Computers & Industrial Engineering 190 (2024) 110088

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Machine learning-based identification of cybersecurity threats affecting autonomous vehicle systems

Furkan Onur [a], Serkan Gönen [b,d,*], Mehmet Ali Barışkan [c,d], Cemallettin Kubat [e], Mustafa Tunay [f], Ercan Nurcan Yılmaz [g]

[a] RSU Consultancy, Istanbul Turkey
[b] Istanbul Gelisim University Software Engineering Dept., İstanbul, Turkey
[c] Istanbul Gelisim University Computer Engineering Dept., İstanbul, Turkey
[d] Istanbul Gelisim University Cyber Security Application and Reseach Center, Istanbul, Turkey
[e] Istanbul Gelisim University Aeronoutical Engineering Dept., İstanbul, Turkey
[f] Cyprus Science University Computer Engineering, Girne, Northern Cyprus, Mersin 10, Turkey
[g] Gazi University Electrics and Electronics Engineering Dept., İstanbul, Turkey

ARTICLE INFO

Keywords:
Autonomous vehicle
Cyber security
Attack detection
Man in the middle
Deauth attack
DoS
DDoS
Wireless communication vulnerabilities
Attack simulation and detection

ABSTRACT

With the advancement of humanity, transportation and trade activities have increased, leading to the development process of basic land vehicles as more than physical power became necessary. Hand tools were developed with the invention of the wheel, followed by animal-powered vehicles, and then steam engine technology. After the advancement of electromechanical technologies, today's modern vehicles have been developed. Those who used these vehicles thought of transferring control from the human to autonomous driving systems to solve their safety and comfort problems. Today, instead of fully autonomous systems targeted for the future, autonomous driving support systems have been developed. Although these systems aim to increase the safety and comfort of passengers, they can become an easy target for malicious people due to network technologies and remote connection features. The most effective method of protection from these attackers is to conduct vulnerability analysis against newly emerging threats for the systems we use and to rectify identified vulnerabilities. In this research paper, the weaknesses of wireless communication towards remote connection usage of the mini electric autonomous vehicle were investigated, which we developed and produced its mechanics, electronics, and software. In this context, a test environment was created, and the problems and threats in autonomous driving technology were revealed through attacks (Deauth Attack, DoS, DDoS and MitM) made on the test environment. Following the exposed vulnerabilities, studies were conducted for the detection of these attacks using Artificial Intelligence. In the study, different algorithms were used to detect the attacks, and random forest algorithm successfully detected 96.1% of attacks. The main contribution to the field of cybersecurity in autonomous vehicles by providing effective solutions for threat identification and defense.

1. Introduction

As technology rapidly advances, research on robots is intensifying, with the range of applications for robots expanding continuously. The widespread use of the Internet has started to impact our phones, vehicles, and even our homes. Robots and autonomous systems, which are beginning to adapt to human life, have also been integrated into land vehicles in recent years. These autonomous technologies adapted to land vehicles bring features that enhance the comfort of human life.

The purpose of electric autonomous vehicles is to enable safe land travel for all individuals, prevent accidents with autonomous driving support systems, allow individuals to utilize their time by delegating driving to fully autonomous systems, and aim to be environmentally friendly by not using fossil fuels. The sustainability issues in global resources and the problems people face in traffic, such as accidents and time losses, demonstrate the importance of electric autonomous vehicles in our lives.

Any technological device with an internet connection is susceptible

* Corresponding author.
E-mail addresses: sgonen@gelisim.edu.tr (S. Gönen), mabariskan@gelisim.edu.tr (M.A. Barışkan), ckubat@gelisim.edu.tr (C. Kubat), mustafatunay@csu.edu.tr (M. Tunay), enyilmaz@gazi.edu.tr (E.N. Yılmaz).

https://doi.org/10.1016/j.cie.2024.110088

Available online 23 March 2024

to cyberattacks. Unfortunately, autonomous vehicles, which aim to ensure people's safety, are also vulnerable to such attacks. Once a malicious hacker gains control of an autonomous vehicle, they can issue any command they desire. For instance, hacktivist groups aiming to assassinate a high-profile individual can take over their vehicle and cause an accident.

In the subsequent sections of the study, the second section examines similar studies conducted on wireless communication in autonomous vehicle platforms. Third section presents the autonomous vehicle platform developed as part of this study, offering an overview of autonomous vehicle platforms within this domain. The fourth section covers vulnerability scanning, attacks, and detection stages carried out on the developed autonomous vehicle platform. The conclusion and recommendations section summarizes the study's findings, proposes measures to ensure communication security.

In this study, we embark on a pioneering journey to address the critical cybersecurity challenges faced by autonomous vehicle systems. By focusing on the prevalent vulnerabilities within wireless communication networks, we explore the real-world implications of various cyber attacks like Deauth, DoS, DDoS, and MitM. Utilizing a uniquely designed mini electric autonomous vehicle for practical simulations, we implement and test the efficacy of advanced machine learning algorithms, particularly the random forest algorithm, in detecting these threats. Our research not only contributes to the theoretical understanding of these vulnerabilities but also offers practical, machine learning-driven solutions, setting a new benchmark in the cybersecurity protocols for autonomous vehicles. The implications of our findings are far-reaching, offering vital insights and tools for stakeholders in the rapidly evolving landscape of autonomous driving technologies. This paper presents an innovative approach to enhancing cybersecurity in autonomous vehicles by integrating machine learning algorithms. It showcases the development of a mini electric autonomous vehicle used as a simulation platform to analyze vulnerabilities in wireless communication networks. Through a series of cyber attacks, including Deauth, DoS, DDoS, and MitM, the study demonstrates the effectiveness of the random forest algorithm in detecting these threats with a high success rate. This research contributes to the theoretical and practical understanding of AI's role in advancing cybersecurity measures in autonomous driving technologies.

2. Literature review

The utilization of autonomous vehicles within the scope of Industry 4.0 has become an increasingly popular area of interest. However, autonomous vehicles possess several factors that make them vulnerable to cyberattacks. Consequently, the issue of autonomous vehicle security has been seriously addressed by researchers in recent times. In this context, Artificial Intelligence-based attack detection systems have been developed, and various studies have been conducted to ensure the security of autonomous vehicles. Kyounggon Kim et al. examined 151 studies related to autonomous vehicles conducted between 2008 and 2019. In their analysis, they classified attacks on autonomous vehicles into three primary groups: those targeting the autonomous control system, those affecting the autonomous driving system's components, and those related to communication and risk evaluation between the vehicle and everything elseThey categorized defense mechanisms against attacks into security architecture, unauthorized intrusion detection, and anomaly detection. The progress in big data and communication technologies, along with advancements in Artificial Intelligence (AI) and machine learning, has enabled the utilization of these technologies for anomaly detection (Kim et al., 2021).

