Available online at www.dergipark.org.tr/en

INTERNATIONAL ADVANCED RESEARCHES and ENGINEERING JOURNAL

Journal homepage: www.dergipark.org.tr/en/pub/iarej

e-ISSN: 2618-575X

International Open Access

Volume 09
Issue 02

August, 2025

Research Article

Performance comparison of machine learning models on a novel in-vehicle controller area network dataset

Batuhan Gul [a]* ***and Fatih Ertam*** [a]

[a] *Department of Digital Forensics Engineering, Faculty of Technology, Firat University, Elazig, 23000, Turkey*

ARTICLE INFO

Article history:
Received 27 December 2024
Accepted 22 July 2025
Published 20 August 2025

Keywords:
Classification algorithms
Cyber attack
Machine learning
Smart vehicle

ABSTRACT

The advancement of technology has significantly enhanced comfort and welfare across all aspects of life, particularly in the field of transportation. One notable development is the growing adoption of autonomous vehicles, driven by the integration of smart systems into automobiles. However, the sophisticated systems and networks within autonomous vehicles have also opened new avenues for cyberattacks. These attacks typically aim to achieve one of three objectives: gaining unauthorized control of system components, overloading the system network to slow its operation, or causing a system crash. The potentially severe consequences of such cyberattacks have underscored the urgent need for robust security measures to protect autonomous vehicles. This study focuses on detecting cyberattacks targeting in-vehicle networks of smart vehicles using machine learning models. A simulation environment was developed to generate cyberattack scenarios, resulting in the creation of a dataset. This dataset was then analyzed using classification algorithms, including XGBoost, LightGBM, Random Forest, and Decision Trees. Performance comparisons revealed that XGBoost achieved the highest accuracy at 86.22% and F1 Score at 79.7%, while the Decision Tree algorithm had the lowest accuracy at 80.7% and F1 Score at 72.5%. In addition, the LightGBM algorithm had an accuracy rate of 85.83% and the Random Forest algorithm had an accuracy rate of 85.84%. The findings of this study are expected to contribute to the efforts of smart vehicle security experts in mitigating cyber threats and raising awareness about the importance of cybersecurity in autonomous vehicles.

1. Introduction

The rapid advancement of technology has led to significant innovations in many areas of daily life. One of the most impactful developments is the emergence of smart vehicles, which have contributed positively to society and the environment. These smart vehicles contribute to society not only in the field of comfort, but also in driving safety, environmental security and economic areas. In smart vehicles, there are in-vehicle networks such as Control Area Network (CAN), Media Oriented System Transport (MOST), Local Interconnect Network (LIN), FlexRay, Ehernet [1]. Among these networks, CAN is the most widely used due to its reliability and simplicity. The task of these in-vehicle networks is to provide communication between Electronic Control Units (ECU) in the vehicle. Electronic control units are embedded systems that control certain functions in the vehicle. Today's vehicles have an average of 70-100 electronic control units [2]. These units have tasks such as vehicle engine control (Engine Control Unit), transmission control (Transmission Control Unit), brake control (Brake Control Unit), airbag control (Airbag Control Module), entertainment system control (Infotainment Control Unit). ECUs play a crucial role in ensuring both driver comfort and operational safety. However, the malfunction of even a single ECU can severely disrupt in-vehicle communication [3]. In-vehicle networks provide communication between electronic control units thanks to their advanced technology. However, these networks also have disadvantages among themselves. CAN network enables secure data communication thanks to the error control mechanisms it contains and problematic equipment can be isolated from the network. In addition, new devices and nodes can be added to the CAN network. However, the fact that the CAN network has a low data transmission speed (1 mbps), does not have security mechanisms such as authentication and data encryption, and can cause message delays during busy times shows that the CAN network is not secure enough [4]. Another in-vehicle network (LIN) was developed as an alternative to the CAN network and is responsible for providing communication between non-critical electronic units. LIN supports a data

* Corresponding author. Tel.:*+90-505-100-2355*.
E-mail addresses: b.gul@firat.edu.tr *(B. GUL),* fatih.ertam@firat.edu.tr *(F. ERTAM)*
ORCID: 0009-0007-1772-5373 *(B. GUL),* 0000-0002-9736-8068 *(F. ERTAM)*
DOI: 10.35860/iarej.1607108

transfer of up to 20 kbps and is sufficient because it is used in applications that require low transfer speed. Despite these advantages, the LIN network has disadvantages such as being usable only for low-priority applications due to its low data transfer speed, having limited error detection mechanisms, and not having security measures such as authentication. Another in-car network, FlexRay, was developed to be a high-speed and reliable communication protocol. This network provides high-speed transfer between control systems and other critical applications. Its data transfer speed of up to 10 Mbps makes FlexRay a faster communication protocol than LIN and CAN. In addition, it offers very good error management thanks to its redundancy mechanism and checksum features [5]. However, it is much more costly than other in-car network protocols such as CAN and LIN, and since the network is more difficult to configure than other networks, it is more complex than other protocols. The MOST network enables high-speed transmission of audio, video, and data in in-car multimedia systems, with up to 150 Mbps via the MOST150 standard and low power use with fiber optics. However, its high cost, complexity, and alternatives like Ethernet and CAN FD limit its widespread adoption. Ethernet offers the highest data transmission speed and was designed for high-speed communication. However, as advanced technological platforms, in-car networks remain vulnerable to cyberattacks. Previous studies [6] have shown that in-car networks are vulnerable to cyber attacks. The first reason for this is that cyber security was not taken into consideration when in-car networks were developed. Therefore, security measures such as data encryption and authentication are quite limited. In-car networks also connect to the Internet for reasons such as operating entertainment systems and receiving updates, and they become vulnerable to cyber attacks.

Various methods have been developed to protect in-vehicle networks against cyber attacks and to perform attack detection, but due to the emergence of new types of cyber attacks day by day, it has become necessary to conduct new studies in this field. In this study, it is aimed to develop a new intrusion detection system that detects attacks in the in-vehicle CAN network. It was analyzed using our own dataset and an artificial intelligence-supported classification model was created by extracting the features of this data.

