Thesis

by Jahangir Mumtaz

Submission date: 02-Jun-2023 06:31AM (UTC+0500)
Submission ID: 2107093187
File name: Proposal__1__1.pdf (1.77M)
Word count: 15558
Character count: 83401

Multiple Attacks Detection scheme Along with the Identification of Vehicle Type in VANET's using Machine learning

Submitted by

Misbah Haider

19I-1242

Supervised by

Dr. Subhan Ullah

Masters of Science (Computer Science)

A thesis submitted in partial fulfillment of the requirements for the degree of

Masters of Science (Computer Science)

at National University of Computer & Emerging Sciences

Department of Computer Science

National University of Computer & Emerging Sciences

Islamabad, Pakistan.

March, 2023

Contents

List of Tables

List of Figures

Abstract

Intelligent Transportation Systems (ITS) is a group of automated technologies that connect automobiles, roads, and public transportation via the Internet. These technologies include wireless and automated systems that enhance safety, efficiency, and sustainability. Attackers in vehicle-to-vehicle communication are a serious threat to the security of modern vehicle networks. As these networks grow more complex, malicious attacks become more likely, making it crucial to implement rigid security protocols. In this paper we proposed a methodology to detect the attacks in vehicle-to-vehicle communication. And the proposed model also detects the 14 different type of attacks. We use five model to detect the attacks and its types. The model includes Decision Tree Classifier, KNN, LSTM, GRU, and random forest. The accuracy of the model on the detection of the attacks is Intelligent Transportation Systems (ITS) is a group of automated technologies that connect automobiles, roads, and public transportation via the Internet. These technologies include wireless and automated systems that enhance safety, efficiency, and sustainability. Attackers in vehicle-to-vehicle communication are a serious threat to the security of modern vehicle networks. As these networks grow more complex, malicious attacks become more likely, making it crucial to implement rigid security protocols. In this paper, we proposed a methodology to detect attacks in vehicle-to-vehicle communication. Moreover, the proposed model also detects 14 different types of attacks. We use five models to detect the attacks and their types. The model includes Decision Tree Classifier, KNN, LSTM, GRU, and random forest. For attack detection, the Decision tree classifier accuracy is 91% , KNN accuracy is 91.5%, LSTM model accuracy is 40%, Random forest accuracy is 96.8%, and GRU model accuracy is 40%. For the detection of attack type, Decision Tree Classifier accuracy is 96%, KNN accuracy is 96%, LSTM model accuracy is 23%, Random forest accuracy is 92%, and GRU model accuracy is 81%.

Chapter 1

Introduction

Intelligent Transportation Systems (ITS) is a collection of automated technologies that connect cars, users, and infrastructure, including roads and public transportation. These technologies include wireless, electronic, and automated systems that can improve transportation safety, efficiency, and sustainability. In advanced data communication technologies, ITS combines information, communication, computers, and mobility to create a network of people, vehicles, and roads that can operate more efficiently [1] With ITS, transportation service providers can move from managing roads to helping drivers, providing real-time and accurate transportation system management. In-vehicle technologies like automated guideways, exact docking of buses, and collision avoidance systems work together with infrastructure to prevent accidents and reduce traffic congestion. By leveraging cutting-edge technology, ITS can send road information to drivers and provide useful services, ultimately helping to make roads bigger and reduce traffic jams [2].

Vehicle-to-vehicle (V2V) communication is an emerging technology that enables cars to communicate with each other via wireless networks. V2V communication is an essential ITS component, which uses advanced technologies to make transportation systems safer, more efficient, and more environmentally friendly. V2V change information like speed, location, and direction of travel with nearby vehicles. V2V communication can enhance safety features, in-

cluding lane departure warnings, automatic emergency braking, and collision avoidance systems. This technology has the potential to significantly improve road safety by enabling cars to ”see” around corners and react to potential hazards in real time [3]

In a VANET, the cars have sensors and ways to communicate with each other, cameras and

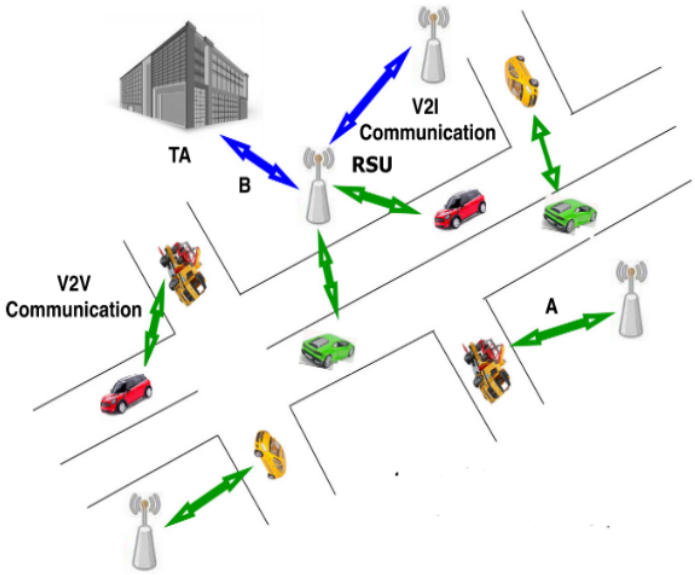

Figure 1.1: Basic Architecture of VANets.

GPS units. Through wireless interfaces, the sensors collect important information about the car, such as its location, acceleration, and speed, and then send that information to other vehicles and roadside units (RSU) [4]. A roadside unit (RSU) is a fixed device on the road's side that is part of a vehicle ad hoc network (VANET). RSUs are speed cameras, base stations for cell phones, and relay nodes. There are two ways to send these messages: first, between cars (V2V) using DSRC protocols, and second between vehicles and infrastructure (V2I) using mobile communication standards (l3G, LTE) to allow for long-distance communications [5]. Millions of vehicles will be on the road worldwide by 2024. Even a mid-range model comes standard with various devices

that help detect mechanical problems while driving and assist drivers, making their commutes safer and more relaxing. Additionally, it increases the risk of threats and cyberattacks, making vehicles more vulnerable to security problems as they become more connected. .

The safety and security of today's vehicular networks depend on being able to find attackers in vehicle-to-vehicle communication. Malicious attacks are more likely as these networks get bigger and more complicated, so it is even more important to put in place strong security measure [6]. This study suggests a two-step prediction method using machine learning and statistical modelling to find attackers in vehicle-to-vehicle communication. Attacks on vehicle networks could have far-reaching effects that could threaten passengers' safety and the systems that control traffic at risk. So, it is essential to ensure that vehicle networks have reliable security systems so that the transportation systems can keep running smoothly and reliably. This study significantly impacts the security of automotive networks because we create and test a new two-step method for predicting where attackers are in vehicular communication. The suggested method improves the accuracy and reliability of 14 different attacks detection.

Due to these advancements, Idea of Vehicular networks starts emerging. In this concept, vehicles interconnect with each other, share each other's resources and carryout the computations more rapidly and efficiently. Although vehicular network is an emerging field, it still requires a lot of efforts to implement it on larger scale. Huge amount of real time data is required, so that it can be processed to obtain exact situational awareness. Three major components of VANets are [7], Trusted Authority (TA), Roadside Units (RSU) and On board Unit (OBU). Trusted Authority is responsible for security of overall network. It is also used for the verification of RSU and vehicles, their authenticity so that a secure communication can be occurred [8]. If a vehicle or roadside unit becomes malicious or started misbehaving, then TA can also revoke its legitimacy. OBU is installed in every vehicle and a vehicle can communicate with other vehicles and Road side units through OBU [9]. Whereas RSA are placed along the road with certain distance. It not only provides internet connectivity to the OBU's but can also broadcast messages to other OBU's and RSU. The Figure 1.1 presents basic architecture of VANets [10]. .

Although this advancement in technology in VANets comes up with a lot of benefits and ease to the drivers, at the same time, these benefits also got some challenges. One of the major challenges is ensuring secure communication between V2V and V2I networks. Assuring the security of VANets messages is significant as the entities in VANets transmit messages through an open remote medium. This presents a chance for malicious persons to gain illegal access to communication channels for catching, replaying, deleting, and changing traffic-related messages sent or impersonating different vehicles. Malicious entities can also send fake messages to other vehicles or RSU to deceive them so that the wrong decision could be taken, resulting in losses like traffic jams and accidents. Therefore, it is important to detect attacks between V2V communication to ensure the communication is secure and reliable.

VANets are vulnerable to certain security attacks, including (Constant Speed, Eventual Stop, Constant Position, Constant Speed Offset, Grid Sybil, Constant Position Offset, Data Replay, DoS, DoS Random Sybil, Random Speed, Delayed Messages, Data Replay Sybil, DoS Disruptive, Random Speed Offset, Disruptive, DoS Disruptive Sybil, DoS Random, Random Position, Random Position Offset). These attacks may arise when communication occurs between the V2V network. Imagine two vehicles, V1 and V2, moving on a congested highway. V1 is controlled by a malicious attacker who wishes to cause vehicular chaos. The attacker has discovered a vulnerability in the VANet system that allows them to inject spoofed messages into the network. As V2 approaches V1, the attacker sends a fake message to V2 that warns of an impending traffic jam. This causes V2 to decelerate and take an alternate route, increasing traffic on the alternative road and causing inconvenience for other drivers. In addition, the attacker continues to spread the false message to nearby vehicles, causing traffic congestion and increasing the likelihood of collisions. As a result of this attack, the attacker can navigate traffic more efficiently and reach their destination faster. Due to the false information spread by the attacker, other drivers are left feeling frustrated and confused. This illustration underlines the importance of implementing robust security standards in VANets to detect and prevent malicious attacks and ensure the safety.

Security techniques are being used to ensure network security to mitigate different attacks.

These schemes have vulnerabilities along with benefits. The existing solution has many limitations. Some are used to detect specific attacks, and some model accuracy needs to increase to make a secure and reliable system. Moreover, some existing work eventually results in large computational overhead and delays. Hence there is a need for a strong mechanism which will not only detect a single attack and a limited range of attacks but also increase the maximum accuracy to develop a secure a reliable solution to detect attacks timely.

This research on multiple anomaly detection scheme is divided into six chapters to address the problem statement thoroughly. The first chapter contains a Case Study that provides a clear understanding of the problem by elaborating on a use-case scenario.

Chapter 2 contains a Literature Review that critically evaluates the current state of the work. It includes a comprehensive review of previous research and a comparison table to compare various approaches. Observing a comprehensive review of the existing literature, subsection 2.1 highlights the identified research gap. Section 2.2 defines the problem statement based on the identified research gap, while section 2.3 contains the research questions that will guide the proposed solution.

The third chapter focuses on the proposed solution to the problem. It briefly describes the proposed technique and a generic illustration of the proposed methodology. In addition, the dataset that will be utilized to implement the proposed method is described in this chapter.

Implementation chapter three focuses on implementing the proposed strategy. The implementation process and the tools and techniques are described in detail. In addition, the chapter describes the experimental setup, which includes the selection of datasets and evaluation metrics. The implementation's results are presented as graphs, tables, and other visual support.

The chapter four contains an analysis and discussion of the implementation chapter's results. The results of different models are compared to propose the best techniques to detect attacks in V2V communication.

This chapter concludes the research by summarizing the proposed methodology's principal contributions, accomplishments, and limitations. Future work is also discussed, including poten-

tial improvements, extensions, and new research directions.

The final section of this research contains a bibliography of all sources cited throughout the thesis. The structure of the thesis provides a comprehensive and clear understanding of the problem, the current state of the art, the research gap, the proposed approach, and the intended dataset.

1.1 Use-case Scenario

Here is an example use case scenario in which vehicle named as “Attacker” has been robbed by a malicious entity and now enters the Vanets network.As the vehicle enters in the network it verifies itself from the trusted agent as legitimate vehicle and receive the traffic congestion message from the other car. The Attacker Vehicle store the message and after some time when the road is cleared for traffic, broadcast the old beacon to other vehicles. RSU verifies the authentication certificate as the legitimate one and allows the message to be broadcasted. Which eventually results in the spread of false information to other nodes. Other vehicles will change their path give the attacker an easy path to move towards its destination.

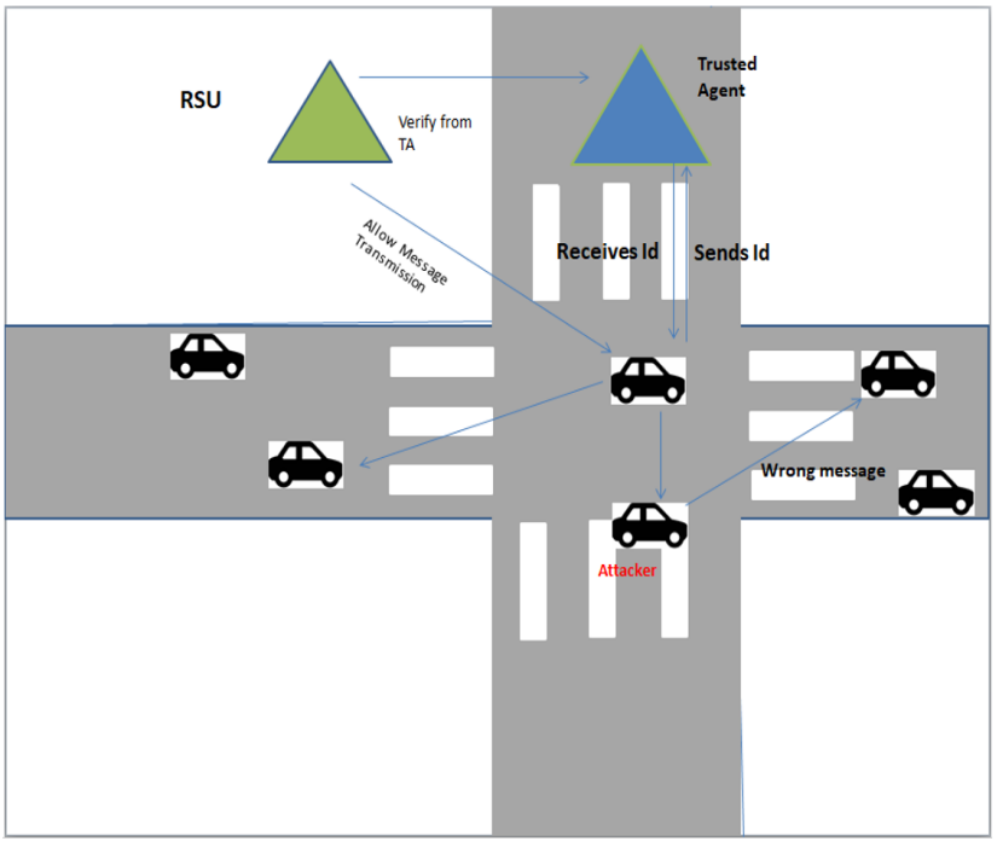

Figure 1.2: Use Case Scenario

Chapter 2

Literature Review

Xie et al. [6] proposed UWPEE, a trust evaluation scheme to find the wireless nodes behind attacks based on traffic. UWPEE is a way to find things based on wavelet packet energy entropy, a path planner, and UAVs that actively gather traffic data. This plan allows them to find attacks based on traffic in a 6G wireless system with the limited transmission, computing, and caching resources. The experiment results show that UWPEE is correct 84.47percent of the time against traffic-based attacks. Most of the time, they found only 4.89 percent of rogue nodes. Compared to the greedy method, the UAV flight distance was down by 15.44percent. Active detection methods based on UAVs are vital in determining trustworthy IoT devices. The results show that UWPEE is very accurate and uses little energy.