Nie et al. demonstrated in September 2016 that they were able to remotely attack the autonomous control system of a Tesla Model S vehicle using wireless connectivity (Nie et al., 2017). Y. Lee and S. Woo presented a new attack method called CEDA attack in their study, which remains undetectable by IDS. This attack method is carried out through CAN signal damping, causing the target ECU to ignore the received signal (Lee & Woo, 2022). Fowler and his team employed Fuzz testing to uncover security weaknesses in CAN prototypes. Their black box fuzz testing on a lab vehicle exposed software defects in the ECU and vulnerabilities within the vehicle's system (Fowler et al., 2019). Lim et al. evaluated potential security vulnerabilities in obstacle detection ultrasonic sensors commonly used in modern and autonomous vehicles (Lim et al., 2018). Jakobsen et al. mentioned their progress towards the fusion of lidar and camera sensors through multi-sensor fusion (MSF). However, they concluded that further analysis is needed to create a secure MSF by discussing potential attacks and countermeasures (Jakobsen et al., 2022). Eriksson and his colleagues worked on the security and adequacy of in-vehicle applications on the Android Automotive operating system. They performed static code analysis of in-vehicle applications using a tool called AutoTame. They conducted a case study on a Spotify application in Volvo Cars' physical test beds (Eriksson et al., 2019). Cai et al. conducted a study on multiple security vulnerabilities in the NBT Head Unit and Telematics Communication Box components of a BMW car. By exploiting these vulnerabilities, they achieved code execution through common external interfaces (USB, Ethernet, and OBD-II) in the Head Unit. In the Telematics Communication Box, they conducted a remote attack via a fake mobile network using a payload transported via HTTP and Short Message Service (SMS) (Cai et al., 2019). Zoppelt and Kolagari analyzed the security architecture of cloud-based remote attacks on autonomous vehicles through a Security Abstraction Model (SAM) designed for automotive software systems. (Zoppelt & Kolagari, 2019). Maple and colleagues introduced a reference model employing a hybrid Functional-Communication approach to assess the attack surface of connected autonomous vehicles. This model encompasses devices, edge computing, and cloud systems. Additionally, they outlined the application of the model through two sample scenarios (Maple et al., 2019). In 2015, Miller and Valasek infiltrated a 2014 model Jeep Cherokee. They exploited a security vulnerability in the main unit, reprogramming the gateway chip in the main unit to generate arbitrary Controller Area Network (CAN) messages. Further research allowed them to gain physical control over the vehicle, including steering and braking functions. Miller provided a detailed analysis of his experience in 2019 (Miller & Valasek, 2015; Miller, 2019). Woo et al. emphasized that the greatest challenge in compromising autonomous vehicles is the security vulnerabilities in the Controller Area Network (CAN). They proposed the use of CAN ID obfuscation through Network Address Shuffling (NAS) as a defense mechanism. They evaluated the performance of the solution through an assessment conducted on a laboratory vehicle (Woo et al., 2019). Shrestha and Nam designed a regional block cipher for Vehicle-to-Vehicle communication (VANET) to detect relay attacks. Their study identified a condition that decreases the likelihood of successful attacks to 51 %, contingent on factors like the quantity of good nodes versus malicious nodes, message delivery time, and puzzle-solving duration. Simulations were conducted to scrutinize parameter effects, highlighting the criticality of a low message delivery time in preserving blockchain system stability for good nodes (Shrestha & Nam, 2019). Nasser and Ma investigated the sensitivity of automotive systems to the Code Reuse security vulnerability. They proposed an HSM-based monitoring system as a solution in their Routine Control case study (Nasser & Ma, 2019). In their research, Zhang and Ma created a hybrid Intrusion Detection System (IDS) utilizing a deep convolutional neural network (DCNN). This DCNN was trained to identify patterns in network traffic and to spot malicious traffic autonomously, without relying on manually crafted features. They designed a DCNN model specific to CAN data traffic using the Inception-ResNet model architecture, which reduced unnecessary complexity while providing high detection performance. The study conducted using real vehicle data demonstrated significantly lower false-negative and error rates compared to classical machine learning-based IDS (Zhang & Ma, 2022). Zhou et al. used slight differences in bit intervals in CAN packets to identify Electronic Control Unit (ECU) fingerprints, utilizing this feature

to detect a new type of masquerade attack. Statistical properties of bit intervals, which serve as fingerprints, were calculated, enabling the detection of a new type of masquerade attack. Testing on a real vehicle dataset resulted in the correct identification of the sender with an average accuracy of 99.76 % (Zhou et al., 2019). Olufowobi and colleagues introduced a novel algorithm designed to detect message spoofing attacks on the CAN bus network. They also proposed a recovery mechanism utilizing a feature known as time intervals to mitigate the impact of such attacks. They exploited the predictable operating times of CAN message packets and the error management capability of the CAN network to enable recovery by rebooting the compromised network node. The algorithm was tested on a CAN controller device, and its performance was evaluated (Olufowobi et al., 2019). Hamad and his team implemented False Data Injection (FDI) attacks on a simulation system for autonomous vehicles and designed an early detection system employing Long Short-Term Memory (LSTM) neural networks based on the data they gathered. This system they developed attained an accuracy level of 99.95 % (Hamad et al., 2019). Ozgur employed Decision Analysis and Resolution, achieving an accuracy rate of 99.95 % (Song et al., 2020). Ahmad et al. developed a machine learning model using LSTM recurrent neural networks to reduce relay attacks on Passive Keyless Entry and Start (PKES) systems and authenticate the driver's identity. The model achieved a 99.8 % accuracy rate (Tang et al., 2019). Gundu and Maleki proposed the use of time intervals as a new feature for detecting attacks on the CAN Bus system. They found that adding time intervals to the feature set of the Random Forest machine learning algorithm improved the accuracy of Random Forest applications (Ahmad et al., 2020). Kumar et al. proposed Blockchain and deep learning-based BDEdge to prevent security vulnerabilities in Mobile Edge Computing (MEC) servers used for data processing in Intelligent Transportation Systems. BDEdge utilized an IDS based on a Sparse Auto-Encoder-enabled Attention Bidirectional Gated Recurrent Unit (SAE-ABIGRU) and achieved a 99 % accuracy rate (Gundu & Maleki, 2022). Alsulami and colleagues conducted FDI attacks on a simulation system for autonomous vehicles and subsequently devised an early detection system employing LSTM neural networks. This system demonstrated an accuracy rate of 99.95 % (Kumar et al., 2022). Ozgur employed Decision Analysis and Resolution, achieving an accuracy rate of 99.95 % (Alsulami et al., 2022). "Considerations for Cyber Security Implementation in Autonomous Vehicle Systems" by Kyung Su Lee, presented at the 21st International Conference on Control Automation and Systems (ICCAS) in 2021, discusses the increasing security threats to autonomous vehicle systems and emphasizes the importance of implementing cyber security measures. It delineates guidelines for implementing cybersecurity according to UN Regulation No. 155, with a focus on secure boot, secure communication, and secure debugging within autonomous vehicle systems. The paper highlights the essential role of international standardization and regulatory activities in the automotive industry to counteract cyber-attacks (Lee, 2021). The paper titled "Security for Autonomous Vehicle Networks" by Aifen Sui and Gordon Muehl, presented at the IEEE 3rd International Conference on Electronic Information and Communication Technology in 2020, discusses the security hurdles encountered in autonomous vehicle networks. It highlights the transformation of automotive E/E architecture, the integration with vehicle-to-everything (V2X) communication, and theimpacts of advancements in AI on security. The paper identifies technical challenges, analyzes mainstream standards and industry best practices, and highlights research areas such as secure E/E architecture, secure V2X communication, and secure perception (Sui & Muehl, 2020). In their publication "A Trustworthy Internet of Vehicles: The DAO to Safe, Secure and Collaborative Autonomous Driving" in the IEEE Transactions on Intelligent Vehicles in December 2023, Jing Yang et al. delve into the concept of leveraging a decentralized Internet of Vehicles (IoV) framework supported by Decentralized Autonomous Organizations (DAOs) to improve the reliability of interactions among vehicles and various entities. The paper proposes a dual-layer DAO structure incorporating blockchain technology and smart contracts to ensure data security, integrity, and accuracy. This novel approach targets major challenges including data breaches, privacy deficiencies, and restricted fault tolerance within centralized management systems, thereby fostering safer, more secure, and cooperative autonomous driving environments (Yang et al., 2023). "ADRC Controller Design for Autonomous Vehicles Queuing Systems in Zero-Trust Environment" by XinRong Li, Yuhong Na, DaRong Huang, and Ling Zhu, presented at the 2023 6th International Conference on Robotics Control and Automation Engineering (RCAE), focuses on improving the security and efficiency of autonomous vehicle (AV) queuing systems in a zero-trust environment. The research introduces an active disturbance rejection control (ADRC) approach aimed at managing unknown input perturbations and maintaining the stability of vehicle formation. The paper illustrates how ADRC, combined with zero-trust architecture principles, enhances the robustness and security of ICV queuing systems against dynamic driving environment disturbances (Li et al., 2023). "A Software Security Testing Model for Autonomous Systems" by Jin Dhang Hu, et al presented at the 2023 10th International Conference on Dependable Systems and Their Applications (DSA), introduces a comprehensive model for software security testing in autonomous systems. The paper emphasizes the integration of static and dynamic testing methods to enhance system robustness against security threats. It covers syntax-level, semantic, and system-level testing approaches, aiming to ensure the security of autonomous systems through a detailed analysis of potential (Hu et al., 2023). In their work on the Internet of Vehicles, Chen, C., et al. present a rear-end collision prediction scheme utilizing deep learning. They introduce a model called CPGN (Collision Prediction model based on GA-optimized Neural Network) specifically designed for predicting rear-end collisions within the IoV. It leverages a BP neural network optimized by a genetic algorithm to predict collision probabilities by analyzing various influential factors, such as vehicle-to-vehicle and vehicle-to-infrastructure communication. The model aims to improve driving safety by accurately forecasting potential rear-end collisions and suggesting preventive measures. Simulation results demonstrate the model's effectiveness in predicting collision risks and its potential to enhance road safety through advanced computational methods (Chen et al., 2018). In a separate publication, Liao, D. et al. examine the safeguarding of location and trajectory privacy within 5G-based Vehicular Social Networks (VSNs). They propose a Dynamic Group Division (DGD) algorithm that incorporates Mobile Femtocell technology for enhanced privacy protection. This approach aims to enhance privacy by dynamically grouping vehicles to exchange pseudonyms, thus obscuring their actual identities and movements. The proposed method is shown to outperform existing solutions in simulations, providing better privacy protection without compromising the real-time requirements of 5G networks (Liao et al., 2018). Another aspect of autonomous vehicle research is efficiency The article by Gung s. et al. explores an energy-saving strategy for parked vehicles to improve connectivity in Vehicle Ad-hoc Networks (VANETs) by employing them as relay nodes. It introduces a Dynamic Group Division (DGD) algorithm for optimizing the selection and operation of parked vehicles based on environmental factors and vehicular communication needs. The algorithm aims to enhance VANET efficiency and sustainability without exhausting the energy resources of parked vehicles. Through simulations, the study demonstrates the effectiveness of this approach in extending the operational time of parked vehicles, thus ensuring better connectivity and service provision in intelligent transportation systems (ITS) (Sun et al., 2019).