The contributions of the study to the literature are as follows:

- Detailed information about in-vehicle networks was given and the most recent studies in the literature were analyzed. These studies were presented with their advantages and disadvantages.
- A novel dataset was created using ICSim, an in-vehicle network simulation tool, and various cyber attacks were carried out in the simulation environment and network packets were analyzed.
- The attacks were classified using Random Forest, LightGBM, XGBoost and decision tree algorithms. As a result of the classification, it was seen that XGBoost achieved the highest accuracy rate with 86.22%, while the decision tree algorithm achieved the lowest accuracy rate with 80.7%.

2. Literature Review

Numerous protection mechanisms have been proposed for in-vehicle networks in the literature; however, as technology advances, new attack vectors continue to emerge, driving the need for ongoing research to detect these threats. Hossain et al. [7] developed an intrusion detection system based on Long Short-Term Memory (LSTM) networks specifically designed to identify attacks on the CAN network. The authors created their dataset by simulating cyber attacks on an actual smart vehicle. Experimental results demonstrated that their approach successfully detected 99.94% of Fuzzing attacks and achieved a 100% detection rate for DoS attacks. Hanselmann et al. [8] proposed a new neural network architecture to detect attacks on the CAN network. The method called CANet was tested on real and synthetic CAN data and plateau, repetition, and flooding attacks were applied. The proposed method has a true negative rate of over 99%. Jin et al. [9] proposed a signature-based intrusion detection system that can be directly applied to electronic control systems in the vehicle. The authors created an artificial dataset using the CANoe software and applied the proposed method on this dataset. As a result of the experiments, the proposed model detected packet drop and replay attacks with a high accuracy rate. It detected tampering attacks with an accuracy rate of 66.2%. The authors stated that the proposed method is more flexible and has less computational time compared to other IDS types. Islam et al. [10] presented a graph-based intrusion detection system named GGNB, which leverages the graphical characteristics of the data. When tested against DoS, Fuzzy, Spoofing, and Replay attacks using the RawCAN dataset, the method achieved detection accuracies of 99.61%, 99.83%, 96.79%, and 93.35%, respectively. In addition to the RawCAN dataset, GGNB achieved an accuracy rate of 100%, 99.92%, 99.92% and 97.75% in DoS, Fuzzing, Replay and Suspension attacks on the OpelAstra dataset, respectively. Deng et al. [11] introduced VoltageIDS, an intrusion detection system that relies on voltage measurements and operates without needing any prior knowledge. The proposed approach avoids utilizing the CAN bus bandwidth. It is implemented in three stages: collecting voltage data, extracting features, and detecting intrusions. VoltageIDS was tested on three vehicles, Opel, Peugeot, and Toyota, and fabricated, Masquerade, and Bus-off attacks were applied to these vehicles. VoltageIDS was able to detect 100% of the attacks. Khandelwal et al. [12] proposed a hybrid intrusion detection system using FPGA technology and a CNN-based model. The system incorporates an IDS accelerator on the Zynq Ultrascale+

platform, enabling CAN messages to be forwarded directly to the ML-IDS without disrupting ECU operations. The deep learning model, QdCNN, detects DoS, Fuzzy, and RPM spoofing attacks with 99.32% accuracy. Experimental results show a 51.8% reduction in message delay and a 94% decrease in power consumption compared to conventional methods. Moreover, QdCNN outperforms other IDS models such as GIDS, DCNN, and iForest.

Desta et al. [13] introduced a novel intrusion detection system based on convolutional neural networks (CNN). Their approach transforms in-vehicle network data into images using recurrence graphs. They trained their model on both the Car Hacking dataset and real vehicle data, utilizing timestamp and arbitration ID features from CAN packets. The method was tested against DoS, fuzzy, gear spoofing, and RPM spoofing attacks, achieving a classification accuracy of 99.9%. Yu et al. [14] introduced an innovative intrusion detection system (IDS) that identifies attacks by analyzing conditional entropy values. This approach computes conditional entropy based on the time intervals between consecutive messages, taking into account both the frame ID and the data field. Under normal conditions, message entropy remains within a specific range, enabling the system to detect malicious messages that deviate from this pattern. The method was experimentally evaluated using the OTIDS dataset against Denial of Service (DoS), Fuzzy, and Impersonation attacks targeting the CAN bus. Results showed high effectiveness, blocking 100% of DoS attacks, 99.9% of Fuzzy attacks, and 100% of Impersonation attacks, all with a minimal delay of just 2 milliseconds. Tanksale [15], who proposed an intrusion detection system for in-vehicle CAN networks, proposed a new machine learning based model based on LSTM. The proposed method consists of two components: anomaly detection engine and decision engine. The first component uses the LSTM model to estimate the time series outputs of various functions, and if the difference between the actual value and the estimated value exceeds the threshold value, it is considered an anomaly. The labels made by the first component are transferred to the second component, the decision engine, where a further analysis is performed using three criteria. The proposed system has higher sensitivity and lower false positive rate compared to other studies. However, the fact that it has not been studied with large amounts of data makes it difficult to apply the proposed method in real-world conditions. To prevent spoofing and replay attacks, Ye et al. [16] proposed GDT-IDS, a graph-based decision tree approach. The proposed method creates graphs using every 200-message window from the CAN network and extracts features such as graph density, degrees, edges, time difference from these graphs. And the proposed model performs anomaly detection using these features. GDT-IDS has been tested on various real datasets such as IVN, Opel Astra and detected spoofing attacks with 99.83% rate and replay attacks with 99.94% rate. In multiple attack scenarios, it had a success rate of 99.43%. The proposed method has a limitation that it detects attacks only in certain time windows.

Unlike previous studies, we are creating a new dataset in our study. ICSim, a simulation tool, was used to create this dataset. The performance of various machine learning algorithms was compared and analyzed on the created dataset. The study aims to give researchers an idea of which machine learning algorithm is more successful.

