Cybersecurity within the Internet of Vehicles (IoV): Using Machine Learning to Protect Against External Attacks

by Monica G. Payne

B.S. in Management, May 1993, Morgan State University
M.B.A., May 2003, University of Baltimore

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

May 19, 2024

Praxis directed by

John Fossaceca
Professorial Lecturer of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Monica Grant Payne has passed the Final Examination for the degree of Doctor of Engineering as of April 3, 2024. This is the final and approved form of the Praxis.

Cybersecurity within the Internet of Vehicles (IoV): Using Machine Learning to Protect Against External Attacks

Monica G. Payne

Praxis Research Committee:

John Fossaceca, Professorial Lecturer of Engineering and Applied Science, Praxis Director

Walid Hassan, Professorial Lecturer of Engineering and Applied Science, Committee Chair

Medhi Hashemipour, Professorial Lecturer of Engineering and Applied Science, Committee member

© Copyright 2024 by Monica G. Payne
All rights reserved

Dedication

I have extreme gratitude to my husband, Eric Payne for your unwavering support and patience throughout my doctoral journey. Thank you have for taking up the slack and ensuring that our family life remained steady despite the demands on our time and resources.

Thank you to my daughter Zoë, who always inspires me and to my parents, Patricia and the late Nathaniel Grant, and brother Nathaniel Grant, who have always encouraged me and have been a source of strength.

Acknowledgments

I wish to acknowledge my advisor Dr. John Fossaceca and co-advisor Dr. Shahryar Sarkani who provided their expertise, guidance and support for this effort. A special thanks to Dr. John Fossaceca for fostering my interest in cyber security within the Internet of Vehicles.

I would also like to acknowledge my fellow cohort members who willingly gave of their time and knowledge to help me through this journey. I give sincere and special thanks to my cohort members Dr. Olusegun "SJ" Jegede and Dr. Madjid Tehrani for their help, encouragement and study support.

Abstract of Praxis

The Internet of Vehicles (IoV), a cloud-connected network comprising vehicles, smartphones, people, and roadside infrastructure, serves as the catalyst for Intelligent Transportation Systems (ITS). Advancements in automotive architecture, increased integration of connected vehicle (CV) technology, and the adoption of wireless communication protocols are the primary drivers of the progression in external vehicle communication.

Efficient improvements in growing transportation services, including enhanced traffic predictions, road safety measures, and driver convenience, are made possible through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication. However, the complexity of the hardware and software components within the connected vehicles protocol gives rise to numerous cybersecurity vulnerabilities. One such concern is the potential for Traffic Sybil cyber-attacks, where an attacker creates multiple identities for the same vehicle to interfere with normal operations, posing serious safety and security problems in V2V communications. This praxis introduces a Machine Learning (ML)-based approach to classify and detect these malicious attacks. The study addresses the solution's approaches, limitations, challenges and lessons-learned associated with using ML-based methods.

Table of Contents

Chapter 1 Introduction 1

1.1 Background 1

1.2 Research Motivation 3

1.3 Problem Statement 5

1.4 Thesis Statement 5

1.5 Research Objectives 6

1.6 Research Questions and Hypotheses 6

1.7 Scope of Research 7

1.8 Research Limitations 8

1.9 Organization of Praxis 8

Chapter 2 - Literature Review 10

2.1 Introduction 10

2.2 Relevant Literature Search Strategy 10

2.3 Communication Within the Internet of Vehicles 10

2.3.1 VANET Communication 11

2.4 Cybersecurity Challenges in the VANET 14

2.5 Misbehavior in VANETs 15

2.6 Misbehavior Detection in VANET 17

2.6.1 Sybil Attack Detection in VANETs 18

2.6.2 ML based Sybil Attack Detection 19

2.7 Summary and Conclusion 32

Chapter 3 - Methodology 34

3.1 Introduction 34
3.2 Data Collection 35
3.3 VeReMi Extension Datasets 37
3.3 Data Preprocessing 39
3.4 Feature Selection 42
3.4.1 Feature Selection Methods 43
3.5 Machine Learning Models 49
3.6 Model Training and Validation 51
3.7 Model Evaluation 52
3.8 Summary 54
Chapter 4 Results 55
4.1 Introduction 55
4.2 Dataset Balancing 56
4.3 Feature Selection Results 56
4.3.1 Forward Feature Selection 56
4.3.2 Decision Tree based Feature Selection 57
4.3.3 Correlation Coefficient Feature Selection 58
4.3.4 Selected Features 62
4.4 VANET Misbehaving Classification Model Results 64
4.4.1 Grid Sybil Dataset Results 65
4.4.2 DoS Random Sybil Dataset Results 66
4.4.3 Random Position Offset Dataset 67
4.5 Cross Validation with Random Forest 69

4.6 Model Results Summary 69

4.7 Research Questions and Hypotheses Results 71

Chapter 5 Discussion and Conclusions 73

5.1 Discussion and Conclusions 73

5.3 Contributions to Body of Knowledge 75

5.5 Recommendations for Future Research 76

References 79

Appendix A 87

List of Figures

Figure 1: Four categories of Sybil traffic attacks..... 23

Figure 2: Research Methodology..... 34

Figure 3: Data Set Preprocessing..... 39

Figure 4: Lasso - Grid Sybil..... 47

Figure 5: Lasso – DoS Random Sybil..... 47

Figure 6: Confusion Matrix 52

Figure 7: Correlation Heat Map - Sybil Grid..... 59

Figure 8: Correlation Heat Map – DoS Random Sybil..... 60

Figure 9: Grid Sybil Feature Correlations 61

Figure 10: Random Position Offset Feature Correlations..... 62

Figure 11: Confusion Matrix Legend 64

Figure 12: Confusion Matrices: Grid Sybil Dataset 65

Figure 13: Confusion Matrices: DoS Random Sybil Dataset 67

Figure 14: Confusion Matrices: Random Position Offset Dataset..... 68

List of Tables

Table 1: Data Description for Model Evaluation 25

Table 2: Dataset Attributes and Features 26

Table 3: Deep Neural Network Model Results 28

Table 4: Detection performance -Basic Set 29

Table 5: Detection performance-Ext Set 29

Table 6: Position Falsification Results 30

Table 7: Overall Comparison of Study Results 31

Table 8: MbRE - Recall Scores 32

Table 9: Identified Gaps and Praxis Contributions 33

Table 10: VeReMi Extension Data Set Description 35

Table 11: Dataset File and Storage Details 37

Table 12: Datasets Feature Descriptions 38

Table 13: Aggregated Dataset Fields 40

Table 14: Feature Selection Techniques-Advantages and Disadvantages 48

Table 15: SVM Hyperparameters 50

Table 16: Random Forest Hyperparameters 51

Table 17: Performance metrics and formulas 53

Table 18: Research Questions 55

Table 19: Dataset SMOTE Results: Before and After 56

Table 20: Wrapper - Forward Feature Selection Results 57

Table 21: Decision Tree based Feature Selection Scores 57

Table 22: Selected Features for Praxis Datasets 63

Table 23: Classifier Results with Grid Sybil Dataset 66

Table 24: Classifier Results with DoS Random Sybil Dataset 67

Table 25: Classifier Results with Random Position Offset Dataset 69

Table 26: Random Forest Cross Validation Results 69

Table 27: Random Forest Comparison - Grid Sybil 70

Table 28: Random Forest Comparison - DoS Random Sybil 70

Table 29: Random Forest Comparison - Random Position Offset 70

Table 30: Summary of Research Questions, Hypotheses and Results A 71

Table 31: Summary of Research Questions, Hypotheses and Results B 71

Table 32: Summary of Research Questions, Hypotheses and Results C 72

Table A-1: ML Output - 70/30 Train/Test Split 87

Table A- 2: List of Selected Dataset Features 88

List of Acronyms

ECU	Electronic Control Unit, Engine Control Unit
IoT	Internet of Things
IoV	Internet of Vehicles
KNN	K Nearest Neighbor
ML	Machine Learning
OBD	Onboard Diagnostics
OEM	Original Equipment Manufacturer
RSU	Road-side Unit
SVM	Support Vector Machine
TCB	Telematics Control Board
TCU	Telematics Control Unit
USB	Universal Serial Bus
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad Hoc Network
VEINS	Vehicles in Network Simulation
VeReMi	Vehicle Reference Misbehavior Detection

Chapter 1 Introduction

1.1 Background

The Internet of Vehicles (IoV), facilitated by wireless communication protocols, is an integration of connected vehicle (CV) technology and interconnected devices smart devices within the transportation infrastructure. Groups of wirelessly connected vehicles operate as Vehicular Ad hoc Networks (VANETs), enabled by On-Board Units (OBUs) that function as receivers, and Roadside Units (RSUs) which are a fixed part of the infrastructure. The RSUs provide connectivity and information support to vehicles within its range.

The information vehicles send in message communications provides the data to determine traffic and road safety, weather, news and entertainment updates within the environment. The connections and communications between CVs and the transportation infrastructure are the core pillars of the Intelligent Transportation System (ITS)

CV technology consists of advanced components within the vehicle's on-board electronic parts. Combined with vehicular mobile internet for internet connectivity, vehicles are able to interact with and process outside information received from smart objects, roadside units, and other vehicles (Wang et al., 2019). The U.S. Department of Transportation defines CV technology as "the equipment and systems that use vehicle to vehicle (V2V) communications to address safety, system efficiency, or mobility on the roadways" (U.S. DOT, 2020). The foundation for CV technology is built on car sensors, mapping and navigation software, communication software, and a robust wireless communication network. Wireless fidelity (Wi-Fi), worldwide interoperability for microwave access (WiMAX), long-term evolution (LTE), and dedicated short-range

communication (DSRC) are examples of the current wireless communication options within the IoV. DSRC is a wireless communication channel designed for short-range use, has a bandwidth of 75MHZ, and is effective at short-range message exchanges. Another component, the Global Positioning Systems (GPS) receiver, provides real-time vehicle location information and enables navigation and route planning.

Enhanced road safety and traffic management is a main benefit of V2V communication. Shared status information between vehicles enables proactive driver responses to identified hazards. Basic safety messages (BSMs) are a main type of communication and contain the following types of information:

- Position data which specifies the latitude, longitude, and elevation of the vehicle.
- Motion data, the vehicles heading, speed, and acceleration.
- Specific vehicle component statuses for lights, brakes, and wipers.
- Safety information which consists of the vehicle's path history and event details.

When vehicles receive the BSMs, through the internal automotive systems, warning signals or indicators are displayed as warnings to the driver so the appropriate safety action can be taken (Liu et al., 2016).

Cybersecurity vulnerabilities within the IoV stem from the increase of external communication interfaces as well as the wireless network platform. Attacks on vehicle networks can result in involuntary personal privacy disclosure, economic loss and endanger the life and safety of people. (Guan, Tian et al.). Another vulnerability is with the broadcast mode of message exchange. Controller area network (CAN) bus messages

are broadcast every ten seconds and all vehicle and roadside infrastructure nodes within the broadcast proximity can accept the messages. These messages don't contain original address information, and when received by the electronic control units (ECUs), the data can't be confirmed as authentic or original. This limitation of the CAN protocol increases the ability for adversaries to employ message spoofing, replay, or unauthorized message injection (Nilsson et al., 2009) via access to one of the many ECU ports. Adversaries can also launch sybil traffic attacks by exploiting the abundance of pseudonyms, which allows them to present as multiple vehicles. To create fake congestion or distort traffic management data, the attackers send false message data to nearby vehicles. Sybil attackers possess a valid cryptographic certificate, but intentionally uses multiple valid pseudonyms simultaneously to execute the attack. Genuine behaving vehicles can only use one pseudonym certificate to sign BSM messages within a certain period of time.

1.2 Research Motivation

As the importance and steady growth of the connected vehicle network increases, so does the frequency and severity of cybersecurity attacks. Road safety is a top priority as fatal vehicle crashes can wreak havoc on the human toll and suffering. The U.S. Department of Transportation has spent more than a decade researching to find solutions which utilize technology to reduce the number of traffic fatalities and stated that "the emergency response to these motor vehicle crashes, and related support systems cost the American taxpayer more than $140 billion per year" (U.S. DOT, 2020). The ITS provides the opportunity to prevent a considerable number of accidents and strengthen the safety of the transportation network within the U.S.

The integration of vehicle sensors and communication capabilities provides the potential for vehicles to have an all-around awareness of threats and hazards along the roadways.

Threats targeting vehicular communications can occur in three major layers of the vehicle infrastructure. The sensing layer contains the lidar, radar, and GPS abilities, the communication layer contains the CAN and OBD for direct communication and Infotainment telematics, while the top layer contains the control functions. These include vehicle speed control, and the control systems for braking and steering. (El-Rewini et al., 2020). The communication layer is where the V2V communication takes place. Vehicles send and receive message transmissions which detail their movements and those of surrounding vehicles. Vehicles then use this information to provide alerts or warnings to the drivers about upcoming activities on the roadway that aid in the notification of perceived threats which can be further out from the driver's line of sight.

The messaging communication platform utilized for the exchanging of these messages is a major cybersecurity vulnerability. This is due to the technology's reliance on wireless technology, Wireless Access in the Vehicular Environment (WAVE), which is built on the 802.11 standard protocols. Several factors expose WAVE to cybersecurity vulnerabilities, such as the openness of the wireless channel, lack of security in vehicle physical security and no encryption mechanisms. Researchers have indicated that certain attacks on VANETs, such as spoofing, DoS and Sybil can endanger the life of passengers. Sybil traffic attacks, one of the greatest cybersecurity threats, can severely disrupt the safety operations of the VANET.

This praxis aims to make a valuable contribution in the area of Sybil traffic attack countermeasure by evaluating various machine learning models to determine which model would be the most effective in detecting and classifying these attacks.

1.3 Problem Statement

Connected vehicles are vulnerable to many cybersecurity attacks such as sybil and position falsification where adversaries create fake identities and broadcast false safety information, which can compromise road safety.

(Kumar et al., 2022) discussed the different ways a sybil attack can be implemented. When an attacker creates new identities or uses spoofed identities, this is known as an identity-based attack. These identities can be fabricated or stolen by spoofing nearby nodes. A communication-based attack occurs when the attacker uses the fake sybil identities to communication with genuine vehicles. Communication based attacks can be direct or indirect. Direct attacks occur when each fake vehicle communicates directly with other legitimate vehicles. With indirect attacks, the fake vehicles do not communicate directly with any legitimate vehicle, instead, they forward data packets through the attacker. Participation based attacks are when the attacker creates multiple Sybil vehicles and each one can communicate with other genuine vehicles either simultaneously or non-simultaneously. Simultaneous participation occurs when the attack vehicle utilizes all its sybil identities. With a non-simultaneous attack, the sybil identities are used in a leave or join manner, and the identities are swapped by multiple sybil attackers.

1.4 Thesis Statement

A machine learning model is required to classify malicious communications within the Internet of Vehicles architecture.

Using a machine learning classification model that can detect misbehavior from Sybil and position falsification attackers can significantly reduce the potential for harm from these attacks, and also increase the likelihood for proactive responses and optimum recovery when applied as a real-time process.

1.5 Research Objectives

The primary objective of this research is to demonstrate a novel approach of using a machine learning classification model for detecting vehicular messaging misbehavior. Details are as follows:

1. To research prior works on identified cyber security vulnerabilities within the IoV and with connected cars to determine gaps and areas for future research.
2. To determine which attributes of V2V messages sent during Sybil and position falsification attacks are the most significant for classifying as malicious.
3. To determine if supervised learning classifiers can be used to accurately classify Sybil and position falsification attack communications.
4. To evaluate the performance of the supervised learning models using the Random Forest, K Nearest Neighbor (KNN), Naïve Bayes, Support Vector Machine (SVM), and Decision Tree algorithms.

1.6 Research Questions and Hypotheses

This study is directed by the following research questions and proposed hypotheses:

RQ1: *Which set of features are the most significant indicators for determining whether a IoV communication is malicious or safe?*

RQ2: *Can a classification machine model be developed to accurately classify malicious and safe IoV communications?*

RQ3: *Which machine-learning algorithm can classify IoV communications with the highest accuracy and low false positive rate and negative rates?*

H1: *The position, speed, acceleration, and header values in the message data are significant indicators for malicious IoV communications.*

H2: *A machine model will accurately classify an IoV communication as malicious or safe.*

H3: *A classification model using the Random Forest algorithm, will achieve an accuracy of at least 95% for classifying IoV communication as malicious or non-malicious.*

H4: *A classification model using the Random Forest algorithm, will accurately classify a IoV communication as malicious with false positive and negative rates of less than 1%.*

1.7 Scope of Research

This research investigates the performance of classification machine learning models for detecting malicious V2V communications sent during Sybil traffic and position falsification attacks. The praxis uses the Veremi Extension datasets to train and test the various machine learning models. The records represent messages sent to and

received from vehicles during a simulated rush hour session. Messages, and includes both misbehaving and safe message exchanges.