In conclusion, the academic studies discussed above reveal the extensive research conducted on the security of autonomous vehicles. Researchers have explored various techniques such as Artificial Intelligence, machine learning, cryptographic protocols, network security, and anomaly detection to enhance the security of autonomous vehicles. These studies contribute to the understanding of potential vulnerabilities and the development of effective defense mechanisms against

cyberattacks in the context of autonomous vehicles.

The significant contributions of the study are:

- The Deauth Attack, DoS, DDoS and MitM attacks, was carried, analyzing the impact of the attackers on autonomous vehicles.
- An expert system integrating continuous monitoring and AI was employed for attack detection purposes.

3. Machine learning and cyber security

Expanding on the role of machine learning in cybersecurity, we delve into a detailed analysis of its applications, challenges, and future prospects. Machine learning (ML), a subset of artificial intelligence, has become a cornerstone in modern cybersecurity strategies. Its ability to process and learn from large volumes of data enables it to identify patterns and anomalies that would be impossible for humans to detect efficiently.

The use of ML in cybersecurity can be broadly categorized into several key areas: malware detection, network security, phishing detection, and fraud prevention. In malware detection, ML algorithms analyze the characteristics of known malware samples to predict and identify new, similar threats. This proactive approach is crucial in combating the ever-evolving nature of malware. In network security, ML is used to monitor network traffic patterns to detect anomalies that could indicate a breach, such as unusual data flows or access requests. This real-time monitoring helps in the early detection of potential threats, thereby reducing the risk of significant data breaches.

Phishing detection is another critical area where ML has shown significant efficacy. By analyzing the content and metadata of emails, ML algorithms can identify patterns consistent with phishing attempts, such as deceptive links or spoofed email addresses, and alert users to these potential threats. In fraud prevention, especially within the financial sector, Machine Learning algorithms scrutinize transaction data to detect patterns that suggest fraudulent activity. This approach aims to safeguard consumers and businesses from financial losses.

Yet, incorporating Machine Learning into cybersecurity encounters challenges, with one primary obstacle being the necessity for extensive, varied, and top-tier datasets to adequately train the algorithms. The quality of these datasets directly impacts the effectiveness of the ML models. Another challenge is the dynamic nature of cyber threats. As cyber attackers become aware of the defense mechanisms, they continually evolve their tactics, requiring constant updates and retraining of ML models.

Moreover, there is the challenge of false positives, where legitimate activities are mistakenly flagged as threats, and false negatives, where actual threats remain undetected. Achieving a balance between sensitivity and specificity in ML models is crucial to mitigate these errors. Additionally, there is a growing concern about adversarial attacks against ML models themselves, where attackers manipulate data inputs to cause the model to make incorrect predictions or classifications.

Looking ahead, the future of ML in cybersecurity is promising but requires ongoing research and development. Advancements in deep learning, a more complex form of ML, are expected to enhance the ability to detect and predict more sophisticated cyber threats. Integrating ML with other emerging technologies like Blockchain and the Internet of Things (IoT) also presents new opportunities for enhancing cybersecurity measures.

4. Methodology

The methodology begins with designing and constructing a mini electric autonomous vehicle tailored to simulate real-world autonomous systems. This involves detailed mechanical, electronic, and software configurations to ensure the vehicle accurately represents larger, more complex systems. The vehicle serves as a practical testbed for simulating cybersecurity threats.

The next phase simulates various cyber attacks, including Deauth, DoS, DDoS, and MitM attacks. The simulations are performed within a controlled environment to guarantee the safety and integrity of the system. The goal is to mimic real-world attack scenarios, allowing for an in-depth study of how these threats affect autonomous vehicle systems. Various parameters of each type of attack are adjusted to observe different threat levels and system responses.

In parallel, the study incorporates machine learning algorithms to detect and analyze the simulated attacks. The choice of algorithms, especially the random forest algorithm, is based on their proven effectiveness in pattern recognition and anomaly detection. The algorithms undergo training using data acquired from the simulations, allowing them to learn and enhance their detection accuracy progressively. This training process entails fine-tuning different algorithm parameters to maximize performance.The last phase of the methodology involves examining the data gathered from simulations, with a specific focus on assessing the performance of machine learning algorithms in identifying simulated attacks. It thoroughly examines the detection rates, false positives, and false negatives, providing a comprehensive understanding of the strengths and weaknesses of the employed cybersecurity measures.

Throughout the methodology, emphasis is placed on replicability and real-world applicability. The autonomous vehicle platform is designed to be as representative of real systems as possible, and the simulated attacks are chosen based on their prevalence and significance in the real world. The use of machine learning algorithms is aligned with current trends in cybersecurity, showcasing a forward-looking approach to threat detection and prevention.

