









# Detection of Cyber Attacks Targeting Autonomous Vehicles Using Machine Learning

Furkan Onur<sup>1</sup> , Mehmet Ali Barışkan<sup>1</sup>  (✉), Serkan Gönen<sup>1</sup> , Cemallettin Kubat<sup>2</sup> , Mustafa Tunay<sup>1</sup> , and Ercan Nurcan Yılmaz<sup>3</sup> 

- <sup>1</sup> Computer Engineering Department, Istanbul Gelisim University, İstanbul, Turkey  
mabariskan@gelisim.edu.tr
- <sup>2</sup> Aeronautical Engineering Department, Istanbul Gelisim University, İstanbul, Turkey
- <sup>3</sup> Electrics and Electronics Engineering Department, Gazi University, İstanbul, Turkey

**Abstract.** The advent of Industry 4.0, characterized by the integration of digital technology into mechanical and electronic sectors, has led to the development of autonomous vehicles as a notable innovation. Despite their advanced driver assistance systems, these vehicles present potential security vulnerabilities, rendering them susceptible to cyberattacks. To address this, the study emphasized investigating these attack methodologies, underlining the need for robust safeguarding strategies for autonomous vehicles. Existing preventive or detection mechanisms encompass intrusion detection systems for Controller Area Networks and Vehicle-to-Vehicle communication, coupled with AI-driven attack identification. The critical role of artificial intelligence, specifically machine learning and deep learning subdomains, was emphasized, given their ability to dissect vehicular communications for attack detection. In this study, a mini autonomous vehicle served as the test environment, where the network was initially scanned, followed by the execution of Man-in-the-Middle, Deauthentication, DDoS, and Replay attacks. Network traffic was logged across all stages, enabling a comprehensive analysis of the attack impacts. Utilizing these recorded network packets, an AI system was trained to develop an attack detection mechanism. The resultant AI model was tested by transmitting new network packets, and its detection efficiency was subsequently evaluated. The study confirmed successful identification of the attacks, signifying the effectiveness of the AI-based model. Though the focus remained on autonomous vehicles, the study proposes that the derived methodology can be extended to other IoT systems, adhering to the steps delineated herein.

**Keywords:** IoT · Cyber Security · Machine Learning · IIoT

## 1 Introduction

The proliferation of Internet of Things (IoT) and Industrial Internet of Things (IIoT) has catalyzed substantial advancements in various sectors, including the automotive industry, exemplified by the emergence of autonomous vehicles. However, the cybernetic integration of IoT and IIoT systems within these vehicles elicits significant cybersecurity

apprehensions. This study aims to scrutinize and counteract cyber threats—Man-in-the-Middle (MitM), Distributed Denial of Service (DDoS), Deauthentication (Death), and Replay attacks—impacting autonomous car networks, proposing a Gradient Boosting-oriented detection mechanism as a viable solution. With autonomous vehicles increasingly prone to cyberattacks due to their dependency on IoT and IIoT for data communication, processing, and decision-making, establishing rigorous security countermeasures is indispensable. The study underscores the utility of machine learning algorithms, namely Support Vector Machines (SVM), Random Forests (RF), and Neural Networks (NN), in the detection and mitigation of such threats. In particular, the superior efficacy of the Gradient Boosting algorithm in addressing these cyber threats within the IoT and IIoT landscape is demonstrated. This paper navigates through relevant literature, provides an overview of the targeted cyber-attacks and proposed detection mechanism, evaluates the performance of each algorithm, and concludes by encapsulating the findings, acknowledging study limitations, and suggesting future research directions.

## 2 Related Works

Within the framework of Industry 4.0, the domain of autonomous vehicles has sparked substantial interest. Despite the transformative potential of these vehicles, they remain susceptible to myriad cyber threats, necessitating focused research on their security. Recent studies have underscored the value of artificial intelligence (AI) in detecting and mitigating these threats. For instance, Kim et al. [1] analyzed the potential vulnerabilities and corresponding countermeasures in autonomous vehicles, advocating for enhanced anomaly detection via AI and machine learning. Nie et al. [2] demonstrated a successful remote attack on a Tesla Model S, exploiting its wireless connection to control the autonomous system.