3. Security Weaknesses of In-Vehicle Networks

Smart vehicle network protocols prioritize low cost and performance but lack protection against modern cyber threats. Studies show that in-vehicle networks remain vulnerable, and increasing connectivity suggests cybersecurity challenges will persist [17, 18]. Attacks on in-vehicle networks are generally categorized into four main types: direct interface-initiated attacks, infotainment system-initiated attacks, telematics-initiated attacks, and sensor-initiated attacks [19]. Since the CAN protocol does not have sufficient defense mechanisms, the communication mechanism between in-vehicle components is also weak. The most common attacks applied to in-vehicle networks are CAN denial of service (CAN DoS) and injection attacks [20]. In this study, the vulnerabilities of in-vehicle CAN networks were investigated and an intrusion detection system (IDS) was developed by compiling the most common cyber attack types.

3.1. CAN Bus Fuzzing Attack

This attack is based on the lack of authentication and data integrity control feature of the CAN protocol. For these reasons, electronic control units accept CAN messages from other units and respond to these messages. In a fuzzing attack, the attacker generates random CAN ID and Data values and sends them to other ECUs [21]. Each CAN ID has different tasks in the vehicle. The fact that the sent packets have a CAN ID and data value and are not sent at a very high frequency makes this attack harder to detect than a denial of service (DoS) attack and causes unexpected behavior in the vehicle. The working principle of the Fuzzy Attack is shown in Figure 1.

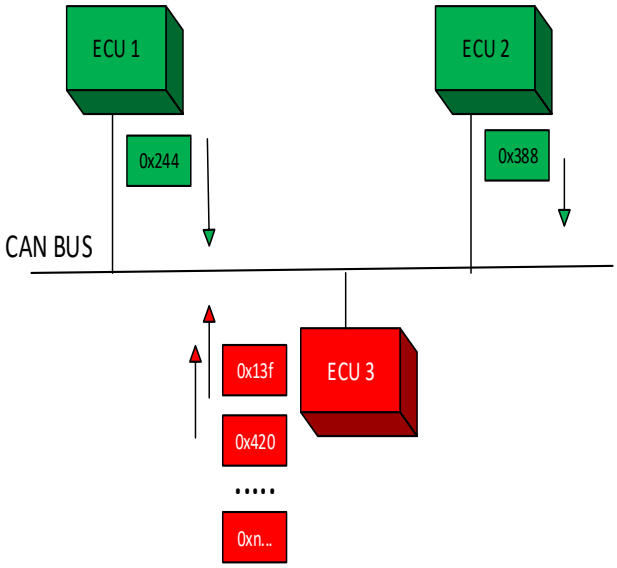

Figure 1. CAN bus fuzzing attack

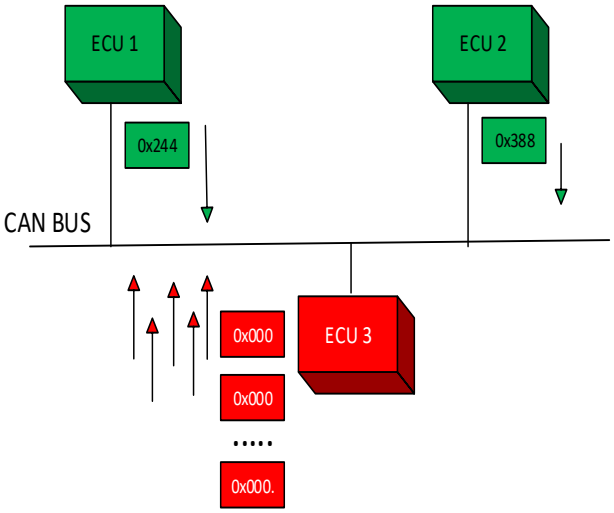

Figure 3. CAN bus denial of service attack

3.2. CAN Bus Spoofing Attack

This attack type exploits knowledge of the CAN ID to target specific vehicle components. Because the CAN protocol lacks strong data integrity and authentication measures, detecting such attacks is difficult. Moreover, the data volume is minimal, as the attacker aims to avoid overwhelming the system. For this reason, it is difficult to detect the attack. In order to prevent such attacks, it is important to integrate authentication mechanisms into the CAN network. Figure 2 shows a spoofing attack.

In the figure below, ECU 3 sends CAN data with predetermined IDs to the CAN bus. Since the data is not sent very quickly and the CAN IDs are legitimate, they are very difficult to detect.

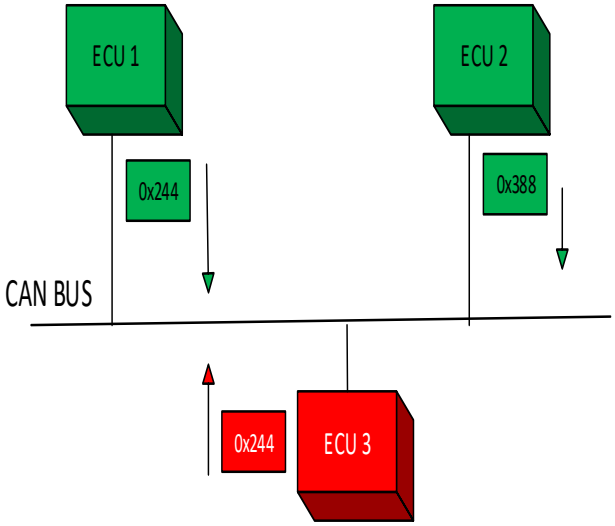

Figure 2. CAN bus spoofing attack

3.3. CAN Bus Denial of Service (DoS) Attack

3.4. On the CAN bus, messages are transmitted based on CAN ID priority [1], with ECUs using the arbitration field to determine which node takes precedence. In DoS attacks, a malicious actor can exploit this by repeatedly sending the highest priority CAN ID (0x000), blocking other nodes and disrupting communication. Denial of service is among the most common attacks targeting CAN networks [22]. DoS attacks are relatively easier to detect than other attacks because attackers usually attack using the highest priority CAN ID. Today, various detection algorithms based on machine learning and deep learning have been developed to prevent DoS attacks [23, 24]. However, since new attack methods emerge day by day, it has become necessary to develop new defense mechanisms. The operation of a denial of service attack is illustrated in Figure 3. In this example, the third electronic control unit rapidly transmits the highest priority CAN ID (0x000) onto the bus. Additionally, a DoS attack can be carried out by sending a legitimate CAN ID at an unusually high frequency, overwhelming the network and disrupting normal communication. Thus, the denial of service attack will be more difficult to detect.

