ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

CYBERSPACE FOR DETECTING ATTACKS IN AUTONOMOUS VEHICLES BASED APPROACHES

MOHAMMED Y. ALZAHRANI[1,*]
[1]Department of Information Technology, Faculty of Computer Science & Information Technology, AlBaha University, AlBaha, Saudi Arabia, imohduni@gmail.com

ID 54650 Submission	Editorial Screening	Conditional Acceptance	Final Revision Acceptance
25-05-2024	25-05-2024	31-05-2024	14-10-2024

ABSTRACT

With the advancement of technology, cities have progressively grown more intelligent. Smart mobility is a vital component of smart cities, and autonomous vehicles play a fundamental role in enabling smart mobility. Nevertheless, the presence of vulnerabilities in autonomous cars might have a detrimental impact on both the overall quality of life and the safety of individuals. Consequently, several security researchers have examined both offensive and defensive strategies against autonomous cars. Machine learning (ML) and deep leaning (DL) is used in these mobile robots to automate repetitive driving chores and make judgments based on their understanding of the scenario. This research presents showcases the utilization of adversarial instances in connected autonomous vehicles (CAVs) to illustrate how adversarial ML and DL techniques are extremely to detect CANs attacks. The CAVs security system was developed using a dataset acquired from standard research. The dataset includes five types of attacks along with normal packets. The decision tree (DT), extra tree, and Gated Recurrent Units (GRUs) were utilized to identify cyber CAVs threats. The empirical data indicate that the DT technique produced a 99% accuracy rate, while the extra tree and GRU achieved 98% and 96% respectively. Technology demonstrates potential in safeguarding vital infrastructure through the analysis of adversary methods. With near-perfect precision, the performance of all the models constructed in this manner outshone that of previously used models. When it comes to in-vehicle networks (IVN) security, the created system is up to the task.

Keyword: *Cybersecurity, Decision Tree, Gated Recurrent Units, Autonomous Vehicle*

1.INTRODUCTION

Contemporary automobiles have surpassed their conventional purpose as simple means of transportation, evolving into intricate electronic systems on wheels, equipped with powerful sensors, linked networks, and software-driven capabilities. In the last decade, automobiles have acquired a growing array of technical functions and capabilities, resulting in enhanced intelligence and efficiency. However, a new kind of security risk has emerged as a result of these technological advancements: cybersecurity vulnerabilities, which may affect vehicles. Automatic gearshifts, power steering, and climate control were just a few of the modern amenities that cars started to include during the middle of the twentieth century. Modern fuel injection and ignition timing controls were made possible with the advent of electronic control units (ECUs) in the 1970s [1]. This concept had a pivotal influence on enhancing fuel efficiency, as well as implementing safety measures such as seatbelts and mitigating pollutants. In addition, several automobiles are now equipped with contemporary safety technologies such as Anti-lock Braking Systems (ABS) and airbags, which have become customary, significantly augmenting the safety of both motorists and occupants[1].

Connected autonomous vehicles (CAVs) are highly dependent on the data collected by sensors. Aberrant sensor data resulting from inaccuracies or cyber intrusions can lead to significant ramifications, such as system failures and disruptions in traffic flow. CAV systems are susceptible to several forms of internal and external cyberattacks [2].

Automobile cyber security refers to the measures taken to safeguard electronic systems, control units, communication networks, and user data from hostile assaults. Security is essential due to the fact that modern cars are essentially computerized vehicles rather than just mechanical devices. If a car were to have a vulnerability, a hacker might readily exploit it, perhaps leading to unpleasant consequences To prove that it is possible to hack a car by deciphering its CAN network, the authors [4] performed an experiment on a real vehicle in 2010. Examine the below instances to demonstrate the importance of cyber security. In 2016, researchers discovered a method to exploit the Wi-Fi interface and many software flaws to compromise the security of the Tesla Model S vehicle [5]. These examples demonstrate that hackers are particularly interested in targeting the weak components of cars for hacking purposes.

The purpose of this project is to advance the field of cybersecurity in autonomous cars by investigating and suggesting advanced ML and DL techniques. In order to keep the autonomous vehicle network reliable and secure, this project aims to establish a new standard for detecting and, hopefully, preventing assaults on autonomous automobiles. IDS refers to the systematic monitoring and analysis of network or computer system activities in order to identify and thwart attempts at unauthorized access [5]. The method seeks to identify signs of prospective intrusions, such as attempts to circumvent security barriers or gain illegal access [6].

Due to their large attack surfaces, self-driving cars are becoming more susceptible to cyberattacks that aim to compromise either their internal or external networks as they grow more advanced and linked. The extensive use of computers and communication networks makes autonomous cars an easy target for cybercriminals [7]. There are vulnerabilities in these systems that might be used by hackers to gain unauthorized access, steal data, or even take over the car [8]. These attacks might lead to accidents, deaths, and significant property damage. Figure 1 provides a comprehensive summary of an IDS designed specifically for autonomous cars. The text emphasizes essential features, such as the ability to filter and detect in real-time. It has the capability to identify and counteract both sensor-based and communication-based assaults, a crucial aspect in ensuring the security of autonomous cars.