Cheng et al. [11] creates a deep evolving stream clustering model (DESC-IDS) for automobile intrusion detection systems by fusing sparse regularization convolutional auto-encoder (SRCAE) and stream clustering. By encoding continuous messages as 2-D data frames and feeding them into the SRCAE constructed by the temporal convolutional network (TCN), we can obtain a low-dimensional non-linear spatial-temporal mapping of the high-dimensional input. After that, the stream clustering model used the spatial and temporal features of average communication messages to describe a baseline contour for the flow of information. DESC-IDS picks

up on any sudden changes in the way cars talk to each other. In particular, the SRCAE is used to reconstruct message matrices, viewed as different attacks from those already known because of reconstruction deviation. Using the HCRL intrusion dataset, the proposed approach achieved an accuracy of 96.44percnt. The ORNL intrusion dataset achieved an accuracy of 98.80percent, demonstrating its superior performance and real-time competitiveness in anomaly detection. The suggested model consistently achieves F1 scores in the 93.48–86.99percent range when subjected to attacks that combine fabrication and masking. The performance against unknown threats is also good, with an F1 score of 97.15percent and a level of accuracy of 98.43percent.

Rani et al. [1] proposed an intelligent transport system for the IOVs-based vehicular network traffic in a smart city scenario by Using Random Forest (RF), tree-based Decision Tree (DT), XGBoost, and Extra Tree (ET) machine learning (ML) models. The results of the simulations show that by using ensemble learning and averaging the essential traits, the proposed system gives very accurate detection. Tree-based ML methods with feature selection did better than those without feature selection for IOV-based vehicle network traffic. They applied Many attacks, like the Dos and Sybil bugs. ML-based network traffic classification methods could help in intrusion detection, user and control traffic data, routing, real-time management of the Internet of Things, and building network profiles for proactive network monitoring. The Stacking model has a higher classification accuracy of 99.05percent than the lowest KNN accuracy of 96.6percent and the best SVM accuracy of 98.1percent.

Zhang et al. [12] proposed a method for detecting a Sybil attack that uses basic security message (BSM) packets. This method takes advantage of the fact that BSM packets are always received from the unique resource and uses the changing locations of vehicles to find and stop malicious activity. A weighted integration technique is proposed as a way to improve the accuracy of detection without using machine learning model prediction. Tests showed that the method could find Sybil's attacks in real-time, no matter how many attacks or how much traffic there was. Also,

it can find Sybil nodes and track down malicious nodes simultaneously with accuracy rates of ¿98percent and ¿94percent, respectively. This model resolves the flaws in previous identification methods.

Anyanwu et al. [13] proposed an IDS for SDN-integrated VANETs. Data about SDN DDoS attacks were used to compare the model. Their method used parameter optimization to reduce absolute error, improve accuracy, and simplify things. The GSCV method was used to avoid overfitting and get good results from the optimization process. The results were good when the RBF-SVM kernel was used with the GSCV method. They also looked at several different ML methods and used different ways to measure how well they worked. The experiment results show that with accurate assault recognition with an accuracy of 99.40percent and a precision of 99.26percent, the model did better than all other ML methods used to solve binary classification problems. Also, the framework could benefit from more ways to find attacks that use more complex data at higher levels.

Dey et al. [14] proposed a way to find and pinpoint Distributed Denial of Service (DoS) attacks in real time over an LTE-based vehicular network made up of mobile network components (like cars, femto access points, etc.). They consider both unintentional (through accurate identification) and intentional (through fake identification) ways cars could send insufficient data. In three important ways, this work adds a lot to the field. First, they find threats using the number of data packets sent and their average Packet Delivery Ratio (PDR). Then, they show how machine learning can help us make a better system for finding attacks. They use a few supervised classification methods based on machine learning to improve the reliability and consistency of detection. Lastly, they propose using the Data Packet Counter (DPC), geolocation, and measurement reports to find out where intentional and accidental DoS attacks are coming from. Using an M/M/m queue model, they determine the average message delay that cars have to deal with. The results of their experiments show that their proposed way does a lot better than the current

gold standard. the accuracy of the xgboost model is 99.15percent with a detection time of 96.50(s).

Anyanwu et al. [15] develops RSO-FDS, a scheme for finding fakes that use an ensemble of random search optimizers. The RSO method was used to make the Ensemble-based Random Forest (RF) model. This research used three datasets to test for and find instances of falsification in IoV. They looked at the six most popular supervised learning (SL) methods to learn how well the suggested RSO-FDS worked, which did the best across all datasets. Performance parameters like precision, recall, and F1 scores, as well as how quickly predictions can be made and how well attacks can be confirmed, are studied. So that the RSO-FDS could be proven to work, its effectiveness was compared to that of more modern research. It was also shown that data balance does not matter for real-world IoV use cases. The results show that the proposed model performs better than the most advanced methods. The model RSO-FDS was detected with 99.53percent accuracy, 99.90percent accuracy, and 99.99percent accuracy on the VeReMi, BurST-ADMA, and V2X fabrication test sets.

Karagiannis et al. [16] introduces a system based on unsupervised machine learning and a unique measure, the fluctuations in the relative speed (RSV) between the vehicle of the jammer and the vehicle of the receiver, to detect and classify RF jamming attacks. In order to evaluate the usefulness of the proposed statistic, we conducted tests replicating three different attack scenarios, two of which involved a mobile jammer and one of which involved interference alone. For both intentional and accidental RF jamming, our technique is able to recognize the distinct features of each attack type. Their research shows that the relative velocity and any changes to it significantly affect the efficacy of jamming detection. We also showed that a system could not distinguish between interference and intentional jamming scenarios or identify the specific aspects of an attack if it only uses standard physical and network layer metrics from wireless receivers, such as PDR, SINR, and RSSI.

Song et al. [17] introduces an intrusion detection system (IDS) based on a deep convolutional neural network (DCNN) to protect the controller area network (CAN) of any vehicle. The DCNN is able to learn network traffic patterns and detect malicious behavior without being programmed with any specific characteristics. By adopting a Deep Convolutional Neural Network (DCNN) model to the CAN bus's data stream, we were able to both boost the detection performance of the Inception-ResNet model and simplify its underlying architecture. We conducted an experimental study using a real car and the datasets we generated to determine how well our detection system worked. The findings of DCNN were compared to those of several other popular machine learning algorithms, including SVM, LSTM, kNN, NB, ANN, and decision trees. Although the LSTM and ANN performed better than other conventional techniques, their ER and FNR are twice as high as the suggested approaches. The suggested DCNN model outperformed the state-of-the-art methods on the fuzzy attack dataset, making it the superior machine learning method for dealing with complex irregular random attacks. Experiments demonstrate that the proposed IDS significantly reduces false negative and error rates compared to conventional machine learning methods.

Pavithra et al. [18] presented a method for message authentication, integrity and non-repudiation and also to address replay attack. PKI based cryptographic protocol is used for this purpose. Three major entities used in this architecture are Vehicles as a source element, Gateways as get- ting element and RA (Registration Authority) acts as a trusted mediator. Digital certificates are used for the authentication and non- repudiation purpose. Whereas they used hashing technique to make sure message integrity. Besides this, timestamps Ts were included in the messages to protect them from being replayed. At the end of paper, Scyther is used for the verification of protocol. Although the scheme is helpful in ensuring message authentication and integrity but it doesn't the confidentiality of the message.

Fotohi et al. [19] compiles a list of the drawbacks and challenges of the current methods.

In addition, we suggest an agent-based self-protective strategy (ASP-UAVN) for UAVNs that draws inspiration from the human immune system (HIS). In ASP-UAS, a self-protective mechanism is used to find the safest route between the UAVs. This tactic involves a group of agents outfitted with an AIS to locate the enemy UAV and choose the safest route. A route request packet (RREQ) is transmitted from the source UAV to the destination UAV in the planned ASP-UAVN to discover existing routes. Agents and the knowledge base are used in a self-protective strategy that, upon receiving the route reply packet (RREP), chooses the safest route and identifies the intruding UAVs. The suggested ASP-UAVN has been analyzed theoretically and simulated computationally for testing and evaluation. Simulation results and theoretical analysis show that the ASP-UAS is better than the SUAS-HIS, SFA, and BRUIDS approaches by more than 17.4percent, 20.8percent, and the detection rate by more than 17.2, 23.1, and 29.3percent, respectively, while lowering the PLR by more than 14.4percent, 16.8percent, and 20.21percent and the false-positive and false-negative rates by more than 16.

Park et al. [20] describes a machine learning-based data analysis approach for real-time detection of aberrant behavior caused by malware in massive network traffic. We specify a detection architecture required by the intrusion detection module to detect and stop malware attacks launched from smartphones. We developed a machine-learning method that successfully detects Android malware in automobiles. Six machine learning techniques were used to evaluate the algorithm's detection accuracy and speed, utilizing the recommended ideal hyperparameters. We also found that the novel score-function model for real-time detection may significantly reduce the overall detection time. We ran simulations to demonstrate our algorithm's suitability for real-time malware detection in an autonomous vehicle context. We found it accurate (92.9 percent of the time) and fast (0.049 seconds).

Wang et al. [21] proposes a distributed anomaly detection system based on hierarchical temporal memory (HTM) to improve the security of a vehicle's controller area network bus. Using

the flow data's learning history, the HTM model can make predictions in real-time. Additionally, we improved the mechanism that generates anomalous scores for use in judging the accuracy of the forecast. We manually synthesized data field tampering and replay attacks. Area under the receiver operating characteristic curve score, precision, and recall show that the HTM-based distributed anomaly detection system is better than recurrent neural networks and hidden Markov model detection models.

Cui et al. [9] proposed a secure privacy-preserving authentication scheme. Trusted agent assigns internal pseudo-identity (IPID) to the vehicle through its original identification. After that, the vehicle chooses an encryption key and places it into TPD alongside IPID. Than after the authentication to trusted agent, the vehicle creates public pseudo-identity (PPID) and afterwards, generates a signature over the given message. The recipient checks the message by confirming the signature. Eventually, the vehicle regenerates IPID values and encryption key periodically for stop the in- formation leakage. Additionally, this method ensures unlinks ability, non-forgery, confidentiality, and provides prevention against the side-channel attack. Nevertheless, it does not specify any details of the end-to-end messages delays.

Sharma et al. [22]In this research, the author advocates for a data-driven, supervised machine learning (ML)-based methodology for identifying behavioral outliers. In addition, six different algorithms were instantiated and compared to the model, and plausibility tests were incorporated utilizing ML approaches. The model does more than detect harmful behavior; it also classifies attacks, which may be used to validate security solutions. This paper focuses on the supervised learning algorithms for IoV behavior identification, analyzing their performance and suggesting how this field should develop further. The VeReMi dataset, a simulation of road traffic that includes a vehicle-to-everything (V2X) position forgery attack, is used to test the proposed model and gauge its efficacy. Some examples of performance metrics are the area under the receiver operating characteristic (ROC) curve and the ratio of precision to recall (PR). The results

demonstrate the value and significance of ML in identifying problematic behaviors in IoV. When plausibility tests are used, the accuracy and recall improve by 5percent and 2percent, respectively.

Barletta et al. [23]proposes a highly-effective and robust unsupervised Kohonen Self-Organizing Map (SOM) network for detecting intrusions via attack messages on the Controller Area Network (CAN) bus. Due to its high detection rate, short training time, and flexibility, the SOM network has found widespread application in intrusion detection. Our research suggests the SOM network might benefit from CAN bus intrusion detection. Combining the SOM network with other clustering algorithms, such as the k-means algorithm, was just one of many hybrid solutions to increase the model's accuracy. We developed a distance-based approach and compared it to the original K-means method to better include the SOM network. A car hacking dataset was used to test the models. The data consisted of CAN bus traffic data messages, notorious for their huge volume, low feature count, and uneven distribution. The experimental findings showed that the proposed method had a much higher detection accuracy than the standard method.

Gyawali et al. [24] provides a machine-learning and reputation-based MDS that may increase detection precision and ensure the reliability of both vehicles and messages. We initially simulate a realistic vehicular network environment to train the proposed MDS. To improve detection rates, we implemented a system based on the Dempster-Shafer (DS) theory that relies on group cooperation to identify inappropriate behaviors. The suggested method uses a Dempster-Shafer-based feedback combination, with the vehicles' reputation scores used as the belief values. Additionally, a beta-distributed reputation removal and update system is proposed. In addition, we show that our proposed method surpasses existing methods in accurately recognizing various harmful actions.

Park et al. [25] proposed a cloud-based security system and a lightweight authentication architecture for car networks. They proposed a communication security design. The edge-based

EDC_V made authentication keys for certifying vehicles in a vehicular cloud and private keys for securely exchanging messages. The EDC_V sent out these keys. In addition, deep learning models were run at a cloud data centre on the network's edge, and the results were then sent to a cloud in a moving car. Many simulations showed that the suggested authentication architecture significantly impacted the level of security. The authentication architecture gets great detection rates depending on how many cars are enrolled in the CAN-based intrusion detection system. Extensive simulations show that the suggested authentication architecture significantly impacted the level of security. The F1-score varies from 94.51 to 99.8percent when traffic on the control area network is used to test the proposed authentication architecture.

Helmi et al. [26] The Vehicular Ad Hoc Network (VANET), or vehicle network, is an Intelligent Transportation Systems (ITS) system made for self-driving cars. VANET's way of talking to each other is based on a radio network that can be easily broken. The Sybil attack is one type of attack that can hurt a network by sending fake data to all of the nodes in the system. Sybil attacks the network by pretending to be a node and sending fake data to other nodes in the area. This study uses a deep learning system to look at the patterns of attacks so that Sybil's attacks can be predicted. The independent factors being tested here are time, place, and the amount of traffic. It finds 94percent of Sybil attacks by using a deep learning algorithm that mimics the pattern of a Sybil attack and combines many different factors, such as time, place, and traffic density .

Velayudhan et al. [27] proposed a Deep Residual Network based on Competitive Dolphin Echolocation Optimization (CDEO) to find a Sybil attack and RSU misbehavior. This is the best way to send data flows through a traffic-aware routing system built on Fractional Glow-Worm Swarm Optimization (FGWSO). At the hub, Sybil attacks are found. Using the suggested CDEO method, a Deep residual network is trained to find Sybil attacks. The CDEO algorithm combines the Dolphin Echolocation Optimization (DEO) and Competitive Swarm Optimizer (CSO) methods. Odd RSU behavior can also be found with the help of the Deep residual network. Precision,

F1-measure, and recall are measured against the performance of the created method and found to be 0.9197, 0.9121, and 0.9046, respectively.