Lee and Woo [3] proposed an insidious attack method, dubbed the CEDA, which stealthily attenuates CAN signals, thereby causing the targeted Electronic Control Unit (ECU) to ignore the received signals. Fowler et al. [4] deployed fuzz testing to identify security vulnerabilities in CAN prototypes, revealing software errors in ECU. Other studies, like those by Lim et al. [5] and Jakobsen et al. [6], focused on the vulnerabilities of the obstacle detection ultrasonic sensors and Lidar-camera sensor fusion, respectively.

Eriksson et al. [7] conducted an examination of in-vehicle Android Automotive application security utilizing static code analysis, while Cai et al. [8] spotlighted vulnerabilities in BMW's NBT Head Unit and Telematics Communication Box, emphasizing the necessity for all-encompassing security precautions. Zoppelt and Kolagari [9] investigated the prospect of cloud-based remote attacks on autonomous vehicles, deploying a Security Abstraction Model. Simultaneously, Maple et al. [10] introduced a hybrid model for attack surface analysis in connected autonomous vehicles, demonstrating its practical application through two use cases.

Other researchers like Miller and Valasek [11, 12] exploited a known Jeep Cherokee vulnerability to gain control of the vehicle, while Woo et al. [13] identified CAN as a security vulnerability and suggested a network address scrambling solution. Shrestha and Nam [14] proposed a regional block cipher for maintaining blockchain stability in VANETs, and Nasser and Ma [15] examined the Code Reuse security flaw, suggesting

an HSM-based monitoring system. Zhang and Ma [16] put forth a hybrid IDS for high detection rates and minimal computational expense.

Subsequent research, including those by Zhou et al. [17], Olufowobi et al. [18], and Hamad et al. [19], concentrated on methods for ECU identification, message forgery attack detection, and intrusion response systems for autonomous vehicle networks. Song et al. [20] developed a deep convolutional neural network-based attack system for detecting malicious CAN traffic, while Tang et al. [21] reviewed machine learning techniques for future 6G vehicle networks.

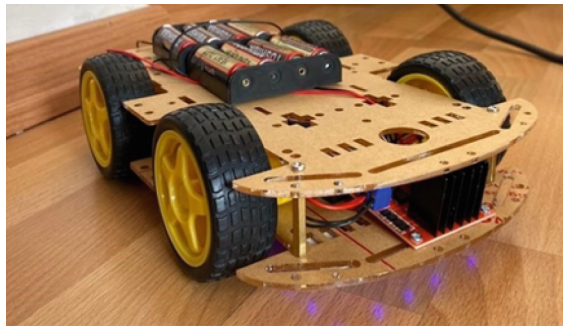
Ahmad et al. [22] leveraged LSTM networks to mitigate relay attacks and verify driver identity, whereas Gundu and Maleki [23] improved Random Forest accuracy by incorporating time intervals. Kumar et al. [24] proposed BDEdge, a Blockchain and deep learning-based system for MEC server security. Alsulami et al. [25] developed a 99.95% accurate LSTM-based early detection system for False Data Injection, and Özgür [26] achieved a similar accuracy rate using Decision Analysis and Resolution.

While these studies have primarily centered around the communication systems of cars or core computer components like CAN and ECU, our research primarily focuses on the control systems such as gas, brake, and steering wheel, thereby offering a unique perspective on the security of autonomous vehicles.

### 3 Testing Infrastructure

#### 3.1 Designed Autonomous System

The autonomous miniature vehicle, engineered utilizing Arduino, incorporates the ESP8266 NodeMCU module. This essential module enables bidirectional communication between the vehicle and peripheral computing devices such as desktops or mobile systems. Upon receiving distinct commands from these external interfaces, the vehicle, by leveraging the capabilities of the integrated communication module, initiates and executes the corresponding actions seamlessly (Fig. 1).