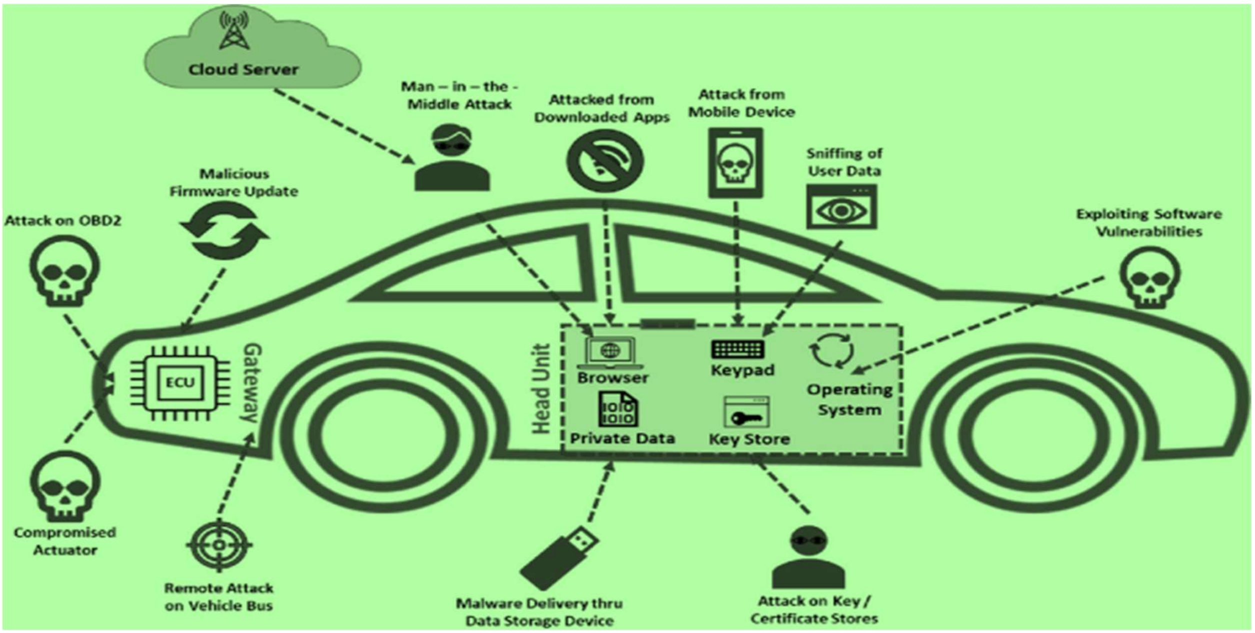
Cloud Server
Man – in – the - Middle Attack
Attacked from Downloaded Apps
Attack from Mobile Device
Sniffing of User Data
Malicious Firmware Update
Attack on OBD2
Exploiting Software Vulnerabilities
ECU
Gateway
Head Unit
Browser
Keypad
Operating System
Private Data
Key Store
Compromised Actuator
Remote Attack on Vehicle Bus
Malware Delivery thru Data Storage Device
Attack on Key / Certificate Stores

Figure 1. IDS In Autonomous Cars

The CAN bus is a protocol that enables reliable and prioritized communication between ECUs in vehicles and other devices. All devices in the network get messages or "frames" without the need for a host computer. The CAN is backed by a comprehensive collection of global standards outlined in ISO 11898. Figure 2 shows the structure of CAN bus.

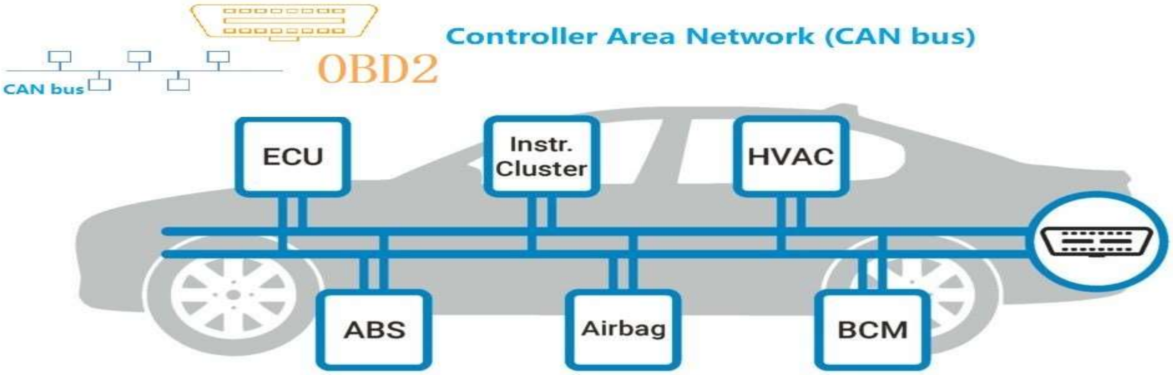

Figure 2. CAN Bus [9]

With the help of potential attack messages and cybersecurity measures, the suggested method aims to address the information security issues with CAVs. To successfully handle cyber threats to IVN communication, an ML and DL framework is a potential option. Given the extensive usage of CAVs in several countries and their incorporation into daily social activities, effective intrusion detection from in-vehicle networks' congestion is vital.

2.BACKGROUND OF STUDY

IDS developed for use in autonomous vehicles are the subject of much recent academic study, which is summarized below. Researchers looked at potential weak spots in both private and public networks in their research. A signature-based IDS and an anomaly-based IDS are combined in their suggested multi-tiered hybrid architecture. Attacks on car networks, both old and new, will be spotted by this system.