Elsayed et al. [28] proposed a BoostGuard, a new behavior detection system. It uses the boosting ensemble method. Decision trees are added to the suggested ensemble to help find fake and strange BSMs in-car networks. BoostGuard is much more useful to cybersecurity experts when looking into a hacked vehicle because it is an effective tool to identify the exact attack type. The experimental evaluation confirms these contributions by showing that BoostGuard has great detection skills (F1-measure of 0.99) and beats state-of-the-art methods (F1-measure of 0.89) in this area. In our study on learning strategies, we will pay special attention to how parallel processing can speed up data analysis. The plan will also think about new ways to attack and how to find them.

Mundhe et al. [10] gives detailed description of the literature in the area of VANets. Authors presented a precise review on the existing solutions that address the security features of Vehicular networks. A comprehensive study on different attacks like Impersonation attack, Sinkhole attack, Sybil attack, Tunneling, GPS spoofing, Message replay attack, Message falsification, Message delay attack, Illusion attack, Message alteration attack, Greedy drivers, Gray hole attack, Illusion attack, Industrial insider attack Forgery etc. and their expected adversaries with proposed solutions has been done. They have also compared and analyzed possible solutions of these attacks. Major cryptographic schemes that are being discussed, Identity Based Cryptography, Symmetric Key Cryptography, Certificate less Cryptography etc. At the end, they have talked about a few open issues which should be an area of research in future.

Anyanwu et al. [29] used the Radial Basis Function (RBF) kernel of the Support Vector Machine (SVM) classifier and the exhaustive parameter search method Grid Search Cross-Validation (GSCV). The proposed architecture is used by each car's On-Board Unit (OBU) to process vehicle

data. It does intrusion detection to determine if a certain message sequence is part of a distributed denial-of-service (DDoS) attack. The proposed algorithm was compared to state-of-the-art ML algorithms using key performance measures. Experiments and models have shown that the proposed system can spot DDoS attacks, which proves its worth. The best results were found when the RBF-SVM kernel settings "C" and "gamma" () were set to 100 and 0.1, respectively. The suggested method beat the best baselines with a total accuracy of 99.33percent, a detection rate of 99.22percent, and an average squared error of 0.007.

Basarvaraj et al. [30] an artificial neural networks (ANN) model of an Intrusion Detection System (IDS) that works well to find problems in the transportation network. This study uses a Control Area Network (CAN) dataset made in real-time. The idea groups the different kinds of car attacks into three categories: reconnaissance, denial of service, and fuzzing. Experiments are completed to compare different classification models based on their accuracy, precision, memory, and F-1 score. Their model meets or exceeds important IDS measures, such as an accuracy of 0.986 (98.68percent), a recall of 0.967, a precision of 0.967, and an F-1 of 0.967. This effectiveness is achieved by fine-tuning key factors while experimenting with how the model works by changing the number of hidden layers, the encoding methods, and the learning rate.

Jeong et al. [31] uses an intrusion detection method to find AVTP stream injection attacks in Ethernet-based car networks. A convolutional neural network (CNN) creates features for the proposed intrusion detection model. Based on Broad R-Reach, we built a real-world testbed and used packet capture to test how well their intrusion detection system worked. The results from the experiments show that the model works better: Both F1-score and recall are greater than 0.9704, and F1-score is 0.9949. Their CNN model works well for real-time identification because it has a low inference time per input and short times between when AVTP traffic is generated.

Han et al. [32] examines how complex value neural networks (CVNNs) find arbitration

fields (CAN IDs) to protect CAN networks. Using the auto-encoder method, they made an encoder to pull out shallow features. They also made a random phase that rotates features in the complex-valued domain to hide the real features. The suggested processing approach then uses an attention-based methodology to extract relevant data. The real-time detection demonstrates that the IDS constructed has a high level of accuracy, up to 98percent, after odd data was introduced into the actual automobile to create the CAN dataset. In particular, the attack experiment demonstrates how their approach hinders the adversary's ability to gather actionable intelligence.

Al et al. [33] says that the objective of Vehicular Networks are to facilitate road users and formulate traffic management well organized. Still, the extensive nature of VANets communication network medium reveals the messages towards security attacks and the data within to privacy breaches. Researchers suggested numerous security schemes and methods to resolve the privacy and security issues in VANets. Whereas, majority of the existing schemes shows high communication and computation overheads. In this paper, a detailed review on the replay attacks and their prevention schemes has been discussed. In conclusion authors suggested creating more schemes for the mitigation of this attack.

Alazzawi et al. [34] presented a conditional anonymity scheme which is based on authentication and integrity so that a secure communication in vehicular networks can be provided. In the proposed scheme, during the authentication phase, pseudo-identity has been used with the RSU. Authors discussed these constraints as the vehicle message signing by a signature which is obtained from the RSU. At joining phase, Vehicle which requires authentication for themselves with trusted authority get these pseudo identities through the RSU. Initially it embeds a timestamp along with the message. Then, the RSU will check the legitimacy of the time stamp before proceeding with the cycle. Additionally, they embed timestamp with the last message in the broadcasting step.

Majid et al. [35] The author proposed RSU-based approach, which uses a temper-proof

device (TPD) to add the master key of trusted agents to the roadside unit. Rather than embedding a master key into all of the OBUs, it makes sense to store the TA's master key in the RSUs, as their communication pathways are more secure and faster. However, it is still vulnerable to replay attack. For example, a malicious entity can capture a last message during the transmission stage in the event that he/she wants to damage the street in the VANet framework. On the other case, when the vehicles are in a tight spot like traffic jam situation, malicious entity can block numerous signature of various vehicles and replay the signature to make a deception of a jam and misdirect different vehicles to stay away from that specific part of the street.

Raya et al. [36] presented an authentication based scheme on public key infrastructure (PKI).In this scheme, a huge number of key pairs and corresponding anonymous certificates need to be preloaded in OBU. Every time during the communication, vehicle chooses a public/secret-key pair randomly. With the help of those keys pairs vehicles can attain integrity along with the authentication. In any case, the TA needs to store every one of the vehicle's certificates and the vehicle needs to preload a great deal of public/private key combines and related advanced certificates which causes immense capacity trouble. Furthermore, when the authority needs to follow the vehicle's original Id, it needs to execute a thorough inquiry in an enormous database. Which causes delay and increase in computational cost of the system.

Kamel et al. [37] created an extension of VeReMi datasets in which they have added new attack sets and data points. Veremi dataset is use to evaluate misbehavior detection in VANets. It consists of data corresponding to security attacks and faulty transmissions. Dataset includes the following nine cybersecurity Attacks Data Replay, Data Replay sybil, Dos (Daniel of service), Traffic congestion sybil. Data coincides to faulty transmission includes the velocity offset/ constant position, constant position/velocity, velocity offset/random position and velocity/random position.

Bereczki et al. [38] firstly discussed an overview multiple attacks that occurs in VANets. Then a brief introduction of machine learning has presented. A comparative study has been conducted in which certain machine learning algorithms that are used in VANets such as KNN, SVM, Naive Bayes, K-means are compared. These algorithms have been compared on the basis of five major criteria including scalability, fastness, outliers, memory, high dimensional data. With the help of this comparative study it becomes easier to select certain ML base algorithm for certain case scenarios in VANets.

Wei et al. [39] presented an authenticated key agreement scheme for securing V2V and V2I communications in VANets. Proposed scheme has been divided into three major steps. The first step is about the authentication step in which trusted agent, RSU and vehicles authenticate them- selves. Next step is of key arrangement and in the last step, tree based Key algorithm has been applied for the prevention of malicious attacks and to secure authenticity of the VANets network.

Vijayakumar et al. [40] proposed a scheme based on dual authentication for vehicles communication in the Network. Methodology depends on the user's fingerprint and secret key of the vehicle. Authors have used CRT based key management system so that computational cost can be reduced. As they've focused on the group communication in the proposed scheme, so the information required for update the group key has also been reduced. which helps in reducing computation cost as well as less communication overhead. In addition to this, purposes scheme also provide resistance for various attacks.

Azees et al. [41]proposed anonymous authentication scheme which prevents the entry of vehicles with malicious behavior in the network. Scheme has been divided into five phases. First phase includes key generation and registration phase. whereas in second phase anonymous certificates have been generated. Third step comprised of signature generation and in forth step

message verification have been done. Last phase is of conditional tracking which will immediately revoke any vehicle which will found doing misbehavior. At the end of the paper, performance analyses of the given approach have been generated in which authors have considered the certificate's computational cost, serving capability of road side units and verification process of signature. Analysis results showed that the purposed scheme generate less computation cost and also resistant again certain security attacks as well.

Raya et al. [8] presented a method in which regardless of one public key, there are anonymous public keys used for authentication and privacy of vehicular communication. With the help of this, message receiver is not able to detect the sender and hence privacy of the sender has been achieved. One of the major limitation of this scheme is that each vehicle has to store huge number of keys as well as certificates which eventually required storage space. And due to the use of alot of anonymous keys, they also require same number of CRL checks as well which leads to large computational overhead. The purposed scheme was also vulnerable to several attacks, one of the major is which is Replay attack.

Singh et al. [42] apply ML based approach for the detection of position falsification in VANets. VeReMi has been used for feature extraction along with SVM and Logistic Regression. SVM gave the finest feature set which include the speed, position and position speed difference between message sender and receiver. As VeReMi data set has been improved and new features are also added to the dataset like acceleration and heading. It will now help in extracting more feature combinations and feature selection.

So et al. [43]studied the VeReMi dataset with SVM and KNN algorithms. They added some changes while labelling the data so that it will ease the classification. During experimentation, they have identified a minor vulnerability in VeReMi data set. On an eventual stop of a vehicle, few time before performing an attack, it is labelled as attacker node. Researchers con-

cluded in their discussion that a node should only be labelled as attacker node when it performs a malicious activity not few time before the misbehavior. The vulnerability has been resolved in the upgraded version of VeReMi and now a node has only been labelled as attacker node when it is actively participating in malicious activity.

Kushardianto et al. [44] develops ways to detect attacks on ITS vehicle-to-vehicle transmission. The 2-Step Prediction method can improve the accuracy of each model ML even more because it focuses on classifying the type of attack after first telling the difference between the attacker's vehicle and the actual vehicle. Regarding grouping and 2-Step Prediction, GRU and LSTM are the most accurate, while DBN is the least accurate. In 2-step Prediction, the LSTM method takes the most time, while the Random Forest method takes the least. When the number of messages per car increases, it has little effect on how fast the model can find things. The higher number of messages per vehicle in DBN makes it slightly less accurate than the other three models, but this is still a big gain. Compared to the other three types, adding more messages per car has less of an effect on the speed of the LSTM. The accuracy of each ML model's predictions will improve if the 19 attacks in the VeReMi dataset are grouped into 14 separate groups. The accuracy of the LSTM model is 95percent, while the accuracy of the GRU model is 86.4percent.

Sun et al. [45]proposed the Convolutional Neural Network-Latent State Transfer Memory model (CLAM), a new way to find intrusions in the CAN network. In order to obtain the abstract features of the signal values at each time step, one-dimensional convolution (Conv1D) is used. The CLAM model's temporal dependence is obtained by feeding these features into the Bi-LSTM. We can juggle the attention processes by calculating the importance of each hidden state output from Bi-LSTM and doing a weighted summing. The model will converge more quickly and produce more accurate predictions. The suggested model can be applied to multiple vehicles without parsing the CAN communication matrix since it uses the bit flip rate to extract continuous signal boundaries from 64-bit CAN data. Their CLAM model has an average F1 score of 0.951 and an error rate of 2.16percent when detecting CAN assault. When compared to other research,

this one catches 2.5percent more attacks.

Table 2.1: Comparison Table

Ref	Year	Model Used	Attacks	Dataset Used	Limitations
[46]	2021	kNN, RF, and an ensemble learning	Position Falsification Attack	Veremi	They only work on a single attack
[47]	2021	LSTM, CNN	Position Falsification Attack	Veremi Extension	Mechanism works for only position falsification attacks, and they Haven't explored other ML models for their technique
[48]	2022	LSTM, GRU , RF	Multiple Attacks	Veremi Extension	Implementation results showed decrease in accuracy,and they Haven't explored other ML models for their technique
[12]	2023	Linear Regression	Sybil Attack	Veremi	Mechanism works only on a single attack and they Haven't explored other ML models for their technique
[36]	2022	ANN	DOS Attack	real-time generated CAN	Mechanism works only on a single attack without exploring other ML models for their technique

2.1 Research Gap

The safety and security of today's vehicular networks depend on being able to find attackers in vehicle-to-vehicle communication. Malicious attacks are more likely as these networks get bigger and more complicated, so it is even more important to put in place strong security measures. Assuring message security in VANets is significant because the nodes in VANets transmit messages through open remote medium. Which easily gives chance for malicious persons to obtain illegal access to communication channels for catching, replayed messages sent. Malicious entities can also send wrong messages to other vehicles or RSU to deceive them so that bad decision could be taken which finally results in losses such as traffic jams and accidents.The min issue that arises is how to identify and detect multiple types of attacks in one simulation along with improved accuracy performance. Another challenge is identify best ML model for detection of these identified attacks and at the same time knowing the advantages and disadvantages of each model in terms of accuracy and timing. Existing security techniques have many limitations including they detected a limited range of attacks many others have less accuracy, and other have high computational cost for attack detection in the vehicle network.

2.2 Problem Statement

Detection of maximum attacks at the same time along with the identification of legitimate or attacker vehicle with improved accuracy performance in Vanets is a challenging task.Also identification of best ML model with maximum performance time and accuracy needs to be determined.

2.3 Research Questions

Following are the research questions which will going to be answered at the end of this research: Q1: What are Vehicular adhoc networks, it's security threats and existing approaches use for the detection of security attacks in VANets?

Q2: How machine learning approach be considered as more reasonable solution in multiple attacks detection and to ensure message integrity as compare to other existing approaches in VANets?

Q3:How does the size and composition of the training dataset impact the performance and generalizability of the anomaly detection system in vehicular networks?

