs," J. Transp. Secur., vol. 13, no. 1–2, pp. 19–31, Jun. 2020, doi: 10.1007/s12198-020-00208-0.

[2] O.-R. A. D. (ORAD) Committee, Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. SAE international, 2021.

[3] P. Gershon, S. Seaman, B. Mehler, B. Reimer, and J. Coughlin, "Driver behavior and the use of automation in real-world driving," Accid. Anal. Prev., vol. 158, p. 106217, Aug. 2021, doi: 10.1016/j.aap.2021.106217.

[4] N. Sharma, N. Chauhan, and N. Chand, "Security challenges in Internet of Vehicles (IoV) environment," in 2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India: IEEE, Dec. 2018, pp. 203–207. doi: 10.1109/ICSCCC.2018.8703272.

[5] K. Aswal, D. C. Dobhal, and H. Pathak, "Comparative analysis of machine learning algorithms for identification of BOT attack on the Internet of Vehicles (IoV)," in 2020 International Conference on Inventive Computation Technologies (ICICT), IEEE, Feb. 2020, pp. 312–317. doi: 10.1109/ICICT48043.2020.9112422.

[6] Q. Zhao, M. Chen, Z. Gu, S. Luan, H. Zeng, and S. Chakrabory, "CAN Bus Intrusion Detection Based on Auxiliary Classifier GAN and Out-of-distribution Detection," ACM Trans. Embed. Comput. Syst., vol. 21, no. 4, pp. 1–30, Jul. 2022, doi: 10.1145/3540198.

[7] H. Sun et al., "CCID-CAN: Cross-Chain Intrusion Detection on CAN Bus for Autonomous Vehicles," IEEE Internet Things J., pp. 1–1, 2024, doi: 10.1109/JIOT.2024.3393122.

[8] E. Aliwa, O. Rana, C. Perera, and P. Burnap, "Cyberattacks and Countermeasures for In-Vehicle Networks," ACM Comput. Surv., vol. 54, no. 1, pp. 1–37, Jan. 2022, doi: 10.1145/3431233.

[9] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," IEEE Internet Things J., vol. 1, no. 1, pp. 10–21, 2014, doi: 10.1109/JIOT.2014.2302386.

[10] S.-H. Chen and C.-H. R. Lin, "Evaluation of DoS Attacks on Vehicle CAN Bus System," in Recent Advances in Intelligent Information Hiding and Multimedia Signal Processing, vol. 110, J.-S. Pan, A. Ito, P.-W. Tsai, and L. C. Jain, Eds., in Smart Innovation, Systems and Technologies, vol. 110. , Cham: Springer International Publishing, 2019, pp. 308–314. doi: 10.1007/978-3-030-03748-2_38.

[11] H. J. Jo and W. Choi, "A Survey of Attacks on Controller Area Networks and Corresponding Countermeasures," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 7, pp. 6123–6141, Jul. 2022, doi: 10.1109/TITS.2021.3078740.

[12] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data," IEEE Access, vol. 8, pp. 58194–58205, 2020, doi: 10.1109/ACCESS.2020.2982544.

[13] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-Vehicle Network Attacks and Countermeasures: Challenges and Future Directions," IEEE Netw., vol. 31, no. 5, pp. 50–58, 2017, doi: 10.1109/MNET.2017.1600257.

[14] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of CAN Bus Security Challenges," Sensors, vol. 20, no. 8, p. 2364, Apr. 2020, doi: 10.3390/s20082364.

[15] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme," in Advances in Cryptology–CRYPTO 2008: 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings 28, Springer, 2008, pp. 203–220.

[16] K. Koscher et al., "Experimental security analysis of a modern automobile," in 2010 IEEE symposium on security and privacy, IEEE, 2010, pp. 447–462.

[17] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," Black Hat USA, vol. 2015, no. S 91, pp. 1–91, 2015.

[18] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," Black Hat Eur., vol. 11, no. 2015, p. 995, 2015.

[19] Z. Zorz, "Backdooring connected cars for covert remote control—Help Net Security (2018)," Retrieved August, 2020.

[20] A. Palanca, E. Evenchick, F. Maggi, and S. Zanero, "A stealth, selective, link-layer denial-of-service attack

against automotive networks," in Detection of Intrusions and Malware, and Vulnerability Assessment: 14th International Conference, DIMVA 2017, Bonn, Germany, July 6-7, 2017, Proceedings 14, Springer, 2017, pp. 185–206.

[21] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2, pp. 993–1006, 2014.

[22] S. Nie, L. Liu, and Y. Du, "Free-fall: hacking tesla from wireless to can bus," Defcon, pp. 1–16, 2017.

[23] S. Mukherjee, H. Shirazi, I. Ray, J. Daily, and R. Gamble, "Practical DoS attacks on embedded networks in commercial vehicles," in Information Systems Security: 12th International Conference, ICISS 2016, Jaipur, India, December 16-20, 2016, Proceedings 12, Springer, 2016, pp. 23–42.

[24] I. Ahmed, G. Jeon, and A. Ahmad, "Deep Learning-Based Intrusion Detection System for Internet of Vehicles," IEEE Consum. Electron. Mag., vol. 12, no. 1, pp. 117–123, Jan. 2023, doi: 10.1109/MCE.2021.3139170.

[25] 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. Netw., vol. 9, no. 5, pp. 1113–1122, Oct. 2023, doi: 10.1016/j.dcan.2022.06.018.

[26] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-Based Intrusion Detection System for In-Vehicle Can Bus Communications," IEEE Access, vol. 8, pp. 185489–185502, 2020, doi: 10.1109/ACCESS.2020.3029307.

[27] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," Veh. Commun., vol. 21, p. 100198, Jan. 2020, doi: 10.1016/j.vehcom.2019.100198.

[28] J. Zhang, F. Li, H. Zhang, R. Li, and Y. Li, "Intrusion detection system using deep learning for in-vehicle security," Ad Hoc Netw., vol. 95, p. 101974, Dec. 2019, doi: 10.1016/j.adhoc.2019.101974.

[29] Zhou, Li, and Shen, "Anomaly Detection of CAN Bus Messages Using A Deep Neural Network for Autonomous Vehicles," Appl. Sci., vol. 9, no. 15, p. 3174, Aug. 2019, doi: 10.3390/app9153174.

[30] P. Mansourian, N. Zhang, A. Jaekel, and M. Kneppers, "Deep Learning-Based Anomaly Detection for Connected Autonomous Vehicles Using Spatiotemporal Information," IEEE Trans. Intell. Transp. Syst., vol. 24, no. 12, pp. 16006–16017, Dec. 2023, doi: 10.1109/TITS.2023.3286611.

[31] E. C. P. Neto et al., "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus," Internet Things, vol. 26, p. 101209, Jul. 2024, doi: 10.1016/j.iot.2024.101209.

[32] E. C. P. Neto, H. Taslimasa, S. Dadkhah, S. Iqbal, P. Xiong, T. Rahmanb, and A. A. Ghorbani, "CIC IoV dataset 2024, Canadian Institute for Cybersecurity, https://www.unb.ca/cic/datasets/iov-dataset-2024.html." 2024. [Online]. Available: https://www.unb.ca/cic/datasets/iov-dataset-2024.html

[33] Sudhakar and S. Kumar, "An emerging threat Fileless malware: a survey and research challenges," Cybersecurity, vol. 3, no. 1, p. 1, Dec. 2020, doi: 10.1186/s42400-019-0043-x.