To identify irregularities in automobile networks, a new IDS for CAN was suggested in reference [10]. One may learn about the intrusion patterns and behaviors that are intrinsic to a system by studying the statistical features of attacks. Experiments proved that the proposed method can accurately and consistently identify denial-of-service (DoS) attacks, even fuzzy ones, with a negligible amount of false positives. It has been shown that the overall inaccuracy lowers for various window widths as the occurrence of assaults rises. Based on the findings, the suggested IDS is a better fit for in-vehicle networks as it successfully lowers the rate of misclassification. The experimental results show that the proposed IDS can successfully distinguish between valid and malicious data in CAN-BUS systems. With a recall rate of 99.64% and an F1-score of 99.56%, the IDS accomplished its goal. In order to identify an attack on the ECU, the suggested IDS compares the window vectors with predefined normality values. Nevertheless, if the infected ECU initiates an attack before returning normal numbers, anomaly diagnostics might provide false findings. The adaptive neuro-fuzzy inference system (ANFIS) and convolutional neural networks (CNNs) were used in the development of a very effective intelligent IDS in [11]. At the moment, methods mostly target known attacks in the context of vehicular ad hoc networks (VANETs). Intelligent intrusion IDS and soft-computing techniques alleviate the limitation. Modules called IDS are part of the suggested solution; they can detect both known and unknown threats, even ones that haven't been found yet. While the module uses ANFIS classification to find known malicious assaults, the UIDS module uses deep learning to find unknown threats in the VANET. According to the attack detection rate (ADR) research, 98.5% of harmful assaults, 99.7% of port scan attacks, 93.9% of brute force attacks, and 98.9% of botnet attacks were all effectively discovered. Without deep learning and sophisticated key management, the IDS's performance on the VANET was subpar.

In [12], the authors presented a novel IDS that integrates deep learning, thresholding, and error reconstruction methods. They trained and tested a wide variety of neural network topologies and then compared their results. Four distinct assault types were used to evaluate the proposed anomaly detection system: fuzzy, RPM spoofing, gear spoofing, and denial of service (DoS). Accuracy, recall, and F1-scores were the metrics used for evaluation. Pretty much every time we tested the deep learning-based model, we got a performance level over 99.90%. As far as prediction times go, it performed well, with a scant 128.73 ms forecast time. Using a hybrid model that included gradient descent momentum and adaptive gain, Zhang et al. [13] developed a system to identify CAN bus intrusions and categorize the messages linked to these attacks. In order to detect intrusions and keep an eye on the CAN bus message frames, Liang et al. [14] used a system that was based on deep neural networks. In order to train the deep learning model, the deep-belief network function was used. With a 98% success rate, the proposed approach is clearly cutting it. The CAN bus IDS was developed by Hoppe et al. [15] to analyze data packets sent across a network and spot novel trends. After that, they checked the IDS system for preexisting patterns that matched these ones. Their method achieved a remarkable degree of accuracy when compared to the conventional methodology. One method that Taylor and colleagues [16] suggested for detecting CAN bus assaults is an LSTM model. To build a distributed anomaly classification system, Wang et al. [17] Used hierarchical temporal memory method. The results of the empirical study shows that able to detect attacks. Potential CAN bus intrusions have been predicted using a variety of ML and DL methods. The deep neural network [18,19], CNNs [20], and ANNs used to build adversarial attacks [21] are all examples of such systems.

3.METHDOLOGY

With the fast development of self-driving cars, several firms have encountered difficulties in safeguarding the CAV system against attacks, leading to a range of problems on the road. Several research have examined methods to ensure the security of systems, but there is a deficiency in the algorithm required to achieve optimal performance. For this work, we employed new methodologies based on ML and DL approaches on actual datasets of connected and autonomous vehicles (CAVs). Figure 3 depicts the suggested framework for identifying and thwarting attacks on a connected and autonomous vehicle (CAV) network.

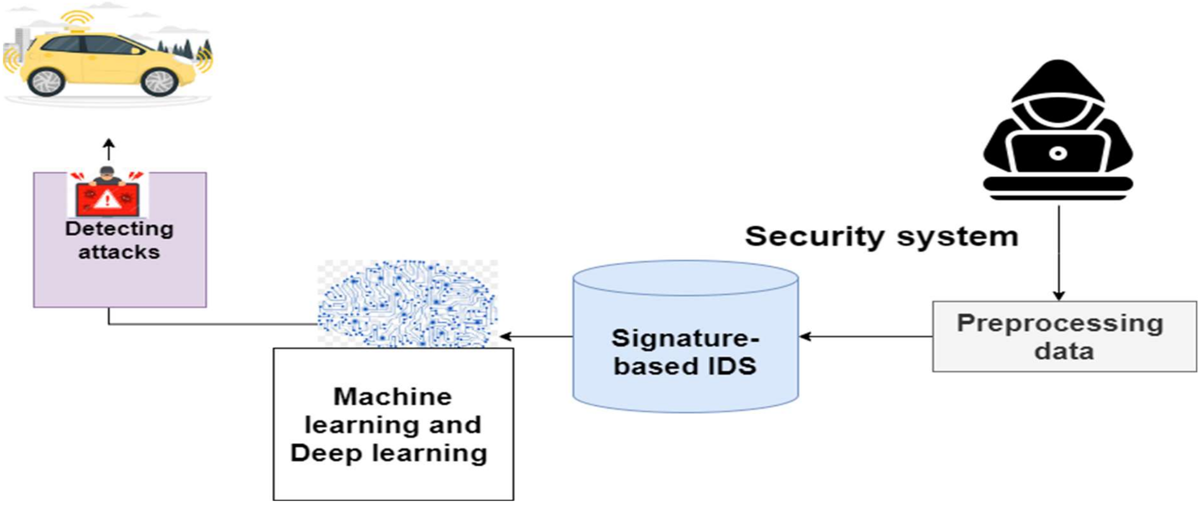

Figure 3. Proposed System

3.1 Dataset

The CAV dataset was compiled using authentic CAN traffic data, which encompasses spoofing, flood, and replaying attacks, as well as benign packets. The dataset was created by constructing a Controller Area Network (CAN) traffic On-Board Diagnostics (OBD-II) port using a genuine Connected and Autonomous Vehicle (CAV). The injected messages included several forms of attack messages. The Open Car Testbed and Network Experiments (OCTANE) utilized the CAN packet generator. The incursions occurred at regular intervals of 3 to 5 seconds, and the CAV traffic lasted for a duration of 30 to 40 minutes. Table 1 displays the occurrence of an injection attack on CAN traffic.