**Fig. 1.** Designed mini autonomous test vehicle.

The ESP8266 NodeMCU module establishes a localized network, wherein an external system can participate as a client, facilitating the transmission of HTTP GET commands. These command requests prompt interactions with the system, enabling the manual manipulation of the miniature autonomous test vehicle. For instance, such functionality can be deployed to remotely operate a stationary vehicle, initiating maneuvers such as exiting a parking space through a mobile device interface. This communication system’s structural framework is depicted in Fig. 2.

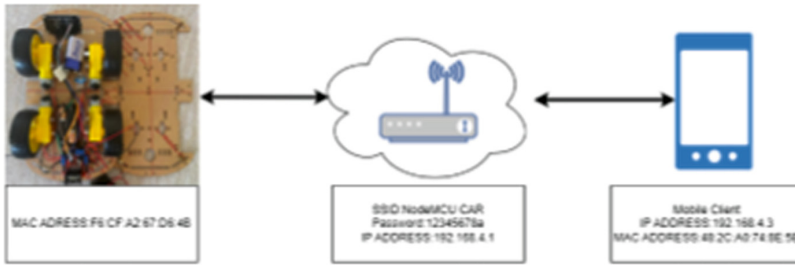


Fig. 2. Communication System

### 3.2 Preparation of Attack System

Deauthentication (Deauth), Denial-of-Service (DoS), Man-in-the-Middle (MitM), and Replay attacks were executed on the miniaturized autonomous test vehicle system utilized in this research. The assault methodologies were facilitated using network analysis and penetration tools including Nmap, hping3, airodump-ng, aireplay-ng, Ettercap, Wireshark, and Burp Suite. These tools were operated within the Kali Linux environment on the network topology described in Fig. 3. The precise attack workflow was depicted in Fig. 4.

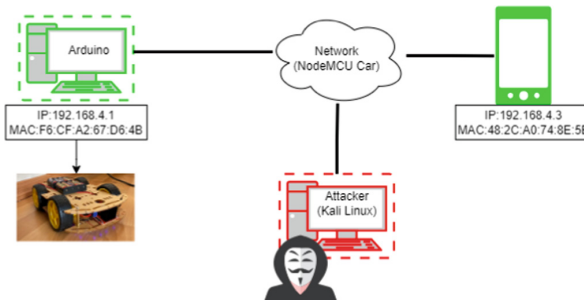


Fig. 3. Network Topology

When examining the general attack flowchart, it can be seen that the process was divided into five main stages. Specifically, the MitM and Replay attack procedures encompassed the first four phases: Discovery, Attack, Observation, and Repeated Attack. The final stage, Attack Detection via Artificial Intelligence, was a unique addition to this study, focusing on the automatic identification of both passive attacks (like MitM) and active attacks (like Replay attacks). This phase allows for early attack notification and immediate system responsiveness.

In the Discovery phase, an initial network scan was conducted to identify the target network, which was then subject to a specific scan. Subsequently, in the Attack phase, three distinct assaults were launched against the identified system: Deauth Attack, DoS, and MitM, as illustrated in the flowchart. The Observation phase followed, monitoring the impacted system to discern the consequences of the executed attacks. The Deauth Attack resulted in re-authentication, the DoS assault increased packet time intervals, and the MitM attack was discerned by observing duplicate packets via Wireshark. The successful modification of the victim device's ARP table by the attacker was also confirmed.

Subsequently, in the Repeated Attack phase, a Replay attack was enacted on the target system using the data gathered during the Observation stage. Finally, during the Detection phase, packet data obtained during the attack period was introduced to the machine learning system. This data encompassed packets from pre-attack, during attack, and post-attack stages, aiding in the detection and mitigation of future assaults.

## **4 Attack Analyses**

### **4.1 Deauth Attack**

A Deauthentication (Deauth) attack is a form of cyber-attack that disrupts network connectivity temporarily, thereby severing ongoing communications. The implications of this attack were evaluated in the context of a mini autonomous test vehicle system. The procedure encompassed several stages.

In the initial stage, the 'airodump-ng' command was utilized to detect all active networks in the proximity, subsequently providing comprehensive information regarding each one. The following stage involved identifying the specific target network for the attack - in this instance, the 'NodeMCU Car' network.