Chapter 3

Proposed Approach

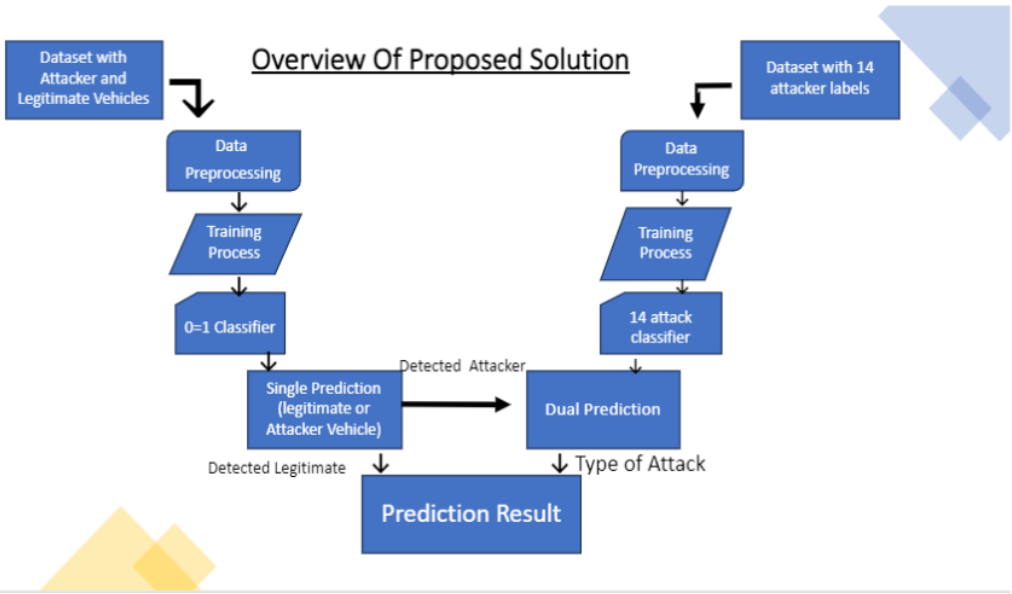

Figure 3.1: Anomaly Detection Scheme

We have implemented a duel-step detection and classification system to distinguish between income messages from legitimate vehicles and those from attack vehicles. First, a binary

classifier (0-1 classifier) is used to classify the vehicle's behavior as legitimate or malicious. This classifier's output is a trigger for the step-2 classification system, which is activated if a vehicle is suspected of being an attacker. Secondly, the data from the suspect vehicle is fed into a second predictor that seeks to identify the type of attack. This secondary classifier is designed to identify various 14 V2V Network attacks. The incoming messages are analyzed and classified using this two-step attack detection and classification system. The first stage identifies the vehicle's overall behavior, while the second stage focuses on identifying the type of attack. This method enables a more thorough analysis of the vehicle's behavior and improves the system's ability to detect and respond to various V2V network attacks, as shown in Figure 3.1

3.1 Dataset

The VeReMi Extension dataset has been created to assess the effectiveness of misbehavior detection methods in Vehicular Ad-Hoc Networks (VANETs). It comprises message logs from onboard units, accompanied by a labelled ground truth generated from a simulated environment. The dataset includes malicious messages that cause incorrect application behaviour, which the misbehaviour detection mechanisms aim to prevent. The initial release of the VeReMi dataset contains simple attacks, but it aims to be a starting point for more complex attacks. The message logs of each vehicle contain GPS data (labelled as type=2) about the local vehicle and Basic Safety Messages (BSM) (labelled as type=3) received from other vehicles via Dedicated Short-Range Communications (DSRC).

VeReMi dataset has two primary objectives. Firstly, it serves as a benchmark for evaluating the performance of misbehaviour detection mechanisms on a city-wide scale. Secondly, it saves computational resources typically required to run VEINS frequently. The VeReMi dataset contains different density levels, different attacks, and different attacker densities. VeReMi-extension, or VeReMi, is a dataset created to test and compare different methods of identifying VANET activity. The VeReMi dataset is an invaluable resource for researchers interested in evaluating

and enhancing the effectiveness of misbehaviour detection mechanisms for VANETs. Several different basic and complicated kinds of attacks are included in the raw data. Misconduct/Attacks Listed in VeReMi.

VeReMi dataset consists of message logs per vehicle and originally it has four main data fields:

- Position
- Velocity
- Acceleration
- Heading

First, we will check that the message on our system is an attacker or non-attacker. We have indicated 0 for non-attacker and 1 for attacker messages.

3.2 Methodology

The proposed methodology for detecting attacks in V2V communication consists of five steps. First, we collect the VeReMi dataset, which contains 14 different types of attacks, and combine it into a single file, extracting only the relevant columns. In the second step, we preprocess the dataset. Furthermore, we convert the position and speed columns into the standard format. Finally, we use feature engineering to extract relevant features from the dataset and divide it into training and testing sets. We use decision tree classification, KNN, LSTM, GRU, and random forest models. In the fourth step, we train the models on the training data and use the trained models to predict whether the message is from an attacker or a non-attacker. Finally, we compare the performance of various models for attack detection in V2V communication. The proposed methodology of the model contains the following steps (Data collection, data preprocessing, feature engineering, model implementation, performance matrix) as shown in figure 3.2.

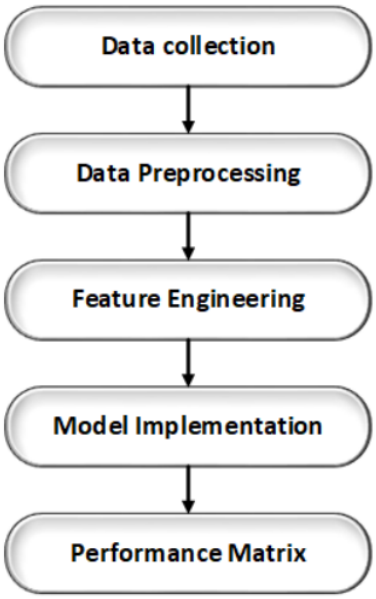

Figure 3.2: Methodology

3.3 Data collection

Data collection is an important part of the research and analysis process. The data collected must be accurate, valid, and represent the community or sample being studied well. We use the VeReMi dataset, a publicly available dataset, to evaluate various attack detection models. The VeReMi dataset provides a valuable resource for researchers interested in developing and evaluating attack detection models in vehicle-to-vehicle communication. Using this dataset, we compare the performance of different models and identify the most effective techniques for attack detection in V2V communication.

3.4 Convert JSON to CSV format

Data preparation is the procedure of converting raw data into a structure data that is appropriate for analysis. The VeReMi dataset is a collection of data in JSON format, and data is compressed into a zipped file. To start working with the data, unzip the folders to extract the contents of the compressed file. It generates a directory containing the VeReMi dataset's files and subfolders. VeReMi dataset includes 14 different types of folders of different attacks. After extracting the dataset, we convert data from JSON to csv format. We aim to convert the dataset to csv format to make it easier to deal with. After converting the data to csv format, it will include several rows and columns. Each row in the dataset represents a specific instance, whereas each column represents a specific data property, as shown in the figure 3.3.

	type	rcvTime	pos	pos_noise	spd
0	2	28800.000000	[1273.113237925505, 975.233376565642, 0.0]	[4.934850850783329, 4.880370582427511, 0.0]	[-1.488384819915062, -0.280261496830701, 0.0]
1	3	28800.229258	[1110.2859500104034, 959.2305027454435, 0.0]	[4.34693211998391, 4.199623283149641, 0.0]	[12.835311402776867, 3.303194608563299, 0.0]
2	3	28800.281468	[1380.171631017499, 1123.3789460767741, 0.0]	[3.508445985803822, 3.5053203032572453, 0.0]	[-5.578824783561503, -12.869926898902007, 0.0]
3	3	28800.511271	[1176.4361986904235, 966.8138388594847, 0.0]	[3.512651622112204, 3.698216342455024, 0.0]	[23.335900525418825, 11.263953772269755, 0.0]
4	3	28800.517869	[1300.9047627185794, 1002.8576331806318, 0.0]	[4.152066965092342, 4.132776279337015, 0.0]	[9.526369066722076, 11.073777306901396, 0.0]
...	...	...	...	...	
2227777	3	32398.659024	[877.472440401282, 709.2629266410954, 0.0]	[4.22164566454426, 4.644622274085652, 0.0]	
2227778	2	32399.000000	[686.2083625386699, 850.1871544868113, 0.0]	[4.939600631307386, 4.783735082155642, 0.0]	
2227779	3	32399.242401	[939.9504508743767, 659.6827113012921, 0.0]	[4.12591974019814, 3.94387385713614, 0.0]	
2227780	3	32399.659027	[886.6934232166074, 700.3926885852319, 0.0]	[4.285544791419161, 4.377745160206796, 0.0]	
2227781	2	32400.000000	[678.447534682301, 855.9610318191689, 0.0]	[4.788545746786137, 4.8450713291553225, 0.0]	

Figure 3.3: CSV Dataset Frame

3.5 Data Preprocessing

After combining all 14 attacker's files and converting them into one file, we collected 10,000 rows from each CSV file in chunks of each attack. Next, we extracted two columns from the VeReMi dataset: speed and position. These two columns are important for detecting attacks in V2V communication. We remove the other column from the dataset. The column of speed and position is in the list.

3.6 Data splitting

We transform the columns of speed and position into separate columns for each of their three elements. We create three new columns for each of these columns and then assign the values of each list element to its corresponding column. This way, each row will have six columns (three for speed and three for position). We extract each list element into a separate column to make the data suitable for the model. So we split the position list data into three columns (pos0, pos1, pos2). And speed list data into (spd0, spd1, spd2), as shown in the figure 3.4.

se	acl	acl_noise	hed	hed_noise	sendTime	sender	senderPseudo	messageID	Attacks	pos_0	pos_1	spd_0	spd_1	pos_2	spd_2
.0]	[2.4853506544000004e-05, 2.4853506544000004e-0...	[0.000896941097843, 0.000896941097843, 0.0]	[0.05705384081456600 6, -9.9903711029713881, 0.0]	[17.93510627825615, 17.93510627825615, 0.0]	NaN	NaN	NaN	NaN	ConstPos	143.932447	1022.904829	0.000000	0.000000	0.0	0.0
51, 14, .0]	[-0.1207047067893 04, 0.9289683444301011, 0.0]	[0.000326565860304060 04, 0.000106924987624, 0.0]	[-0.042720860595233, 0.999087047293679, 0.0]	[9.477199332411711, 1.5148029680231 01, 0.0]	26991.013798	4773.0	1047732.0	11437406.0	ConstPos	354.420706	641.637176	-2.108532	16.227503	0.0	0.0
.0]	[4.6772491805e-05, 4.6772491805e-05, 0.0]	[0.0, 0.0, 0.0]	[-0.024017036939449003, -0.9997115493664401, 0.0]	[11.698944553208877, 11.698944553208877, 0.0]	26991.234059	4923.0	1049232.0	11439177.0	ConstPos	153.586103	900.726664	0.000149	0.000149	0.0	0.0
.0]	[3.4836388096e-05, 3.4836388096e-05, 0.0]	[0.0, 0.0, 0.0]	[0.053922819258641, -0.9985451064239411, 0.0]	[11.567081013381452, 11.567081013381452, 0.0]	26991.311351	5073.0	1050732.0	11439523.0	ConstPos	928.797437	1091.099487	0.000020	0.000020	0.0	0.0
08, 41, .0]	[0.056580695793518, -0.3529195561332 76, 0.0]	[6.612632522000001e-06, 2.9681184266850194e-07...	[-0.07367577035818901, 0.9972822473413061, 0.0]	[7.836759568774746, 1.628267345895968, 0.0]	26991.335942	4797.0	1047972.0	11439877.0	ConstPos	155.782562	771.833062	-1.968970	13.744119	0.0	0.0

Figure 3.4: Data Splitting

After extracting these columns, we remove the actual speed and position columns from the dataset,

as shown in the figure 3.5.

	type	pos_0	pos_1	spd_0	spd_1	Attacks
0	0	143.932447	1022.904829	0.000000	0.000000	ConstPos
1	0	354.420706	641.637176	-2.108532	16.227503	ConstPos
2	0	153.586183	900.726664	0.000149	0.000149	ConstPos
3	0	928.797437	1091.099487	0.000020	0.000020	ConstPos
4	0	155.782562	771.833062	-1.968970	13.744119	ConstPos
...	...	...	...	...	...	...
159995	1	132.119616	889.446690	0.000505	0.000505	DoSDisruptive
159996	1	510.372108	860.120894	8.855281	8.277780	DoSDisruptive
159997	1	630.371460	420.831258	14.423676	5.560019	DoSDisruptive
159998	1	800.305077	791.123899	12.422521	-3.610628	DoSDisruptive
159999	1	838.941172	742.790481	11.638469	-9.348271	DoSDisruptive

[160000 rows x 6 columns]

Figure 3.5: Filtered Dataset

3.7 Check Null Value

We use the isnull() function in Pandas DataFrame to find and check for null values column by column. This method is used to find out if a given number is NaN. The output of the null values shows that the type and attacks column has no null values, but the other columns have null values, as shown in Figure 9.

In this step, we find the information of the complete dataset, like their attack names, types, non-null values and data types of the dataset, as shown in figure 10.

we selected specific columns, including type, pos0, pos1, spd0, spd1, and attacks, as shown in Figure 6. The type column is the target column, while the other columns are used to train the model. In this step, we dropped the attacks column as it was not used, as shown in figure 11. When scaling the data, we indicate that we are not considering the initial index items, which include the column of attacks and the type of attacks.

3.8 Feature Engineering

We have applied feature engineering to split the dataset. We divided the data and used 80percent for the training of the model and 20% for testing the model. The dataset values to train and testing on x, y are shown in Figure 3.6. The x-training contains the position elements, and the y-training contains the labelled dataset.

```
(128000, 4)
(32000, 4)
(128000,)
(32000,)
```

Figure 3.6: Training and Testing Data

Shuffle Dataset

We convert the dataset into shuffle form to test the five different models. We consider the dataset in pair (0,1) rather than consider only 1 or 0. After shuffling the dataset, the indexing of the dataset becomes irrelevant. Then we reindex the indexing and remove the index column. And we collect 30 rows of dataset. Firstly, we check the accuracy of the models on the 5 rows, then on 10 rows, on 15 rows, on 20 rows, on 25 rows and lastly on 30 rows of the dataset.

Type as a Target

In this step, we differentiate between the messages from attackers and non-attackers. We assign a binary label of 0 for non-attackers and 1 for attackers. The model is then trained using

these labels along with the other features in the dataset. Once the model is trained, we test it on a subset of the dataset. Specifically, we test it on the first five elements of the dataset and then on 10 elements, 15 elements, and 20 elements. This ensures the model performs as expected and correctly identifies attackers and non-attackers in the dataset. We also record the start and end times and the model's accuracy during testing. This is important information as it allows us to assess the model's efficiency and determine whether it is suitable for use.

3.9 Model selection

Model selection is an important phase in the V2V communication attack detection process. Several models are used for attack detection in V2V communication, including Decision Tree classification, KNN, LSTM, GRU, and Random Forest.

Decision Tree classification

We implement a decision Tree classification model that is specific yet effective for regression and classification issues. It is a tree-structured model with internal nodes representing features, branches representing decision rules, and leaf nodes representing the result. The model is simple to understand and handles category and numerical data.

KNN

Another model we used for classification jobs is KNN or K-Nearest Neighbors. It is a non-parametric model without assumptions about the underlying data distribution. Instead, it simply examines a new data point's K-nearest neighbors and assigns it to the most prevalent class among those neighbors.