Table 1. Attacks of dataset

Flooding_attack	3,665,771
Fuzzing_attack	4,443,142
Normal	4,621,702
Spoofing (gear)_attack	3,838,860

3.3 Preprocessing approach

The categorical variables were translated and then the maximum-minimum normalization procedures were applied to eliminate any potential overlap in the training process resulting from manipulating huge datasets. As part of the normalization procedure, we applied scaling to the dataset using a range of 0 to 1. This was done to ensure that all values are within a consistent range.

$$z_n = \frac{x - x_{min}}{x_{max} - x_{min}} \qquad (1)$$

The variable x_{min} represents the smallest value in the dataset. The variable x_{max} represents the highest value in the dataset.

3.4 algorithms

3.4.1 Decision tree

When it comes to supervised learning algorithms, decision trees are your best bet for solving classification and regression issues. Following a flowchart-like layout, the algorithm builds a hierarchical structure where each internal node stands for a property test, each branch for a potential result of the test, and every leaf node (terminal node) has a class label. Starting with the maximum tree depth or the minimum number of samples needed to split a node, the building procedure repeatedly divides the training data into subsets using attribute values until a stopping criteria is fulfilled.

Using metrics like entropy or Gini impurity, the Decision Tree algorithm selects the best characteristic to partition the data for training. These measures provide a numerical representation of the subsets' level of impureness or unpredictability. Finding the attribute that minimises impurity after the split or maximizes information gain is the goal.

$$Entropy = (S) = \sum_{i=1}^{C} p_i \ log_2 \, p_i \qquad (2)$$

$$entropy \ (S \ |B) = \sum_{j=1}^{j} \frac{|s_i|}{|S_i|} \ entropy \ (S_i) \qquad (3)$$

$$\text{Gain} \ (S \ |B) = entropy(S) - entropy(S \ |B) \qquad (4)$$

Where can I find a training dataset that includes both attack and normal classes? P_i represents the probability of a simple event indicating class C. S_i: represents the samples of subsets of a class in the feature set B.

A decision tree is a hierarchical structure like a flowchart with core nodes representing features, branches representing rules, and leaf nodes representing algorithm results. The flexible supervised machine learning method may be used for classification and regression. This algorithm is powerful. Random Forest also uses this method to train on different subsets of training data, making it one of the most powerful machine learning algorithms.

3.4.2 Extra tree approach

An ensemble learning technique, the Extra Trees Classifier classifies using the results of multiple uncorrelated decision trees in a "forest". Its main difference from a Random Forest Classifier is how it builds decision trees within the forest. The Extra Trees Forest builds every Decision Tree from the original training sample. Every tree receives a random sample of k feature-set features at each test node. Each decision tree then chooses the best feature to partition the data using a mathematical criteria, generally the Gini Index. The random selection of features creates numerous uncorrelated

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

decision trees. The forest structure is formed and the normalized total decrease in the mathematical criterion used to separate the feature (Gini Index if utilized) is computed for each feature to choose features. This value is called "Gini". Characteristic significance.

3.4.3 Gated Recurrent Unit (GRU)

The GRU is an RNN design that can replace LSTM networks, simplifying things. Like LSTM, GRU can process sequential text, audio, and time-series data. GRU works by selectively changing the network's hidden state at each time step via gating. Gate systems may regulate network data entry and exit. The GRU has two typical gates: reset and update. The reset gate chooses how much to ignore the prior concealed state, while the update gate controls how much the current input changes it. From the updated hidden state, the GRU calculates output. The GRU's design is shown Figure 4.

These two gates together govern network data flow. The update gate determines how much past data to process. The reset gate decides how much data to delete. The equations below summarize a GRU.

$$\mu_t = \sigma(V_\mu x_t + W_\mu o_{t-1} + b_\mu) \quad (5)$$

$$r_t = \sigma(V_r x_t + W_r o_{t-1} + b_\mu) \quad (6)$$

$$i_t = tanh(V_o x_t + W_o(r_t \odot o_{t-1}) + b_0) \quad (7)$$

$$o_t = \sigma(\mu_t \odot o_{t-1} (1 - \mu_t) \odot i_t) \quad (8)$$

The variables in the equation are defined as follows: x_t represents the input, o_t represents the output, μ_t represents the output of the update gate, r_t represents the output of the reset gate, and the $\odot$ symbol indicates the Hadamard product. The parameters or weight matrices are denoted by V, W, and b.