In conclusion, the methodology of this paper presents a holistic approach to understanding and mitigating cybersecurity threats in autonomous vehicles. By merging practical simulations with sophisticated machine learning techniques, the study provides valuable insights into the dynamic realm of automotive cybersecurity. The anticipated impact of the study's discoveries is substantial, as they are poised to substantially enhance the development of autonomous vehicle systems, bolstering their dependability and safety against escalating cyber risks.

5. Experiment and analysis

The prototype autonomous vehicle, built with Arduino, integrates the ESP8266 NodeMCU module. This component facilitates communication between the miniature test vehicle and either a computer or mobile device. When necessary, it receives commands from the computer or mobile device and controls the movement of the system through the communication module as shown in Fig. 1.

The ESP8266 NodeMCU module establishes a local network. By connecting as a client to the local network, the system receives requests through HTTP GET commands. These requests allow manual control of the mini test autonomous vehicle. For instance, the vehicle can be controlled from a mobile device to maneuver it out of a parking spot. Fig. 2 illustrates the communication system.

The "autonomous" aspects of the autonomous vehicle primarily consist of software-based driving assistance systems. These systems can include features such as Lane Keeping, Automatic Emergency Braking, and Automated Parking. All these functionalities can be combined in more advanced systems to enable both driving assistance and total autonomous driving.

The implemented autonomous vehicle setup relies on camera and sensor data. The camera captures the real-time video feed and then transmits it to a Raspberry Pi for further processing. Utilizing Python and OpenCV, the software algorithm identifies road lanes and computes the required steering angle for the vehicle. Subsequently, the steering control is adjusted based on the algorithm's determination.

The autonomous system evaluates the sensor data obtained while the vehicle is in motion. When the target vehicle shares the same lane as the autonomous vehicle, a real-time collision calculation is executed, taking

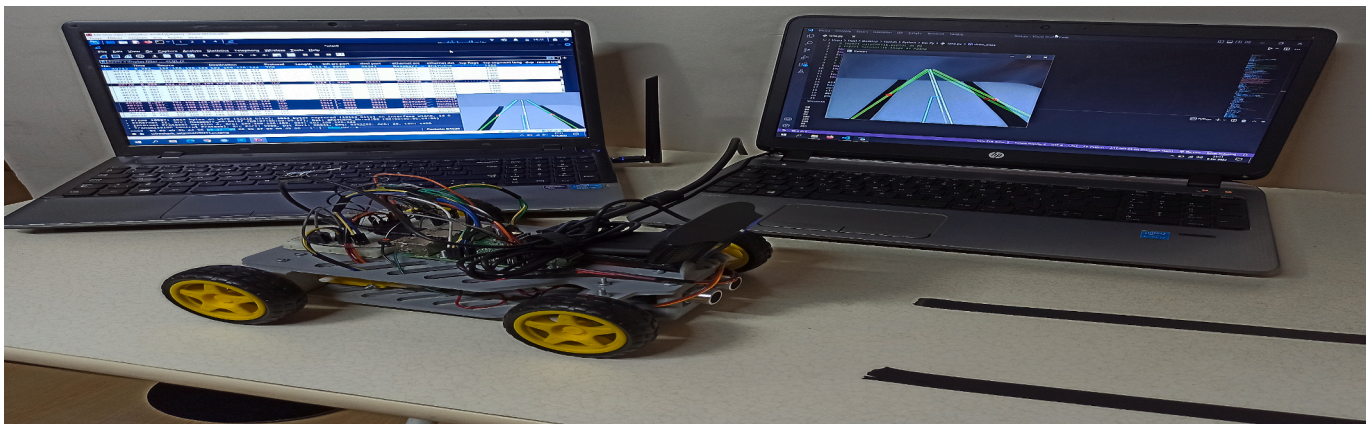

Fig. 1. Autonomous Vehicle and Test System during MitM Attack.

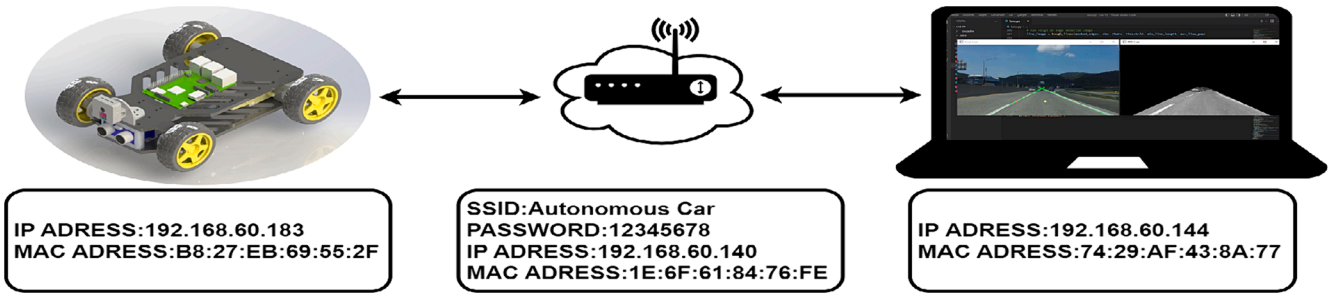

Fig. 2. Communication System.

into account the speed of the target vehicle. If the calculated collision time is less than the braking time, the Automatic Emergency Braking system is activated.

The lane-keeping system, shown in Fig. 3, ensures that the vehicle stays on the intended path during autonomous travel. It utilizes a camera mounted on the front windshield to detect the road and maintain the vehicle's trajectory. The system is a safety measure to prevent accidents caused by driver distractions or drowsiness, especially during long journeys. If the vehicle deviates from the lane, the lane-keeping system alerts the driver through auditory cues, steering wheel vibrations, or seatbelt tensioning. Depending on the car manufacturer, the system is typically engaged when the vehicle exceeds an average speed of 60 km/h and disengaged when the driver initiates a lane change.

The Automatic Emergency Braking system, which shown in Fig. 4, relies on RADAR, LIDAR, or camera sensors to detect obstacles and distances around the vehicle. It initiates automatic braking when necessary. Once the vehicle reaches a speed of 30 km/h or higher, the system starts monitoring the preceding vehicles and provides driver alerts based on the selected following distance.

For the communication system of the autonomous vehicle, a Raspberry Pi serves as the main onboard computer, while a ground station and the Python library OpenCV are utilized for processing the camera data. The communication between the Raspberry Pi and the ground station begins with the creation of a network based on the IEEE 802.11 standard, allowing the computers to connect to the network. With the computers connected, the system is completed, and data transmission between them is established using the TCP protocol. This data transfer

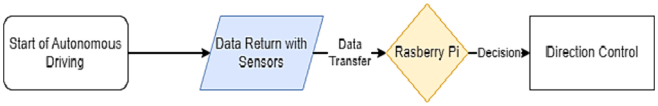

Fig. 3. Algorithm Flowchart of the Lane-Keeping System.

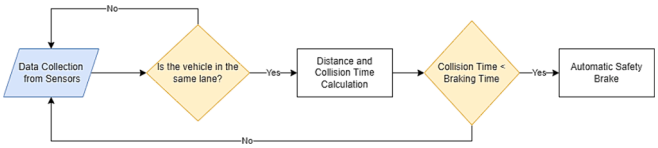

Fig. 4. Algorithm Flowchart of the Automatic Emergency Braking System.

involves the Raspberry Pi transmitting the video feed to the ground station, which then processes the images to detect lanes and obstacles, calculates the vehicle's direction and speed, and sends the resulting information back to the Raspberry Pi.

The described components and functionalities of the autonomous vehicle demonstrate the integration of software-based driving assistance systems and their reliance on camera and sensor data. These features contribute to enhancing the safety and autonomy of the vehicle, paving the way for advanced autonomous driving capabilities.