The main goal of the praxis is to identify which machine learning model has the highest rate of accuracy, and the lowest false negative and positive rates for classifying misbehaving messages sent during a Sybil or position falsification attack.

1.8 Research Limitations

This praxis focuses on examining the Random Forest, KNN, Naïve Bayes, SVM, and Decision Tree machine learning algorithms, and is limited to the detection and classification of the Sybil traffic and position falsification attacks through the use of the VeReMi Extension simulated datasets. The following limitations exist:

- The Veremi Extension Grid Sybil (traffic), DoS Random Sybil and Random Position Offset datasets are imbalanced between the number of misbehaving and genuine records. SMOTE synthetic resampling was used to offset this limitation.
- The research does not include an assessment of the protocols each vehicle manufacturer uses to transmit and receive messages.

1.9 Organization of Praxis

This praxis is organized into five chapters.

Chapter 1 contains the introduction, which provides and overview of the research topic, objective and motivation for the praxis. The research questions, hypotheses, scope and research limitations are also detailed.

Chapter 2 provides a review of the research literature and provides a critical analysis of prior related work and existing knowledge gaps. Included is published literature that is specific to the subject matter and methodology used in this praxis.

Chapter 3 details the methodology approach to solve the problem presented in this praxis. In-depth descriptions of the data cleanse and preparation steps, the classification model characteristics and settings, as well as the experimental approach for the research study.

Chapter 4 summarizes the results generated from the machine learning models used in the study. The feature selection process, classification metrics of accuracy, precision, recall, and the confusion matrix are discussed in detail.

Chapter 5 holds the conclusion of this praxis and summarizes the key findings. Discussed are the implications and suggested avenues for future research. Provided is a comprehensive culmination to the study.

Chapter 2 - Literature Review

2.1 Introduction

The practical approach of using classification models to improve the detection of Sybil cyber-attacks within the Internet of Vehicles (IoV) extends upon previously studied foundational works which form this body of literature review. This study builds upon the knowledge gained from analyzing peer reviewed journals, articles, book chapters, and other published content, demonstrates that more strategic work is needed to mitigate misbehavior within the IoV. This review of relevant literature helped inform the selection of appropriate research methodologies and identify gaps in the existing knowledge. Demonstrated in the literature are supported and evaluated methods that use machine learning models and classification algorithms and have guided the research approach in this praxis.

2.2 Relevant Literature Search Strategy

To ascertain the existing knowledge and historical research for this study, a search strategy was developed to identify relevant sources of literature. The adopted strategy consists of search databases, terms and keywords. Database searches were performed on the Gelman Library search engine, IEEE Xplore and Google Scholar. Some of the keywords and search terms that were used to query the sources were: Internet of Vehicles, VANET, Cyber Vulnerabilities, Cyber Security, Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Sybil Attacks in the VANET. Results were then refined based on literature type and year published.

2.3 Communication Within the Internet of Vehicles

The primary goal of IoV is to create an all-inclusive and interconnected transportation environment where vehicles, persons, smart devices and road-side infrastructure can exchange data and services, and improve transportation, safety, and convenience outcomes. The IoV incorporates an extensive range of communication technologies and services to facilitate and meet the goal and utilizes multiple communication standards and protocols. Leveraged technologies include Dedicated Short-Range Communications (DSRC), Cellular Vehicle-to-Everything (C-V2X), Wi-Fi, and 4G/5G cellular networks. Available services within the IoV include entertainment, navigation, remote diagnostics, and over-the-air software updates. There are several modes of vehicle communication within the IoV network, which include V2X, V2V and V2I. V2X encompasses all the communication exchanges and interactions between vehicles and the other smart and connected entities within the environment. (Olovsson et al., 2022). V2V communication is the exchange of information between vehicles on the road and V2I communication is the exchange of information between vehicles and the infrastructure, which includes traffic lights, road signs, and other elements within the transportation system. Both communication types contribute to the construct of V2X.

2.3.1 VANET Communication

The Vehicular Ad-Hoc Network (VANET) operates as a subset communication technology within the IoV is. VANET originated an extension of the mobile ad-hoc network (MANET) in the late 1990's. (Olovsson et al., 2022). During the early 2000s, significant contributions by researchers and advancements in technology the VANET evolved into the broader landscape of connected and autonomous vehicles. Communication standards were formalized by the Institute of Electrical and Electronics

Engineers (IEEE) and the Society of Automotive Engineers (SAE), which helped ensure interoperability and compatibility among different systems.

The VANET primarily relies on the DSRC or C-V2X communication technology, which is consistent and reliable, and is better suited for the low-latency ad-hoc communication exchanges between vehicles and roadside units. In this open platform, vehicles which act as nodes, are free to enter and exit the network, creating a tiny web system of simultaneous communication, which continues to grow (Krishna, K., et al., 2022). The four basic categories of VANET communication are: In-Vehicle, Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Vehicle to Broadband Cloud (V2B).

In-Vehicle Communication is specific to the communication that takes place within a vehicle. The Application Unit (AU) serves as the device responsible for managing the essential safety applications of the vehicle and has primary responsibility for the alerts and warnings sent to the driver to maintain safety.

V2V communication refers to the exchange of information between vehicles within the VANET. Vehicles communicate with each other wirelessly, sharing information about their speed, position, direction and road conditions. This communication is vital for enhancing road safety. Vehicles use this data to detect potential collision risks and issue warnings to drivers, which significantly reduces the chances of accidents. Vehicles sharing information about hazardous road conditions, sudden braking, or unexpected obstacles helps improve road safety because drivers can be proactive and react more effectively when warned in advance. (Mohamed et al., 2021). Benefits realized from V2V communications include better road safety and traffic management due to the real-time position, speed and behavior information provided from

nearby vehicles. Information about road conditions, weather alerts, and traffic congestion is routinely exchanged. Vehicles use this information to warn or alert the drivers of potential hazards or obstacles so that preventive actions can be taken to avoid accidents. Another safety benefit of V2V occurs when the vehicle's cooperative adaptive cruise control (CACC) functionality is enabled. In this mode, vehicles communicate with other vehicles to coordinate their speeds and positioning which allows them to follow each other at safe and consistent speeds and distance. Furthermore, vehicles can provide information about approaching emergency vehicles, allowing other vehicles to quickly and safely move out of the way (Ismath et al., 2019).

Communication exchanges with traffic signals and roadside sensors, categorized as V2I communication, are a critical part of the Intelligent Transportation System (ITS) (Ismath et al., 2019). The real time information provided to vehicles from Road Side Units (RSUs), is designed to improve traffic flow, increase the overall management of traffic and promote safer driving behavior. There are many benefits realized by the infrastructure nodes. These include maximized timing of signal changes by the traffic signals, alerts from road and pedestrian intersection sensors to warn drivers to adjust their speed, and improved traffic efficiency for traffic patterns, and road conditions.

Vehicle to Broadband Cloud (V2B) communication refers to the technology that enables vehicles to connect and communicate with cloud-based services over broadband networks, and is a subcategory of the broader V2X, or Vehicle-to-Everything communication. In V2B communication, vehicles use broadband Internet connections to access a wide range of services and data stored in the cloud. V2B technology also allows

for the uploading of data from the vehicle to the cloud, such as telematics, diagnostic information, and driving behavior data.

This praxis primarily focuses on message exchanges in the V2V and V2I communication modes.

2.4 Cybersecurity Challenges in the VANET

Today's vehicles contain systems architecture developed on over 100 million lines of code. Advanced connectivity and computerization provide direct advantages to the driver and passenger experience, however, more opportunities exist for hackers to take command of vehicles cause havoc within the transportation network (Zeinab et al., 2019). One of the four major vehicular cybersecurity challenges (Zeinab et al., 2019), is categorized as "critical risk for drivers or passengers' lives". Illegitimate messages or misinformed sensor data released into the communication streams can cause receiving vehicles to experience malfunctions or abnormalities that place the lives of drivers, passengers, and pedestrians at risk. Malicious actors can easily manipulate, disrupt, or delete transmitted data in VANET due to the open nature of wireless communication. When a hacker modifies data and sends misleading signals, it has the potential to cause severe traffic issues, leading to accidents or redirecting drivers along the unauthorized route preferred by the hacker (Krishna et al., 2022). Dedicated short range communication (DSRC) technology is used to broadcast basic safety messages (BSMs) periodically by all vehicles. Each message contains the vehicle's pseudonym, its temporary identity used to prevent tracking and to protect privacy, along with the vehicle sensor data. A public key infrastructure (PKI) is used to manage the cryptographic certificates. Each vehicle is equipped with one long-term certificate and multiple short-

term certificates known as pseudonym certificates. These pseudonym certificates are utilized for signing the basic safety messages (BSMs). The vehicles can change pseudonyms frequently but can only use one at a time during a certain time period. PKI is an effective countermeasure against external attackers, but not in preventing Sybil attacks.

To ensure confidentiality in the VANET, public-key Elliptic Curve Cryptography (ECC) is used with conditional privacy-preserving authentication. Authentication and verification can be performed without revealing sensitive information, conditionally, based on certain predefined conditions being fulfilled. This helps maintain privacy while allowing necessary authentication or verification processes to take place securely (Sultana et al., 2022). The cryptography mechanisms do not detect when internal entities misbehave by performing data forging, data replay and Denial-of-Service (DoS) attacks. Data integrity and network resources still face potential cyber threats. Many researchers believe that misbehavior detection systems are the prominent solution for detecting the data from transmitted V2X messages.

2.5 Misbehavior in VANETs

The usual objectives of cyber-attacks in the VANET are to obstruct and steal information. In (Krishna et al., 2022) the authors classified attackers of VANETs into distinct categories. The first category is determined by their location in terms of the network. Attackers can be inside or outside of the network based on authentication status and whether the attacks are launched outside the network. Inside attackers are usually well versed with and have direct access to the network's configuration. Attacks originating from inside the network are also harder to detect because they involve entities

that are already authenticated and supposed to be there. Data tampering and insider information disclosure of two types of inside based attacks. The second categorization is determining if the attack is active or passive. Unauthorized observation and data gathering is a form of passive attack, while direct manipulation or disruption of data signifies an active attack. Both can be equally as harmful. Attackers can also exhibit a combination of both malicious and rational behaviors, which is the third defined category. Factors based on the attackers' goals or motivations, can be used to determine malicious intent, but is often challenging. Malicious behavior has characteristics of harmful intentions and or deception. Examples include spreading malware and hacking.

GPS spoofing, also known as a tunnel attack, is where an attacker provides false location data to another vehicle by using GPS simulators. Attackers generate counterfeit signals that mimic authentic GPS signals and then broadcast these through the network. The result is that receiving devices mistakenly believe they are in a location different from the actual position.

Replay Attacks occur when an adversary captures legitimate communication between vehicles, then replays or retransmits those messages at a later time. Replayed messages often contain false location or speed information. Consequences of this type of attack include the misleading of other vehicles or the overall traffic management system, ultimately leading to traffic accidents or other disruptions.

One of the most devastating attacks in VANETs is the Sybil assault. A node or vehicle may appear to have numerous identities in a Sybil attack. Other vehicles in the network are not able to tell if the data originated from a single vehicle or a fleet of vehicles. In sybil attacks, attackers can influence the behavior of other automobiles by

compelling them to veer off their intended course. Hallmarks of these attacks include the appearance of one vehicle being in numerous locations and giving false information about the location. These difficult to detect attacks are particularly risky on systems that use geographic navigation (Krishna et al., 2022).

2.6 Misbehavior Detection in VANET

Researching security in VANETs has emerged as a notable area of study, with researchers proposing novel security procedures to classify, detect and prevent intruder attacks (Krishna et al., 2022). Faisal et al., (2022) explored commonly recommended Sybil defense mechanisms and determined that a comprehensive defense is lacking. They concluded that mitigation of a VANET sybil attack would require a combination of mechanisms rather than any singular approach and designed a high-level algorithmic strategy for identifying potential sybil vehicles. This approach is advantageous in the initial phase of communication, after which, the trust certification process ensures that all messages sent and received are totally safe from Sybil attacks.

The misbehavior detection system (MDS) has been widely studied for use in detecting possible cyber-attacks. There are two types of MDS detection mechanisms, node-centric and data-centric based. Node-centric based detection mechanisms are essential for maintaining overall security and trustworthiness of the network. Vehicle node abnormalities are based on credibility from historical behavior. Data-centric based detection mechanisms analyze data patterns to identify irregular or suspicious behaviors within the VANET. Most detection schemes combine both node and data centric mechanisms because they are generally inseparable. (Xu et al. 2022) prepared a novel categorization of detection mechanisms for the different approaches to misbehavior

detection. The authors categorized machine learning as a mechanism technique based on collaborative consistency, which is defined as a pairwise comparison of messages while examining packet sequences from vehicles. Resolving conflicting data involves employing secure aggregation techniques to merge information from multiple vehicles, and provides a mechanism for identifying conflict nodes.

2.6.1 Sybil Attack Detection in VANETs

Noman et al., (2023) discussed important sybil attack detection techniques in their research paper. Many detection techniques include the use of radar sensor data. Radar captures adjacent vehicles' physical details, and when systems archive this data, a comprehensive log of vehicle movement history can be used to classify vehicles in categories of trusted, untrusted, or suspicious. Utilizing timestamps is also commonly used as a detection technique. As part of the RSU infrastructure, a digital timestamp is generated for each vehicle that passes through an RSUs range. Transmitted vehicle messages include these timestamps and system can use this information to track how many messages are being sent per vehicle within a specific RSU range. The active position detection technique uses the relative angle, position, and velocity of each vehicle in the VANET (Baza et al., 2022) to determine the precise location of vehicles in real-time and involves the transmission and reception of signals to ascertain the position. It is a critical component for vehicle communication that ensures collision avoidance and employs traffic management. The use of a trusted certificate authority is a very effective technique that can be used to mitigate Sybil Attacks. As a form of trust establishment between two nodes, a Certificate Authority (CA) is used as the mechanism to verify the one-to-one communication between the vehicle and the

network (Reddy et al., 2017), (Soni et al., 2018). A CA also has the authority to revoke the certificate of any node found to be suspicious in the network. The Random Key Distribution with Validation Key Test technique provides a framework for securing v2v and V2I communications through the use of key validation. A cryptographic key is distributed amongst the vehicles in a random manner and then a validation key test is employed to ensure the authenticity and the integrity of the communication.[22]. This technique is effective in preventing Sybil attacks as malicious nodes will not pass the validation test. The keys associated with a random identity do not overlap enough with the compromised key.

2.6.2 ML based Sybil Attack Detection

Prior research has revealed an increase in ML based misbehavior detection solutions. Several papers documented detection schemes using ML to classify and identify sybil attacks and position misbehavior in messages sent by nodes within the VANET (Hamdi et al., 2022). Azam et al., (2022) detailed the vital role that machine learning is playing in combatting Sybil attacks within the VANET.

Commonly used supervised classification algorithms include, k-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Machine (SVM), Extreme Gradient Boosting (XGB), Naïve Bayes and Decision Tree, Stochastic Gradient Boosting (SGB) and Multi-Layer Perception (MLP).

KNN, useful for classification and regression tasks, is as an instance-based learning algorithm. When used for classification tasks, KNN classifies a data point based on how its neighbors are classified. The K value determines how many neighbors influence the classification outcome and can be set through many different approaches,

including trial and error, square root rule and cross validation. KNN is effective for categorizing items into discrete groups based on similar characteristics. KNN can be computationally expensive for large datasets and may perform poorly with high-dimensional data due to its dimensionality aspect (Sarker, I. H. 2021).

RF is an ensemble learning method that generates numerous decision trees throughout the training phase. The input space is then divided into multiple subsets, which are fed into individual decision trees running in parallel. The final decision is then made by combining the prediction outcome of the trees. Some benefits for using RF include high accuracy in making predictions, good for handling non-linear data, feature ranking and handling of missing values. Limitations of being computationally intensive and low performance with very high-dimensional, sparse data are (like text data) exist for RF. Another con, is that model outcomes are not as interpretable as simpler models, like a single decision tree (Sarker, I. H. 2021).

An extension of the perceptron algorithm, SVM maximizes the margin between the decision boundary (hyperplane) and instances closest to the hyperplane (support vectors). A support vector is defined as the data points closest to the hyperplane, where removal of these points would impact the position of the dividing hyperplane. The margin is the gap between two lines on the closet class points and the concept of maximizing the margin is to select the hyperplane with the maximum distance between data points of each class. SVMs can be computationally intensive with large datasets and datasets with a high number of features. Another drawback is that SVMs are less effective on very large datasets or datasets with a lot of overlapping classes (Sarker, I. H. 2021).