LSTM

We used LSTM deep learning method that is specifically developed for sequential data. LSTM models understand long-term dependencies and model complex data patterns; however, they necessitate a huge amount of data.

GRU

We used GRU model that is specifically developed for sequential data. GRU models understand long-term dependencies and model complex data patterns; however, they necessitate a huge amount of data.

Random Forest

We used Random Forest, an ensemble model combining numerous decision trees to improve performance while minimizing overfitting. The model builds each decision tree by randomly selecting a subset of the characteristics and data and then aggregates the forecasts of all trees to generate the final prediction.

Chapter 4

Performance Evaluation

4.1 Step 1

4.1.1 Decision Tree Classification Results attack detection

Decision Tree classification model to detect the attack type and achieved an accuracy score for different messages. Specifically, the accuracy rate of 5 messages is 85%. In the same way, the accuracy of 10, 15, 20, 25, and 30 messages are 90percent and for others is 81%%. However, differences were observed in the time taken to classify these messages. The classification accuracy for 5 messages is 85%, and the processing time is 0.00297ms. Similarly, the accuracy score is 81% for 10, 15, 20,25, and 30 messages, and the processing time is 0.00399ms, 0.001995ms, 0.000997ms, 0.000998ms, and 0.000998ms respectively as shown in Figure 4.1.

Compare Time and Accuracy:

The Decision Tree Classification model shows that predicting the attack of 5 messages requires more time than 10 messages. In the same way, predicting the attack of 10 messages

	5msg	10msg	15msg	20msg	25msg	30msg
DT	0.850000	0.81250	0.812500	0.812500	0.812500	0.812500
Time	0.002974	0.00399	0.001995	0.000997	0.000998	0.000998

Figure 4.1: Decision Tree classification results

requires less time than predicting the attack of 5 messages. However, we also found that detecting attacks for 10, 15, 20, 25, and 30 messages remains constant. Finally, the prediction time of attacks in V2V networks varies slightly at the startup of 5 messages, but on more than 5 messages, the time remains almost constant, as shown in Figure 4.2.

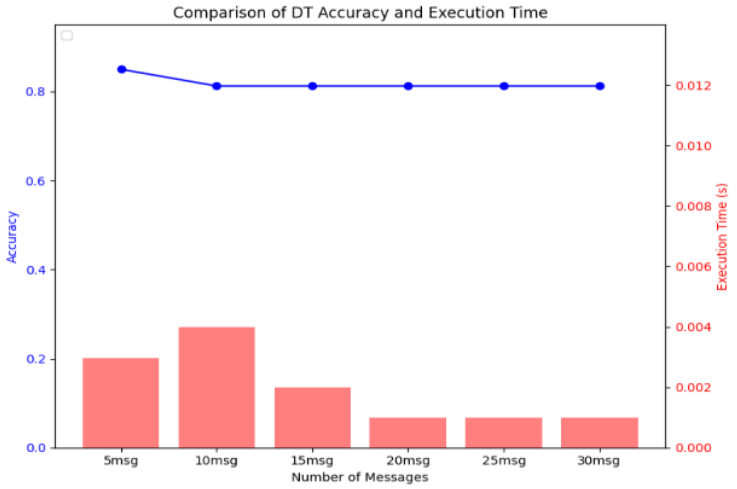

Figure 4.2: Step1 Accuracy Vs Time Graph OF Decision Tree classification Model

4.1.2 KNN Results

KNN model detects the attack with an accuracy of 85% for 5 messages. In the same way, the accuracy of 10, 15, 20, 25, and 30 messages is 87.5%, for each message. However, differences were observed in the time taken to classify these messages. The classification accuracy for 5 messages is 85%, and the processing time is 0.018ms. Similarly, the accuracy score for the above 10 messages is 87.5%, the processing time for 10 messages is 0.0169ms, and the processing time for 15 messages is 0.0112ms. In the same way, the processing time for 20 messages is 0.096ms, for 25 messages is 0.00956 seconds; and for 30 messages, the processing time is 0.0069ms, as shown in Figure 4.3.

	5msg	10msg	15msg	20msg	25msg	30msg
KNN	0.85000	0.875000	0.875000	0.875000	0.875000	0.875000
Time	0.01895	0.016963	0.011244	0.009686	0.009561	0.006982

Figure 4.3: KNN results

Compare Time and Accuracy:

The KNN Classification model shows that predicting the attack of 5 messages requires more time than 10 messages. In the same way, predicting the attack of 10 messages requires less time than predicting the attack of 5 messages. However, we also found that detecting attacks for 10, 15, 20, 25, and 30 messages decreases gradually. Finally, the prediction time of attacks in V2V networks varies slightly at the startup of 5 messages, but on more than 5 messages, the time starts to decrease, as shown in Figure 4.4

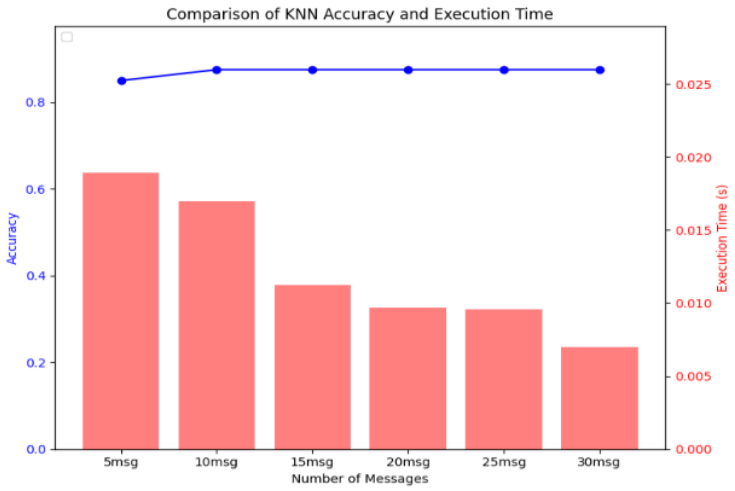

Figure 4.4: Step1 Accuracy Vs Time Graph OF KNN Model

4.1.3 Random Forest Results

We implemented the Random Forest model to detect the attack and achieved an accuracy score for different messages. The accuracy of the model on 5 messages is 95%, and on 10, 15, 20, and 25 messages is 96.87%. However, differences were observed in the time taken to classify these messages. The classification accuracy for 5 messages is 95%, and the processing time is 0.0160ms. Furthermore, the accuracy score is 96.87% for 10, 15, 20, and 25 messages, with processing times of 0.312ms, 0.0156ms, 0. 0160ms, 0. 0156ms, and 0.0156ms, respectively, as shown in Figure 4.5.

	5msg	10msg	15msg	20msg	25msg	30msg
RF	0.950000	0.968750	0.96875	0.968750	0.96875	0.968750
Time	0.016024	0.031223	0.01562	0.016018	0.01562	0.015623

Figure 4.5: Random Forest Results

Compare Time and Accuracy:

The Random forest Classification model shows that predicting the attack of 5 messages requires less time than 10 messages. In the same way, predicting the attack of 10 messages requires more time than predicting the attack of 5 messages. However, we also found that detecting attacks for 10, 15, 20, 25, and 30 messages remains constant. Finally, the prediction time of attacks in V2V networks varies slightly at the startup of 10 and 15 messages, the time remains constant for 15,25 and 30 messages, as shown in Figure 4.6.

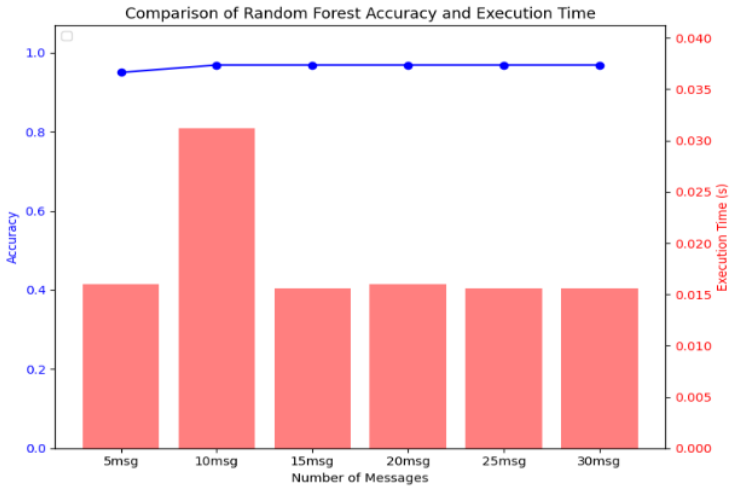

Figure 4.6: Step1 Accuracy Vs Time Graph OF Random Forest Model

4.1.4 LSTM Results

We use the LSTM model to detect the attack and achieve an accuracy score for different messages. Specifically, the accuracy rate of 5 messages is 35%. In the same way, the accuracy of 10, 15, 20, 25, and 30 messages is 34.37%. However, differences were observed in the time taken to classify these messages. The processing time is 1.615ms for 5 messages. Similarly, the processing time is 0.0937ms for 10 messages. For 15 and 20 messages, the processing time is 0.0624ms. For 25 messages, the processing time is 0.109ms. Finally, the processing time is 0.0647ms for 30 messages, as shown in Figure 4.7.

	5msg	10msg	15msg	20msg	25msg	30msg
lstm	0.350000	0.343750	0.343750	0.343750	0.343750	0.343750
Time	1.615334	0.093728	0.062485	0.062486	0.109349	0.064748

Figure 4.7: LSTM Results

Compare Time and Accuracy:

The LSTM Classifier model reveals that predicting the attack of 5 messages requires more time than 10 messages. In the same way, predicting the attack of 10 messages requires less time than predicting the attack of 5 messages. However, we also found that detecting attacks for 15 and 20 messages remains constant. Finally, the prediction time of attacks in V2V networks are high at the startup of 5 messages, but on more than 5 messages, the required time is less and remains constant as compare to stating time, as shown in Figure 4.8.

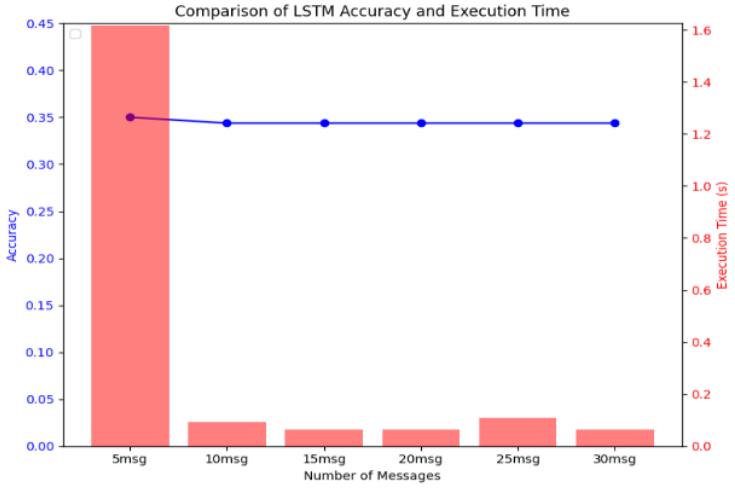

Figure 4.8: Step1 Accuracy Vs Time Graph OF LSTM Model

4.1.5 GRU Results

We applied the GRU model to detect the attack type and achieved an accuracy score for different messages. Specifically, the accuracy rate of 5 messages is 35%. The accuracy of 10, 15, 20, 25, and 30 messages is 34.37%. However, differences were observed in the time taken to classify these messages. The processing time is 2.08ms for 5 messages. Similarly, the processing time is 0.0954ms for 10 messages. For 15 messages, the processing time is 0.0803ms. For 20 messages, the processing time is 0.0812ms. For 25 messages, the processing time is 0.0628ms. Finally, the processing time is 0.0624ms for 30 messages, as shown in Figure 4.9.

	5msg	10msg	15msg	20msg	25msg	30msg
GRU	0.350000	0.343750	0.343750	0.343750	0.34375	0.343750
Time	2.082007	0.095402	0.080391	0.081278	0.06286	0.062485

Figure 4.9: GRU Results

Compare Time and Accuracy:

The GRU model reveals that predicting the attack of 5 messages requires more time than 10 messages. In the same way, predicting the attack of 10 messages requires less time than predicting the attack of 5 messages. However, we also found that processing time for detecting attacks for 10, 15, 20, 25, and 30 messages started decreasing gradually. Finally, the prediction time of attacks in V2V networks are high at the startup of 5 messages, but on more than 5 messages, the required time is less as compare to stating time, as shown in Figure 4.10.

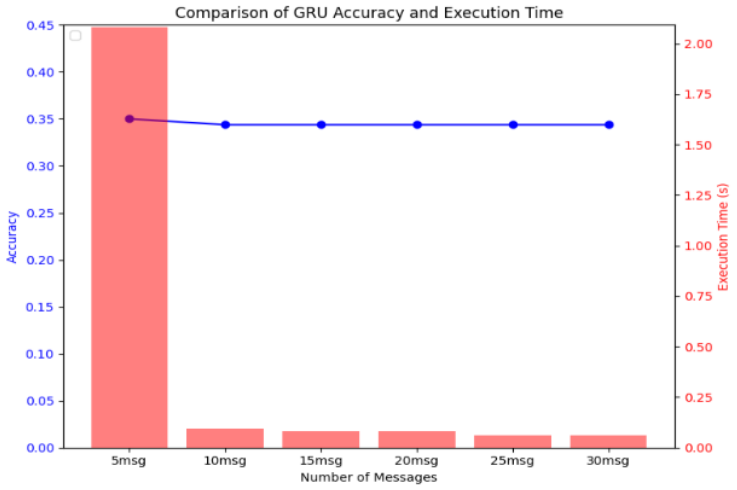

Figure 4.10: Step1 Accuracy Vs Time Graph OF GRU Model

4.2 Step 2

In this step, we target the attacks column to check the type of attack from (Constant Speed, Eventual Stop, Constant Position, Constant Speed Offset, Grid Sybil, Constant Position Offset,

Data Replay, DoS, DoS Random Sybil, Random Speed, Delayed Messages, Data Replay Sybil, DoS Disruptive, Random Speed Offset, Disruptive, DoS Disruptive Sybil, DoS Random, Random Position, Random Position Offset). The model does not accept the string value, so we do label encoding. As a result, the values come in the form of digits

Split data

Again, we split the dataset and used 80% for the training of the model and 20% for testing the model on the attacks column. The others are dropped. The dataset values to train and test on x and y are shown in Figure 4.11.

```
(128000, 3)
(32000, 3)
(128000,)
(32000,)
```

Figure 4.11: Splitting Of DataSet

Shuffle data

We convert the dataset into shuffle form to test the five different models. We consider the dataset in pair (0,1) rather than consider only 1 or 0. After shuffling the dataset, the indexing of the dataset becomes irrelevant. Then we re-index the indexing and remove the index column. And we collect 13 rows of dataset. Firstly, we check the accuracy of the models on the 5 rows, then on 10 rows, on 15 rows, on 20 rows, on 25 rows and lastly on 30 rows of the dataset, as shown in figure 4.12.