5.1. Preparation of the attack system

For the purpose of this study, Deauth Attack, DoS, and MitM attacks were conducted on the autonomous vehicle system using various penetration testing tools available in the KALI LINUX operating system, including Nmap, HPing3, airodump-ng, aireplay-ng, Ettercap, Arpspoof, FFmpeg, and Wireshark. These attacks were performed on the system described in the network topology outlined in Fig. 5. The attacks were carried out by following the steps detailed in the Attack Network Topology in Fig. 6.

The Deauth Attack aimed to disrupt the communication between the autonomous vehicle and the connected devices by sending deauthentication packets, causing disconnections. This attack was conducted using the hping3 tool to generate deauthentication packets targeted at the wireless access point. The DoS (Denial of Service) attack was performed

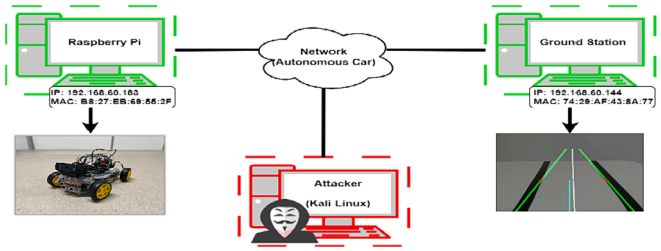

Fig. 5. Network Topology.

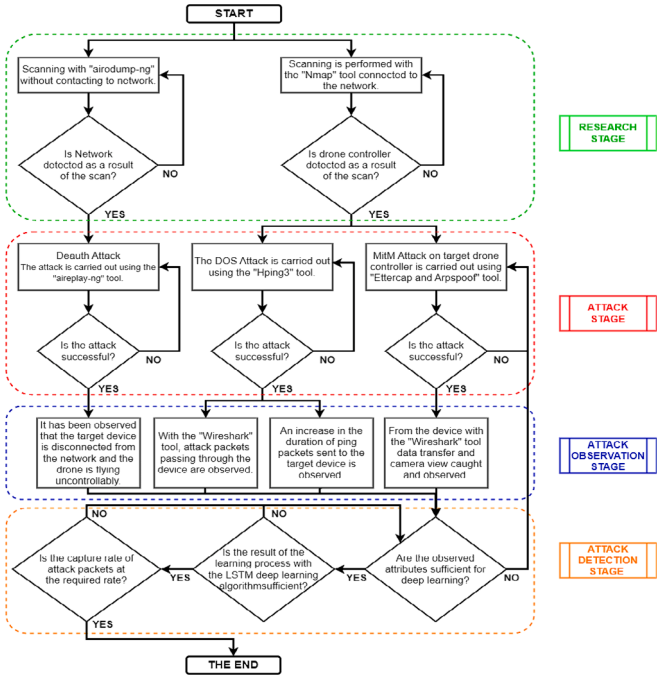

Fig. 6. Attack Flowchart.

to overload the system and render it unable to respond to legitimate requests. The attack utilized the airodump-ng and aireplay-ng tools to capture and inject excessive network traffic, effectively flooding the network and causing service disruption. The MitM (Man-in-the-Middle) attack involved intercepting and altering communication between the autonomous vehicle and its intended recipients. This attack was executed using the Ettercap and Arpspoof tools to manipulate ARP (Address Resolution Protocol) tables and redirect network traffic through the attacker's machine. This allowed the attacker to eavesdrop on the communication and even modify the transmitted data.

To capture and analyze the network traffic during the attacks, the Wireshark tool was used. It enabled the monitoring and inspection of the packets exchanged between the autonomous vehicle, connected devices, and the attacker's machine. FFmpeg was employed to record the screen during the attacks, providing visual evidence of the attack scenarios and their impact on the system.

By conducting these attacks on the autonomous vehicle system, the vulnerabilities and weaknesses in its security were identified, and potential countermeasures and improvements could be explored to enhance the system's resilience against such threats.

When examining the flow diagram in Fig. 5, it can be observed that it consists of four main stages. The attack-related parts, which are the discovery, attack, and observation stages, form the first three stages. However, our study's main focus is on the final stage, which involves the detection of passive attacks such as MitM through an AI algorithm and ensuring the system remains operational and protected from attacks at the earliest possible time to prevent information disclosure.

In the initial stage, the network is scanned, and once the target system is identified, the authenticity of this target system's presence in the network is verified by confirming its brand and model information. Subsequently, as shown in the flow diagram, three different attacks (Deauth Attack, DoS, MitM) are performed on the identified target system. The impacts of these attacks on the system are observed in the third stage, the attack stage. It is observed that in the Deauth Attack, authentication is disrupted, in DoS attacks, there is an increase in packet delays, and in MitM attacks, duplicate packets are received via Wireshark, allowing successful manipulation of the ARP tables of the victim devices by entering the attacker's ARP table. In the final stage, the attack detection stage, packets obtained through Wireshark are used to train the system by introducing both normal network packets and packets identified as attacks. After the machine learning stage is completed, the success rate of detecting attack packets is examined using the training set. The examination reveals a successful detection rate of 96.1 % for attack packets.

Overall, this methodology provides an automated and Artificial Intelligence-based approach for the early detection and prevention of MitM attacks, allowing the system to remain protected and preventing information disclosure.

5.2. Attack detection

In this study, three types of attacks (Deauth Attack, DoS, MitM) were conducted. However, the main focus of our work lies in the detection of passive attacks, such as MitM, through an AI algorithm, aiming to automatically detect and mitigate attacks in the earliest possible stage to prevent information disclosure. This section investigates attack detection using Artificial Intelligence.

The key aspect in AI is the training and identification of features on the test packets. The determination and training of features based on the type of attack are crucial in this stage. By imparting the acquired knowledge to the machine, the system is expected to be able to issue alerts during an attack by comparing incoming packets with comparable patterns. Essential information and protocols necessary for the attack typically encompass the target IP, MAC, and port addresses. Additionally, the flowing protocol characteristic and packet size are important factors as they can influence the success rate of the attack against the target.

In the MitM attack, a crucial point is the successful interception of duplicate packets via Wireshark, allowing the attacker to manipulate the ARP tables of the victim devices and effectively insert their own malicious ARP entry.

6. Detection of attacks using artificial intelligence

AI Intelligence algorithms on network traffic. The recorded network traffic is passed through various AI algorithms. In this study, the Artificial Intelligence-based attacker detection model consists of four stages. In the first stage, the data obtained from network traffic is processed through data preprocessing steps to create a suitable dataset. This dataset is divided into 10-millisecond time intervals before loading onto the model to enhance the accuracy of the algorithms. In the second stage, the created dataset is divided into 70 % training data and 30 % validation data. This dataset is analyzed using different AI algorithms such as Stochastic Gradient Descent, Gradient Boosting, Neural Network, SVM, Random Forest, and kNN, using the 10-fold cross-validation method. In the third stage, visualization techniques are employed to better understand the results of the AI algorithms. This enables a more visual evaluation of the obtained data. In the final evaluation stage, the Random Forest algorithm, which exhibits the highest accuracy, F1 score, recall, and time values across all attack types, is selected for attack detection and recorded for use in real-time data as shown in Fig. 7.

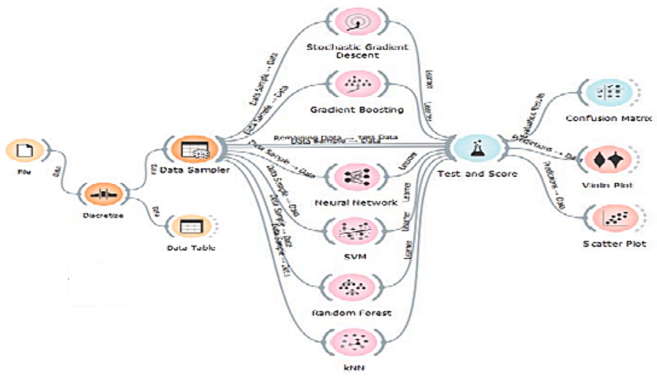

Fig. 7. Artificial Intelligence Algorithm.