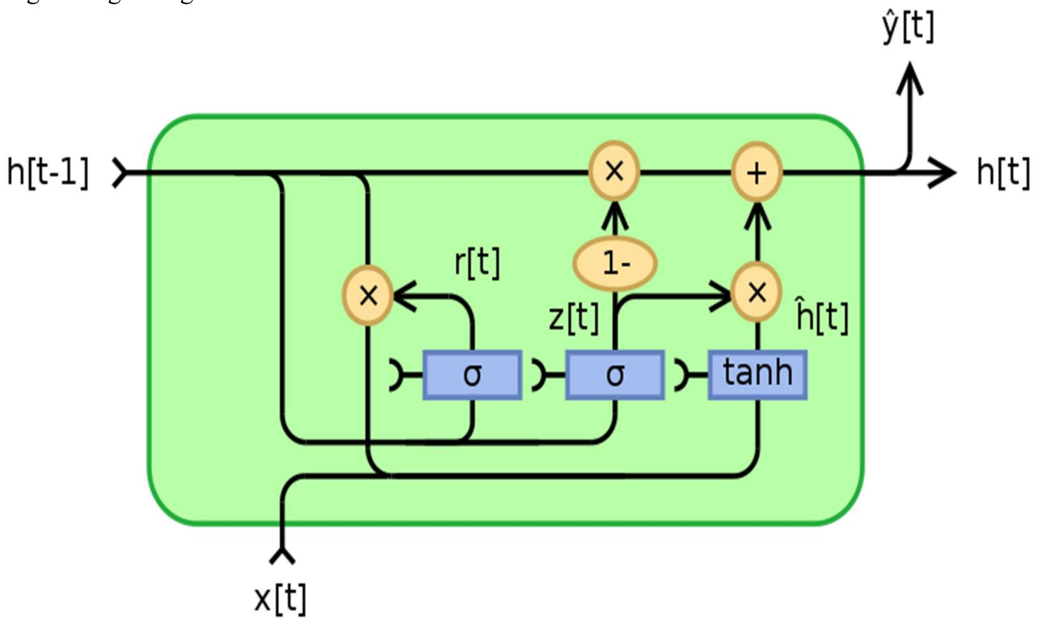

Figure 4. GRU Model

3.5 Evaluation metrics

Assessing the performance of ML and DL models is crucial for gauging the effectiveness of the models. There are several metrics used to quantify performance, including as accuracy, sensitivity, precision, recall, F1 score, ROC curve, and confusion matrix. The evaluation measures provide an alternative perspective on the model's advantages and disadvantages.

$$Accuracy = \frac{TP+TN}{FP+FN+TP+T} \times 100 \quad (9)$$

$$F1 - score = 2 * \frac{precision \times Recall}{precision + Recall} \text{x}100\% \qquad (10)$$

$$Recall = Sensitivity = \frac{True\ Positives}{True\ Positives + \ False\ positives} x100\% \qquad 11)$$

$$Specificity = \frac{True\ \ Negatives}{True\ Negatives + False\ Negatives} x100\% \qquad (12)$$

4. EXPERIMENTAL RESULTS

This section presents the findings from different investigations investigating the effectiveness of machine learning and deep learning techniques in detecting cyberattacks in networks of connected and autonomous vehicles (CAVs). We specifically look at how well the Decision Tree, Extra Tree, and GRU approaches perform in identifying malicious activity in CAV communication networks. A dataset consisting of network traffic logs gathered from simulated CAV settings is used in these investigations. Our goal is to evaluate each method's accuracy in identifying different kinds of assaults, such as replay attacks, flooding, fuzzing, and spoofing. We use important metrics including precision, recall, and F1-score to assess each approach's robustness and accuracy.

4.1 *Decision Tree Classification Results*

This subsection introduces the Decision Tree classification report which are presented in table 2below demonstrates strong predicting across a variety of attack types on the Connected Autonomous Vehicle (CAV) network. With excellent precision and recall scores for "Flooding" and "Spoofing" attacks, the model identifies these threats correctly.

Table 2: Testing Classification Report Of The Decision Tree Model..

Attack Type	Precision	Recall	F1-score	Support	Accuracy
Flooding	100	100	100	7727	99
Fuzzing	97	94	95	4525	
Normal	99	100	99	146721	
Replay	72	46	0.556	2114	
Spoofing	100	1.00	1.00	191	

However, its performance is slightly inferior for "fuzzing" and "replay" attacks, which have lower precision and recall values. Despite these variances, the Decision Tree model has an overall accuracy of 99%, demonstrating its usefulness in detecting cyber risks in the CAV environment.

4.2 Extra Tree classification Results

Similar to the Decision Tree model, the Extra Trees classification report provides strong performance across several attack types within the CAV network. The model has good precision and recall scores for "Flooding" and "Spoofing" attacks, showing that it can reliably recognize these threats. However, its performance is slightly poorer for "Fuzzing" and "Replay" attacks, which have lower precision and recall values. Despite these variations, the Extra Trees model retains an overall accuracy of 99%, demonstrating its efficacy in detecting cyber threats in the CAV environment. Table 3summarizes the testing classification results of the Extra Trees model.

Table 3: Testing Classification Report Of The Extra Trees Model

Attack Type	Precision	Recall	F1-score	Support	Accuracy
Flooding	100	100	100	7727	99
Fuzzing	98	94	96	4525	
Normal	99	100	99	146721	
Replay	73	47	57	2114	
Spoofing	100	100	100	191	

4.3 GRU classification Results

The GRU model classification report presented in table 4 shows that the model performance varies depending on the type of attack. The model performs noticeably worse for "Fuzzing," "Replay," and "Spoofing" attacks than it does for "Flooding" attacks, when it achieves flawless precision and recall. Notably, the "Fuzzing" attack precision and recall scores indicate that the model has difficulty correctly identifying occurrences of this attack type. In a similar vein, the model's recall and precision levels for "Replay" and "Spoofing" assaults are below par. The GRU model's total accuracy of 96% is maintained in spite of these difficulties, highlighting its capacity to identify cyberthreats in the CAV environment.

Table 4: Testing Classification Report Of The GRU Model

Attack Type	Precision	Recall	F1-score	Support	Accuracy
Flooding	100	100	100	7727	99
Fuzzing	000	000	000	4525	
Normal	96	100	98	146721	
Replay	96	003	005	2114	
Spoofing	000	000	000	191	

These reports provide insights into the performance of each model in detecting cyberattacks within the CAV network, enabling stakeholders to make informed decisions regarding security measures and system improvements. Figure 5 displays the performance of GRU model.