4. In-Vehicle Network Datasets

There are various data sets for researchers to test their work on CAN networks. These data sets are obtained from real vehicles or vehicles that simulate CAN networks. In our study, the most commonly used CAN data sets are analyzed and compared.

4.1. OTIDS

Lee et al. [25] introduced the OTIDS dataset produced by HCRL in 2017. The dataset collected from the Kia Soul

vehicle includes DoS, Fuzzy, and Impersonation attacks. It contains 656,579 DoS, 591,990 Fuzzy, and 995,472 Impersonation attack entries, along with 2,369,868 normal CAN messages. Each record includes features such as timestamp, CAN ID, DLC, and payload. It is not recommended to use it because the injected messages are not labeled and the features specified by the authors are contradictory (for example, although it is stated that the Fuzzy attack started after 250 seconds, fuzzy attack data was encountered at 0.1565 seconds [26]).

4.2. HCRL CH

Another CAN dataset, HCRL Car-Hacking (HCRL CH) dataset, was developed by HCRL in 2018. It is one of the most widely used datasets by researchers while developing intrusion detection systems [27][28]. The dataset, collected from a Hyundai Sonata vehicle, includes 3,665,771 DoS, 3,838,860 Fuzzy, 4,443,142 gear spoofing, and 4,621,702 tachometer spoofing attack instances, along with 988,987 normal data entries. Each entry contains features such as timestamp, CAN ID, DLC, data, and a label indicating attack (T) or normal (R) status. However, there are gaps in the timestamps in this dataset. Therefore, it is not a good choice for researchers working especially in the field of time-based IDS. Berger et al. [29] stated the fluctuations in the timestamps in the HCRL CH dataset in their study.

4.3. HCRL SA (Survival Analysis Dataset)

The HCRL SA dataset developed by Han et al. [30] in 2018 includes attack and normal data from three different vehicles (Hyundai YF Sonata, KiaSoul and Chevrolet Spark). Han et al. developed an anomaly-based attack detection system from this dataset. The dataset includes DoS, Fuzzy and Malfunction (also known as frame spoofing attack) attack data. It has timestamp, CAN ID, DLC, data and label features. Attack datasets were created by injecting 5-second attack data every 20 seconds. It contains a total of 1 million 735 thousand data. It has disadvantages such as the attack data not being well hidden.

4.4. SYNCan

SYNCan, which was developed by Hanselmann et al. [8] in 2019 to be used in the test environment of their intrusion detection system, LSTM-Based CANet, is a signal-based dataset. U32nlike other datasets, SYNCan contains timestamp signal data instead of raw data. With these features, it is similar to the ROAD dataset. The dataset includes fabrication, suspension and masquerade attacks and contains a total of 41 million 900 thousand synthetic data. Since the authors do not provide a raw version of the data in the dataset, many CAN intrusion detection systems that need CAN data in normal format cannot use this dataset. In addition, synthetic datasets are not a good option for evaluating intrusion detection systems.

4.5. ROAD (Real ORNL Automotive Dynamometer CAN intrusion dataset)

ROAD, one of the most widely used datasets by researchers, was produced by Verma et al. [26] in 2020. It was created by taking data from a real vehicle for 3.5 hours and includes fuzzing, spoofing and masquerading attack data. The dataset consists of timestamp, CAN channel, CAN ID and data (in hexadecimal). The data was converted to signals and the CAN-D algorithm was used for this. SYNCan dataset and ROAD are the most widely used signal-based datasets. Kvaser Leaf Light V2 tool was used to collect the data and the data was collected using SocketCAN software. The authors refrained from giving information about the vehicle from which the data was collected and preserved the anonymity of the vehicle. There are a total of 28 million 245 thousand data in the dataset.

4.6. TTIDS

One of the latest CAN network datasets, TTIDS, was produced by Lee et al. [31] in 2022. This is the first dataset featuring a masquerading attack on a real vehicle using a bus-off technique. To generate the dataset, a suspension attack was first executed on the vehicle's CAN bus, followed by masquerading attacks. To perform the attacks, a message requesting the target ECU to be suspended is transmitted, while at the same time malicious CAN messages designed to be transmitted from the target ECU are injected. The dataset contains a total of 3 million 610 thousand data.

4.7. Our Dataset

We used ICSim [32], a CAN bus simulation tool, to create our dataset. The dataset we created includes DoS, fuzzy, speedometer attack, and normal state data. The CAN payload part of the dataset we created is divided into 8 bytes and each byte is determined as a feature. In addition, timestamp and class labels are also selected as features and consist of 11 features in total. There are 25750 speedometer spoof attacks, 61682 DDoS attack data, 13574 fuzzy attack data, and 42982 normal data, totaling 143,998 data.

Among the analyzed datasets, HCRL CH is still widely used, but it consists of easy attack scenarios obtained when the vehicle is parked. SynCAN dataset is the only dataset consisting of synthetic data, but the proposed dataset is unrealistic because they change the data labels to perform the attacks. Among the analyzed datasets, ROAD and HCRL CH are the most used datasets due to their labels and realistic attack scenarios.

5. Material and Method

In this study, cyber attacks were carried out on the simulated in-vehicle CAN network and the classification of these cyber attacks with machine learning models was carried out and the performance of these models was

compared.

5.1. Material

In this study, cyber attacks and algorithms used were tested on a computer with Intel(R) Core(TM) i7-7700HQ 3.6GHz CPU, 16 GB RAM. Kali Linux operating system was installed on this computer using VmWare, an operating system virtualization software. ICSim tool, which has the ability to simulate a CAN network, was installed on the Kali Linux operating system. With the ICSim tool, it is possible to simulate a CAN traffic, create new packets in the network traffic and record the traffic. The dashboard of the ICSim tool is shown in Figure 4. Using the control panel of the ICSim tool (on the left side), the speedometer, signal indicators and door open/closed information on the right side can be changed. The commands used in the control panel are given in Table 1. Using the commands in Table 1, the speed on the vehicle dashboard can be increased, right and left signals can be given, and the vehicle door locks can be opened and closed. While performing these operations, packets are sent to the CAN network in the background. The "cansniffer –c vcan0" command was used to see the simulated CAN network. Figure 5 shows the CAN network packets.