6.1. Random forest

Random Forest is a powerful algorithm commonly used in the field of machine learning and optimization. It utilizes the Gradient Boosting method to optimize the model's parameters (Özgür, 2021). Random Forest divides the dataset into smaller subsets and performs gradient computations on each subset. Based on these computations, the parameters are updated. This approach offers a faster and lighter computational load by working on randomly selected subsets of the data instead of performing gradient computation and parameter updates using the entire dataset. Random Forest performs effectively, especially with large datasets or high-dimensional data. It is also more resistant to sudden or noisy gradients as it calculates gradients for each mini-group or sample and smooths out fluctuations between gradients. One disadvantage of Random Forest is that fluctuations may occur due to the computation of gradients from randomly selected mini-groups, which can introduce noise to the optimization process. However, this disadvantage is balanced by significant advantages such as faster learning rates and lower memory requirements. Random Forest is widely preferred among AI algorithms and has achieved significant success in the field of attack detection. According to the results of studies, the Random Forest algorithm has been found to exhibit high accuracy, F1 score, recall value, and time performance for attack detection. Due to these characteristics, Random Forest is an important optimization method capable of effectively working in the context of security and successfully detecting attacks.

6.2. Creating and training the model

In this section, the focus was on the second stage where attack analyses were conducted and the data packets transmitted to the expert system were processed using machine learning and AI algorithms. In the expert system used for attack detection, captured data packets were classified as either attack or normal network packets. This classification process was introduced to the expert system and used as a 70 % training dataset for system training. After the classification stage was completed, various AI algorithms were applied to a 30 % validation dataset, which consisted of labeled data packets classified by the expert system, for testing purposes.

Upon examining the values presented in Table 1, it can be observed that accuracy rate (CA) alone is not sufficient for evaluating the performance of the AI algorithm. This is mainly due to the importance of factors such as F1 score, precision, recall, and especially test time. Once the learning phase is completed, the system learns the definitions related to the data and usually performs well in this phase. However, the test time, which refers to the rapid and accurate determination of the class to which a new incoming packet belongs, is crucial. The ability to make a correct detection as quickly as possible is of great importance. When examining the table, it can be observed that the accuracy rates, F1 score, precision, and recall scores of the Random Forest algorithm are higher compared to the others, indicating a higher success rate in detection. Therefore, the decision was made to use the Random Forest algorithm in the expert system.

According to the Random Forest algorithm, the network traffic of source hosts was evaluated based on the time interval indicated in Fig. 8 (X-axis) and represented on the Y-axis. During this evaluation, traces of packets sent by the adversary models were tracked, and these packets were represented in red. The red packets were considered as markers indicating the attacks and were successfully visually detected in the reference model without attackers. The red packets flagged as attack packets were linked to the influence of the defined feature on the model. This observation indicates that the adversary models exhibit specific patterns or behaviors in the network traffic, and these patterns can be successfully visually detected in the reference model without attackers. This information demonstrates that the methods used for attack detection can effectively identify adversary activities and differentiate attack packets from other normal traffic. These findings highlight the effective functioning of attack detection systems in terms of security and their ability to visually monitor the impact of adversary models on the network.

When examining the values presented in Table 2, it shows the accuracy of the system in correctly classifying the test data as true positive and false positive after achieving a 70 % learning rate. In this table, the data marked as 0 represents non-attack packets. In the first row, the system was able to correctly classify non-attack packets with an accuracy of 93.6 %, indicating that these packets were correctly identified as non-attacks. When a total of 315,881 packets were sent, it can be observed that legal packets were classified correctly with a 93.6 % accuracy, while 6.4 % of them were falsely identified as attacks.

Moving on to the second row, when it comes to attack packets, it is observed that attack packets were correctly identified as attacks with an accuracy of 87.6 %, while 12.4 % of them were falsely classified as legal packets. These ratios indicate that the system successfully detects attack packets with an 87.6 % accuracy and produces a 12.4 % error rate.

These findings demonstrate that the system effectively detects attack packets with a high level of accuracy and mostly correctly classifies legal packets. This indicates that the system operates effectively in terms of security and provides robust protection against attacks.

7. Discussion and results

In this study, the effectiveness of the Random Forest algorithm for intrusion detection was evaluated based on the analysis of network traffic data. The goal was to assess the ability of the algorithm to

Table 1
Model Comparation.

Model	Train Time[s]	Test Time [s]	AUC	CA	F1	Precision	Recall
Random Forest	3.724	0.442	0.961	0.922	0.923	0.925	0.922
Gradient Boosting	35.890	0.378	0.951	0.882	0.888	0.906	0.882
kNN	2.063	4.575	0.904	0.877	0.874	0.873	0.877
Neural Network	57.525	0.386	0.927	0.872	0.879	0.908	0.872
SGD	2.536	0.359	0.880	0.862	0.869	0.893	0.862
SVM	32.966	0.833	0.449	0.292	0.196	0.819	0.262