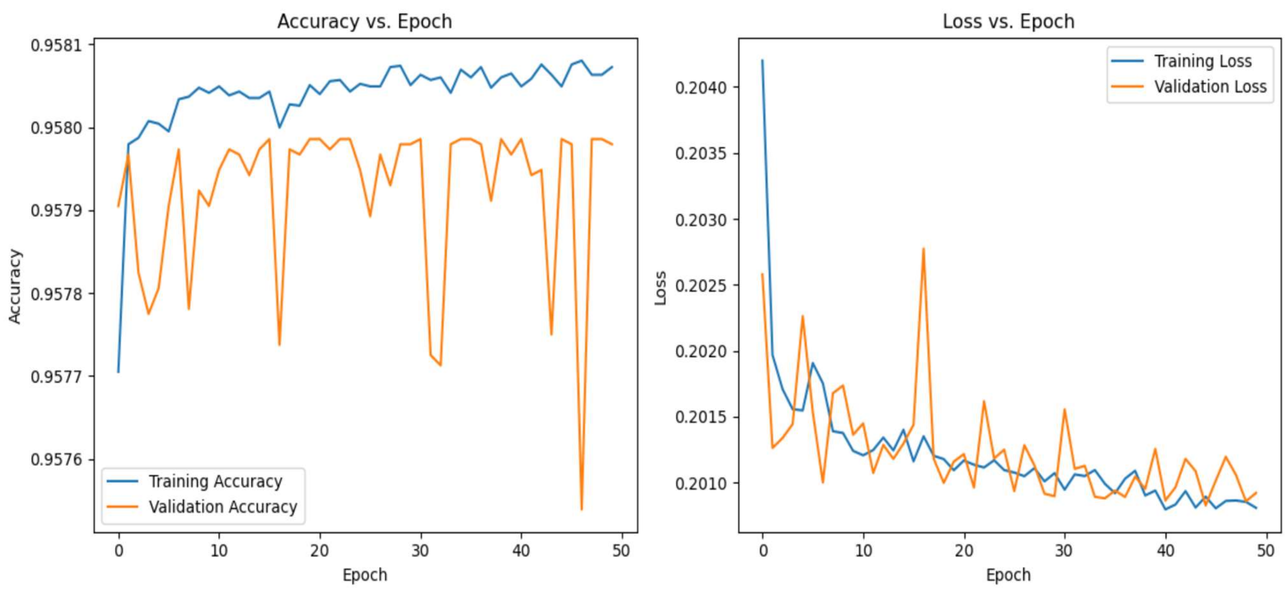

Figure 5. Accuracy Of GRU Model

5. RESULTS DISCUSSION

The subsection gives discussion of the results obtained from the different experiments employing Decision Tree, Extra Trees, and GRU (Gated Recurrent Unit) models for detecting cyberattacks in connected and autonomous vehicle (CAV) networks provide valuable insights into the performance of these algorithms.

Starting with Decision Tree, the classification report reveals high accuracy levels across most attack categories, with precision, recall, and F1-score metrics consistently exceeding 0.90. This indicates that Decision Tree effectively distinguishes between different types of cyberattacks in CAV networks, demonstrating its capability to detect attacks such as flooding, fuzzing, replay, and spoofing with minimal false positives and negatives. However, the model's performance, especially in terms of recall for the "Replay" attack category, appears to be slightly lower compared to other categories.

Similarly, the Extra Trees classifier exhibits robust performance, achieving high precision, recall, and F1-score values across all attack categories. With accuracy levels consistently exceeding 0.99, Extra Trees effectively identifies various cyber threats in CAV networks, including flooding, fuzzing, replay, and spoofing. The model's ability to maintain perfect precision and recall for the "Flooding" and "Spoofing" attack categories further underscores its reliability in accurately classifying attack messages without false positives or negatives.

In contrast, the findings of the GRU model paint a different picture. Specifically, the "Fuzzing" and "Replay" attack categories show much poorer precision, recall, and F1-score values. This implies that, possibly as a result of its recurrent nature and the difficulties involved in processing sequential data, GRU may find it difficult to generalize effectively to specific kinds of intrusions in CAV networks. Although the model's accuracy of 0.96 suggests that it still performs rather well overall, it doesn't seem to be very good at precisely identifying particular assault types.

The decision tree and extra tree classifiers perform well overall in identifying cyber threats in CAV networks, proving that they can correctly categorize attack messages with a high degree of recall and precision. However, in contrast to ensemble learning techniques like Decision Tree and Extra Trees, the GRU model performs poorly in recognizing specific attack types, even when it attains a moderate level of accuracy. To increase the GRU architecture's effectiveness in managing the complexity of a real-world CAV cybersecurity scenario, more optimization and fine-tuning may be required. Figure 6 displays ROC performance of GRU model.