XGB, an ensemble method based on a distributed gradient-boosted decision tree, is also beneficial for classification and regression tasks. It works by creating an ensemble of decision trees in a sequential manner, where each tree corrects the errors made by the previous ones. XGB is known for its high efficiency and rapid processing capabilities, making it well-suited for managing large datasets and complex sets of features, as well as its regularization methods for preventing overfitting. Drawbacks are that XGB can be memory intensive and computationally complex and might overfit smaller datasets even though it has built in regularization (Sarker, I. H. 2021).

Naïve Bayes is a probabilistic machine learning algorithm commonly used for classification and text classification tasks. Based on Bayes' theorem, it makes a "naive" assumption that the features used for classification are conditionally independent. The presence of one feature is independent of the presence of another feature and its influence on the class label. Naïve Bayes is computationally efficient for high-dimensional datasets, but can perform less accurately compared to decision trees, random forests, or deep neural networks (Sarker, I. H. 2021).

Decision Tree algorithms have a tree-like structure used for both classification an prediction tasks, and makes sequences of decisions to arrive at a decision or to predict an output. The tree consists of nodes that represent features, edges that represent the decision rules, and for classification tasks, leaves that represent the class labels. The starting point is the root node where all the data points from the training dataset are considered. The algorithm selects the feature that best splits the data based on some criterion. Gini impurity (Zang et al., 2021) is the criterion mechanism used for binary classification tasks and determines the purity of a dataset with respect to the class labels.

It measures the chance of misclassifying a randomly selected element in the dataset if it were labeled randomly based on the class distribution within that dataset. When a node satisfies the stopping criteria, it becomes a leaf node and is assigned a class label in classification. This class label assigned to a leaf node is often the majority class of the data points in that node. For classification tasks, the output is the class label associated with the reached leaf node (Sarker, I. H. 2021).

SGB is an advanced form of the traditional Gradient Boosting method and is known for its efficiency and effectiveness as a classification method. It is an ensemble learning technique that builds multiple models and combines them into a robust model. Its gradient boosting framework adds new models to correct the errors made by existing models. Each tree is built one at a time and in a way to reduce residual errors made by previous trees. To control the level of its sensitivity, the SGB hyperparameters can be tuned by indicating the number of trees, tree depth, learning rate, and the size of the subsample, however finding the most efficient balance can be challenging. Its learning rate parameter which is used to scale the contribution of each tree contributes to its robust nature but also slows it down. Recognized for its effectiveness in achieving high accuracy in predictive modeling tasks, its complexity and need for careful tuning of its parameters make it one of the more challenging algorithms to use (Sarker, I. H. 2021).

MLP is a deep, feedforward artificial neural network. It has multiple layers of nodes arranged and connected to each other. It typically has three types of layers, input, hidden and output. The input layer is the first layer and receives the data, also known as the input signal. The next one or more layers considered the hidden layers and is where ethe computational processing occurs. A system of weighted connections applies a

nonlinear activation function to the inputs they receive. The final layer is the output layer, which produces the output of the system. MLPs are foundational to the field of neural networks and deep learning, and are beneficial for modeling complex relationships in data. Limitations include the propensity to overfit training data if there are too many layers or neurons, inability to process data that contains images or sequential text where spatial or temporal relationships are important (Sarker, I. H. 2021).

The authors of the VeReMi Extension dataset (Kamel et al. 2020), proposed an ML based linking Misbehavior Detection solution to detect Sybil attacks (Kamel et al. 2019). Since misbehavior detection on the local component, which is done on every vehicle, has been well studied, the authors proposed a solution with a global detection component in the cloud. Their proposed solution was designed to identify and detect the categorized group of Sybil attacks shown in **Figure 1**. The design incorporated a system model with misbehavior detection and local detection check processes, and a Misbehavior Authority (MA) as an investigative process.

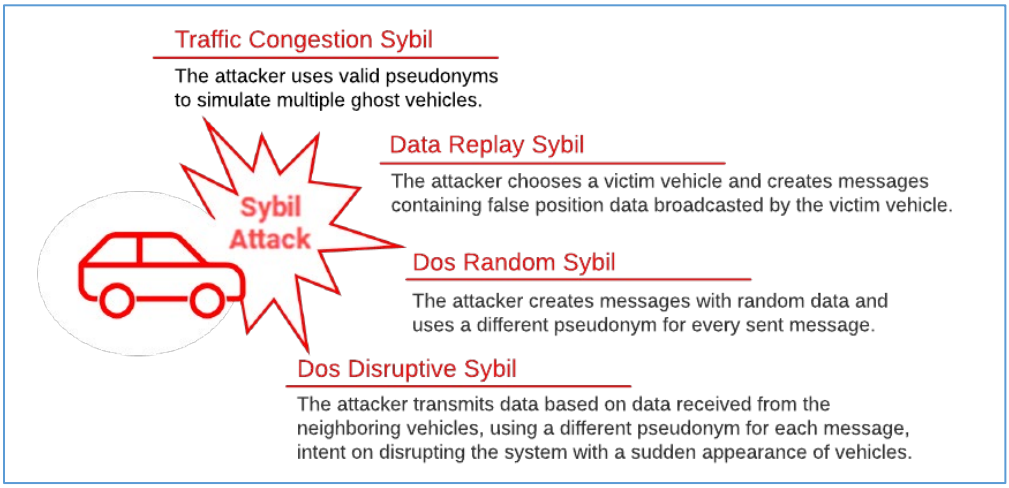


Figure 1: Four categories of Sybil traffic attacks(Kamel et al., 2019)

The misbehavior detection process started with OBUs and RSUs local detection of misbehaving entities, and will all routed messages being processed by the continuously

running plausibility and consistency checks on the Intelligent Transportation System Station (ITS-S) (Kamel et al. 2019). These checks for malicious behavior included plausibility checks to assess the stated values in the messages, determining if the range was within the maximum radio reception, and the likelihood of the stated position based on terrain and obstacles (Kamel et al. 2019). Speed was check to determine if it was within the predefined threshold (Kamel et al. 2019). Consistency checks compared the stated speed, position and beacon between two consecutive beacons from the same ITS-S (Kamel et al. 2019). There were also checks to confirm that two positions didn't intersect or suddenly appear. If malicious activity was detected by the checks, the ITS-S would initiate the second detection process by sending a Misbehavior Report (MBR) to alert the MA. The MA acted as a central authority in the and is in charge of receiving and processing the MBRs. During this misbehavior investigation phase, the MA would determine what type of reported misbehavior was present. To globally detect a Sybil attack, the linking of several pseudonyms to the original attacker is required. For U.S. standards there is an architecture that integrates a Linkage Authority (LA) entity, and provides the results based on a straightforward association between them. European standards do not have this architecture available, so the authors designed an LA-like function based on ML technique to mimic the process.

The MBRs are stored in a spatial database and filtered for aggregation. The authors used 30 extracted key features, comprised of the results from the plausibility and consistency checks, the position, speed, acceleration, heading and time of the last beacon data points, the delta values between the last 2 beacons for the position, speed,

acceleration, heading and time, the delta between the last 2 received reports. were also included, and the number of times the same MBR data was received.

To evaluate their proposed model, the authors used a dataset developed from the output of vehicle trace logs, which were generated by the Luxembourg SUMO Traffic (LuST) scenario. LuST is an open source synthetic traffic scenario that is validated with real data from the Vehicular Lab of the University of Luxembourg (Kamel et al. 2019). **Table 1** describes the datasets used for training and testing the model. The Recurrent Neural Network (RNN) prediction algorithm was fed the previously calculated features and LSTM was for testing purposes.

Table 1: Data Description for Model Evaluation(Kamel et al., (2019)

Data	# of Vehicles	# of V2X Messages	Transmitted MBRs	Attacker Rate
Training Set	82,146	301,082,858	5,209,072	5%
Testing Set	24,663	17,051,860	294,160	5%

The conclusion found that ML algorithms and hyper-parameters should be investigated further. If the pseudonym link was detected, the output of the MA algorithm indicated a Sybil Attack along with the type of misbehavior. Using a detection accuracy calculation of the ratio of the true classified reported vehicles over all reported vehicles, the result was 94.97% accuracy across all Sybil attack types when just using the LA, and 88.83% when using the Linkage AI Model.

Slama et al., (2022) contributed a ML based Misbehavior Detection System for VANETs using the VeReMi Extension dataset to test the model's efficacy. For the binary classification, they grouped all the misbehavior datasets into a single category name "misconduct". For the multi-class classification, they had three classes, "attacks", "faults" and non-misbehaving. Malfunctions or "faults" were attributed to defective

OBUs or vehicle sensors and attack involves the intentional transmitting of fake information. Each causes the incorrect position, velocity, acceleration, and heading values to be communicated. To categorize a message as a fault, they used the constant, random, constant offset, and random offset values for the position and velocity values, and delayed messaging. The identified datasets from the VeReMi extension collection of datasets: Denial of Service (DoS), DoS random, DoS disruptive, DoS random sybil, DoS disruptive sybil, data replay, disruptive, data replay sybil and traffic congestion sybil and have the following statistics as shown in **Table 2**.

Table 2: Dataset Attributes and Features (Slama et al., 2022)

Dataset Attributes		Selected Features
Sample Data Size	3,194,808	• posx, • posy, • spdx, • spdy, • spdx_noise, • spdy_noise, • aclx, • acly, • hedx, • hedy, • hedx_noise, • hedy_noise
# of Classes	20	
# Genuine Vehicles	1900539	
# Misbehavior Vehicles	1294269	
# of Features	30	

The authors noted that the imbalanced nature of the dataset created challenges with the multi-class classification, which resulted in inconsistent attributed lost efficiency for one class and improved efficiency for another. The authors' solution was their novel "Guided Learning Approach for Multi-Class Classification", which reduced the number of classes by combining comparable inappropriate behaviors into one. They concluded that based on the precision, recall, and F1-score assessment measures, the Random Forest classifier outperformed the other classifiers for detecting traffic sybil attacks.

Kushardianto et al. (2021) proposed a two-step prediction technique with two different classifiers, using two different training datasets. The first dataset, with only two labels, legitimate and attacker, trained the model to produce a classifier the authors named “0-1 Classifier”. The second dataset contained 14 attack labels and did not include any legitimate vehicles and was used to train the model to produce a classifier named “14 Attack Classifier”. The authors two-step process model first predicted if a vehicle was legitimate or an attacker. If the vehicle was detected as legitimate, processing stopped, however, if the vehicle was detected as an attacker, the vehicle data was passed on to the “14 Attack Classifier” prediction phase to determine the type of attacker.

In their approach to safeguarding IoT-enabled VANETs, (Alladi et al., 2021) employed the VeReMi Extension dataset within their proposed deep neural network anomaly detection scheme. Their network model was comprised of a vehicular network containing vehicle OBUs and RSUs, with Cellular V2X or Dedicated Short-Range Communication (DSRC) wireless protocols. When the RSUs received the data broadcast by the vehicles, that data was passed the author’s detection scheme, which was deployed directly on the RSU. Their dataset included all 19 attack and fault types for the misbehavior data and the non-misbehaving data was classified as normal. Their selected features from the dataset were the sending timestamp, X and Y position coordinates, and the associated differentials of velocity, acceleration, and heading. The normal type data was used for training the models, and the sequenced anomaly data was used for testing the models. The recall results are described in **Table 3.**

Table 3: Deep Neural Network Model Results(Alladi et al., 2021)

Overall Results for all Behavior Types				
Measure	**Model1**		**Model2**	
	Normal	**Anomaly**	**Normal**	**Anomaly**
Precision	0.970	0.996	0.930	0.995
Recall	0.997	0.956	0.996	0.894
F1-Score	0.983	0.976	0.962	0.942
Accuracy	0.980		0.954	

The authors concluded that their proposed CNN-LSTM model with an overall accuracy of 0.98 outperformed the stacked-LSTM model, which had an overall accuracy of 0.954. They also stated that the computed precision, recall, and F1-scores for each class solidified the correctness of their CNN-LSTM model.

Kim et al., (2023) investigated the challenges associated with location spoofing attacks by leveraging the VeReMi dataset, which offered a blend of authentic data alongside location spoofing attack data. Their research revealed several studies that used this dataset to assess the detection performance concerning spoofing attacks with conventional machine learning (ML) methods. The results from the researched studies indicated high accurate detection rates of greater than 95% using the Random Forest classifier against five predefined attack types in the original VeReMi dataset. The authors were not able to replicate the research results using the same methodology and achieved somewhat lower performance than what was reported. They noted a continued need for investing location spoofing attacks using the identified attacks in the data, as well as attacks involving minor modifications of coordinates as these can greatly decrease the detection performance. Using the position and velocity three-dimensional coordinate values, they designed a data-driven approach to accurately and reliably detect location spoofing attacks. Their methodology also introduced a novel set of features for enhanced

examination of mobility constraints and inconsistencies and employed a “2-sequence” approach by referencing the location information from two consecutive messages. By comparing their scheme to previous research, they demonstrated that the utilization of these features consistently enhances detection performance. The authors depicted the result comparisons in **Table 4** and **Table 5**, noting that the Ext set had lower respective false positive and negative rates compared to the Basic set. Only the MLP results showed a slightly higher FNR, which the authors attributed to the MLP’s tendency to classify instances as normal, based on the zero FPR. The authors conclude that their evaluation results confirm the effectiveness of their detection scheme. Their results showed significant improvement with detection performance across a diverse set of attack variants.

Table 4: Detection performance -Basic Set(Kim et al., 2023)

Basic Set								
Classifier	**TPR**	**TNR**	**FPR**	**FNR**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
KNN	0.883	0.965	0.035	0.117	0.924	0.961	0.883	0.920
SVM	0.267	**1 .000**	0.000	0.733	0.633	**1.000**	0.267	0.421
RF	0.906	0.974	0.026	0.094	0.940	0.972	0.906	0.937
SGB	0.912	0.978	**0.022**	0.088	**0.945**	0.976	0.912	**0.943**
MLP	**0.996**	0.832	0.168	**0.004**	0.914	0.855	**0.996**	0.920

Table 5: Detection performance-Ext Set(Kim et al., 2023)

Ext Set								
Classifier	**TPR**	**TNR**	**FPR**	**FNR**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
KNN	0.948	0.994	0.006	0.052	0.971	0.993	0.948	0.970
SVM	0.894	**1.000**	**0.000**	0.106	0.947	**1.000**	0.894	0.944
RF	0.978	0.998	0.002	0.022	0.988	0.998	0.978	0.987
SGB	0.980	0.997	0.003	0.020	0.988	0.996	0.980	0.988
MLP	**0.983**	**1.000**	**0.000**	**0.017**	**0.991**	**1.000**	**0.983**	**0.991**

Sharma A et al. (2020) proposed an ML based solution to detect misbehavior, with the detection framework residing on the RSUs, and enhanced processing available in

each vehicle OBU. New BSMs received from a vehicle, were checked against the shared database to retrieve the last received BSM from the same sender. The information was then combined and applied to the ML models for classification as legitimate or attacker. When a vehicle was classified as an attacker, the RSU informed the nearby vehicles and infrastructures about the misbehaving vehicle. The vehicles would have built in processing for the OBU to include this information in a stored log.

Ercan et al., (2021) introduced a novel Misbehavior Detection Model using features from the VeReMi extension dataset, and leveraging the estimated angle of arrival and estimated distance via path loss as fresh features for the detection process. The included features provided the relationship between sender and receiver during the message exchanges, and were used to classify a vehicle as an attacker or not by using the Random Forest and k-Nearest Neighbor Machine learning algorithms. Using two different feature-sets, the authors achieved the results in shown in **Table 6**.

Table 6: Position Falsification Results (Ercan et al., 2021)

Measurement	KNN		Random Forest	
	Proposal 1	**Proposal 2**	**Proposal 1**	**Proposal 2**
Precision	0.8552	0.8392	**0.9042**	0.8941
Recall	**0.8559**	0.8300	0.7183	0.7567
Accuracy	**0.9501**	0.9415	0.9382	0.9408
F1 Score	**0.8556**	0.8346	0.8006	0.8197

In (Bassiony et al., 2022), the authors focused their review of studies utilizing the original Vehicular Reference Misbehavior Dataset (VeReMi) dataset and its performance against position falsification attacks using various misbehavior detection techniques. They believed position falsification attacks to be one of the most severe impacts on VANET services due to the detriment that could occur when position information is manipulated in vehicle message exchanges. Identified in the authors study, So et al

(2018), proposed a misbehavior detection and categorization system framework based on plausibility checks. Their approach relied on crafting six feature vectors derived from location and movement plausibility checks, and quantitative metrics such as velocity differentials, total displacement, and time differentials in both the x and y directions. The dataset was then evaluated using the KNN and SVM learning models. Additional studies reviewed by the authors applied different plausibility checks, and the use of additional machine learning models, including Naïve Bayes, Random forest, ensemble-boosting, and ensemble-voting. The authors concluded with their opinion that none of the proposed solutions they reviewed satisfactorily detected position falsification. An overall comparison of the correct classification rate (CCR), precision and recall results are described in **Table 7**.