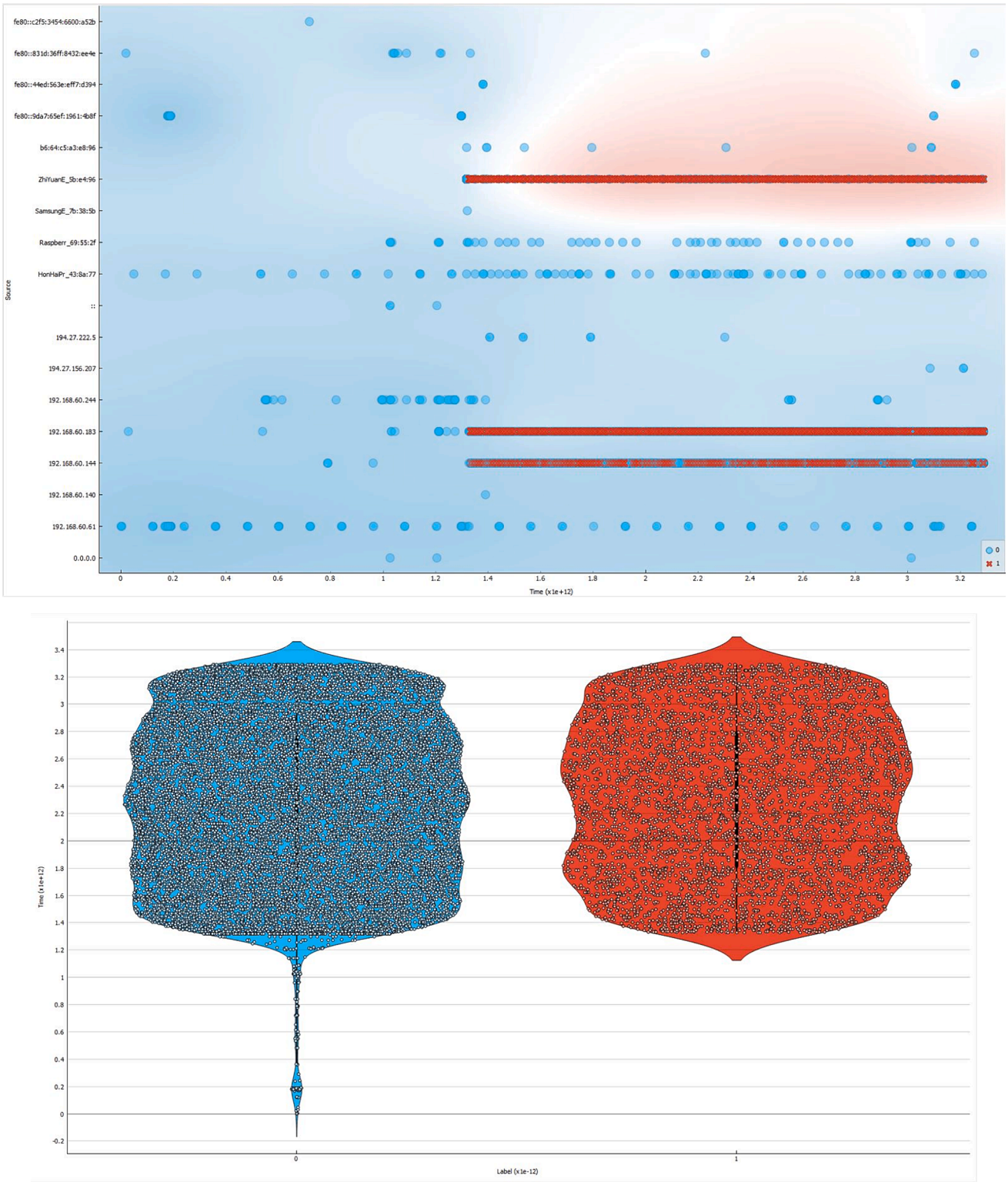

Fig. 8. **A.** Detection of attackers using the Artificial Intelligence model. **B.** Detection of attackers using the Artificial Intelligence model.

accurately classify packets as either attack or non-attack. The results obtained from the evaluation were discussed in relation to the system's performance in terms of accuracy, precision, recall, and false positive rate. The findings from Table 2 revealed that the Random Forest algorithm achieved a high level of accuracy in classifying packets. Non-attack packets were correctly classified with an accuracy of 93.6 %, indicating the algorithm's ability to accurately identify legitimate traffic. However, a small percentage (6.4 %) of non-attack packets were falsely identified as attacks, leading to a false positive rate. When it comes to attack packets, the Random Forest algorithm exhibited a lower accuracy of 87.6 %. This indicates that a certain proportion of attack packets were misclassified, resulting in false negatives. Additionally, a small percentage (12.4 %) of non-attack packets were falsely classified as attacks, leading to false positives. These results demonstrate that the Random Forest algorithm is effective in detecting attack packets, as it achieved a relatively high accuracy rate. However, it's important to

Table 2
Confusion Matrix.

	Predicted			
Actual		0	1	∑
	0	93.6 %	6.4 %	164,311
	1	12.4 %	87.6 %	151,570
	∑	160,280	155,601	315,881

acknowledge that there is potential for enhancing the reduction of false positive and false negative rates. Fine-tuning the algorithm's parameters and exploring additional features may help enhance its performance in distinguishing between attack and non-attack packets. The study's limitations primarily revolve around the scope of attack simulations and the generalizability of the machine learning model's effectiveness. The research focuses on a specific set of cyberattacks (Deauth Attack, DoS, DDoS, and MitM), which, while comprehensive, does not encompass all possible cybersecurity threats to autonomous vehicles. Additionally, the Random Forest algorithm's performance, although promising, was tested in a controlled environment, which may not fully replicate real-world conditions.

8. Future work

In future research, expanding the types of cyber threats analyzed beyond Deauth Attacks, DoS, DDoS, and MitM will be crucial for a comprehensive cybersecurity framework for autonomous vehicles. Additionally, integrating the Random Forest algorithm with other emerging technologies, such as blockchain and quantum computing, could potentially enhance its predictive accuracy and robustness. Experimenting in more dynamic and real-world environments will be essential to validate the model's effectiveness further and to explore its scalability for broader applications. Collaborative efforts with automotive manufacturers and cybersecurity experts are recommended to develop more resilient autonomous vehicle systems.

9. Limitations

This study's primary limitation lies in its focus on a specific set of cyberattacks and the controlled environment in which the Random Forest algorithm's performance was evaluated. The selected cyberattacks, while representative, do not encompass all potential cybersecurity threats facing autonomous vehicles today. Moreover, the controlled environment may not accurately replicate the complex network interactions and attack vectors encountered in real-world scenarios. The diversity and scale of the dataset could also constrain the applicability of the findings. Future research endeavors should strive to overcome these constraints by encompassing a broader spectrum of cyber threats and conducting assessments in more diverse and authentic environments.

Overall, the Random Forest algorithm shows promise in intrusion detection, and its effectiveness can be enhanced through continuous refinement and exploration of feature selection techniques. By addressing the limitations identified in this study, the algorithm has the potential to be an effective tool in detecting and mitigating network attacks, contributing to improved cybersecurity measures.

10. Conclusion

This study meticulously evaluates the Random Forest algorithm for intrusion detection within autonomous vehicle systems, focusing on its efficacy in accurately distinguishing between attack and non-attack network traffic. The algorithm's high precision in identifying legitimate traffic is commendable, showcasing a 93.6 % accuracy for non-attack packets. However, the detection of attack packets reveals room for improvement, with an 87.6 % accuracy indicating a propensity for false negatives. Despite these challenges, the Random Forest algorithm's overall performance is promising, highlighting its potential as a robust tool for cybersecurity in autonomous vehicles. The research underscores the importance of balancing accuracy with the minimization of false positives and negatives, suggesting further refinements to enhance the algorithm's capability. The findings advocate for continuous advancements in machine learning techniques to fortify cybersecurity measures against the evolving threats targeting autonomous vehicle systems.

The evaluation of the Random Forest algorithm for intrusion detection based on network traffic data yielded promising results. The algorithm demonstrated a high accuracy in classifying non-attack packets, indicating its ability to accurately identify legitimate traffic. However, the algorithm exhibited a slightly lower accuracy in classifying attack packets, resulting in false negatives. The results underscore the significance of taking into account both accuracy and false positive rates during the evaluation of intrusion detection systems. While achieving high accuracy is crucial, minimizing false positives is equally important to prevent misclassification of non-attack packets as attacks. The results suggest that further refinements and optimizations of the Random Forest algorithm are necessary to improve its performance in correctly identifying attacks.

Competing interests

As authors we declare that they have no competing interests that could have appeared to influence the work reported in this paper.

Funding information

The authors wish to disclose that this research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Author contribution

Furkan Onur was responsible for the system design, hardware preparation, and software programming, demonstrating a comprehensive understanding of the technical aspects of the research. Mehmet Ali Barışkan and Serkan Gönen conducted the attacks on the system and collected the corresponding data, showcasing their expertise in cybersecurity testing and data acquisition. Cemalettin Kubat and Mustafa Tunay performed the data analysis using AI algorithms, highlighting their contributions to the application of AI in cybersecurity. Ercan Nurcan Yılmaz verified the data accuracy and conducted comparisons with previous studies, ensuring the reliability and relevance of the research findings. Every author has endorsed the submitted version of the manuscript and accepts responsibility for all facets of the work, demonstrating their dedication to upholding the integrity and quality assurance of the research. These individual roles and responsibilities collectively contribute to the holistic success of the research presented in the paper.

CRediT authorship contribution statement

Furkan Onur: Conceptualization, Formal analysis, Investigation, Writing – original draft. **Serkan Gönen:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Mehmet Ali Barışkan:** Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Cemallettin Kubat:** Conceptualization, Supervision, Writing – review & editing. **Mustafa Tunay:** Supervision. **Ercan Nurcan Yılmaz:** Conceptualization, Data curation, Funding acquisition, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

The data supporting the findings of this study are sourced from a live system and are currently under ongoing research analysis. Due to the sensitive and proprietary nature of the data, it is not publicly available. Access to the data will be considered by the corresponding author upon reasonable request, subject to confidentiality agreements and ethical considerations.