To determine whether a message is from an attacker or non-attacker, we assign the label '0' to non-attackers and '1' to attackers. We then calculate the total time it takes to predict an attack. We use a conditional statement to check if the message type is '1', which indicates that

	0	1	2	3
0	0.449677	-0.218825	1.194675	4
1	0.060510	-0.164586	0.972102	5
2	1.684549	-0.080430	-1.400782	7
3	0.263935	-0.877144	-1.026163	11
4	0.683047	-0.000994	-0.016629	4
...	...	...	...	...
159995	0.296257	0.246305	1.153304	11
159996	-0.479709	-0.269558	1.508737	1
159997	-0.570123	-0.001014	-0.016647	10
159998	-0.449987	-0.001016	-0.016649	1
159999	0.206268	-0.174057	0.329904	12

160000 rows × 4 columns

Figure 4.12: Output of Shuffled DataSet

it is from an attacker. If so, we use another model to predict the attack and record the prediction start time, the accuracy of the model, and the final prediction time. The accuracy of the model on 5 messages is 100%, and also 100% on 10, 15, 20, 25, and 30 messages.

4.2.1 Decision Tree Classification Results

We applied the Decision Tree classification model to detect the attack type and achieved an accuracy score for different messages. The accuracy rate of 25 is 96%, and the accuracy increased as we increased the number of messages until 45 msgs, at 45 messages, the accuracy

of the RF model decreased. The accuracy for 30 messages is 96.6%; for 35 messages is 97.14%; for 40 messages is 95%; for 45 messages is 93.55%; and for 50 messages, the accuracy is 94%. However, differences were observed in the time taken to classify these messages. We found that for 25 messages, the processing time is 0.0029ms, for 30 messages, the processing time is 0.00673ms; for 35 messages, the processing time is 0.001995 ms; for 40 messages, the processing time is 0.000998ms; for 45 messages, the processing time is 0.000998ms and for 50 messages the processing time is 0.00498ms as shown in figure 4.13

	25msg	30msg	35msg	40msg	45msg	50msg
DT	0.960000	0.966667	0.971429	0.950000	0.933333	0.940000
Time	0.002994	0.006737	0.001995	0.000998	0.000998	0.004986

Figure 4.13: Decision Tree Classification Results

and the graph comparison for this model is shown in Figure 4.14

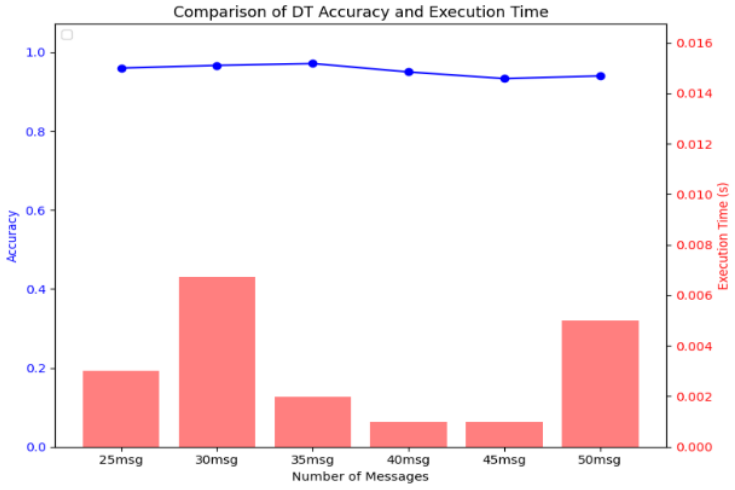

Figure 4.14: Step2 Accuracy Vs Time Graph OF Decision Tree Classification Model

4.2.2 KNN Results

We applied the KNN model to detect the attack type and achieved an accuracy score for different messages. The accuracy rate of 25 is 96%, and the accuracy increased as we increased the number of messages, but at 35 messages, the accuracy of the KNN model decreased. The accuracy for 30 messages is 96%; for 35 messages is 94.2%; for 45 messages is 91%; and for 50 messages, the accuracy is 92%. However, differences were observed in the time taken to classify these messages. We found that the processing time for 25,30,35,40,45, and 50 messages, is 0.0156ms, 0.0156ms, 0.0156ms, 0.0156ms, 0.01568ms, and 0.00ms respectively as shown in figure 4.15.

	25msg	30msg	35msg	40msg	45msg	50msg
KNN	0.960000	0.966667	0.942857	0.925000	0.911111	0.92
Time	0.015621	0.015621	0.015623	0.015619	0.015622	0.00

Figure 4.15: KNN Results

and the graph comparison for this model is shown in Figure 4.16

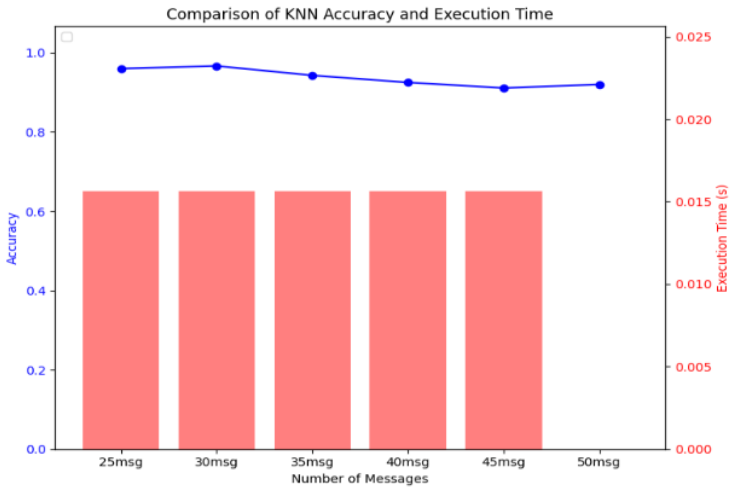

Figure 4.16: Step2 Accuracy Vs Time Graph OF KNN Model

4.2.3 Random Forest Results

We applied the Random forest model to detect the attack type and achieved an accuracy score for different messages. The accuracy rate of 25msgs is 96%, and the accuracy increased as we increased the number of messages, but above to 30 messages, the accuracy of the RF model decreased. The accuracy for 30 messages is 96%; for 35 messages is 94.2%; for 40 messages is 92.5%; for 45 messages is 91.1%; and for 50 messages, the accuracy is 92%. However, differences were observed in the time taken to classify these messages. We found that the processing time for 25,30,35,40,45, and 50 messages, is 0.02992ms, 0.0209ms, 0.0913ms, 0.0156ms, 0.0247ms, and 0.0169ms respectively as shown in figure 4.17.

and the graph comparison for this model is shown in Figure 4.18

	25msg	30msg	35msg	40msg	45msg	50msg
RF	0.96000	0.966667	0.942857	0.925000	0.911111	0.920000
Time	0.02992	0.020942	0.009135	0.015622	0.024749	0.016955

Figure 4.17: Random Forest Results

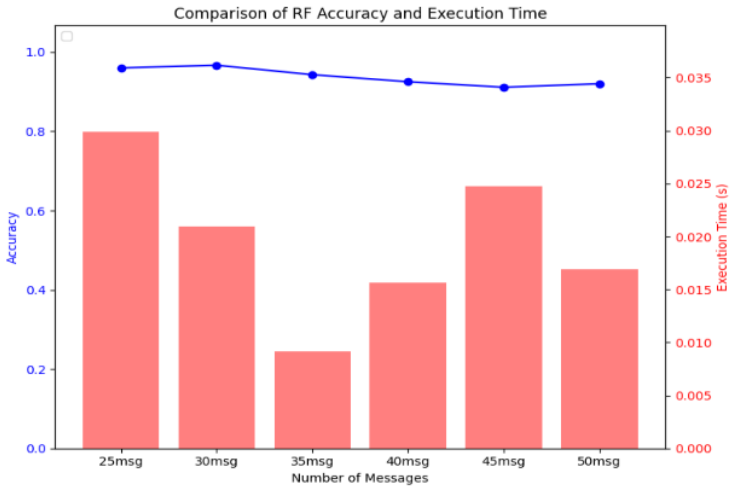

Figure 4.18: Step2 Accuracy Vs Time Graph OF Random Forest Model

4.3 Compare the Best Accuracy Results of Step-1 and Step -2

4.3.1 Decision Tree classifier

The decision tree classifier model accuracy for predicting an attack is less than the attack type detection for the different number of messages. Moreover, the overall accuracy of the decision tree classifier model for attack detection is 91.72% and for attack type detection is 97.1%, as shown in Figure 4.19.

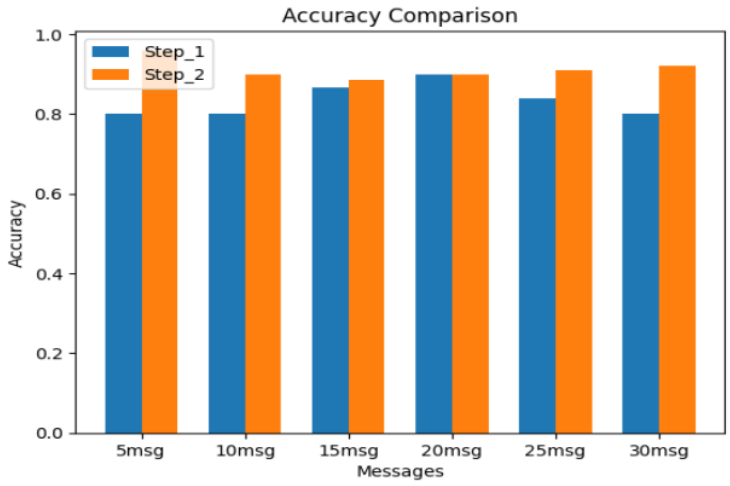

Figure 4.19: Decision Tree classifier Comparison 1 and 2 Step Prediction

4.3.2 Random Forest Classifier

The Random forest classifier accuracy for predicting an attack is better than the attack type detection for the different number of messages. Moreover, the overall accuracy of Random Forest Classifier model for attack detection is 96.6% and for attack type detection is 96.8%, as shown in Figure 4.20.

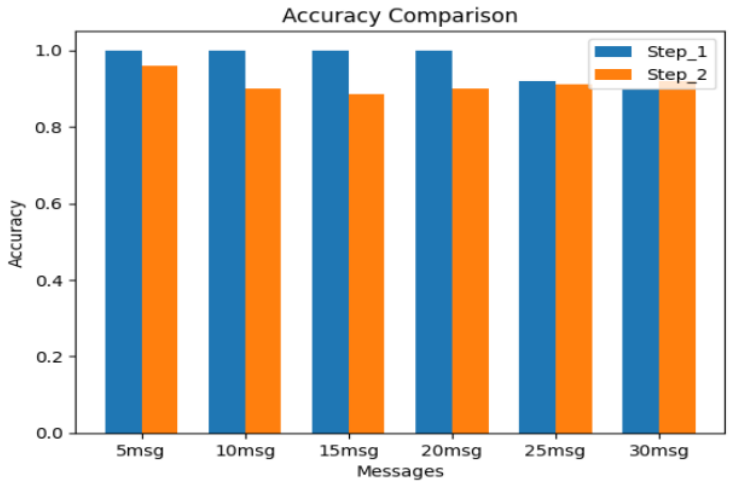

Figure 4.20: Random Forest Classifier Comparison 1 and 2 Step Prediction

4.3.3 KNN

The K-Nearest Neighbors classifier accuracy for predicting an attack is better than the attack type detection for the different number of messages. Moreover, the overall accuracy of the decision tree classifier model for attack detection is 96.6% and for attack type detection is 87.5%, as shown in Figure 4.21.

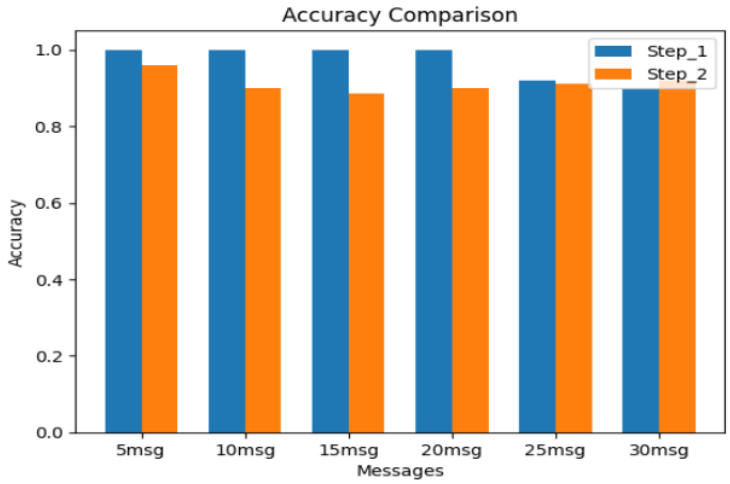

Figure 4.21: KNN Comparison 1 and 2 Step Prediction

Chapter 5

Evaluation of Step 1 and Step 2

The format of the confusion matrix is used to present the output results of the attack and attack types detection. The metrics offer valuable information regarding the model's ability to detect attackers. The confusion matrix comprises four key elements: True Positive, False Positive, True Negative, and False Negative.

Precision:

Precision is a metric that quantifies the accuracy of positive predictions by computing the ratio of true positives to the total number of instances predicted as positive. The emphasis is on the model's capacity to prevent false positives.

$$Precision = \frac{TP}{TP+FP}$$

Recall:

It is also referred to as sensitivity or true positive rate, is a metric that quantifies the ratio of accurately predicted positive instances (true positives) to the total number of positive instances in the dataset. The focus is on the model's ability to detect all positive instances accurately.

$$Recall = \frac{TP}{TP+FN}$$

F1-score:

It is a statistical measure representing the harmonic mean of precision and recall. The metric offers a suitable measure that achieves a balance between precision and recall. This feature proves to be particularly advantageous in cases where the dataset exhibits imbalance.

$$F1 - score = \frac{2*Precision*Recall}{Precision+Recall}$$

Accuracy:

Accuracy evaluates the degree of correctness of a model's predictions. The metric calculates the accuracy of predicted instances by determining the proportion of true positives and true negatives out of all instances.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

The precision metric is inversely proportional to the number of false alarms, while recall is directly proportional to detecting actual attacks. The F1 score is a balanced metric that considers both precision and recall. We use 2-Step Prediction techniques on different models to detect attackers from V2V communication.

5.1 Step 1

5.1.1 Decision Tree Classifier (DTC)

The accuracy of the decision tree classifier for 5 messages is 85%, and for 10, 15, 20, and 25 messages, it is 81.25%. Our decision tree classifier's precision for 5 messages is 87.8% and 10, 15, 20, and 25 messages is 85.4%. The recall of the decision tree classifier for 5 messages is 57%, and for 10, 15, 20, and 25 messages, it is increased by 45.45%, as shown in the figure5.1.