Table 7: Overall Comparison of Study Results(Bassiony et al, 2022).

Study	Detection Technique	Misbehavior					
		Random Position			Random Position Offset		
		CCR	Precision	Recall	CCR	Precision	Recall
So et al., (2018)	ML-KNN	0.9463	0.9506	0.8363	0.9471	0.9627	0.8253
	ML-SVM	0.9116	0.8149	0.8860	0.9177	0.8035	0.8755
So et al., (2019)	First-BSM	0.9832	0.9537	1.0000	0.9307	0.9126	0.8068
	Majority-BSM	0.9999	0.9997	1.0000	0.8143	0.9986	0.2762
	Weighted-BSM	0.9886	0.9681	1.0000	0.9368	0.9402	0.8049

Kohli et al., (2022) proposed an intelligent, statistical Intrusion Detection System (IDS) called Multi-branch Reconstruction Error (MbRE). The IDS was designed to learn only from normal behavior, and when detecting different behavior, classify the deviations into eight generalized buckets. Using only the ground truth data file from the VeReMi extension dataset, the authors achieved recalls, shown in **Table 8,** were between 97.5% and 100% for detecting the normal and misbehavior attack types.

Table 8: MbRE - Recall Scores(Kohli et al., 2022)

Behavior Type	Encoding			Recall %		
	Frequency	Identity	Motion	Frequency	Identity	Motion
Normal	0	0	0	100	100	98
Random Position	0	0	1	100	100	100
Random Speed	0	0	1	100	100	100
Disruptive	0	0	1	100	100	98.5
DoS	1	0	0	100	100	98
DoS Disruptive	1	0	1	100	100	99.5
Traffic Congestion	0	1	1	100	98.5	97.5
DoS Random Sybil	1	1	1	100	100	100
DoS Disruptive Sybil	1	1	1	100	99	99

Chen et al., (2022) examined how the density of traffic and attackers impacted the effectiveness of current schemes involving plausibility checks and observing neighboring nodes. The authors proposed a "malicious attack detection scheme based on traffic-flow information fusion" (Chen et al., 2022), without the need for neighboring observer nodes. Utilizing the basic safety message, they were able to establish the traffic flow characteristics to represent the current traffic state. Subsequently, these features were integrated with road detector data and fusion features, then classified using machine learning algorithms. The authors stated results showed Sybil attack detection rates with greater than 90% accuracy rates. Future plans include the incorporation of greater amounts of credible sensor device data which can be used to further improve the detection performance of Sybil attacks.

2.7 Summary and Conclusion

This section encompassed a comprehensive literature examination and review of the published strategies and techniques used to detect and classify Sybil cyber-attacks within the IoV. Approaches for effective classification of malicious communications within the VANET through the use of machine learning models were discussed. Many of

the reviewed studies focused on misbehavior detection for all attack types, and not solely on Sybil traffic attack incidents. Another noted area for future analysis, was the handling of the imbalanced state of the simulated VeReMi Extension dataset. Some studies applied methods that included assessment measurements and others proposed resampling techniques such as SMOTE to the simulated VANET dataset, but yielded a decrease in respective classifier performance. As outlined in **Table 9**, this research work aims to apply the SMOTE balancing technique to the VeReMi Extension simulation dataset specifically for Sybil attack detection in the VANET. The dataset will have an aggregated feature set and will be trained and tested on multiple machine learning classifier algorithms.

Table 9: Identified Gaps and Praxis Contributions

Identified Gaps	**Praxis Contributions**
Lower accuracy for classifying sybil attacks. No information on false rates or all confusion matrix measurements.	Train, test and validate three VeReMi (Kamal et. al 2021) datasets, with custom feature engineering and classifier hyperparameter turning. Describe results using confusion matrix measurements.
Plausibility checks and classification tasks performed on imbalanced dataset.	Apply the SMOTE resampling technique to the custom preprocessed and aggregated VeReMi (Kamal et. al 2021) datasets.
Studies performed on all VANET attack types with accuracy averaged across all types.	Perform classification tasks for each individual attack type represented in the VeReMi (Kamal et. al 2021) datasets.

Therefore, the described additional research on *Cybersecurity within the Internet of Vehicles (IoV): Using Machine Learning to Protect Against Sybil Traffic Attacks* will add to the body of work in improving the cyber resilience of VANETs within the IoV.

Chapter 3 - Methodology

3.1 Introduction

This chapter describes the methodology applied in this research process for the praxis. As described in Chapter 1, this praxis aims to build a machine learning based classification model for Sybil traffic attacks using the VeReMi Extension data set (Kamel et al., 2020). Outlined in this chapter are the data collection, processing and analysis steps, and the details on training, testing and validating the classification models. Finally, the research is evaluated based on successfully answering the research questions. A high-level overview of the research process is depicted below in **Figure 2**.

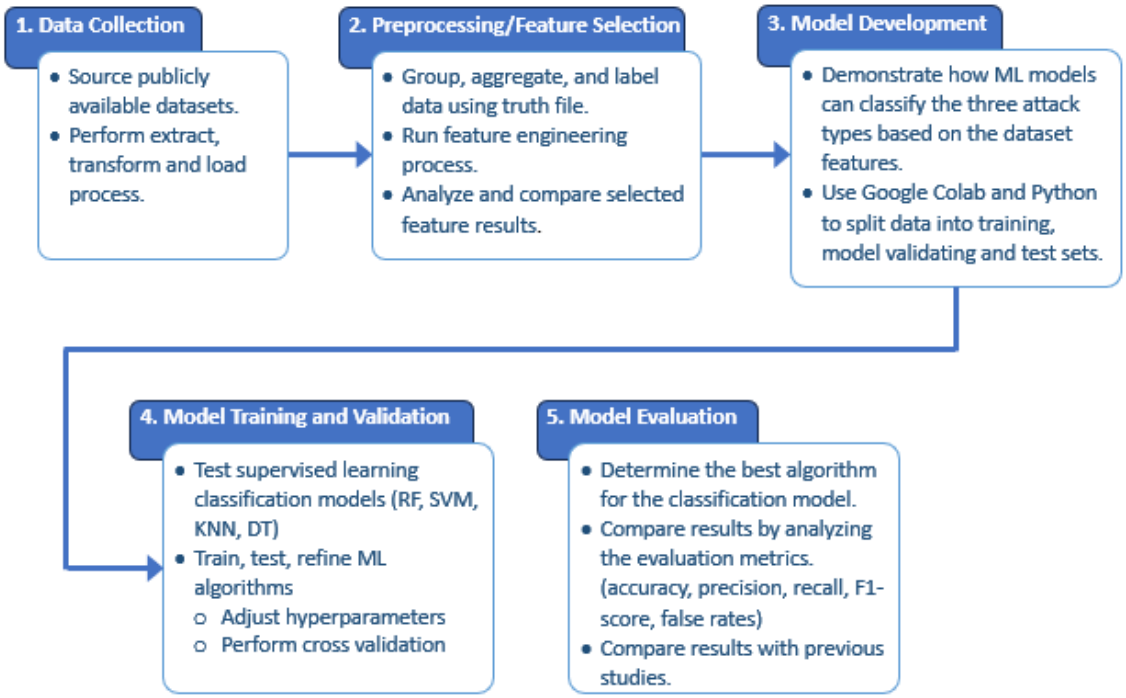


Figure 2: Research Methodology

3.2 Data Collection

The publicly available dataset repository titled Veremi Extension (Kamel et al., 2020) contains vehicle simulation datasets generated for the evaluation of misbehavior detection mechanisms for VANETs. Using their custom Framework For Misbehavior Detection (F_2MD), the authors extended the original VeReMi dataset (Van Der Heijden et al., 2018), by adding four vehicle sensor error models: position, velocity, acceleration, and heading, as well as additional attack types: DoS, DoS random, data replay, disruptive, eventual stop, and traffic congestion Sybil. These datasets are categorized in a folder subset structure of malfunction or attack types as described in **Table 10**.

Table 10: VeReMi Extension Data Set Description(Kamel et al., 2020)

VeReMi Folder Name	Behavior Type	Description
ConstPos	Constant Position Malfunction	Position malfunction mimicking possible positioning system failure (i.e. GPS) where the position is constant throughout the simulation.
ConstPosOffset	Constant Position Offset Malfunction	Position malfunction mimicking possible positioning system failure (i.e. GPS) where the position has a constant offset added to it throughout the simulation.
ConstSpeed	Constant Speed Malfunction	Speed malfunction mimicking possible OBU error or physical sensor failure where the speed is constant throughout the simulation.
ConstSpeedOffset	Constant Speed Offset Malfunction	Speed malfunction mimicking possible OBU error or physical sensor failure where the speed has a constant offset added throughout the simulation.
DataReplay	Data Replay Attack	Simulated attack where the replayed information is signed with the attacker's

		certificate.
DataReplaySybil	Data Replay Sybil Attack	Simulated attack where the replayed information is signed with the attacker's certificate in Sybil mode.
DelayedMessages	Delayed Message Malfunction	Mimicking a large or stressed network or slow OBU unit, messages are sent with all the correct and required data, but with a calculated delay function throughout the simulation.
Disruptive	Disruptive Attack	Attack vehicles replay previously received data from random neighbors.
DoS	DoS Attack	Attack vehicles transmit messages at a rate that exceeds the defined limit specified by IEEE or ETSI standards.
DoSDisruptive	DoS Disruptive Attack	Attack vehicles replay previously received data from random neighbors.
DoSDisruptiveSybil	DoS Disruptive Sybil Attack	Attack vehicles replay previously received data from random neighbors, changing their identities for every sent message to avoid detection.
DoSRandom	DoS Random Attack	Attack vehicles send messages with all the message fields set to random values.
DoSRandomSybil	DoS Random Sybil Attack	Attack vehicles send messages with all the message fields set to random values, changing their identities for every sent message to avoid detection.
EventualStop	Eventual Stop Attack	Attack vehicles simulate a sudden stop by freezing the position values and setting the speed values to null.
GridSybil	Grid Sybil or Sybil Traffic Attack	Attack vehicles create fake traffic congestion by generating a grid of fake vehicles in a chosen position.
RandomPos	Random Position Malfunction	The position is random at every time step for specified vehicles.

RandomPosOffset	Random Position Offset Malfunction	The position is random with a generated offset at every time-step for specified vehicles.
RandomSpeed	Random Speed Malfunction	The speed is random at every time step for specified vehicles.
RandomSpeedOffset	Random Speed Offset Malfunction	The speed is random with a generated offset at every time-step for specified vehicles.
MixAll_0024	All Malfunction and Attacks	Contains a mix of all the previously described misbehavior attacks for a whole simulation day (0h-24h).

3.3 VeReMi Extension Datasets

To demonstrate an improvement upon the results from previous research studies, three VeReMi Extension data subsets were chosen. There are two simulation variations for each subset folder, a rush hour timeframe with the suffix "_0709" and a low traffic timeframe with the suffix "_". One additional folder contains a combined mix of all the behavior types with the simulation duration of a whole day. Each subset folder holds a .JSON file for each vehicle in the simulation and contains the respective vehicle's Onboard Unit (OBU) message exchange data. The subset folder also contains a ground truth .JSON file that contains only the true or genuine messages exchanges that occurred during the simulation. The datasets outlined in **Table 11** were selected to demonstrate this praxis.

Table 11: Dataset File and Storage Details(Kamel et al., 2020)

Dataset Name	Reason for Inclusion	Details
GridSybil_0709	The Grid Traffic Sybil dataset contains data that supports a sybil attack within VANET.	• 4070 JSON .files. • # of Records: 5.6 Million • Storage size: ~ 4.36 GB
DosRandomSybil_0709	The Denial of Service Random Sybil dataset contains data that supports a Random Denial of Service attack in Sybil mode. The position data is also	• 4070 JSON .files. • # of Records: 3.9 Million • Storage size: ~ 2.40 GB

	falsified.	
RandomPoSOffset_0709	The Random Position Offset dataset contains data that supports sensor malfunction as a type of misbehavior where the position data is incorrect.	• 4070 JSON .files. • # of Records: 3.4 Million • Storage size: ~ 1.91 GB

The datasets contain the same feature set, and the name and description of each feature is given in **Table 12.**

Table 12: Datasets Feature Descriptions(Kamel et al., 2020)

Feature Name	Description
type	Type of communication: Type 2 = Data as detected by the sensors Type 3 = Data as received by nearby vehicles
rcvTime	Message received timestamp
sendTime	Message sent timestamp
sender	Sender Identifier
senderPseudo	Sender Pseudo Identifier
messageID	Message Identifier
Pos(x,y,z)	Position in vector format
pos_noise(x,y,z)	Position noise in vector format
spd(x,y,z)	Speed in vector format
spd_noise(x,y,z)	Speed noise in vector format
acl(x,y,z)	Acceleration in vector format
acl_noise(x,y,z)	Acceleration noise in vector format
hed(x,y,z)	Header in vector format
hed_noise(x,y,z)	Header noise in vector format

Within each dataset, records coded as “Type: 2” refer to TYPE_GPS 2, which is the data detected by the sensors. The records coded as “Type: 3” refer to TYPE_TAG 3, and is the data received by nearby vehicles (Kamel, J. 2020). The data in the ground truth file is coded as “Type: 4”, which refers to TYPE_TRUTH_BEACON 4 and is the “truth” or true data sent by each vehicle. To transform the data into a relational format,

each .JSON file was bulk inserted into the database, where the preprocessing and aggregation steps were carried out.

3.3 Data Preprocessing

Data preprocessing helps ensure that the data provided to the machine learning model is clean, relevant and appropriately formatted. Removing noise reduction, the errors, outliers and irrelevant information, helps improve the model's ability to learn meaningful patterns. Normalizing or scaling the dataset ensures that certain features don't dominate the learning process. Handling missing data, which can also arise after preprocessing steps, ensures a complete dataset. Techniques such as over or under sampling and resampling or synthetic data generation can address imbalanced data sets. Splitting the data into training and testing sets is also considered a preprocessing step and is crucial for assessing model performance.

The process for transforming the data from the Sybil grid JSON files into the dataset for the model input involved the steps depicted in **Figure 3**. Step 1 of the process

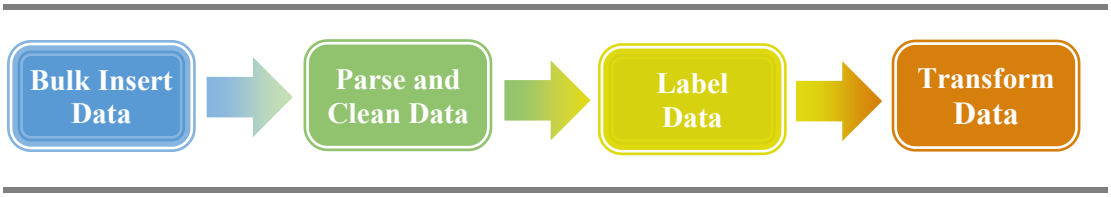


Figure 3: Data Set Preprocessing

is to bulk insert the JSON files into the database. The vehicle labeled files are inserted into a "trace" staging table and the ground truth files are inserted into a "truth" staging table. The "type 2" vehicle sensor data is removed and inserted into a different from table for later processing. Step 2 is to parse the individual fields into separate columns, and then clean the data by stripping the element names and character data. The position vector values are parsed out into the respective x and y columns during this step. This

step is performed on the “trace” and “truth” tables. Each vehicle record is then updated with it’s own sensor data based on the max sensor recorded time that is closest to but less than the “sendTime” of the message received by the vehicle. The labeling process takes place with Step 3. The messages in the trace table are compared to the messages in the ground truth table to determine which messages are misbehaving. The “Misbehavior” class label is populated with “Yes” for misbehaving or “No” if the message is found to be Genuine. This was done because not all of the messages in the attack JSON files are misbehaving. Step 4 is the grouping and aggregation process. **Table 13** shows the fields that were included in the aggregation step for each dataset.