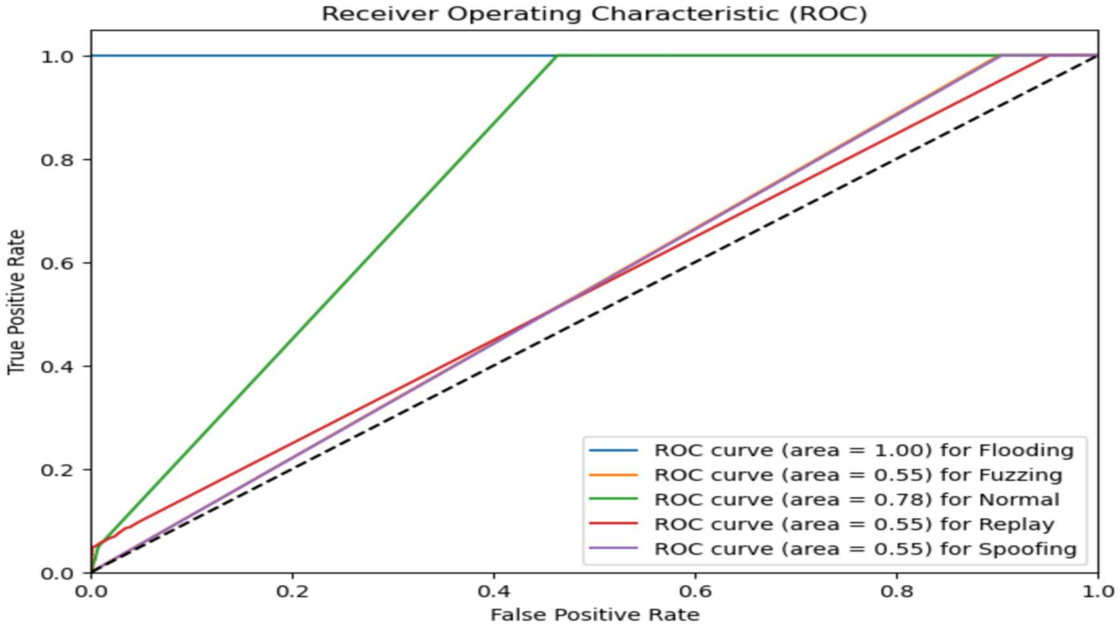

Figure 6. ROC Of GRU Model

Table 5 presents the comparative categorization performance between the proposed system and current models. The suggested framework achieved a 99% accuracy rate, surpassing all existing methods for identifying Intrusion Detection methods (IDS) on vehicle networks.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5. Comparison Between The Our Results And Existing Systems

References, authors	Approach	Acuuarcy	Dataset
Zhu [22]	LSTM	80%	CAN dataset
Avatefipour [23]	ML	90%	CAN dataset
Yang [24]	NN-LSTM	90%	CAN dataset
Aldhyani [25]	CCN-LSTM	97%	CAN dataset
Our proposed	Proposed system	99%	CAN dataset

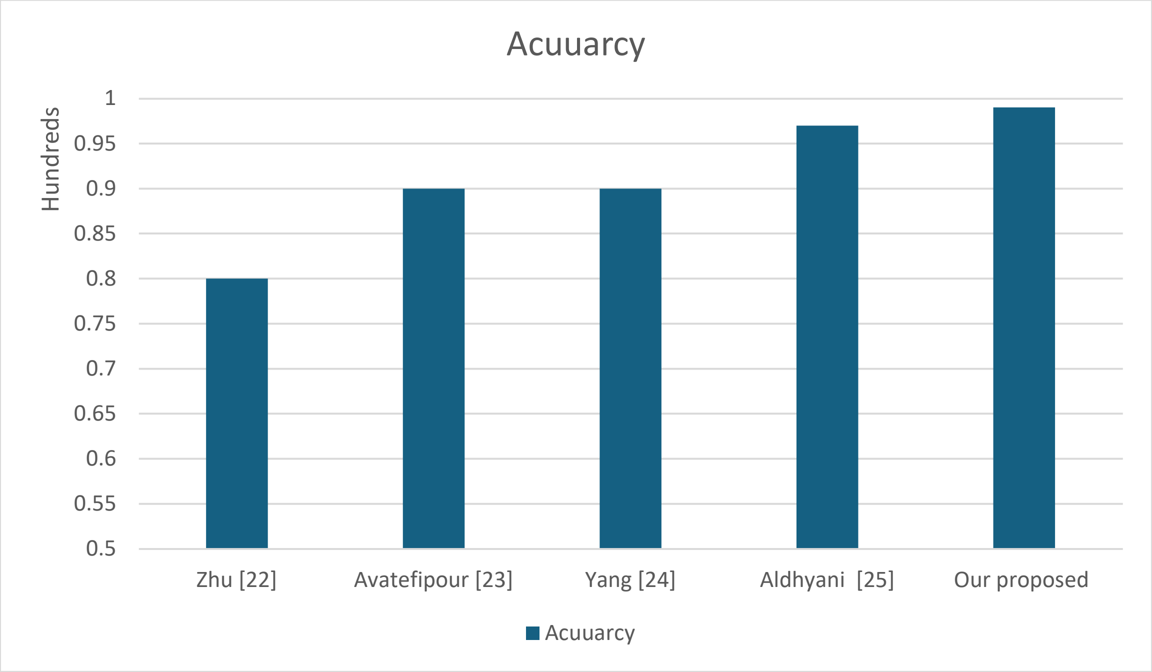

Figure 7. Performance Our CAN -IDS System

6. CONCLUSION

Autonomous vehicles function using intricate computerized systems that are susceptible to cyber assaults. These systems govern all aspects of the vehicle's operation, including its velocity, steering, braking, and acceleration. If these systems are infiltrated or breached, the repercussions might be disastrous. A cybercriminal may remotely seize control of an autonomous vehicle and deliberately induce a collision or deviate from its intended path. Consequently, this paper set out to develop, build, and test an artificial intelligence approaches for anomaly detection system for driverless vehicles. The system was developed based on ML and GRU models, we provide an intrusion detection system (IDS) capable of detecting CAN bus abnormalities in intra-vehicular networks. An extensive analysis of the CAN bus's weaknesses, the field's need for AIDS, the various types of attacks that CAN buses are susceptible to, and the consequences of these attacks on vehicles and their drivers were all covered in this study. A 99% success rate was attained by the suggested CAN-IDS system that used a decision tree technique.

The suggested solutions demonstrated the capability to efficiently identify abnormal packets for the purpose of protecting the CAN bus. They may also be applied to the development of different security systems integrated into the complicated network infrastructures of autonomous vehicles to provide secure data processing. In the near future, our system will further develop with the assistance of advanced artificial intelligence.

REFERENCES

[1]. Eckermann, E. *World History of the Automobile*; SAE Press: Warrendale, PA, USA, 2001.