	5msg	10msg	15msg	20msg	25msg	30msg
Accurcay	0.850000	0.812500	0.812500	0.812500	0.812500	0.812500
Preciouse	0.878125	0.854167	0.854167	0.854167	0.854167	0.854167
Recall	0.571429	0.454545	0.454545	0.454545	0.454545	0.454545

Figure 5.1: DTC Result of 1st Prediction with 20% Density Attacker

5.1.2 KNN

The accuracy of the KNN for 5 messages is 85%, and for 10, 15, 20, and 25 messages, it is increased to 87.5%. Our KNN's precision for 5 messages is 87.8% and 10, 15, 20, and 25 messages increase by 89.5%. The recall of the KNN for 5 messages is 57.1%, and for 10, 15, 20, and 25 messages, it is increased by 63.63%. The F1_score of our model of 5 messages is 72.72%, and the recall of our model has increased by 77.7% for 10, 15, 20, and 25 messages, as shown in the figure 5.2.

	5msg	10msg	15msg	20msg	25msg	30msg
Accurcay	0.850000	0.875000	0.875000	0.875000	0.875000	0.875000
Preciouse	0.878125	0.895000	0.895000	0.895000	0.895000	0.895000
Recall	0.571429	0.636364	0.636364	0.636364	0.636364	0.636364
F1_Score	0.727273	0.777778	0.777778	0.777778	0.777778	0.777778

Figure 5.2: KNN Result of 1st Prediction with 20% Density Attacker

5.1.3 Random forest

The Random Forest model achieved best accuracy, precision, recall, and F1_score .For accuracy of 5 messages is 95% and then for 10, 15, 20, 25, and 30 messages it is 96.87 respectively.Similarly Precision of 5 msgs is 95.3% while for remaining10, 15, 20, 25, and 30 messages is 97.01%. Recall for 5 msgs is 85% and for remaining 10, 15, 20, 25, and 30 messages is 90.9% .Lastly F1-score for 5 msgs is 92.3% whereas for 10, 15, 20, 25, and 30 messages is 95.23% as shown in the figure 5.3.

The results of first step have shown that the model accuracy of the random forest and KNN is better than the Random Forest classifier. The results show that the accuracy of DTC is 85%, the accuracy of KNN classifies for step 1 is 87%, and the accuracy of Random forest is 96.8% on different number of messages.

	5msg	10msg	15msg	20msg	25msg	30msg
Accurcay	0.950000	0.968750	0.968750	0.968750	0.968750	0.968750
Preciouse	0.953571	0.970170	0.970170	0.970170	0.970170	0.970170
Recall	0.857143	0.909091	0.909091	0.909091	0.909091	0.909091
F1_Score	0.923077	0.952381	0.952381	0.952381	0.952381	0.952381

Figure 5.3: Random forest Result of 1st Prediction with 20% Density Attacker

5.2 Step 2

5.2.1 Decision Tree Classifier

The Decision tree classifier model results for precision, recall, and F1 score for 25, 30,35,40,45, and 50 messages are 96%, 96%, 97.1%, 95%, 93%, and 94%. However, upon reaching 40 messages, the precision, recall, and F1 score results decreased to 94%, as shown in the figure 5.4.

	25msg	30msg	35msg	40msg	45msg	50msg
Preciouse	0.96	0.966667	0.971429	0.95	0.933333	0.94
Recall	0.96	0.966667	0.971429	0.95	0.933333	0.94
F1_score	0.96	0.966667	0.971429	0.95	0.933333	0.94

Figure 5.4: Decision Tree Classifier Result of 2nd Prediction with 20% Density Attacker

5.2.2 KNN

The KNN classifier model results for precision, recall, and F1 score for 25, 30,35,40,45, and 50 messages are 96%, 96.6%, 94.28%, 92.5%, 91.1%, and 92%. However, upon reaching 35 messages, the precision, recall, and F1 score results decreased to 92%, as shown in the figure 5.5.

	25msg	30msg	35msg	40msg	45msg	50msg
Preciose	0.96	0.966667	0.942857	0.925	0.911111	0.92
Recall	0.96	0.966667	0.942857	0.925	0.911111	0.92
F1_Score	0.96	0.966667	0.942857	0.925	0.911111	0.92

Figure 5.5: KNN Result of 2nd Prediction with 20% Density Attacker

5.3 Random Forest

The Random Forest model results for precision, recall, and F1 score for 25, 30,35,40,45, and 50 messages are 96%, 96.6%, 94.28%, 92.5%, 91.1%, and 92%. However, upon reaching 35 messages, the precision, recall, and F1 score results decreased to 92%, as shown in the figure 5.6.

The result of step-2 shows that the accuracy of the decision tree is better than the KNN

	25msg	30msg	35msg	40msg	45msg	50msg
Preciose	0.96	0.966667	0.942857	0.925	0.911111	0.92
Recall	0.96	0.966667	0.942857	0.925	0.911111	0.92
F1_Score	0.96	0.966667	0.942857	0.925	0.911111	0.92

Figure 5.6: Random Forest Result of 2nd Prediction with 20% Density Attacker

classifier and RFC model. The results show that the accuracy of DTC reaches up to 97%, the accuracy of KNN classifies for step-2 is 96.6% max, and the accuracy of Random forest is 73.23% on different numbers of messages.

Chapter 6

Conclusion

In this research, we propose a method for detecting vehicle-to-vehicle communication attacks. The proposed model detects fourteen distinct types of attacks. We employ five models to detect and classify attacks. The model comprises the Decision Tree Classifier, KNN, LSTM, GRU, and random forest algorithms. The model's accuracy in detecting attacks is. As the complexity of these networks increases, the likelihood of malicious attacks increases, making it imperative to implement stringent security protocols. This paper presents a method for detecting attacks in vehicle-to-vehicle communication. The proposed model for attack detection achieves 99.41% accuracy on the Decision tree classifier, accuracy on the KNN model is 92.18%, on the LSTM model is 40%, Random forest accuracy is 99.38%, and a GRU model accuracy is 40%. For the detection of attack type, the Decision Tree Classifier, KNN, and Random Forest models all achieve 73.05%, 70.55%, and 73.23% accuracy respectively. The model random forest and decision tree classifier achieve maximum accuracy for attack detection and classification.

References

[1] P. Rani and R. Sharma, "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities," *Computers and Electrical Engineering*, vol. 105, p. 108543, 2023.

[2] R. Miucic, *Connected vehicles: Intelligent transportation systems*. Springer, 2018.

[3] J. Li, R. Xu, X. Liu, J. Ma, Z. Chi, J. Ma, and H. Yu, "Learning for vehicle-to-vehicle cooperative perception under lossy communication," *IEEE Transactions on Intelligent Vehicles*, 2023.

[4] F. Ahmad, A. Adnane, and V. N. Franqueira, "A systematic approach for cyber security in vehicular networks," *Journal of Computer and Communications*, vol. 4, no. 16, pp. 38–62, 2016.

[5] A. R. Khan, M. F. Jamlos, N. Osman, M. I. Ishak, F. Dzaharudin, Y. K. Yeow, and K. A. Khairi, "Dsrc technology in vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) iot system for intelligent transportation system (its): a review," *Recent Trends in Mechatronics Towards Industry 4.0: Selected Articles from iM3F 2020, Malaysia*, pp. 97–106, 2022.

[6] Z. Xie, Z. Li, J. Gui, A. Liu, N. N. Xiong, and S. Zhang, "Uwpee: Using uav and wavelet packet energy entropy to predict traffic-based attacks under limited communication, computing and caching for 6g wireless systems," *Future Generation Computer Systems*, vol. 140, pp. 238–252, 2023.

[7] A. K. Malhi, S. Batra, and H. S. Pannu, "Security of vehicular ad-hoc networks: A comprehensive survey," *Computers & Security*, vol. 89, p. 101664, 2020.

[8] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, 2005, pp. 11–21.

[9] J. Cui, W. Xu, Y. Han, J. Zhang, and H. Zhong, "Secure mutual authentication with privacy preservation in vehicular ad hoc networks," *Vehicular Communications*, vol. 21, p. 100200, 2020.

[10] P. Mundhe, S. Verma, and S. Venkatesan, "A comprehensive survey on authentication and privacy-preserving schemes in vanets," *Computer Science Review*, vol. 41, p. 100411, 2021.

[11] P. Cheng, M. Han, and G. Liu, "Desc-ids: Towards an efficient real-time automotive intrusion detection system based on deep evolving stream clustering," *Future Generation Computer Systems*, vol. 140, pp. 266–281, 2023.

[12] Z. Zhang, Y. Lai, Y. Chen, J. Wei, and Y. Wang, "Detection method to eliminate sybil attacks in vehicular ad-hoc networks," *Ad Hoc Networks*, p. 103092, 2023.

[13] G. O. Anyanwu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Rbf-svm kernel-based model for detecting ddos attacks in sdn integrated vehicular network," *Ad Hoc Networks*, vol. 140, p. 103026, 2023.

[14] M. R. Dey, M. Patra, and P. Mishra, "Efficient detection and localization of dos attacks in heterogeneous vehicular networks," *IEEE Transactions on Vehicular Technology*, 2023.

[15] G. O. Anyanwu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Falsification detection system for iov using randomized search optimization ensemble algorithm," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[16] D. Karagiannis and A. Argyriou, "Jamming attack detection in a pair of rf communicating vehicles using unsupervised machine learning," *vehicular communications*, vol. 13, pp. 56–63, 2018.

[17] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020.

[18] T. Pavithra and B. Nagabhushana, "A survey on security in vanets," in *2020 second international conference on inventive research in computing applications (ICIRCA)*. IEEE, 2020, pp. 881–889.

[19] R. Fotohi, E. Nazemi, and F. S. Aliee, "An agent-based self-protective method to secure communication between uavs in unmanned aerial vehicle networks," *Vehicular Communications*, vol. 26, p. 100267, 2020.

[20] S. Park and J.-Y. Choi, "Malware detection in self-driving vehicles using machine learning algorithms," *Journal of advanced transportation*, vol. 2020, pp. 1–9, 2020.

[21] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, "A distributed anomaly detection system for in-vehicle network using htm," *IEEE Access*, vol. 6, pp. 9091–9098, 2018.

[22] P. Sharma and H. Liu, "A machine-learning-based data-centric misbehavior detection model for internet of vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4991–4999, 2020.

[23] V. S. Barletta, D. Caivano, A. Nannavecchia, and M. Scalera, "Intrusion detection for in-vehicle communication networks: An unsupervised kohonen som approach," *Future Internet*, vol. 12, no. 7, p. 119, 2020.

[24] S. Gyawali, Y. Qian, and R. Q. Hu, "Machine learning and reputation based misbehavior detection in vehicular communication networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8871–8885, 2020.

[25] H. Park, "Edge based lightweight authentication architecture using deep learning for vehicular networks," *Journal of Internet Technology*, vol. 23, no. 1, pp. 193–200, 2022.

[26] Z. Helmi, R. Adriman, T. Y. Arif, H. Walidainy, M. Fitria *et al.*, "Sybil attack prediction on vehicle network using deep learning," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 3, pp. 499–504, 2022.

[27] N. C. Velayudhan, A. Anitha, and M. Madanan, "An optimisation driven deep residual network for sybil attack detection with reputation and trust-based misbehaviour detection in vanet," *Journal of Experimental & Theoretical Artificial Intelligence*, pp. 1–24, 2022.

[28] M. A. Elsayed and N. Zincir-Heywood, "Boostguard: Interpretable misbehavior detection in vehicular communication networks," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–9.

[29] G. O. Anyanwu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Optimization of rbf-svm kernel using grid search algorithm for ddos attack detection in sdn-based vanet," *IEEE Internet of Things Journal*, 2022.

[30] D. Basavaraj and S. Tayeb, "Towards a lightweight intrusion detection framework for in-vehicle networks," *Journal of Sensor and Actuator Networks*, vol. 11, no. 1, p. 6, 2022.

[31] S. Jeong, B. Jeon, B. Chung, and H. K. Kim, "Convolutional neural network-based intrusion detection system for avtp streams in automotive ethernet-based networks," *Vehicular Communications*, vol. 29, p. 100338, 2021.

[32] M. Han, P. Cheng, and S. Ma, "Ppm-invids: Privacy protection model for in-vehicle intrusion detection system based complex-valued neural network," *Vehicular Communications*, vol. 31, p. 100374, 2021.

[33] M. A. Al-shareeda, M. Anbar, I. H. Hasbullah, S. Manickam, N. Abdullah, and M. M. Hamdi, "Review of prevention schemes for replay attack in vehicular ad hoc networks

(vanets)," in *2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP)*. IEEE, 2020, pp. 394–398.

[34] M. A. Alazzawi, H. Lu, A. A. Yassin, and K. Chen, "Efficient conditional anonymity with message integrity and authentication in a vehicular ad-hoc network," *IEEE Access*, vol. 7, pp. 71 424–71 435, 2019.

[35] B. Majid, P. Morteza, R. Majid, and B. Mostafa, "Nera: A new and efficient rsu based authentication scheme for vanets," *Wireless networks*, vol. 26, no. 5, pp. 3083–3098, 2020.

[36] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of computer security*, vol. 15, no. 1, pp. 39–68, 2007.

[37] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[38] N. Bereczki and V. Simon, "A novel machine learning based traffic congestion recognition system," in *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE, 2023, pp. 1–6.

[39] L. Wei, J. Cui, H. Zhong, Y. Xu, and L. Liu, "Proven secure tree-based authenticated key agreement for securing v2v and v2i communications in vanets," *IEEE Transactions on Mobile Computing*, 2021.

[40] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2015.

[41] M. Azees, P. Vijayakumar, and L. J. Deboarh, "Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2467–2476, 2017.

[42] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, and S. Nandi, "Machine learning based approach to detect position falsification attack in vanets," in *International Conference on Security & Privacy*. Springer, 2019, pp. 166–178.

[43] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in vanet," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 564–571.

[44] N. C. Kushardianto, Y. El Hillali, and C. Tatkeu, "2-step prediction for detecting attacker in vehicle to vehicle communication," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–5.

[45] H. Sun, M. Chen, J. Weng, Z. Liu, and G. Geng, "Anomaly detection for in-vehicle network using cnn-lstm with attention mechanism," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 880–10 893, 2021.

[46] S. Ercan, M. Ayaida, and N. Messai, "Misbehavior detection for position falsification attacks in vanets using machine learning," *IEEE Access*, vol. 10, pp. 1893–1904, 2021.

[47] T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "Securing the internet of vehicles: A deep learning-based classification framework," *IEEE networking letters*, vol. 3, no. 2, pp. 94–97, 2021.

[48] N. C. KUSHARDIANTO, C. Tatkeu *et al.*, "Vehicular network anomaly detection based on 2-step deep learning framework."

[49] A. Gauher, A. Umrani, and Y. Javed, "Communication security in vanets," in *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*. IEEE, 2020, pp. 63–67.

[50] M. Bayat, M. Pournaghi, M. Rahimi, and M. Barmshoory, "Nera: A new and efficient rsu based authentication scheme for vanets," *Wireless networks*, vol. 26, no. 5, pp. 3083–3098, 2020.