Table 13: Aggregated Dataset Fields

ID	Feature	Not Included	Aggregation Description
1	filename(receiver vehicle)	X	
2	type	X	
3	rcvTime		Group by whole number
4	**sendTime**		**group by whole number, min, max**
5	sender		count: number of senders
6	**senderpseudo**		**count: number of sender pseudo per sender**
7	**messageid**		**count: number of messages sent**
8	**sensor position X**		**max value at time of message**
9	sensor position Y		max value at time of message
10	sensor position Z	X	
11	sensor position noise X		max value at time of message
12	sensor position noise Y		max value at time of message
13	sensor position noise Z	X	
14	**sensor speed X**		**max value at time of message**
15	**sensor speed Y**		**max value at time of message**
16	sensor speed Z	X	
17	sensor speed noise X		max value at time of message
18	sensor speed noise Y		max value at time of message
19	sensor speed noise Z	X	
20	sensor acceleration X		max value at time of message

21	sensor acceleration Y		max value at time of message
22	sensor acceleration Z	X	
23	sensor acceleration noise X		max value at time of message
24	sensor acceleration noise Y		max value at time of message
25	sensor acceleration noise Z	X	
26	sensor header X		max value at time of message
27	**sensor header Y**		**max value at time of message**
28	sensor header Z	X	
29	sensor header noise X		max value at time of message
30	sensor header noise Y		max value at time of message
31	sensor header noise Z	X	
32	message position X		Average value within message timeGroup
33	**message position Y**		**Average value within message timeGroup**
34	message position Z	X	
35	message position noise X		Average value within message timeGroup
36	message position noise Y		Average value within message timeGroup
37	message position noise Z	X	
38	message speed X		Average value within message timeGroup
39	**message speed Y**		**Average value within message time group**
40	message speed Z	X	
41	message speed noise X		Average value within message timeGroup
42	message speed noise Y		Average value within message timeGroup
43	message speed noise Z	X	
44	message acceleration X		Average value within message timeGroup
45	message acceleration Y		Average value within message timeGroup
46	message acceleration Z	X	
47	**message acceleration noise X**		**Average value within message time group**
48	**message acceleration noise Y**		**Average value within message time group**
49	message acceleration noise Z	X	
50	message header X		Average value within message timeGroup
51	**message header Y**		**Average value within message time group**
52	message header Z	X	
53	message header noise X		Average value within message time group
54	message header noise Y		Average value within message time group

55	message header noise Z	X	
56	**Misbehavior**		**Class Label (Yes, No)**

The time gap field is created by grouping the messages sent within the same pico second (e.g. 28850.186 is between 28850.0 and 28850.9). The position, speed, acceleration and heading fields are averaged for each time gap. The next step for the data table is the feature selection process.

3.4 Feature Selection

The preprocessed praxis dataset contains 32 features and needs to undergo a feature selection process to determine which features will have significant contribution to the model's performance. The use of formal feature selection methods can help avoid serious model training issues in supervised machine learning. One such issue is called overfitting, an issue where the machine learning model is trained too well on the training data and starts to fit the noise or aspects of the data that are not generalizable outside the dataset instead of continued use of the underlying patterns. Another issue that can be avoided is the problem of high dimensionality where there is increased computational complexity due to the dataset having a large number of dimensions. Reducing the number of features encourages optimized model performance where the output is simpler and easier to interpret.

The research studies identified in the literature review employed the use of different feature selection methods to varying results. To achieve optimum results using the Traffic Sybil misbehavior classification model, multiple feature selection methods were analyzed to identify a preeminent set of features that can be used for detecting traffic sybil misbehavior in VANETs.

Supervised feature selection methods use the target variable when selecting the subset of relevant features as opposed to unsupervised methods. A supervised feature selection method was adopted for use in this praxis.

The supervised feature selection process starts with data preparation where the labeled dataset is divided into two parts: features (X) and the target variable (y). Next, ranking features are applied where each feature is individually assessed based on its relevance to the target variable.

3.4.1 Feature Selection Methods

Described in this section is an overview of common techniques used for selecting the best subset of features for machine learning models. The three main categories of feature selection methods are filter, wrapper and embedded. Highlighted within each category, are the selection methods geared towards classification tasks.

Filter methods work by evaluating the relevance of each feature independently of a machine learning model. They rely on statistical or information-theoretic metrics to score features based on their relationship with the target variable. Filter methods are computationally efficient and are often used as a preprocessing step to quickly identify potentially relevant features. Filter methods for classification tasks include:

- **Chi-Squared Test:** Selects features by assessing the independence between categorical features and the target variable and then quantifying the difference between the observed and expected frequencies.
- **Correlation Coefficient Scores:** Selects features based on linear dependence between two variables.

- **Information Gain**: Selects features by quantifying the reduction in uncertainty of the target variable when a feature is used for splitting.

Wrapper methods work by using a machine learning model to evaluate the importance of features by their impact on model performance. Different subsets of features are evaluated, with the selected features yielding the best performance. Wrapper methods are computationally intensive because they involve training and evaluating the model multiple times for different feature subsets. However, they tend to yield more accurate feature selection results compared to filter methods. Wrapper methods identified as beneficial for classification tasks are:

- **Recursive Feature Elimination (RFE):** Feature selection technique based on recursive model training. A ranking system based on a features contribution to each model's performance is used to eliminate low performing features. An iterative process is used until a predetermined criterion is met. Useful for datasets with large number of features.
- **Forward Selection:** Selects features through an iterative process that involves training the model by adding one feature at a time. Each time a new feature is added, the model performance is evaluated and assessed with performance metrics such as accuracy and F1 score. The feature set is selected based on each features impact to the model performance.
- **Backward Elimination:** Uses an iterative feature selection process that begins with all available features and then removes them one at a time. This method is also commonly employed within wrapper methods for feature selection.

- **Exhaustive Search:** Selects features by systematically evaluating all possible subsets of features. All possible feature set combinations are generated and used to train the model. The feature set is selected based on the performance of each subset.

Embedded methods work incorporating feature selections as an integral part of the model training process. The relevant features are automatically selected while the model is being trained. Embedded methods are computationally efficient and often strike a balance between the filter and wrapper methods, as they select features during model training. Identified embedded methods are:

- **L1 Regularization (Lasso):** Lasso, which stands for Least Absolute Shrinkage and Selection Operator, performs feature selection by using a regularization or scaled version of the absolute values of the model's coefficients. The objective is to minimize the sum of two terms: The first term is the typical loss function that measures the model's ability to fit the training data and the absolute sum of the model's coefficients (weights) multiplied by a regularization parameter, usually denoted as λ (lambda).
- **Decision Tree based:** Features are selected using a machine learning algorithm. The decision tree algorithm splits the features based on their importance in making classification or regression decisions. For classification trees, Gini impurity is the criterion used for making splitting decisions during the construction of the tree. The criterion quantifies the probability of incorrectly classifying a randomly selected data point when it is assigned a class label based on the distribution of classes within a dataset.

- **Support Vector Machine (SVM) based Feature Selection:** SVM based feature selection uses SVMs to rank or score features based on feature weights, ranking, or the absolute values of the SVM's coefficients. These criteria quantify how much each feature contributes to the SVM's decision boundary. SVM based feature selection is primarily used for classification tasks and can be applied to both linear and non-linear SVMs.
- **Ridge Regression:** Selects features in a similar manner as the LASSO method, but uses an L2 penalty, which is the sum of the squares of the coefficients. Usage in classification is beneficial for handling multicollinearity and model overfitting.
- **Elastic Net selection methods:** This feature selection method is a hybrid of the L1 (Lasso) and L2 (Ridge) methods and is useful when multiple features are correlated with each other.

Shown in Error! Reference source not found. and Error! Reference source not found. are the Lasso feature selection results for each dataset. Using a very small alpha value (1e-05) allows for a very light amount of regularization for feature selection capabilities. Utilizing Lasso linear regression for feature selection can be beneficial for classification tasks by indicating which features are less important or irrelevant, which can simplify the model and reduce overfitting.

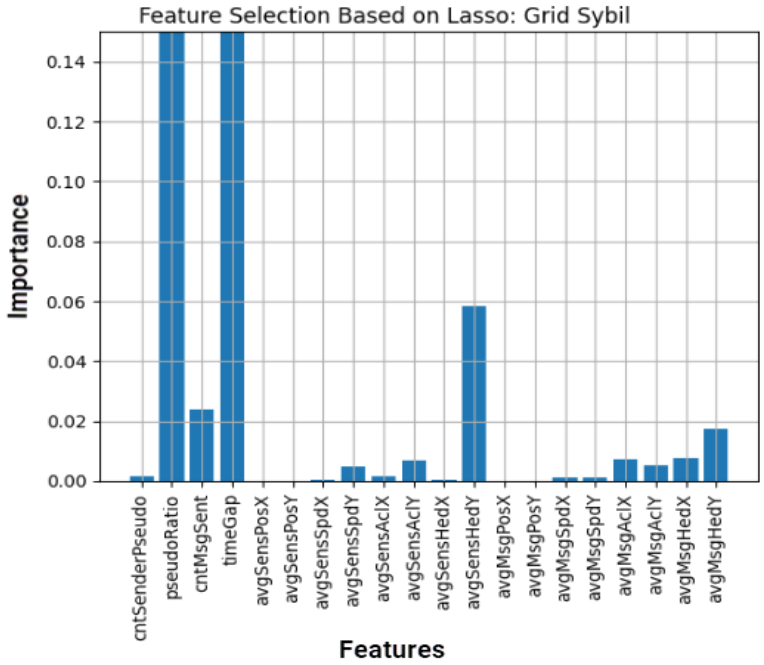


Figure 4: Lasso - Grid Sybil

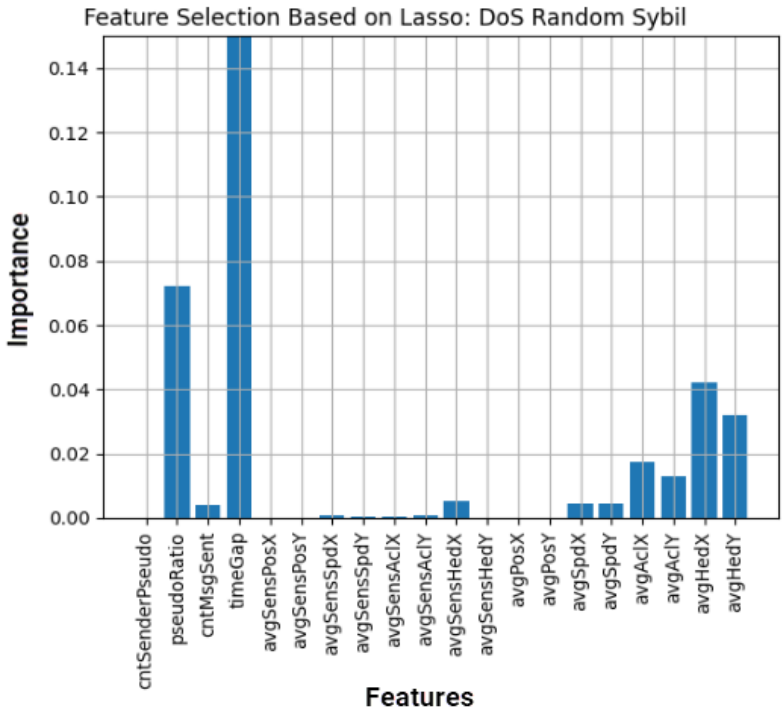


Figure 5: Lasso – DoS Random Sybil

Advantages and disadvantages of the discussed feature selection techniques are detailed in **Table 14.**

Table 14: Feature Selection Techniques-Advantages and Disadvantages

Category	Method	Advantages	Limitations
Filter	Chi-Squared Test	• Identifies features with significant impacts on the outcome. • Has feature ranking and results are easily interpreted.	• Assumes data is categorical. • Not suitable for datasets containing continuous features. • Has issues with non-linear relationships.
	Correlation Coefficient Scores	• Good for quantifying the linear relationship between features. • Easy to determine redundant features for removal and • reducing overfitting.	• Inability to capture non-linear dependencies. • Usually combined with other methods for better feature selection strategy.
	Information Gain	• Selects features with the most valuable information about the target variable. • Features selected are the most relevant.	• May not perform well if with multiple categories or continuous variables with wide range of values.
Category	**Method**	**Advantages**	**Limitations**
Wrapper	Recursive Feature Elimination (RFE)	• Least important features are removed iteratively. • Uses an automatic feature selection approach.	• Can be computationally intensive. • Tends to favor linear models over non-linear ones.
	Forward Selection	• Easy to understand and implement. • Reduced risk of overfitting. • Finds highly informative features earlier.	• Due to being a greedy algorithm, makes decisions based on current context without revisiting previous decisions.
	Backward Elimination	• Easy to understand and implement. • Removes features one by one, easy to understand the order of elimination.	• Makes decisions based on the current context without revisiting previous decisions.
Category	**Method**	**Advantages**	**Limitations**
Embedded	L1 Regularization (Lasso):	• Has automatic model simplification process. • Good for dimensionality reduction.	• Not suitable for all data types. • Primarily designed for linear models. • Has arbitrary feature selection.

	Decision Tree based	• Selects the most relevant features for the specific model. • Captures non-linear relationships between features and the target variable. • Identifies interactions between features.	• Feature selection is model specific. • Prone for overfitting. • Computationally intensive for larger datasets. • May not detect feature redundancy.
	Support Vector Machine based	• Robust and high-performing. • Infers importance of features by magnitude of associated coefficients. • Effective with High-Dimensional Data	• High computational complexity. • Needs careful tuning of hyperparameters. • Can be difficult to interpret. • Feature selection is not the primary function.

After careful analysis of the feature selection methods best suited for classification problems, this praxis utilizes a hybrid approach of techniques to yield the best results. A combination of methods from each category were used to obtain the best feature subset.

3.5 Machine Learning Models

This praxis tests the performance of five supervised ML models to determine which model has superior performance in terms of classification accuracy, F1 score, and lowest false alarm rates. The five models examined in this praxis are Decision Tree, KNN, SVM, RF, and Multilayer Perceptron (MLP) which is an Artificial Neural Network (ANN).

These Machine Learning (ML) models are evaluated using the Python programming language. Python's popularity in ML development stems from its straightforward syntax, extensive library and framework ecosystem and platform independence, which enables scientists and researchers to concentrate on the conceptual aspects of their models rather than the complexities of coding

Decision Tree: This algorithm is useful for classification and regression tasks. Its hierarchical structure represents a sequence of decisions and their possible outcomes. The dataset is recursively split into subsets based on the features, with decisions being made at each note of the tree until a stopping criterion. For classification tasks, each leaf node of the tree represents a class label.

k -Nearest Neighbors (KNN): This instance-based learning algorithm is characterized by memorizing training instances. KNN performs the classification by finding the parameter specified k nearest neighbors of the sample. The k data points closest to the query data point are identified. For classification tasks, a process called majority voting is employed, where KNN counts the number of neighbors in each class among the K neighbors and assigns the class label with the highest count to the query data point.

Support Vector Machine (SVM): An extension of the perceptron algorithm, SVM maximizes the margin between the decision boundary (hyperplane) and instances closest to the hyperplane (support vectors). The SVM hyperparameters used in this praxis are described in **Table 15**.

Table 15: SVM Hyperparameters

SVM-Kernal
Kernal=Poly
C = 1.0

Random Forest (RF): This algorithm is an ensemble of decision trees. It works by dividing the input space into multiple subsets, which are then fed into individual decision trees running in parallel. A prediction is made for the class label of a data point, and for classification tasks, the final decision is made by combining the prediction

outcome of the trees. Employing the majority voting concept, the class label that receives the most votes among the trees is selected as the final prediction. The Random Forest hyperparameters used are described in **Table 16.**

Table 16: Random Forest Hyperparameters

ccp_alpha = 0.0
class_weight = None
Criterion = Gini
n_estimators = 100

Multi-layer Perceptron (MLP): A multi-layer feed-forward neural network consisting of an input layer, one or more hidden layer(s), and an output layer. MLP is trained to perform input-output mapping, often with a non-linear (activation) function.

3.6 Model Training and Validation

Model training and validation are essential steps in the machine learning model development process. With supervised models, the process involves teaching the model to make classifications based on labeled data, then using a separate data set to assess and validate the performance. The following key steps help to achieve a highly effective and reliable model that performs well against new and unseen data:

- Split the data into training and validation sets.
- Initialize the parameters and determine hyperparameter settings for the selected model.
- Perform cross-validation to help mitigate overfitting.
- Perform iterative training using the training data set.
- Monitor the model's performance with calculation metrics.
- Use a random subset of the data to validate the model. This data should be similar to the training set.

- Compare the predictions with the actual target values.
- Compute the performance metric values.

Training and validating different models allow for the comparison of each model's output, and the determination of which is best suited. Performing cross validation helps lesson the likelihood for overfitting as its process involves splitting the dataset into several smaller train/test datasets, then iteratively training the model on those subsets. For each split, the model starts with its initial parameters and learns from the training data, with the goal of estimating how well the model will perform on unseen data.

3.7 Model Evaluation

This Sybil misbehavior classification model has two classes, misbehaving and genuine, thus four classification possibilities. Shown in **Figure 6** is the binary confusion matrix applied in this praxis.