In the penultimate step, the 'aireplay-ng' command was employed to consistently transmit packets to the device tethered to the network until a threshold of 10,000 packets was reached, resulting in the device being ejected from the network.

This attack manifested in the disruption of communication between the smartphone and the autonomous vehicle, hindering real-time data transfer—a critical concern due to the halted flow of pertinent information regarding the autonomous vehicle. In both of the conducted tests, this disruption in communication was observed, thereby affirming the successful execution of the Deauth attack.

### **4.2 Denial of Service (DoS) Attack**

A Denial of Service (DoS) attack is a form of cyber offensive aimed at overloading a computer system's resources, which consequently results in the denial of access services.

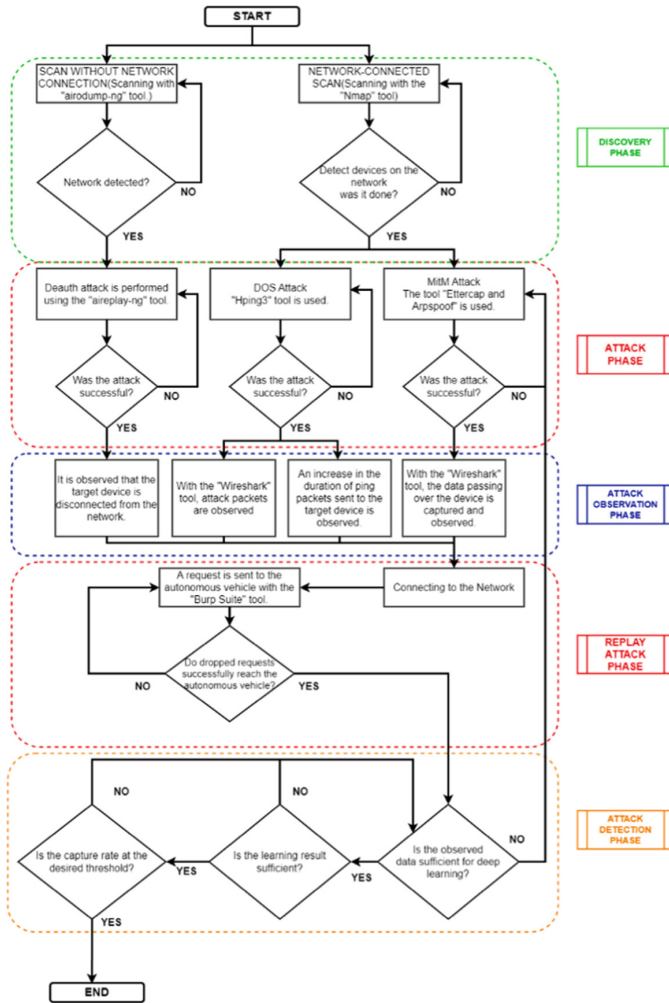


Fig. 4. Attack Flow Diagram

The ramifications of this attack form are assessed in the context of the designed mini autonomous test vehicle system.

### 4.3 Man-in-the-Middle (MitM) Attack

A Man-in-the-Middle (MitM) attack constitutes the interception, alteration, or manipulation of communication between two entities by an unauthorized third party. In wireless networks, packets are broadcast, enabling an attacker to capture all packets without necessitating preprocessing. A MitM attack on the developed mini autonomous car system is scrutinized herein. Initially, information concerning the target device was gleaned using the Nmap scan. As Wireshark initiated network listening, the network packets between the autonomous car and the phone weren't fully perceptible.

To gain access to all transmitted network packets, the Ettercap tool was utilized. Connected devices were identified via the Ettercap tool, and subsequently, an ARP poisoning attack was initiated by designating the phone as a target. Concurrently, all network packets became visible with the onset of the attack, facilitated by the Wireshark tool's listening function.

Upon manipulating one of the intercepted packets, a request was uncovered. Examination of the disclosed information ascertained that the data transmitted to the autonomous vehicle was both instantaneous and accurate.