[2]. AlSalem, T.S.; Almaiah, M.A.; Lutfi, A. Cybersecurity Risk Analysis in the IoT: A Systematic Review. *Electronics* 2023, *12*, 3958.

[3]. Koscher, K.; Czeskis, A.; Roesner, F.; Patel, S.; Kohno, T.; Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; et al. Experimental security analysis of a modern automobile. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 447–462.

[4]. Miller, C.; Valasek, C. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* 2015, *91*, 1–91.

[5]. Nie, S.; Liu, L.; Du, Y. Free-fall: Hacking tesla from wireless to can bus. *Brief. Black Hat USA* 2017, *25*, 1–6.

[6]. Liao, H.-J.; Lin, C.-H.R.; Lin, Y.-C.; Tung, K.-Y. Intrusion detection system: A comprehensive review. *J. Netw. Comput. Appl.* 2013, *36*, 16–24.

[7]. Al-Jarrah, O.Y.; Maple, C.; Dianati, M.; Oxtoby, D.; Mouzakitis, A. Intrusion Detection Systems for Intra-Vehicle Networks: A Review. *IEEE Access* 2019, *7*, 21266–21289.

[8]. Alsulami, A.A.; Abu Al-Haija, Q.; Alqahtani, A.; Alsini, R. Symmetrical Simulation Scheme for Anomaly Detection in Autonomous Vehicles Based on LSTM Model. *Symmetry* 2022, *14*, 1450.

[9]. Aldhyani, T.H.H.; Alkahtani, H. Cyber Security for Detecting Distributed Denial of Service Attacks in Agriculture 4.0: Deep Learning Model. *Mathematics* 2023, *11*, 233. https://doi.org/10.3390/math11010233

[10]. Khan, J.; Lim, D.-W.; Kim, Y.-S. Intrusion Detection System CAN-Bus In-Vehicle Networks Based on the Statistical Characteristics of Attacks. *Sensors* 2023, *23*, 3554.

[11]. Karthiga, B.; Durairaj, D.; Nawaz, N.; Venkatasamy, T.K.; Ramasamy, G.; Hariharasudan, A. Intelligent Intrusion Detection System for VANET Using Machine Learning and Deep Learning Approaches. *Wirel. Commun. Mob. Comput.* 2022, *2022*, 1–13.

[12]. Agrawal, K.; Alladi, T.; Agrawal, A.; Chamola, V.; Benslimane, A. NovelADS: A Novel Anomaly Detection System for Intra-Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.* 2022, *23*, 22596–22606.

[13]. Zhang, Y.; Chen, X.; Jin, L.; Wang, X.; Guo, D. Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data. *IEEE Access* 2019, *7*, 37004–37016. [

[14]. Liang, L.; Ye, H.; Li, G.Y. Toward Intelligent Vehicular Networks: A Machine Learning Framework. *IEEE Internet Things J.* 2019, *6*, 124–135.

[15]. Hoppe, T.; Kiltz, S.; Dittmann, J. Security threats to automotive CAN networks Practical examples and selected short-term countermeasures. *Reliab. Eng. Syst. Saf.* 2011, *96*, 11–25.

[16]. Taylor, A.; Leblanc, S.; Japkowicz, N. Anomaly detection in automobile control network data with long short-term memory networks. In Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA 2016), Montreal, QC, Canada, 17–19 October 2016; pp. 130–139.

[17]. Wang, C.; Zhao, Z.; Gong, L.; Zhu, L.; Liu, Z.; Cheng, X. A Distributed Anomaly Detection System for In-Vehicle Network Using HTM. *IEEE Access* 2018, *6*, 9091–9098.

[18]. Bezemskij, A.; Loukas, G.; Gan, D.; Anthony, R.J. Detecting Cyber-Physical Threats in an Autonomous Robotic Vehicle Using Bayesian Networks. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 98–103.

[19]. Kang, M.-J.; Kang, J.-W. A Novel Intrusion Detection Method Using Deep Neural Network for In-Vehicle Network Security. In Proceedings of the 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), Nanjing, China, 15–18 May 2016; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2016; pp. 1–5.

[20]. Kalash, M.; Rochan, M.; Mohammed, N.; Bruce, N.D.B.; Wang, Y.; Iqbal, F. Malware Classification with Deep Convolutional Neural Networks. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018; pp. 1–5.

[21]. Lin, Z.; Shi, Y.; Xue, Z. IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. *arXiv* 2018, arXiv:1809.02077.

[22]. Zhu, K.; Chen, Z.; Peng, Y.; Zhang, L. Mobile Edge Assisted Literal Multi-Dimensional Anomaly Detection of In-Vehicle Network Using LSTM. *IEEE Trans. Veh. Technol.* 2019, *68*, 4275–4284.

[23]. Avatefipour, O.; Al-Sumaiti, A.S.; El-Sherbeeny, A.M.; Awwad, E.M.; Elmeligy, M.A.; Mohamed, M.A.; Malik, H. An Intelligent Secured Framework for Cyberattack Detection in Electric Vehicles' CAN Bus Using Machine Learning. *IEEE Access* 2019, *7*, 127580–127592

[24]. Yang, Y.; Duan, Z.; Tehranipoor, M. Identify a Spoofing Attack on an In-Vehicle CAN Bus Based on the Deep Features of an ECU Fingerprint Signal. *Smart Cities* 2020, *3*, 17–30.

[25]. Alkahtani, H.; Aldhyani, T.H.H. Developing Cybersecurity Systems Based on Machine Learning and Deep Learning Algorithms for Protecting Food Security Systems: Industrial Control Systems. *Electronics* 2022, *11*, 1717. [Google Scholar] [CrossRef]