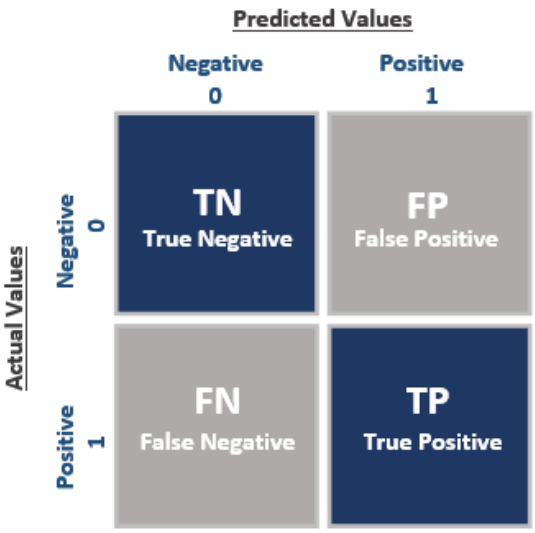


Figure 6: Confusion Matrix

As with (Kamel et al., 2020), the following measurements were observed:

- True positive (TP): the number of observations correctly classified as misbehaving. The model correctly detected the misbehaving V2V message.
- True negative (TN): the number of observations correctly classified as genuine. The model correctly classified the genuine V2V message.
- False positive (FP): the number of observations incorrectly classified as misbehaving. The model incorrectly classified the V2V message as misbehaving.
- False negative (FN): the number of observations incorrectly classified as legitimate. The model incorrectly classified a V2V message as genuine.

The precision, recall, F1, and accuracy metrics were used to evaluate the Sybil traffic misbehavior classification model. The confusion matrix was used as the data inputs to assess the model's performance and identify the best performing classifier.

The performance metrics and formulas (Kamel et al., 2020) are provided below in **Table 17.**

Table 17: Performance metrics and formulas(Kamel et al., 2020)

Metric	Description	Formula
False Negative Rate (FNR)	The percentage of messages incorrectly classified as genuine over all messages.	$\text{FNR} = \frac{\text{FN}}{(\text{FN}+\text{TP})}$
False Positive Rate (FPR)	The percentage of messages incorrectly classified as genuine over all messages	$\text{FPR} = \frac{\text{FP}}{(\text{FP}+\text{TN})}$
Accuracy	The ratio of correctly classified messages over all the considered classifications.	$Accuracy = \frac{TN+TP}{TN+TP+FN+FP}$
Precision	The classifiers ability to distinguish between misbehaving and genuine communications.	$\text{Precision} = \frac{\text{TP}}{(\text{TP}+\text{FP})}$

Recall	The classifiers ability to detect a misbehaving communication. A low recall means an attack is difficult to detect.	$\text{Recall} = \frac{\text{TP}}{(\text{TP}+\text{FN})}$
F1 Score	The harmonic mean between Recall and Precision. The quality of the overall classification.	$F1\ score = 2\ \text{X} \frac{Precision\ \text{x}\ Recall}{Precision + Recall}$

3.8 Summary

This chapter presented the research methodology process used for performing binary classification on the GridSybil_0709, DoSRandomSybil_0709 and RandomPosOffset_0709 VeReMi Extension Datasets. Details about the data collection and pre-processing steps were provided, as well as a thorough evaluation and comparison of feature selection methods.

Supervised machine learning models were leveraged for this research, using the key steps outlined in **section 3.5**, to achieve the most efficient and effective development. The utilization of cross-validation during the model training process was vital step to prevent overfitting and to ensure the model performed effectively on new data. The methodology adopted for training the model included splitting the data set into 70% training and 30% testing subsets, as well as training and testing with a 5-fold cross-validation technique as a variation. The results are discussed next, in Chapter 4.

Chapter 4 Results

4.1 Introduction

This chapter presents the results of this praxis and the evaluation of the methodology described in Chapter 3. The methodology results are assessed using the VeReMi extension Grid Sybil, DoS Random Sybil and Random Position Offset datasets, and answer the research questions and hypotheses discussed in Chapter 1 and shown in **Table 18**.

Table 18: Research Questions

Research Questions		
Research Question #1	**Research Question #2**	**Research Question #3**
Which set of features are the most significant indicators for determining whether a IoV communication is malicious or safe?	Can a classification machine model be developed to accurately classify malicious and safe IoV communications?	Which machine-learning algorithm can classify IoV communications with the highest accuracy and low false positive rate and negative rates?

The main sections of this chapter examine the methodology steps followed in this research. **Section 4.2** describes the dataset preprocessing steps. **Section 4.3** contains the results from the feature selection methods and provides the answers to research question #1. **Section 4.4** reveals the metrics applied to assess the pre-selected classification algorithms, and the results for the binary classification specified in Chapter 3, and addresses research questions 2 and 3. **Section 4.5** identifies and evaluates the best-performing algorithm while exploring any correlations between binary and multi-class classification with ensemble learning. The chapter concludes with **Section 4.6,** which provides a summary of the model results and conclusion. The research questions and hypotheses are appraised and validated based on the research results.

4.2 Dataset Balancing

The Synthetic Minority Over-sampling Technique (SMOTE) method was used to balance the datasets by generating synthetic data for the minority class. **Table 19** displays the before and after SMOTE results for each dataset.

Table 19: Dataset SMOTE Results: Before and After

Dataset	Class	Before Smote	After Smote
Grid Sybil	**Genuine**	303,531	303,531
	Misbehaving	94,631	303,531
	Total # of Records	**398,162**	**607,062**
DoS Random Sybil	**Genuine**	279,420	279,420
	Misbehaving	115,191	279,420
	Total # of Records	**394,611**	**558,840**
Random Position Offset	**Genuine**	277,664	277,664
	Misbehaving	114,481	277,664
	Total # of Records	**392,145**	**555,328**

4.3 Feature Selection Results

Three of the feature selection methods discussed in **Section 3.4** were applied to the praxis dataset. The results from the Forward Feature Selection, Decision Tree based and Correlation Coefficient Scores feature selection methods are described in, and are used as the basis to address the first research question: “Which set of features are the most significant indicators for determining whether a IoV communication is malicious or safe?”.

4.3.1 Forward Feature Selection

Forward feature selection is a technique used in feature selection where you start with an empty set of features and gradually add one feature at a time, evaluating the performance of the model at each step. This process continues until a predefined stopping

criterion is met. Shown in **Table 20** are the results from the forward feature selection method for each dataset.

Table 20: Wrapper - Forward Feature Selection Results

Grid Sybil		DoS Random Sybil		Random Position Offset	
Feature	**Accuracy**	**Feature**	**Accuracy**	**Feature**	**Accuracy**
avgMsgSpdX	0.9995	cntSenderPseudo	0.9992	avgHedY	0.99547
cntSenderPseudo	0.9996	avgSpdX	0.9990	timeGap	0.99545
avgAclY	0.9998	avgSpdY	0.9984	avgPosNoiseX	0.99544
avgAclX	0.9998	cntMsgSent	0.9969	avgPosNoiseY	0.99532
		pseudoRatio	0.9931	avgPosY	0.99532
		timeGap	0.9922	avgAclX	0.99531
		Feature	0.9992	avgAclY	0.99529
				avgSpdX	0.99525
				avgSpdY	0.99485
				avgPosX	0.99402
				avgHedX	0.99173
				cntMsgSent	0.99153
				pseudoRatio	0.89279

4.3.2 Decision Tree based Feature Selection

The embedded type Decision Tree based selection feature method identified a similar set of features, but applied a different rank. The higher the ranking, the better the feature split the data into the correct class. Illustrated in **Table 21** are the Importance rankings for the selected features in descending order.

Table 21: Decision Tree based Feature Selection Scores

Grid Sybil		DoS Random Grid		Position Offset	
Feature	**Rank**	**Feature**	**Rank**	**Feature**	**Rank**
pseudoRatio	0.7902	timeGap	0.9726	avgPosY	0.2903
timeGap	0.0950	avgSpdY	0.0132	avgPosX	0.1999
cntMsgSent	0.0452	avgSpdX	0.0044	timeGap	0.1419
avgSpdY	0.0320	pseudoRatio	0.0035	avgHedY	0.0930
		cntMsgSent	0.0010	avgHedX	0.0767
				avgSpdX	0.0757
				avgSpdY	0.0684

4.3.3 Correlation Coefficient Feature Selection

The Correlation Coefficient Scores (Pearson) feature selection method was applied to create the correlation coefficient matrix. Usually when features are indicated as highly correlated, they are removed from the feature set to increase the model's efficiency. For the Grid Sybil dataset, some highly correlated features were not removed from the feature set due to a key element of traffic Sybil attacks within the VANET. Attackers attempt this misbehavior by forging of multiple identities. Both the senderPseudo and Sender feature are highly correlated, but the senderPseudo to sender ratio and number of messages sent are key to detecting if an abnormal number of identifiers is being used. The model performed higher with these specific highly correlated features than without, as the other features on their own don't represent false identities. Shown in **Figure 7** is the correlation map for the Sybil Grid dataset**,** and shown in **Figure 8** is the correlation map for the DoS Random Sybil dataset. Illustrated are the similarities and differences of the strong feature pairs for the datasets.

Grid Sybil Correlogram of Features

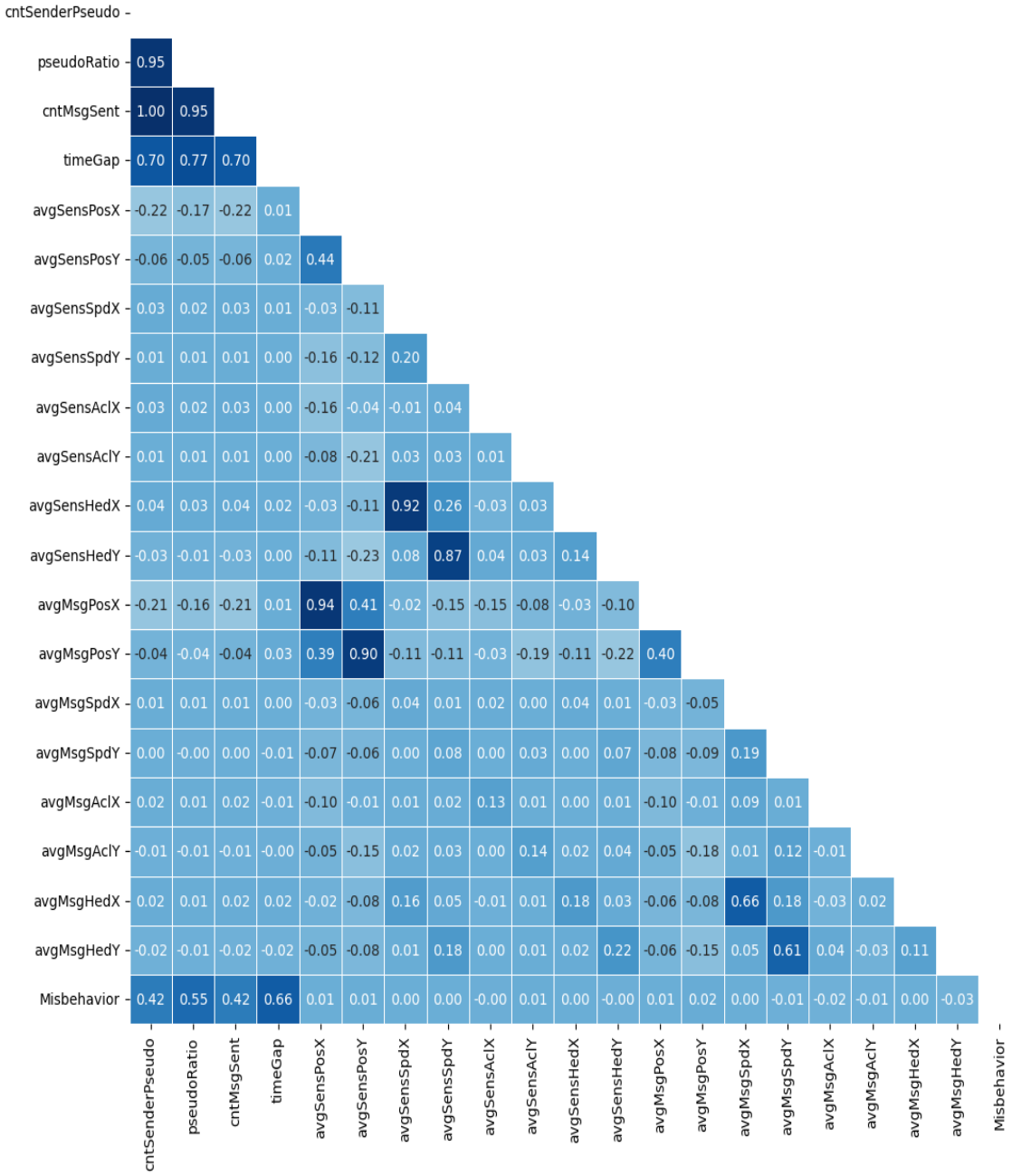


Figure 7: Correlation Heat Map - Sybil Grid

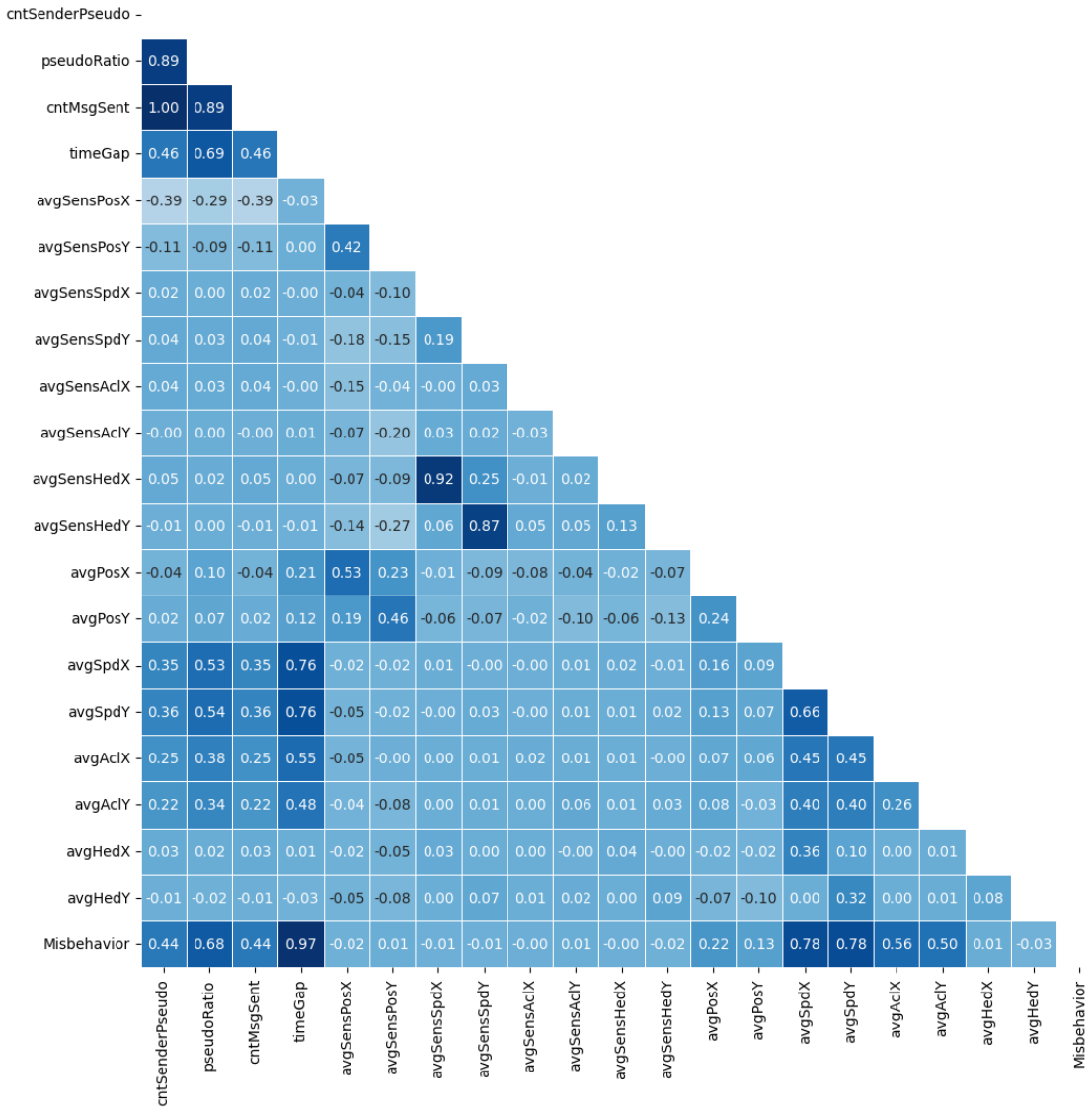


Figure 8: Correlation Heat Map – DoS Random Sybil

An example of highly correlated feature pairs in the Sybil Grid dataset are the features “pseudoRatio and cntMsgSent” and “avgMsgPosX and avgSensPosX”. The first feature pair represents the number of messages sent and the ratio of pseudo identifiers the sender is using during the send time interval of the message. The second feature pair represents the average position x-axis value the sender sent in the message and the receiving vehicle’s average sensor reading for the position x-axis during the same time

interval. When Sybil attacks are implemented, the attacker creates many more pseudo identifiers and sends out more messages. **Figure 9** illustrates the correlation of the pseudoRatio feature to the other features in the Grid Sybil dataset.