Figure 4. ICSim dashboard

Table 1. ICSim control commands

Function	Command
Accelerate	Up arrow key
Left/Right Signal	Left, right arrow keys
Unlock front left, right doors	Right shift+A, Right shift+B
Unlock rear left, right doors	Right shift+X, Right shift+Y
Lock all doors	Right shift+ Left shift
Unlock all doors	Left shift+ Right shift

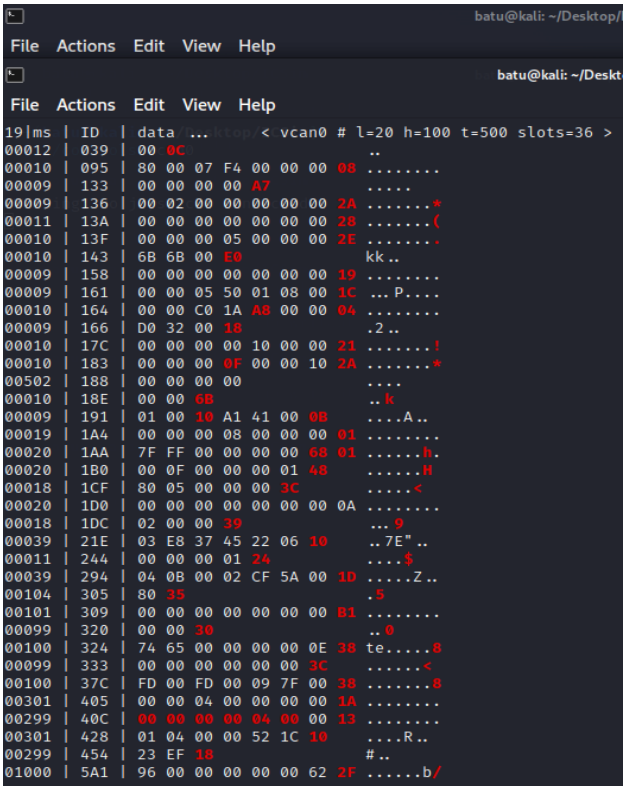

Figure 5. CAN network packets captured on ICSim

The obtained packets can be recorded with the "tcpdump" command or with Wireshark, a packet capture tool. In this study, the SavvyCAN tool was used to perform attacks on the CAN network. SavvyCAN is frequently used in areas such as reverse engineering applications and data analysis on the CAN bus [33]. Cyber attacks were performed on the virtual CAN network using the SavvyCAN tool, and then these packets were recorded with Wireshark, a packet capture tool. Then, these packets were separated according to their features and classified using classification algorithms on the Python programming language.

5.2. Method

In this study, an in-car network, the CAN bus, was simulated and Fuzzing, DoS and Speedometer attacks were made on this bus. Then, the attack packets were recorded and classified by machine learning classifiers and the performance of these classifiers was compared. In the first stage of the study, a Fuzzy, DoS and Speedometer spoofing attack was performed on a virtual CAN network created using the ICSim tool using the SavvyCAN tool. In a Fuzzy attack, the attacker sends random CAN ID and random CAN Data data to the target system. Since each CAN ID sent has a task in the system and the attack data is not sent very quickly in this type of attack, it is difficult for intrusion detection systems to detect these attacks. In order to perform the Fuzzy attack, CAN IDs between 0x133-0x188 were sent to the simulated CAN network using the SavvyCAN tool with a time interval of 0.5ms and

the attack was organized in a way that each CAN Data was different.

In a denial of service (DoS) attack, the attacker sends high priority (0x000) messages to the CAN network at a very high frequency, and the electronic control units that receive these messages respond to these high priority messages first. Since these malicious messages are sent at a very high frequency, after a while the system cannot respond to these messages and the system becomes disabled.

DoS attacks are easier to detect than other attacks because the attackers inject the highest priority messages with the CAN ID of 0x000 into the network. For this reason, in this study, in order to create a more realistic scenario, the performance of the classifiers was evaluated by sending the CAN ID data of 0x095, which is used quite intensively in normal network traffic, to the network at a frequency of 0.2ms.

Finally, a speedometer attack (speedometer spoofing) was performed. In order to perform this attack, it is necessary to know the CAN ID of the speed data in the vcan0 virtual CAN network that we created via ICSim. As a result of the studies conducted using Cansniffer, it was seen that the 0x244 CAN ID data was related to the speedometer [34]. The data in this CAN ID is randomly sent to the target network, changing the speedometer panel in the vehicle in a meaningless way. In the attack scenario we created, a speedometer spoofing attack was carried out by sending the 0x244 CAN ID data to the target network with a frequency of 1ms. While the attacks were being carried out, network packets were recorded using the Wireshark tool and this data was labeled according to the attack names. The data amounts and labels in the data set are shown in Table 2.

Features were extracted from the virtual CAN network dataset for classification, using each byte and the timestamp as features and it is illustrated inTable 3. XGBoost, Random Forest, LightGBM, and Decision Tree models were used to classify four classes (three attack types and one normal) on data generated with the ICSim tool. The dataset was split into 80% training and 20% testing. The classification models used in the study have different advantages. The decision tree algorithm is widely used because it has a simple structure and is easy to interpret. The risk of overfitting is low in the Random Forest algorithm and it performs better in noisy data sets. In addition, since it consists of more than one decision tree, its accuracy rate is high. The XGBoost algorithm was preferred due to its high performance in classification applications, reducing overfitting and operating fast by providing memory management. Finally, the LightGBM algorithm was preferred in this study because it showed high performance in large-sized data sets, its training and prediction processes are fast and it uses less memory compared to other classification algorithms.