References

Kim, K., Kim, J. S., Jeong, S., Park, J. H., & Kim, H. K. (2021). Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers & Security, 103*, Article 102150.

Nie, S., Liu, L., & Du, Y. (2017). Free-fall: Hacking tesla from wireless to can bus. *Briefing, Black Hat USA, 25*, 1–16.

Lee, Y., & Woo, S. (2022). CAN Signal Extinction-based DoS Attack on In-Vehicle Network. Security and Communication Networks, 2022.

Fowler, D. S., Bryans, J., Cheah, M., Wooderson, P., & Shaikh, S. A. (2019). A method for constructing automotive cybersecurity tests, a CAN fuzz testing example. In *In 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 1–8). IEEE.

Lim, B. S., Keoh, S. L., & Thing, V. L. (2018). Autonomous vehicle ultrasonic sensor vulnerability and impact assessment. In *In 2018 IEEE 4th World Forum on Internet of Things (WF-IoT)* (pp. 231–236). IEEE.

Jakobsen, S. B., Knudsen, K. S., & Andersen, B. (2022). Analysis of sensor attacks against autonomous vehicles. *In 25th International Symposium on Wireless Personal Multimedia Communications*. IEEE.

Eriksson, B., Groth, J., & Sabelfeld, A. (2019, May). On the Road with Third-party Apps: Security Analysis of an In-vehicle App Platform. In VEHITS (pp. 64-75).

Cai, Z., Wang, A., Zhang, W., Gruffke, M., & Schweppe, H. (2019). 0-days & mitigations: Roadways to exploit and secure connected BMW cars. *Black Hat USA, 2019*, 39.

Zoppelt, M., & Kolagari, R. T. (2019). UnCle SAM: Modeling cloud attacks with the automotive security abstraction model. *Cloud Computing*, 67–72.

Maple, C., Bradbury, M., Le, A. T., & Ghirardello, K. (2019). A connected and autonomous vehicle reference architecture for attack surface analysis. *Applied Sciences, 9*(23), 5101.

Miller, C., & Valasek, C. (2015). Remote exploitation of an unaltered passenger vehicle. Black Hat USA, 2015(S 91), 1-91.

Miller, C. (2019). Lessons learned from hacking a car. *IEEE Design & Test, 36*(6), 7–9.

Woo, S., Moon, D., Youn, T. Y., Lee, Y., & Kim, Y. (2019). Can id shuffling technique (cist): Moving target defense strategy for protecting in-vehicle can. *IEEE Access, 7*, 15521–15536.

Shrestha, R., & Nam, S. Y. (2019). Regional blockchain for vehicular networks to prevent 51% attacks. *IEEE Access, 7*, 95033–95045.

Nasser, A., & Ma, D. (2019, March). Defending AUTOSAR safety critical systems against code reuse attacks. In Proceedings of the ACM Workshop on Automotive Cybersecurity (pp. 15-18).

Zhang, L., & Ma, D. (2022). A hybrid approach toward efficient and accurate intrusion detection for in-vehicle networks. *IEEE Access, 10*, 10852–10866.

Zhou, J., Joshi, P., Zeng, H., & Li, R. (2019). Btmonitor: Bit-time-based intrusion detection and attacker identification in controller area network. *ACM Transactions on Embedded Computing Systems (TECS), 18*(6), 1–23.

Olufowobi, H., Hounsinou, S., & Bloom, G. (2019, November). Controller area network intrusion prevention system leveraging fault recovery. In Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy (pp. 63-73).

Hamad, M., Tsantekidis, M., & Prevelakis, V. (2019, May). Red-Zone: Towards an Intrusion Response Framework for Intra-vehicle System. In VEHITS (pp. 148-158).

Song, H. M., Woo, J., & Kim, H. K. (2020). In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications, 21*, Article 100198.

Tang, F., Kawamoto, Y., Kato, N., & Liu, J. (2019). Future intelligent and secure vehicular network toward 6G: Machine-learning approaches. Proceedings of the IEEE, 108(2), 292-307.

Ahmad, U., Song, H., Bilal, A., Alazab, M., & Jolfaei, A. (2020). Securing smart vehicles from relay attacks using machine learning. *The Journal of Supercomputing, 76*, 2665–2682.

Gundu, R., & Maleki, M. (2022). Securing CAN bus in connected and autonomous vehicles using supervised machine learning approaches. In *In 2022 IEEE International Conference on Electro Information Technology (eIT)* (pp. 042–046). IEEE.

Kumar, P., Kumar, R., Gupta, G. P., & Tripathi, R. (2022). BDEdge: Blockchain and deep-learning for secure edge-envisioned green CAVs. *IEEE Transactions on Green Communications and Networking, 6*(3), 1330–1339.

Alsulami, A. A., Abu Al-Haija, Q., Alqahtani, A., & Alsini, R. (2022). Symmetrical simulation scheme for anomaly detection in autonomous vehicles based on LSTM model. *Symmetry, 14*(7), 1450.

Lee, K. S. (2021). Considerations for Cyber Security Implementation in Autonomous Vehicle Systems. In Proceedings of the 21st International Conference on Control Automation and Systems (ICCAS 2021) (pp. 1-6). IEEE. https://ieeexplore.ieee.org/document/9649850.

Sui, A., & Muehl, G. (2020). Security for autonomous vehicle networks. *In 2020 IEEE 3rd International Conference on Electronic Information and Communication Technology (ICEICT)*. IEEE.

Yang, J., Ni, Q., Luo, G., Cheng, Q., Oukhellou, L., & Han, S. (2023). A trustworthy internet of vehicles: The DAO to safe, secure and collaborative autonomous driving. *IEEE Transactions on Intelligent Vehicles, 8*(12).

Li, X., Na, Y., Huang, D., & Zhu, L. (2023). ADRC Controller Design for Autonomous Vehicles Queuing Systems in Zero-Trust Environment. In 2023 6th International Conference on Robotics Control and Automation Engineering (RCAE). IEEE. https://doi.org/10.1109/RCAE59706.2023.10398802.

Hu, J. D., Sun, K., Yang, S., Hui, Z., & Huang, S. (2023). A Software Security Testing Model for Autonomous Systems. In 2023 10th International Conference on Dependable Systems and Their Applications (DSA). IEEE. https://doi.org/10.1109/DSA59317.2023.00019.

Chen, C., Xiang, H., Qiu, T., Wang, C., Zhou, Y., & Chang, V. (2018). A rear-end collision prediction scheme based on deep learning in the internet of vehicles. *Journal of Parallel and Distributed Computing, 117*, 192–204. https://doi.org/10.1016/j.jpdc.2017.08.014

Liao, D., Li, H., Sun, G., Zhang, M., & Chang, V. (2018). Location and trajectory privacy preservation in 5G-enabled vehicle social network services. *Journal of Network and Computer Applications, 110*, 108–118. https://doi.org/10.1016/j.jnca.2018.02.002

Sun, G., Yu, M., Liao, D., & Chang, V. (May 2019). Analytical exploration of energy savings for parked vehicles to enhance VANET connectivity. *IEEE Transactions on Intelligent Transportation Systems, 20*(5), 1749–1761. https://doi.org/10.1109/TITS.2018.2834569

Özgür, A. (2021). Classifier selection in resource limited hardware: decision analysis and resolution approach. *Journal of Intelligent Systems: Theory and Applications, 4*(1), 37–42. https://doi.org/10.38016/jista.755419