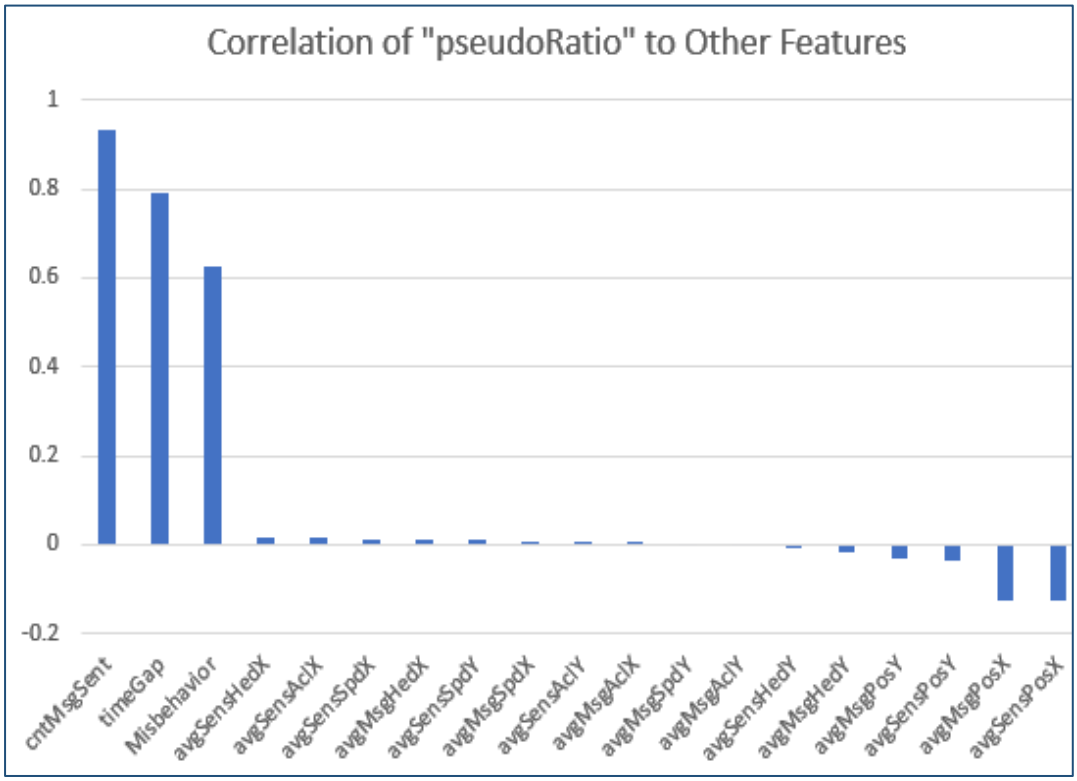


Figure 9: Grid Sybil Feature Correlations

Similar to the Grid Sybil data set, the highly correlated feature pairs in the DoS Random Sybil are “avgMsgPosX and avgSensPosX”, “cntMsgSent and pseudoRatio”. As position falsification is part of the attack due to random position values being sent in the messages, the correlation feature selection method identified the high correlation between the “avgSensHedX and avgSensSpdX” and “avgMsgPosX and avgSensPosX” feature pairs as well.

The feature pairs with the highest correlations in the Random Position Offset dataset are “avgPosX and avgSensPosX”, “avgPosY and avgSensPosY”, “pseudoRatio

and cntSenderPseudo", and "avgSensPosX and avgSensPosX". The high correlation values for average position x-axis and average sensor position x-axis features are expected as the sensors values are the true values registered from the vehicle receiving the messages. The average position x-axis represents the value sent by the sender in the message. **Figure 10** illustrates the correlation of the "avgPosX" feature to the other features in the Random Position Offset dataset.

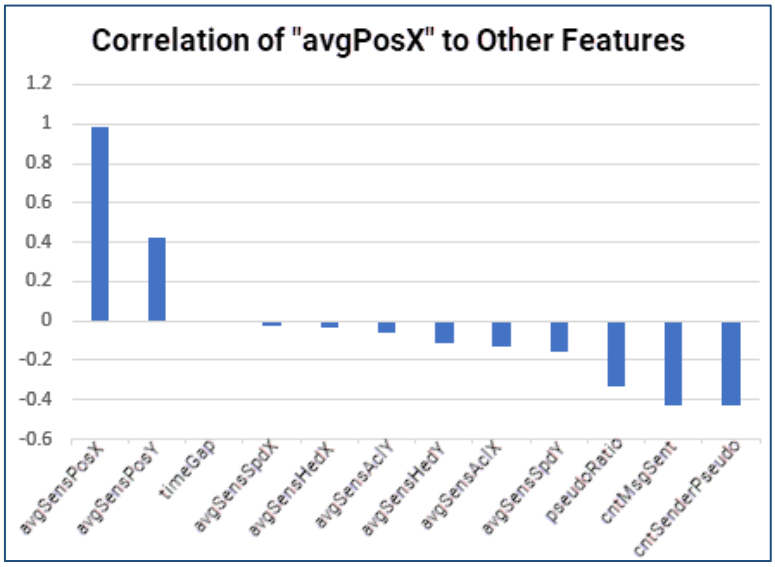


Figure 10: Random Position Offset Feature Correlations

Although these features are highly correlated, their inclusion adds to the success of the model based on the type of attacks being classified.

4.3.4 Selected Features

The chosen features were selected from the combined results of the feature selection methods. Upon careful consideration of the model's objective to accurately classify a specific type of Sybil attack communication, and the need to strike a balance between efficiency and complexity, **Table 22** reveals the selected features employed for the praxis model.

Table 22: Selected Features for Praxis Datasets

ID	Dataset Feature Name	Selected Feature Name
3	rcvTime	
4	sendTime	timeGap
5	sender	cntSenderPseudo
6	senderpseudo	pseudoRatio
7	messageid	cntMsgSent
8	sensor position X	avgSensPosX
9	sensor position Y	
11	sensor position noise X	
12	sensor position noise Y	
14	sensor speed X	avgSensSpdX
15	sensor speed Y	avgSensSpdY
17	sensor speed noise X	
18	sensor speed noise Y	
20	sensor acceleration X	
21	sensor acceleration Y	
23	sensor acceleration noise X	
24	sensor acceleration noise Y	
26	sensor header X	
27	sensor header Y	avgSensHedY
29	sensor header noise X	
30	sensor header noise Y	
32	message position X	avgPosX
33	message position Y	avgPosY
35	message position noise X	
36	message position noise Y	
38	message speed X	avgSpdX
39	message speed Y	avgSpdY
41	message speed noise X	
42	message speed noise Y	
44	message acceleration X	avgAclX
45	message acceleration Y	avgAclY
47	message acceleration noise X	
48	message acceleration noise Y	
50	message header X	avgHedX
51	message header Y	avgHedY
53	message header noise X	
54	message header noise Y	
56	Misbehavior	Misbehavior

4.4 VANET Misbehaving Classification Model Results

Detailed in this section is an explanation of the outcomes from the VANET misbehaving classification model. The goal of the model is to solve the binary classification problem of accurate classification of genuine and misbehaving vehicle to vehicle communications. The following research questions are addressed:

"Can a classification machine model be developed to accurately classify malicious and safe IoV communications?"

"Which machine-learning algorithm can classify IoV communications with the highest accuracy and low false positive rate and negative rates?"

As discussed in Chapter 3, the five machine learning algorithms used for the classification problem were Decision Tree, KNN, SVM, Random Forest, and MLP.

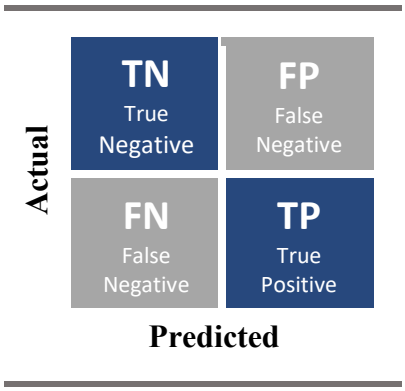


Figure 11: Confusion Matrix Legend

The question, "Is this vehicle communication message misbehaving?" is represented in the four confusion matrix categories **(Figure 11)**. The "True Negative" category represents "Genuine" messages that were correctly classified as "Genuine". These messages are negative for being misbehavior. The "False Positive" category represents "Genuine" messages incorrectly classified as "Misbehaving". "Misbehaving"

messages classified as “Misbehaving” are categorized as “True Positive”, and “Genuine” messages incorrectly classified as “Misbehaving” are categorized as “False Positive”.

4.4.1 Grid Sybil Dataset Results

The confusion matrices by classifier for the Grid Sybil dataset are shown in **Figure 12**.

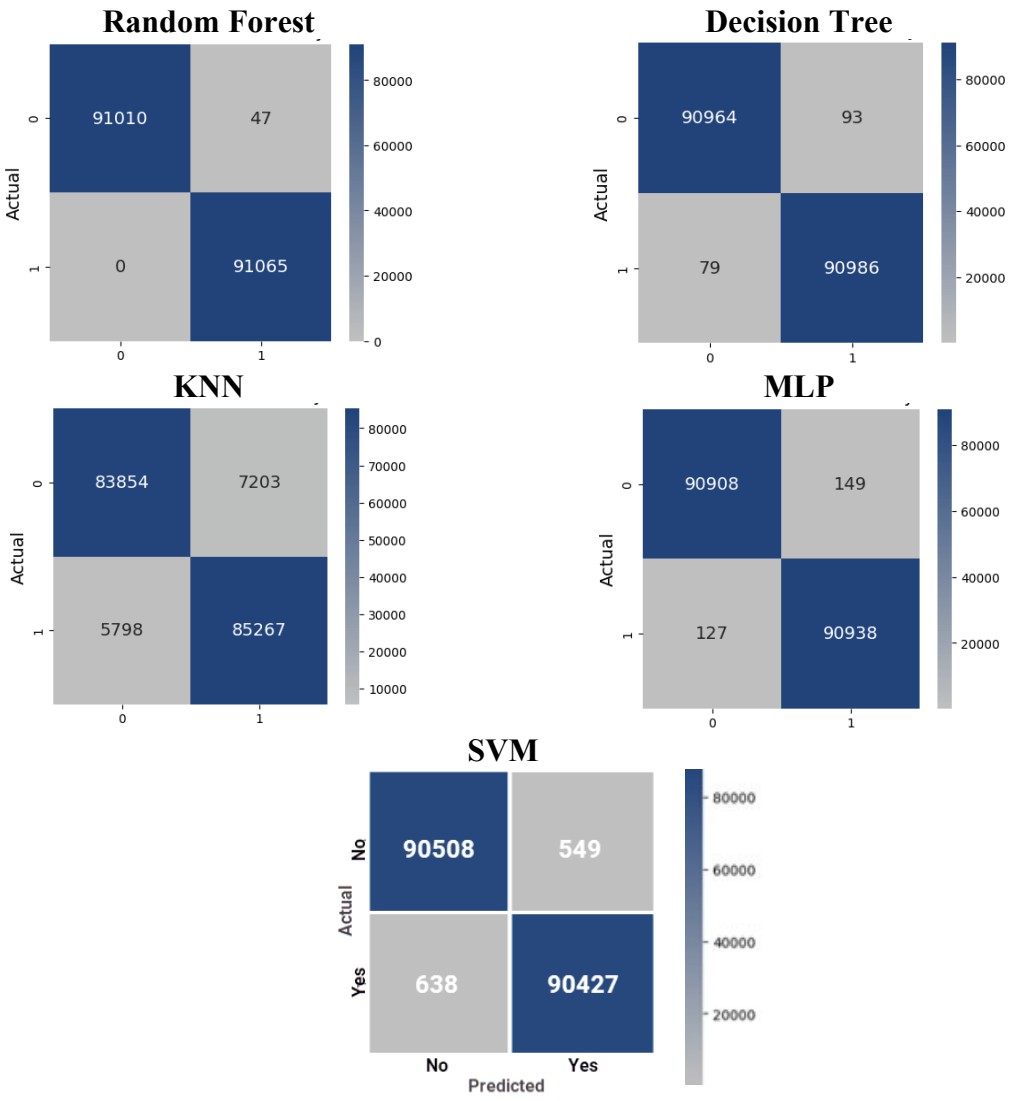


Figure 12: Confusion Matrices: Grid Sybil Dataset

The accuracy, recall, precision and F1-Score evaluation measurements shown in **Table 23,** reveal the Random Forest classifier had the best performance across all the

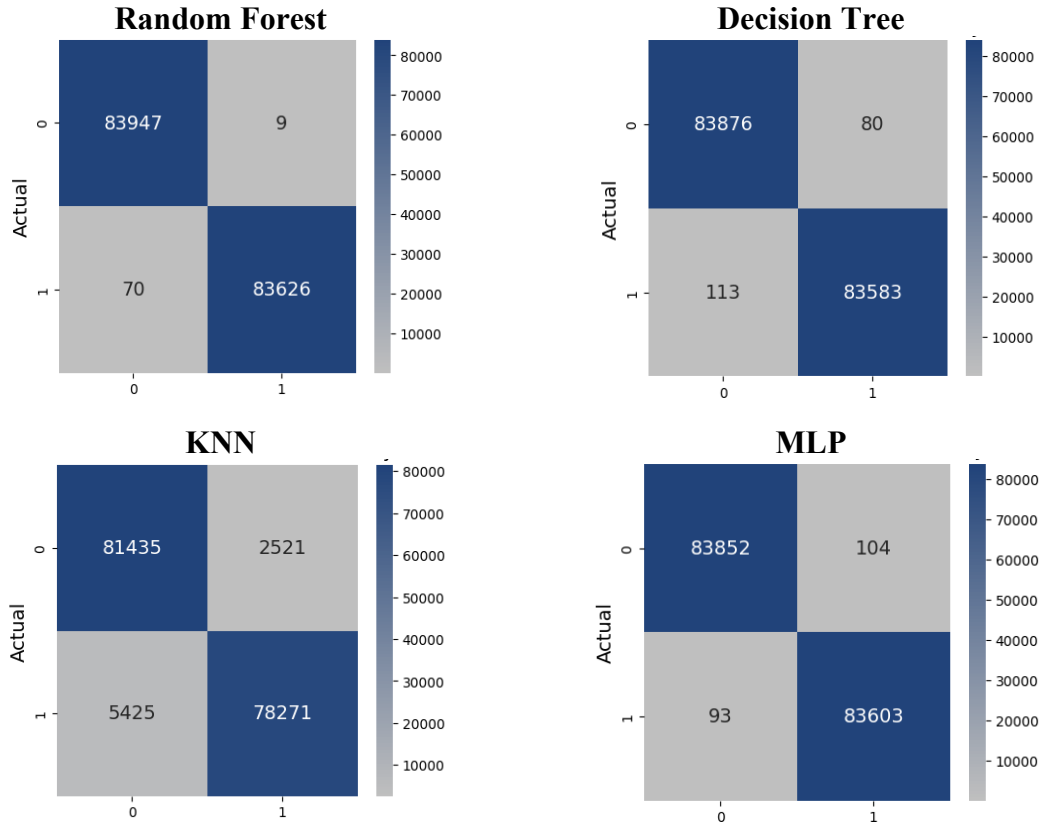
metrics. The KNN classifier was the least performing as it had the highest false positive rate and the lowest accuracy.

Table 23: Classifier Results with Grid Sybil Dataset

Classifier	Accuracy	Recall (TPR)	Precision	F1-Score	FPR	TNR	FNR	Time (min)
Random Forest	**99.97%**	**100.00%**	**99.94%**	**99.97%**	**0.06%**	**99.94%**	**0.001 %**	2.40
Decision Tree	99.91%	99.91%	99.91%	99.91%	0.09%	99.91%	0.09%	0.14
KNN	92.86%	93.63%	92.21%	92.92%	7.91%	92.09%	6.37%	0.37
SVM	99.35%	99.30%	99.40%	99.35%	0.60%	99.40%	0.70%	120.00
MLP	99.85%	99.86%	99.84%	99.85%	0.16%	99.84%	0.14%	2.70

4.4.2 DoS Random Sybil Dataset Results

The confusion matrices by classifier for the DoS Random Sybil dataset are shown in **Figure 13**.