#### **4.4 Replay Attack**

A Replay attack is a type of cyber offensive which involves obtaining unauthorized access or circumventing the authentication process by repetitively transmitting the same data employed in a preceding successful communication. This segment examines a Replay attack on the mini autonomous test vehicle system.

In order to modify the packets sent to the autonomous vehicle and transmit packets in the desired quantity and format, the Burp Suite tool was employed.

### **5 Detecting Attacks Through Artificial Intelligence Algorithms**

In this section, the network traffic associated with the mini-autonomous test vehicle is subjected to various artificial intelligence algorithms to facilitate attack detection, as depicted in Fig. 14. The attack detection model employed in this study encompasses four stages. Initially, data amassed over the network is processed, and the dataset file, subsequent to the preprocessing stage, is integrated into the model. Subsequently, the prepared dataset is partitioned into 70% training and 30% validation data, and subsequently subjected to analysis using Neural Network (NN)-ReLU, kNN, Random Forest, Gradient Boosting, SVM, and Stochastic Gradient Descent artificial intelligence algorithms. The third stage involves visualization of the data procured from the artificial intelligence algorithms to enhance analysis. Finally, after evaluation, Gradient Boosting is selected as the artificial intelligence algorithm for attack detection, on account of its superior accuracy, F1, recall, and time values across all attacks, and preserved for application to real-time data.

#### **5.1 Gradient Boosting**

Gradient Boosting is a renowned machine-learning algorithm, which is utilized to tackle classification and regression problems. This algorithm represents an ensemble learning method, which amalgamates multiple weak learners into a singular strong learner. The fundamental concept of gradient boosting entails the construction of an ensemble of decision trees, where each subsequent decision tree aligns with the residual errors of the preceding tree.

The algorithm initiates with a simplistic decision tree tailored to the data. Subsequently, the model's residuals are computed, and a new tree aligns with these residuals. This process is iterated multiple times, with each new tree aligning with the residuals

of the preceding trees. The final prediction is garnered by aggregating the predictions of all trees within the ensemble.

Gradient Boosting provides several advantages over alternative machine learning algorithms. It exhibits particular efficacy when handling high-dimensional data and can accommodate both numerical and categorical data. It also displays relative resistance to overfitting, which may pose an issue for other algorithms. Furthermore, Gradient Boosting is highly adaptable and compatible with numerous loss functions, rendering it a versatile algorithm for a myriad of problem types.

### 5.2 Model Creation and Training

Prior to the application of artificial intelligence algorithms, network packets underwent scrutiny. The dataset was divided, with 70% allocated to training and 30% to testing. Following the training of the model, results were evaluated based on various performance metrics such as training time, testing time, AUC, CA, F1, Precision, and Recall, as presented in Table 1.

Table 1. Model Comparison

Model	Train Time[s]	Test Time [s]	AUC	CA	F1	Precision	Recall
Gradient Boosting	332.383	1.404	0.997	0.987	0.987	0.987	0.987
Random Forest	18.112	0.979	0.986	0.981	0.981	0.981	0.981
Neural Network	382.897	0.9696	0.989	0.976	0.975	0.975	0.976
SGD	12.917	8.857	0.917	0.970	0.969	0.969	0.970
kNN	6.863	89.698	0.702	0.824	0.788	0.762	0.824
SVM	255.419	4.923	0.217	0.217	0.177	0.881	0.217

According to the time interval specified in Fig. 5 (X-axis), the network traffic of source hosts (Y-axis) was appraised. The red network packets, indicative of the packets dispatched by the attacker models, were tracked, leading to the successful visual detection of the attacker models in the reference model devoid of an attacker. The red packets marked as attack packets were attributed to the impact of the defined feature on the model.