Table 2. Attack classes and amount of the data

Label	Class	Amount of Data
0	Speedometer spoofing	25750
1	DDoS	61682
2	Fuzzy	13574
3	Normal	42982

Table 3. Extracted data features of our dataset

No. of Feature	Description
1	First byte of CAN data
2	Second byte of CAN data
3	Third byte of CAN data
4	Fourth byte of CAN data
5	Fifth byte of CAN data
6	Sixth byte of CAN data
7	Seventh byte of CAN data
8	Eleventh byte of CAN data
Frame.time_delta	Time difference between the frame and the previous frame

6. Findings and Discussions

In this study, a dataset was obtained from a simulated CAN network and then the data in this dataset was separated into its features and subjected to classification by LightGBM, RandomForest, XGBoost and Decision tree classifiers.

First, the classification of the LightGBM algorithm was applied. The experiments showed that the LightGBM algorithm predicted the DDoS and normal class with a high accuracy rate, but it could not show a satisfactory performance especially in predicting the fuzzy attack. Although the prediction rate of the speedometer spoofing attack was not as high as the DDoS and normal class, it gave a better result than the fuzzy attack. LightGBM had an overall accuracy rate of 85.83%.

Table 4 presents the model's performance metrics, while Figure 6 displays the confusion matrix. In the figure, the x-axis corresponds to the predicted labels and the y-axis to the actual labels.

The Decision Tree model showed lower overall performance compared to LightGBM. While its performance was similar in classifying the normal class, it performed significantly worse in detecting Fuzzy attacks.

It also showed slightly lower performance than LightGBM in classifying speedometer spoofing and DDoS attacks. It was seen that the best classification performance was obtained with 99% F1-score in the classification of the normal class, while the lowest performance was obtained with 44% F1-score in the classification of Fuzzy attacks. It has an accuracy rate of 80.70%. The performance values of the decision trees algorithm are given in Table 5. The confusion matrix is given in Figure 7.

Table 4. LightGBM performance metrics

Label	Class	Precision	Recall	F1-Score
0	Speedometer spoofing	%69	%82	%75
1	DDoS	%88	%86	%86
2	Fuzzy	%66	%47	%55
3	Normal	%99	%99	%99

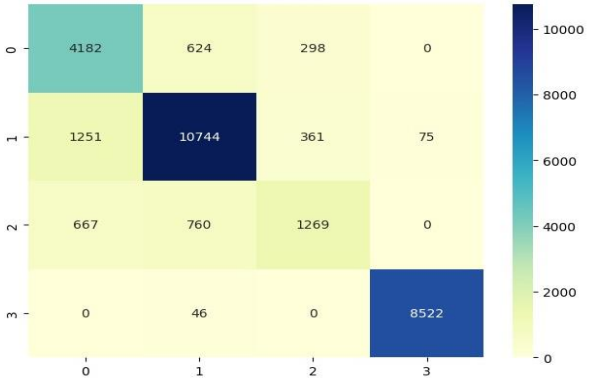

Figure 6. Confusion matrix of LightGBM algorithm

Table 5. Decision tree performance metrics

Label	Class	Precision	Recall	F1-Score
0	Speedometer spoofing	%65	%63	%64
1	DDoS	%83	%83	%83
2	Fuzzy	%43	%46	%44
3	Normal	%99	%99	%99

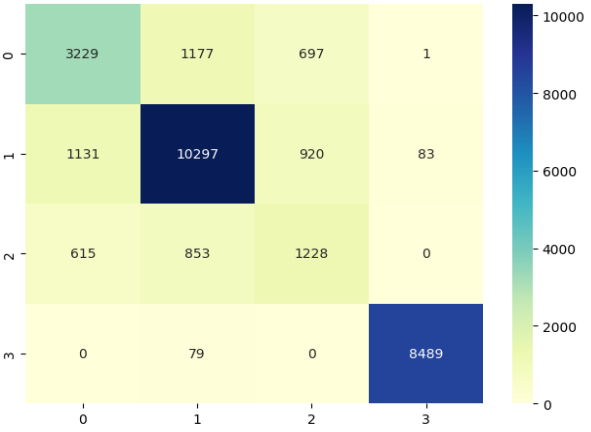

Figure 7. Confusion matrix of Decision Tree algorithm

Table 6. XGBoost performance metrics

Label	Class	Precision	Recall	F1-Score
0	Speedometer spoofing	%69	%82	%75
1	DDoS	%89	%86	%88
2	Fuzzy	%66	%51	%57
3	Normal	%99	%99	%99

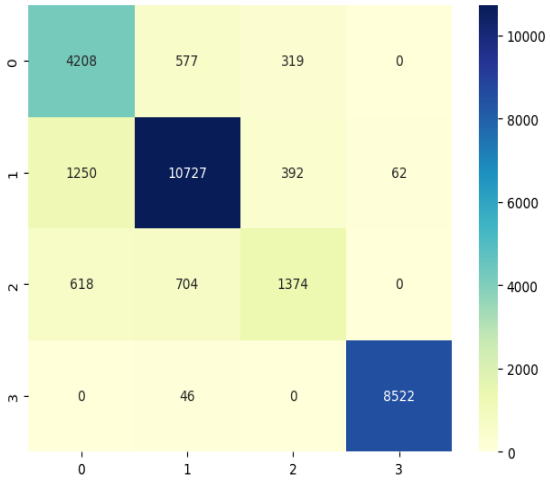

Figure 8. Confusion matrix of XGBoost algorithm

As in other models, the XGBoost model had the highest accuracy rate when predicting the normal class. The model achieved its highest performance in classifying the normal class, with an F1-score of 99%, while its lowest performance was observed in the fuzzy class, with an F1-score of 0.57. The overall accuracy of the model was calculated as 86.22%. Detailed performance metrics are presented in Table 6, and the confusion matrix is illustrated in Figure 8.Another algorithm was used in the study is the Random Forest. This algorithm generally shows similar features to LightGBM. It performed better than LightGBM in the classification of DDoS attacks and it was seen that it had very similar performance values in other classes. It performed much better than the decision tree algorithm. It was seen that it performed better than the decision tree algorithm, especially in speedometer spoofing and fuzzy attacks. It was understood that it showed its best performance in the classification of this class by providing 99% accuracy rate like other algorithms in the classification of the normal class. The model has an accuracy rate of 85.84%. It was the algorithm that showed the best performance after the XGBoost algorithm. The performance values of the model are given in Table 7. The confusion matrix is shown in Figure 9.