[51] M. A. SHAHID, "Survey on malware detection approaches in vanet."

[52] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," *arXiv preprint arXiv:1804.06701*, 2018.

[53] M. M. Hamdi, L. Audah, M. S. Abood, S. A. Rashid, A. S. Mustafa, H. Mahdi, and A. S. Al-Hiti, "A review on various security attacks in vehicular ad hoc networks," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 5, pp. 2627–2635, 2021.

[54] S. Ftaimi and T. Mazri, "A comparative study of machine learning algorithms for vanet networks," in *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, 2020, pp. 1–8.

[55] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks," *IEEE Internet of things journal*, vol. 1, no. 6, pp. 570–577, 2014.

[56] S. S. Manvi and S. Tangade, "A survey on authentication schemes in vanets for secured communication," *Vehicular Communications*, vol. 9, pp. 19–30, 2017.

[57] D. T. Le, K. Q. Dang, Q. L. T. Nguyen, S. Alhelaly, and A. Muthanna, "A behavior-based malware spreading model for vehicle-to-vehicle communications in vanet networks," *Electronics*, vol. 10, no. 19, p. 2403, 2021.

[58] A. Sharma and A. Jaekel, "Machine learning approach for detecting location spoofing in vanet," in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–6.

[59] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in vanet," in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, 2007, pp. 19–28.

[60] S. Bao, W. Hathal, H. Cruickshank, Z. Sun, P. Asuquo, and A. Lei, "A lightweight authentication and privacy-preserving scheme for vanets using tesla and bloom filters," *ICT Express*, vol. 4, no. 4, pp. 221–227, 2018.

[61] Q. Wang, D. Gao, and D. Chen, "Certificate revocation schemes in vehicular networks: A survey," *IEEE Access*, vol. 8, pp. 26 223–26 234, 2020.

Thesis

ORIGINALITY REPORT

16%	10%	13%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	2%
2	Nur Cahyono Kushardianto, Yassin El Hillali, Charles Tatkeu. "2-Step Prediction for Detecting Attacker in Vehicle to Vehicle Communication", 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 2021 Publication	1%
3	Mahmood A. Al-shareeda, Mohammed Anbar, Iznan H. Hasbullah, Selvakumar Manickam, Nibras Abdullah, Mustafa Maad Hamdi. "Review of Prevention schemes for Replay Attack in Vehicular Ad hoc Networks (VANETs)", 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP), 2020 Publication	1%
4	www.mdpi.com Internet Source	1%
5	arxiv.org Internet Source	1%

6	Preeti Rani, Rohit Sharma. "Intelligent transportation system for internet of vehicles based vehicular networks for smart cities", Computers and Electrical Engineering, 2023 Publication	1 %
7	scholar.uwindsor.ca Internet Source	<1 %
8	Pravin Mundhe, Shekhar Verma, S. Venkatesan. "A comprehensive survey on authentication and privacy-preserving schemes in VANETs", Computer Science Review, 2021 Publication	<1 %
9	"International Conference on Innovative Computing and Communications", Springer Science and Business Media LLC, 2021 Publication	<1 %
10	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
11	www.hindawi.com Internet Source	<1 %
12	Dimitrios Karagiannis, Antonios Argyriou. "Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning", Vehicular Communications, 2018 Publication	<1 %

13	"Intelligent Systems and Applications", Springer Science and Business Media LLC, 2022 Publication	<1 %
14	Sohan Gyawali, Yi Qian, Rose Qingyang Hu. "Machine Learning and Reputation based Misbehavior Detection in Vehicular Communication Networks", IEEE Transactions on Vehicular Technology, 2020 Publication	<1 %
15	jit.ndhu.edu.tw Internet Source	<1 %
16	Prinkle Sharma, Hong Liu. "A Machine-Learning-Based Data-Centric Misbehavior Detection Model for Internet of Vehicles", IEEE Internet of Things Journal, 2021 Publication	<1 %
17	eprints.mdx.ac.uk Internet Source	<1 %
18	Amit Kumar Tyagi, Niladhuri Sreenath. "Intelligent Transportation Systems: Theory and Practice", Springer Science and Business Media LLC, 2023 Publication	<1 %
19	Submitted to National College of Ireland Student Paper	<1 %

20	Yuanyuan Tan. "Feature Recognition and Style Transfer of Painting Image Using Lightweight Deep Learning", Computational Intelligence and Neuroscience, 2022 Publication	<1 %
21	Zhaoyi Zhang, Yingxu Lai, Ye Chen, Jingwen Wei, Yuhang Wang. "Detection method to eliminate Sybil attacks in Vehicular Ad-hoc Networks", Ad Hoc Networks, 2023 Publication	<1 %
22	vitalflux.com Internet Source	<1 %
23	irep.ntu.ac.uk Internet Source	<1 %
24	core.ac.uk Internet Source	<1 %
25	dspace.cc.tut.fi Internet Source	<1 %
26	ijeecs.iaescore.com Internet Source	<1 %
27	Submitted to The University of the West of Scotland Student Paper	<1 %
28	downloads.hindawi.com Internet Source	<1 %

29	www.cs.ru.nl Internet Source	<1 %
30	Submitted to University of Northumbria at Newcastle Student Paper	<1 %
31	projekter.aau.dk Internet Source	<1 %
32	www.diva-portal.org Internet Source	<1 %
33	"Advances in Cyber Security", Springer Science and Business Media LLC, 2021 Publication	<1 %
34	Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Jae Min Lee, Dong-Seong Kim. "RBF-SVM kernel-based model for detecting DDoS attacks in SDN integrated vehicular network", Ad Hoc Networks, 2022 Publication	<1 %
35	Zhixia Zhang, Yang Cao, Zhihua Cui, Wensheng Zhang, Jinjun Chen. "A Many-objective Optimization based Intelligent Intrusion Detection Algorithm for Enhancing Security of Vehicular Networks in 6G", IEEE Transactions on Vehicular Technology, 2021 Publication	<1 %
36	e-space.mmu.ac.uk Internet Source	<1 %

37	Aekta Sharma, Arunita Jaekel. "Machine Learning Based Misbehaviour Detection in VANET Using Consecutive BSM Approach", IEEE Open Journal of Vehicular Technology, 2021 Publication	<1 %
38	Mahmood A. Al-shareeda, Mohammed Anbar, Iznan H. Hasbullah, Selvakumar Manickam. "Survey of Authentication and Privacy Schemes in Vehicular ad hoc Networks", IEEE Sensors Journal, 2020 Publication	<1 %
39	Submitted to University of Hertfordshire Student Paper	<1 %
40	Submitted to Kingston University Student Paper	<1 %
41	Submitted to Liverpool John Moores University Student Paper	<1 %
42	www.econstor.eu Internet Source	<1 %
43	Chundong Wang, Zhentang Zhao, Liangyi Gong, Likun Zhu, Zheli Liu, Xiaochun Cheng. "A Distributed Anomaly Detection System for In-Vehicle Network Using HTM", IEEE Access, 2018 Publication	<1 %

44	J. Naskath, G. Sivakamasundari, A. Alif Siddiqua Begum. "A Study on Different Deep Learning Algorithms Used in Deep Neural Nets: MLP SOM and DBN", Wireless Personal Communications, 2022 Publication	<1 %
45	etd.aau.edu.et Internet Source	<1 %
46	Submitted to Khalifa University of Science Technology and Research Student Paper	<1 %
47	Sunilkumar S. Manvi, Shrikant Tangade. "A survey on authentication schemes in VANETs for secured communication", Vehicular Communications, 2017 Publication	<1 %
48	Submitted to University of Nottingham Student Paper	<1 %
49	pdfs.semanticscholar.org Internet Source	<1 %
50	www.cs.cornell.edu Internet Source	<1 %
51	1library.net Internet Source	<1 %
52	ijctet.org Internet Source	<1 %

53	medialibrary.uantwerpen.be Internet Source	<1 %
54	www.truprojects.in Internet Source	<1 %
55	Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Jae-Min Lee, Dong-Seong Kim. "Falsification Detection System for IoV Using Randomized Search Optimization Ensemble Algorithm", IEEE Transactions on Intelligent Transportation Systems, 2023 Publication	<1 %
56	Hasanien Ali Talib, Raya Basil Alothman, Mazin S. Mohammed. "Malicious attacks modelling: a prevention approach for ad hoc network security", Indonesian Journal of Electrical Engineering and Computer Science, 2023 Publication	<1 %
57	Sahil Garg, Kuljeet Kaur, Georges Kaddoum, Syed Hassan Ahmed, Dushantha Nalin K. Jayakody. "SDN-Based Secure and Privacy-Preserving Scheme for Vehicular Networks: A 5G Perspective", IEEE Transactions on Vehicular Technology, 2019 Publication	<1 %
58	Seonghoon Jeong, Boosun Jeon, Boheung Chung, Huy Kang Kim. "Convolutional neural	<1 %

network-based intrusion detection system for AVTP streams in automotive Ethernet-based networks", Vehicular Communications, 2021
Publication

59	Shafika Showkat Moni, D. Manivannan. "CREASE: Certificateless and REused-pseudonym based Authentication Scheme for Enabling security and privacy in VANETs", Internet of Things, 2022 Publication	<1 %
60	Tejasvi Alladi, Varun Kohli, Vinay Chamola, F. Richard Yu. "Securing the Internet of Vehicles: A Deep Learning based Classification Framework", IEEE Networking Letters, 2021 Publication	<1 %
61	Youness Arjoune, Saleh Faruque. "Real-time Machine Learning Based on Hoeffding Decision Trees for Jamming Detection in 5G New Radio", 2020 IEEE International Conference on Big Data (Big Data), 2020 Publication	<1 %
62	ijns.jalaxy.com.tw Internet Source	<1 %
63	uwspace.uwaterloo.ca Internet Source	<1 %
64	web.archive.org Internet Source	<1 %

65	www.ttc.ca Internet Source	<1%
66	Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Jae-Min Lee, Dong-Seong Kim. "Optimization of RBF-SVM Kernel using Grid Search Algorithm for DDoS Attack Detection in SDN-based VANET", IEEE Internet of Things Journal, 2022 Publication	<1%
67	Jiabin Li, Zhi Xue, Changlian Li, Ming Liu. "RTED-SD: A Real-Time Edge Detection Scheme for Sybil DDoS in the Internet of Vehicles", IEEE Access, 2021 Publication	<1%
68	Nitha C Velayudhan, A. Anitha, Mukesh Madanan. "An Optimisation driven Deep Residual Network for Sybil attack detection with reputation and trust-based misbehaviour detection in VANET", Journal of Experimental & Theoretical Artificial Intelligence, 2022 Publication	<1%
69	Pengzhou Cheng, Mu Han, Gongshen Liu. "DESC-IDS: Towards an efficient real-time automotive intrusion detection system based on deep evolving stream clustering", Future Generation Computer Systems, 2023 Publication	<1%

70	jurnal.iaii.or.id Internet Source	<1 %
71	manuscriptlink-society-file.s3.ap-northeast-1.amazonaws.com Internet Source	<1 %
72	mjee.iaumajlesi.ac.ir Internet Source	<1 %
73	uis.brage.unit.no Internet Source	<1 %
74	wrap.warwick.ac.uk Internet Source	<1 %
75	www.diva-portal.se Internet Source	<1 %
76	Avleen Kaur Malhi, Shalini Batra, Husanbir Singh Pannu. "Security of vehicular ad-hoc networks: A comprehensive survey", Computers & Security, 2020 Publication	<1 %
77	Jalawi Sulaiman Alshudukhi, Badiea Abdulkarem Mohammed, Zeyad Ghaleb Al-Mekhlafi. "An Efficient Conditional Privacy-Preserving Authentication Scheme for the Prevention of Side-Channel Attacks in Vehicular Ad hoc Networks", IEEE Access, 2020 Publication	<1 %

78 Marwa A. Elsayed, Nur Zincir-Heywood. "BoostGuard: Interpretable Misbehavior Detection in Vehicular Communication Networks", NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, 2022 <1%
Publication

79 Muhammad Arif, Guojun Wang, Md Zakirul Alam Bhuiyan, Tian Wang, Jianer Chen. "A survey on security attacks in VANETs: Communication, applications and challenges", Vehicular Communications, 2019 <1%
Publication

80 Philip Asuquo, Haitham Cruickshank, Jeremy Morley, Chibueze P. Anyigor Ogah, Ao Lei, Waleed Hathal, Shihan Bao, Zhili Sun. "Security and Privacy in Location-Based Services for Vehicular and Mobile Communications: An Overview, Challenges, and Countermeasures", IEEE Internet of Things Journal, 2018 <1%
Publication

81 R Muthumeenakshi., A Anu Monisha., K Murugan.. "DAKAA: Double Authentication and Key Agreement Algorithm for securing Vehicular Ad hoc Networks", 2018 Tenth International Conference on Advanced Computing (ICoAC), 2018 <1%

	Publication	
82	Sagheer Ahmed Jan, Noor Ul Amin, Mohamed Othman, Mazhar Ali, Arif Iqbal Umar, Abdul Basir. "A Survey on Privacy-Preserving Authentication Schemes in VANETs: Attacks, Challenges and Open Issues", IEEE Access, 2021 Publication	<1 %
83	Yakub Kayode Saheed. "Machine learning-based blockchain technology for protection and privacy against intrusion attacks in intelligent transportation systems", Institution of Engineering and Technology (IET), 2022 Publication	<1 %
84	ebin.pub Internet Source	<1 %
85	herschel.esac.esa.int Internet Source	<1 %
86	scholar.archive.org Internet Source	<1 %
87	uu.diva-portal.org Internet Source	<1 %
88	uwe-repository.worktribe.com Internet Source	<1 %
89	www.infocommunications.hu Internet Source	<1 %

90	www.ojp.gov Internet Source	<1%
91	www.politesi.polimi.it Internet Source	<1%
92	Heng Sun, Miaomiao Chen, Jian Weng, Zhiquan Liu, Guanggang Geng. "Anomaly Detection for In-Vehicle Network Using CNN-LSTM With Attention Mechanism", IEEE Transactions on Vehicular Technology, 2021 Publication	<1%
93	Vita Santa Barletta, Danilo Caivano, Antonella Nannavecchia, Michele Scalera. "Intrusion Detection for In-Vehicle Communication Networks: An Unsupervised Kohonen SOM Approach", Future Internet, 2020 Publication	<1%
94	Aekta Sharma, Arunita Jaekel. "Machine Learning Approach for Detecting Location Spoofing in VANET", 2021 International Conference on Computer Communications and Networks (ICCCN), 2021 Publication	<1%
95	Yong Xie, Fang Xu, Dong Li, Yu Nie. "Efficient Message Authentication Scheme with Conditional Privacy-Preserving and Signature Aggregation for Vehicular Cloud Network",	<1%

Wireless Communications and Mobile Computing, 2018

Publication

Exclude quotes	Off	Exclude matches	< 6 words
Exclude bibliography	On		