3. Lee, Y., Woo, S.: CAN Signal Extinction-based DoS Attack on In-Vehicle Network. *Secur. Commun. Netw.* 2022 (2022)
4. Fowler, D.S., Bryans, J., Cheah, M., Wooderson, P., Shaikh, S.A.: A method for constructing automotive cybersecurity tests, a CAN fuzz testing example. In: 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), pp. 1–8. IEEE (2019)
5. Lim, B.S., Keoh, S.L., Thing, V.L.: Autonomous vehicle ultrasonic sensor vulnerability and impact assessment. In: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT) (pp. 231–236). IEEE (2018)
6. Jakobsen, S.B., Knudsen, K.S., Andersen, B.: Analysis of sensor attacks against autonomous vehicles. In: 25th International Symposium on Wireless Personal Multimedia Communications. IEEE (2022)
7. Eriksson, B., Groth, J., Sabelfeld, A.: On the road with third-party apps: security analysis of an in-vehicle app platform. In: VEHITS, pp. 64–75 (2019)
8. Cai, Z., Wang, A., Zhang, W., Gruffke, M., Schweppe, H.: 0-days & mitigations: roadways to exploit and secure connected BMW cars. *Black Hat USA* 2019, 39 (2019)
9. Zoppelt, M., Kolagari, R.T.: UnCle SAM: modeling cloud attacks with the automotive security abstraction model. *Cloud Comput.* 67–72 (2019)
10. Maple, C., Bradbury, M., Le, A.T., Ghirardello, K.: A connected and autonomous vehicle reference architecture for attack surface analysis. *Appl. Sci.* 9(23), 5101 (2019)
11. Miller, C., Valasek, C.: Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015(S 91), 1–91 (2015)
12. Miller, C.: Lessons learned from hacking a car. *IEEE Design & Test* 36(6), 7–9 (2019)
13. Woo, S., Moon, D., Youn, T.Y., Lee, Y., Kim, Y.: Can id shuffling technique (cist): moving target defense strategy for protecting in-vehicle can. *IEEE Access* 7, 15521–15536 (2019)
14. Shrestha, R., Nam, S.Y.: Regional blockchain for vehicular networks to prevent 51% attacks. *IEEE Access* 7, 95033–95045 (2019)
15. Nasser, A., Ma, D.: Defending AUTOSAR safety critical systems against code reuse attacks. In: Proceedings of the ACM Workshop on Automotive Cybersecurity, pp. 15–18 (2019)
16. Zhang, L., Ma, D.: A hybrid approach toward efficient and accurate intrusion detection for in-vehicle networks. *IEEE Access* 10, 10852–10866 (2022)
17. Zhou, J., Joshi, P., Zeng, H., Li, R.: Btmonitor: bit-time-based intrusion detection and attacker identification in controller area network. *ACM Trans. Embed. Comput. Syst. (TECS)* 18(6), 1–23 (2019)
18. Olufowobi, H., Hounsinou, S., Bloom, G.: Controller area network intrusion prevention system leveraging fault recovery. In: Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy, pp. 63–73 (2019)
19. Hamad, M., Tsantekidis, M., Prevelakis, V.: Red-Zone: Towards an intrusion response framework for intra-vehicle system. In: VEHITS, pp. 148–158 (2019)
20. Song, H.M., Woo, J., Kim, H.K.: In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.* 21, 100198 (2020)
21. Tang, F., Kawamoto, Y., Kato, N., Liu, J.: Future intelligent and secure vehicular network toward 6G: Machine-learning approaches. *Proc. IEEE* 108(2), 292–307 (2019)
22. Ahmad, U., Song, H., Bilal, A., Alazab, M., Jolfaei, A.: Securing smart vehicles from relay attacks using machine learning. *J. Supercomput.* 76, 2665–2682 (2020)
23. Gundu, R., Maleki, M.: Securing CAN bus in connected and autonomous vehicles using supervised machine learning approaches. In: 2022 IEEE International Conference on Electro Information Technology (eIT), pp. 042–046. IEEE (2022)
24. Kumar, P., Kumar, R., Gupta, G.P., Tripathi, R.: BDEdge: blockchain and deep-learning for secure edge-envisioned green CAVs. *IEEE Trans. Green Commun. Netw.* 6(3), 1330–1339 (2022)

25. 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* **14**(7), 1450 (2022)
26. Özgür, A.: Classifier selection in resource limited hardware: decision analysis and resolution approach. *J. Intell. Syst. Theory Appl.* **4**(1), 37–42 (2021). <https://doi.org/10.38016/jista.755419>