Table 7. Random Forest performance metrics

Label	Class	Precision	Recall	F1-Score
0	Speedometer spoofing	%68	%84	%75
1	DDoS	%89	%85	%87
2	Fuzzy	%65	%49	%56
3	Normal	%99	%99	%99

Figure 9. Confusion matrix of Random Forest algorithm

Table 8. Comparison of the machine learning models

Model	Accuracy	Precision	Recall	F1-Score
LightGBM	%85.83	%80.5	%78.5	%78.7
Decision Trees	%80.7	%72.5	%72.7	%72.5
XGBoost	%86.22	%80.7	%79.5	%79.7
Random Forest	%85.84	%80.2	%79.2	%79.2

The comparison of the algorithms used in the study is shown in table 8. Among the classification algorithms evaluated, XGBoost demonstrated the highest accuracy rate, achieving 86.22%.

It also performed better than other algorithms in other performance metrics such as Precision, Recall, F1-Score. The lowest accuracy rate among the compared algorithms belonged to the decision tree algorithm with 80.7%. It was seen that it performed lower than other algorithms in other parameters such as Precision, Recall, and F1-Score.

6. Conclusions

The rapid advancement of technology has led to the rise of smart vehicles, which enhance driving comfort and safety. However, they are also vulnerable to cyber threats. In this study, cyber attacks were simulated on an in-car CAN network using ICSim and SavvyCAN tools. Three types of attacks—speedometer spoofing, DDoS, and fuzzy—were performed, and the resulting data was classified using XGBoost, LightGBM, Random Forest, and Decision Tree models. The DDoS attack had the highest detection accuracy, while normal data was most accurately classified overall. Fuzzy attacks showed lower accuracy due to the use of legitimate CAN IDs and low packet frequency. Among the models, XGBoost achieved the highest accuracy at 86.22%, and Decision Trees the lowest at 80.7%.

This study aims to contribute to the detection of cyber attacks against smart vehicles and the development of new security strategies. The performances of the compared models can help cyber security experts who want to develop attack detection systems for smart vehicles in the future.

Smart vehicles are gaining more and more space in societies and their use is increasing with the development of technology today. However, the development of technology causes new attack platforms to emerge against these vehicles. Therefore, security experts should follow the innovations in the sector and new security measures should be considered and integrated from the design stage. In this study, cyber attacks were carried out by simulating a CAN network and network packets were recorded. The data obtained from the simulated network may not always be as accurate as the data in a real vehicle. In addition, the process of extracting features from simulated data is more difficult than real vehicle network data. Therefore, it is recommended that future researchers work on data obtained from real vehicles.

The highest accuracy rate in the study was obtained with the XGBoost algorithm. This algorithm is faster than other algorithms in large data sets and can be preferred due to its low memory usage. Although the decision tree algorithm has a simple structure and is fast, it is not recommended because it has a low accuracy rate.

In this study, an intrusion detection system is developed to protect the in-vehicle CAN network against cyber attacks and this system is tested with a dataset obtained using ICSim, a CAN network simulation tool. It is important to acknowledge that although synthetic datasets provide good results, they may not fully reflect the complexity and noise of real-world data. Real CAN traffic may vary due to unpredictable data traffic between ECUs, environmental conditions, etc. Therefore, future studies should aim to use real CAN network datasets.

Security mechanisms alone may not be sufficient to protect smart vehicles against cyber threats. In addition, driver awareness campaigns and raising awareness on this issue are of great importance.

Declaration

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article. The authors also declared that this article is

original, was prepared in accordance with international publication and research ethics, and ethical committee permission or any special permission is not required.

Author Contributions

B. Gul contributed to the writing of the original draft and experiments. F. Ertam conceptualized the study and was involved in supervising.

References

1. Huang, T., J. Zhou, Y. Wang, and A. Cheng, *On the security of in-vehicle hybrid network: Status and challenges*, in Lecture Notes in Computer Science, 2017. Melbourne: p.621-637.
2. Chakraborty, S., M. Lukasiewycz, C. Buckl, S. Fahmy, N.Chang and S. Park, *Embedded systems and software challenges in electric vehicles*, in *2012* Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012. Dresden: p. 424–429.
3. Du, X., S. Jiang, D. Zhou, A. B. Milhim, and H. Sadjadi, *Ground Fault Diagnostics for Automotive Electronic Control Units*, Int. J. Progn. Heal. Manag., 2023. **14**(3): p. 1-13.
4. Buttigieg, R., M. Farrugia, and C. Meli, *Security issues in controller area networks in automobiles*, in 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), IEEE, Dec. 2017. Monastir: p. 93–98.
5. Shaw, R. and B. Jackman, *An introduction to FlexRay as an industrial network*, in IEEE International Symposium on Industrial Electronics, IEEE, Jun. 2008. Cambridge: p. 1849–1854.
6. Wolf, M., A. Weimerskirch, and C. Paar, *Security in Automotive Bus Systems*, Proc. Work. Embed. Secur. Cars, 2004. **2004**: p. 1–13.
7. Hossain, M. D., H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, *LSTM-Based Intrusion Detection System for In-Vehicle Can Bus Communications*, IEEE Access, 2020. **8**: p. 185489–185502.
8. Hanselmann, M., T. Strauss, K. Dormann, and H. Ulmer, *CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data*, IEEE Access, 2020. **8**: p. 58194–58205.
9. Jin, S., J.-G. Chung, and Y. Xu, *Signature-Based Intrusion Detection System (IDS) for In-Vehicle CAN Bus Networ*k, in 2021 IEEE International Symposium on Circuits and Systems (ISCAS), IEEE, May 2021. Daegu: p. 1–5.
10. Islam, R., M. K. Devnath, M. D. Samad, and S. M. Jaffrey Al Kadry, *GGNB: Graph-based Gaussian naive Bayes intrusion detection system for CAN bus*, Veh. Commun., 2022. **33**: p. 100442.
11. Deng, Z., J. Liu, Y. Xun, and J. Qin, *IdentifierIDS: A Practical Voltage-Based Intrusion Detection System for Real In-Vehicle Networks*, IEEE Trans. Inf. Forensics Secur., 2024. **19**: p. 661–676,
12. Khandelwal, S., E. Wadhwa, and S. Shreejith, *Deep Learning-based Embedded Intrusion Detection System for Automotive CAN*, in Proceedings of the International Conference on Application-Specific Systems, Architectures and Processors, 2022. Gothenburg: p. 88–92
13. Desta, A. K., S. Ohira, I. Arai, and K. Fujikawa, *Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots*, Veh. Commun., 2022. **35**: p. 100470.
14. Yu, Z., Y. Liu, G. Xie, R. Li, S. Liu, and L. T. Yang, *TCE-IDS: Time Interval Conditional Entropy- Based Intrusion Detection System for Automotive Controller Area Networks*, IEEE Trans. Ind. Informatics, 2023. 19(**2**): p. 1185–1195.
15. Tanksale, V., *Intrusion detection system for controller area network*, Cybersecurity, 2024. **7**(1): p. 1-21.
16. Ye, P., Y. Liang, Y. Bie, G. Qin, J. Song, Y. Wang and W. Liu, *GDT-IDS: graph-based decision tree intrusion detection system for controller area network*, J. Supercomput., 2025. **81**(4): p. 591.
17. Checkoway, S., D. McCoy, B. Kantor,D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, *Comprehensive experimental analyses of automotive attack surfaces*, in Proceedings of the 20th USENIX Security Symposium, 2011. San Fransisco: p. 1-16.
18. Petit, J., B. Stottelaar, M. Feiri, and F. Kargl, *Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR*, [cited 2025 14 May]; Available from: Blackhat.com, p. 1–13.
19. Aliwa, E., O. Rana, C. Perera, and P. Burnap, *Cyberattacks and Countermeasures for In-Vehicle Networks*, ACM Comput. Surv., 2021. **54**(1): p. 1–37.
20. Deng, J., L. Yu, Y. Fu, O. Hambolu, and R. R. Brooks, *Security and Data Privacy of Modern Automobiles*, in Data Analytics for Intelligent Transportation Systems, Elsevier, 2017. **2017**: p. 131–163.
21. Bari, B. S., K. Yelamarthi, and S. Ghafoor, *Intrusion Detection in Vehicle Controller Area Network (CAN) Bus Using Machine Learning: A Comparative Performance Study*, Sensors, 2023. **23**(7): p. 3610.
22. Gao, S., L. Zhang, L. He, X. Deng, H. Yin, and H. Zhang, *Attack Detection for Intelligent Vehicles via CAN- Bus: A Lightweight Image Network Approach*, IEEE Trans. Veh. Technol., 2023. **72**(12): p. 16624–16636.
23. Lo, W., H. Alqahtani, K. Thakur, A. Almadhor, S. Chander, and G. Kumar, *A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic*, Veh. Commun., 2022. **35**: p. 100471.
24. Wang, K., A. Zhang, H. Sun, and B. Wang, *Analysis of Recent Deep-Learning-Based Intrusion Detection Methods for In-Vehicle Network*, IEEE Trans. Intell. Transp. Syst., 2022. **24**(2): p. 1–12.
25. Lee, H., S. H. Jeong, and H. K. Kim, *OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame*, in Proceedings - 2017 15th Annual Conference on Privacy, Security and Trust, PST 2017, IEEE. 2018. Calgary: p. 57–66.
26. Verma, M. E., R. A. Bridges, M. D. Iannacone, S. C. Hollifield, P. Moriano, S. C. Hespeler, B. Kay and F. L. Combs, *A comprehensive guide to CAN IDS data and introduction of the ROAD dataset*, PLoS One, 2024. **19**(1): p. e0296879.
27. Seo, E., H. M. Song, and H. K. Kim, *GIDS: GAN based Intrusion Detection System for In-Vehicle Network*, in 2018 16th Annual Conference on Privacy, Security and Trust (PST), IEEE. 2018. Belfast: p. 1–6.
28. Rajapaksha, S., H. Kalutarage, M. O. Al-Kadri, A. Petrovski, G. Madzudzo, and M. Cheah, *AI-Based Intrusion Detection Systems for In-Vehicle Networks: A Survey*, ACM

Comput. Surv., 2023. **55**(11): p. 1–40.

29. Berger, I., R. Rieke, M. Kolomeets, A. Chechulin, and I. Kotenko, *Comparative study of machine learning methods for in-vehicle intrusion detection*, in Lecture Notes in Computer Science, 2018. Barcelona: p. 85-101.
30. Han, M. L., B. Il Kwak, and H. K. Kim, *Anomaly intrusion detection method for vehicular networks based on survival analysis*, Veh. Commun., 2018. **14**: p. 52–63.
31. Lee, S., H. J. Jo, A. Cho, D. H. Lee, and W. Choi, *TTIDS: Transmission-Resuming Time-Based Intrusion Detection System for Controller Area Network (CAN)*, IEEE Access, **10**: pp. 52139–52153.
32. GitHub - zombieCraig/ICSim: Instrument Cluster Simulator. Accessed: May 25, 2024. [Online]. Available: https://github.com/zombieCraig/ICSim.
33. GitHub - collin80/SavvyCAN: QT based cross platform canbus tool. Accessed: May 26, 2024. [Online]. Available: https://github.com/collin80/SavvyCAN.
34. Mercaldo, F., R. Casolare, G. Ciaramella, G. Iadarola, F. Martinelli, F. Ranieri and A. Santone, *A Real-time Method for CAN Bus Intrusion Detection by Means of Supervised Machine Learning,* in Proceedings of the International Conference on Security and Cryptography, 2022. Lisbon: p. 534-539.