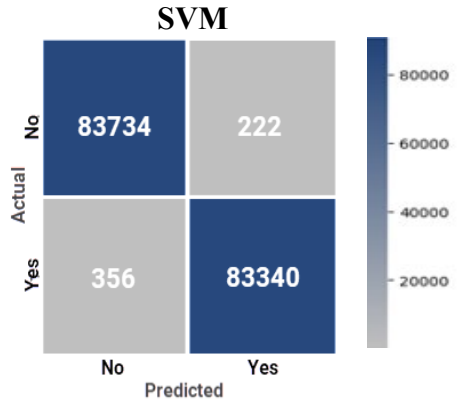


Figure 13: Confusion Matrices: DoS Random Sybil Dataset

The accuracy, recall, precision and F1-Score evaluation measurements shown in **Table 24,** reveal the Random Forest classifier had the best performance across all the metrics. The KNN classifier was the least performing as it had the highest false positive rate and the lowest accuracy.

Table 24: Classifier Results with DoS Random Sybil Dataset

Classifier	Accuracy	Recall (TPR)	Precision	F1-Score	FPR	TNR	FNR	Time (min)
Random Forest	**99.95%**	**99.92%**	**99.99%**	**99.95%**	**0.01%**	**99.99%**	**0.08%**	2.49
Decision Tree	99.88%	99.87%	99.90%	99.89%	0.09%	99.90%	0.13%	0.11
KNN	95.26%	93.52%	96.88%	95.17%	3.00%	97.00%	6.48%	1.29
SVM	99.66%	99.57%	99.73%	99.65%	0.26%	99.74%	0.43%	360.00
MLP	99.88%	99.88%	99.88%	99.88%	0.12%	99.88%	0.11%	6.69

The Decision Tree was a close second, with an accuracy of 99.89% and low false positive rate of 0.09%. The output from the decision tree shows how the position values were used in determining if a message communication was genuine or misbehaving. The position, speed, acceleration features were prominent in the decision making.

4.4.3 Random Position Offset Dataset

The confusion matrices by classifier for the Random Position Offset dataset are shown in **Figure 14.**

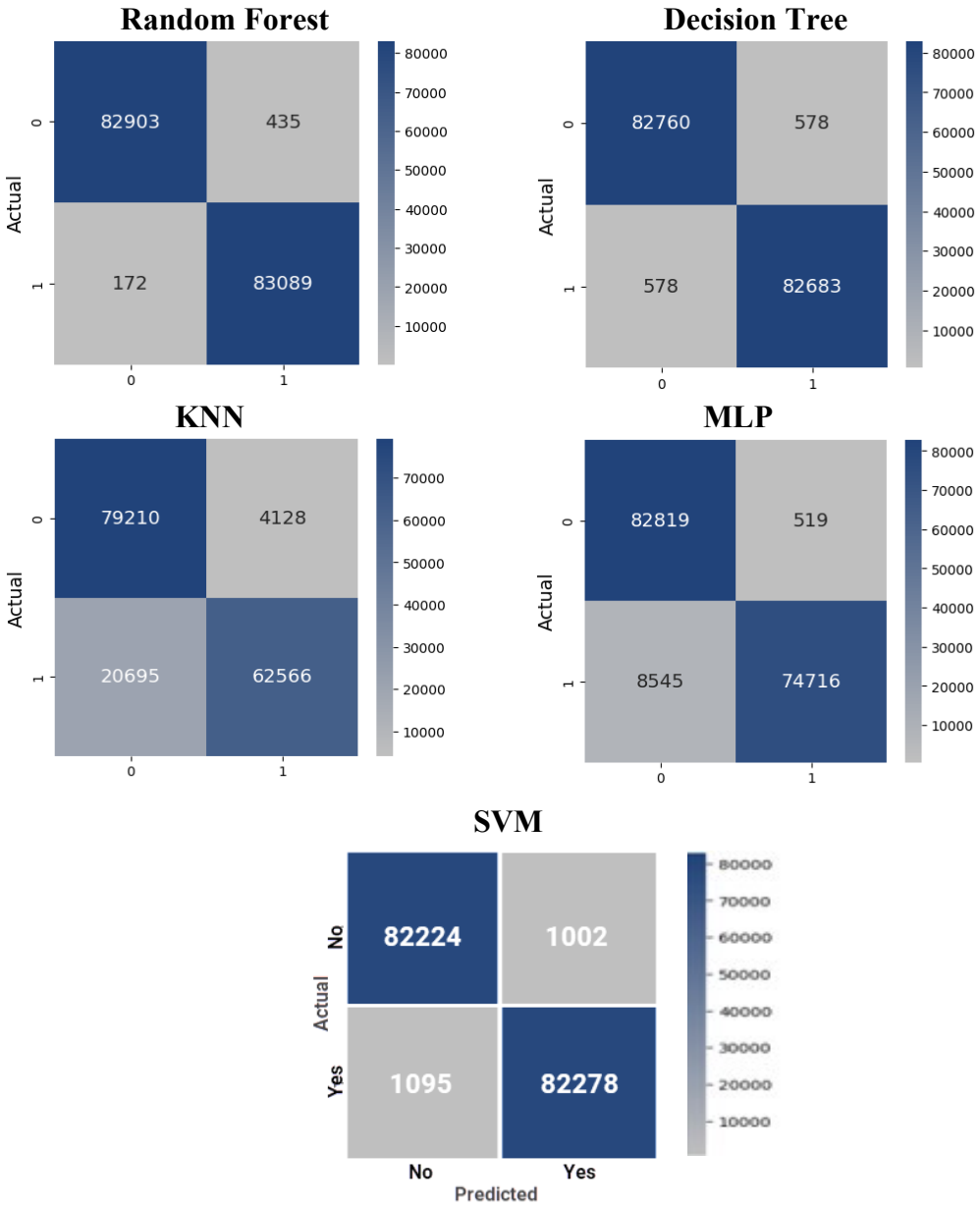


Figure 14: Confusion Matrices: Random Position Offset Dataset

Based on the accuracy, recall, precision and F1-Score, the results depicted in **Table 25** reveal the Random Forest classifier had the best performance across all the metrics. Similar to the Grid Sybil dataset, the KNN classifier was the lowest performer.

Table 25: Classifier Results with Random Position Offset Dataset

Classifier	Accuracy	Recall (TPR)	Precision	F1-Score	FPR	TNR	FNR	Time (min)
Random Forest	**99.64%**	**99.80%**	**99.35%**	**99.57%**	**0.66%**	**99.34%**	**0.20%**	3.79
Decision Tree	99.25%	99.27%	99.23%	99.25%	0.77%	99.22%	0.73%	0.21
KNN	85.10%	75.14%	93.81%	83.45%	4.95%	95.05%	24.86%	0.43
SVM	98.74%	98.69%	98.80%	98.74%	1.20%	98.80%	1.31%	420.00
MLP	98.66%	98.24%	99.07%	98.65%	0.92%	99.07%	1.76%	2.71

4.5 Cross Validation with Random Forest

The Random Forest classifier performed the best across all datasets. To validate the achieved results, cross validation was with 5 folds (k=5) on all three datasets. The accuracy scores, mean scores and 95% confidence intervals are presented in **Table 26.**

Table 26: Random Forest Cross Validation Results

Dataset	Accuracy Score for each Fold					Mean Score	95% CI
	1	2	3	4	5		
Grid Sybil	0.9944	0.9999	0.9999	0.9999	0.9995	0.9987	0.0044
DoS Random Sybil	0.9986	0.9982	0.9999	0.9999	0.9999	0.9993	0.0015
Random POS Offset	0.9618	0.9879	0.9999	0.9978	0.9999	0.9895	0.0291

4.6 Model Results Summary

The praxis used three publicly available VeReMi Extension datasets to train and test the vehicle communication misbehavior classification model. The random forest model achieved the highest metrics scores. The comparison of results between this praxis and studies discussed in the literature review section are detailed in **Tables 27-29**.

Table 27: Random Forest Comparison - Grid Sybil

Random Forest Results - Grid Sybil Dataset				
Research Study	**Accuracy**	**Precision**	**Recall (TPR)**	**F1-Score**
Praxis Study	**99.97%**	**99.94%**	**100.00%**	**99.97%**
Kim et al, 2023 (All Attacks-Classic)	N/A	66.00%	N/A	N/A
Kim et al, 2023 (All Attacks-OVA-BT)	N/A	71.00%	N/A	N/A
Slama et al, 2022 (All Attacks-BCA- M1)	N/A	94.15%	93.53%	93.81%
Slama et al, 2022 (All Attacks-BCA- M2)	N/A	94.08%	93.47%	93.75%
Slama et al, 2022 (All Attacks-BCA- M3)	N/A	94.31%	93.79%	94.03%

Table 28: Random Forest Comparison - DoS Random Sybil

Random Forest Results - DoS Random Sybil Dataset				
Research Study	**Accuracy**	**Precision**	**Recall (TPR)**	**F1-Score**
Praxis Study	**99.95%**	**99.99%**	**99.92%**	**99.95%**
Kim et al, 2023 (All Attacks-Classic)	N/A	66.00%	N/A	N/A
Kim et al, 2023 (All Attacks-OVA-BT)	N/A	71.00%	N/A	N/A
Slama et al, 2022 (All Attacks-BCA- M1)	N/A	94.15%	93.53%	93.81%
Slama et al, 2022 (All Attacks-BCA- M2)	N/A	94.08%	93.47%	93.75%
Slama et al, 2022 (All Attacks-BCA- M3)	N/A	94.31%	93.79%	94.03%

Table 29: Random Forest Comparison - Random Position Offset

Random Forest Results – Random Position Offset Dataset				
Research Study	**Accuracy**	**Precision**	**Recall (TPR)**	**F1-Score**
Praxis Study	**99.64%**			
Kim et al, 2023 (All Attacks-Classic)	N/A	66.00%	N/A	N/A
Kim et al, 2023 (All Attacks-OVA-BT)	N/A	71.00%	N/A	N/A
Sultana et al, 2022 (Without)	81.10%	63.50%	70.10%	63.70%
Sultana et al, 2022 (With)	81.12%	63.60%	70.40%	66.80%
Sultana et al, 2022 (VeReMi Extension)	N/A	97.10%	99.70%	N/A
Sharma et al, 2021 (VeReMi Original Low)	N/A	99.20%	96.70%	97.10%
Sharma et al, 2021 (VeReMi Original Med)	N/A	99.10%	95.20%	97.10%
Slama et al, 2022 (All Attacks-BCA- M1)	N/A	94.15%	93.53%	93.81%
Slama et al, 2022 (All Attacks-BCA- M2)	N/A	94.08%	93.47%	93.75%
Slama et al, 2022 (All Attacks-BCA- M3)	N/A	94.31%	93.79%	94.03%

4.7 Research Questions and Hypotheses Results

This section discusses the results related to the research questions and hypotheses with comparisons to the results featured in the Chapter 2 literature review. The study results for the research questions and hypotheses is summarized in **Tables 30-32.**

Table 30: Summary of Research Questions, Hypotheses and Results A

Research Question # 1
Which set of features are the most significant indicators for determining whether a IoV communication is malicious or safe?
Hypothesis # 1
H1. The position, speed, acceleration, and header values in the message data are significant indicators for malicious IoV communications.
Result
No, the position, speed, acceleration, and header values are not the significant indicators for all attack types represented in this research. The result per dataset: **Grid Sybil** No, the engineered features of sender pseudo ratio and message count per timeframe were the significant indicators. **DoS Random Sybil** No, due to the Sybil component of this attack type, the position, speed, and acceleration are slightly less significant than the engineered features of sender pseudo ratio and message count per timeframe. **Random Position Offset** Yes, the position, speed, and acceleration values in the message data are significant indicators.

Table 31: Summary of Research Questions, Hypotheses and Results B

Research Question # 2	
Can a classification machine model be developed to accurately classify malicious and safe IoV communications?	
Hypothesis	**Results**
H2. A machine model will accurately classify an IoV communication as malicious or safe.	Yes, the machine model achieved high accuracy with classifying an IoV communication as malicious or safe.

Table 32: Summary of Research Questions, Hypotheses and Results C

Research Question # 3	
Which machine learning algorithm can classify IoV communications with the highest accuracy and the lowest false positive and negative rates?	
Hypothesis	**Results**
H3. A classification model using the Random Forest algorithm will achieve an accuracy of at least 95% for classifying an IoV communication as malicious or non-malicious.	Yes, the Random Forest classifier accuracy rates for each dataset are: **Sybil Grid:** 99.98% **DoS Random Sybil:** 99.95% **Random Position Offset:** 99.57%
H4. A classification model using the Random Forest algorithm, will accurately classify an IoV communication as malicious with false positive and negative rates of less than 1%.	Yes, the Random Forest classifier achieved the following: **False Positive Rates** Grid Sybil: 0.05% DoS Random Sybil: 0.01% Random Position Offset: 0.66% **False Negative Rates** Grid Sybil: 0.004% DoS Random Sybil: 0.09% Random Position Offset: 0.20%

Chapter 5 Discussion and Conclusions

5.1 Discussion and Conclusions

Protecting the IoV and VANET infrastructure from external cybersecurity attacks is critical to the growth and safety of connected and autonomous vehicle networks. The specialized communication within the network supports traffic management and accident prevention applications, and the attackers' ability to spread of false information and perform other misbehavior is an increasing vulnerability. Machine learning solutions have shown to be an effective technique for detecting misbehavior in VANETs, but as described in the Chapter 2 literature review section, there is still a gap in achieving high accuracy and low false classification rates across the different types of vehicle communication attacks. Machine learning can play several roles in increasing VANET security. With the ability to learn from historical data, ML can determine what is considered "normal" behavior. Adoption of ML algorithms with the VANET would enable the continuous real time analysis of vehicle communications.

This research praxis tested and confirmed that a machine learning model could be effective in detecting and accurately classifying sybil and position falsification cyber security attacks within the VANET. The praxis model results indicate that the Random Forest model had the best performance with classifying both sybil and position falsification attacks, with average accuracy rates of 99% and low false rates of .01% across all datasets.

The VANET is vulnerable to external attacks due to its technology infrastructure and ease for attackers to appear as authenticated vehicles. As such, Sybil attacks remain a high-risk attack due as current compliance checks cannot detect this malicious

behavior. Machine learning is a proven component to use to strengthen the VANET from cybersecurity attacks. Machine learning algorithms can analyze vehicle communication exchanges in real-time to identify patterns, anomalies, and correlations that humans might miss. The automation of data analysis of the vehicle messages allows for proactive VANET fortification, and more efficient response times when reacting to potential threats. The enhancement in anomaly detection is achieved by machine learning models that learn the baselines of normal communication patterns, allowing for the identification and flagging of activities that diverge from these established norms. Machine learning enhances the efficiency and scalability of cybersecurity analytics, providing the VANET with robust protection amidst the expanding IoV.

The literature review analysis revealed that machine learning research disproportionately focuses on specific VANET attacks, resulting in insufficient attention to other types of attacks. There is also limited data available on the false rates attained in those studies. Consequently, this research focused on employing an applied feature set technique to enhance the classification of attack types and improve the effectiveness of machine learning classifiers. The adopted feature selection strategy of pooling the outcomes of various feature selection techniques, facilitated the identification of the optimal feature sets for each type of attack. Identified, were a core set of features that were applicable to each attack type, with additional features to be included based on attack type, which further enhanced the classification ability of the model. The position, speed, and acceleration values in the message data are significant indicators for the Random Position Offset dataset since this attack types involved falsifying those values within the messages. The position, speed, and acceleration values in the message data are

slightly less significant on their own for the DoS Random Sybil dataset because this attack also involves a sybil component where the sender and number of messages sent during a time gap are important. However, the sender pseudo ratio, number of messages and time gap features had the most significance for the Grid Sybil dataset. Including the sensor data of the vehicle receiving the messages provides a mechanism to compare and analyze the position and speed between the sender and the receiver. A core set of features can be utilized as a first step for detecting and classifying misbehaving messages, with a next step involving the additional feature sets to determine the type of misbehavior.

Government transportation agencies and automobile manufacturers could look into adopting the use of machine learning within the VANET and vehicle network technology to detect and classify misbehavior in vehicle communications using the identified feature sets. The vehicle communication misbehavior classification model can be developed and deployed as a component within the VANET infrastructure to monitory real time vehicle message exchanges. As the messages are logged, data aggregation and historical patterns can be processed real time.

5.3 Contributions to Body of Knowledge

The following contributions are made by this praxis to the field of cybersecurity within vehicle networks:

- A novel approach to identifying sybil attack and position falsification misbehavior in vehicle message communications using a combination of custom preprocessing steps and feature engineering methods on the simulated VeReMi datasets (Kamel et al., 2020). Each dataset went through an

aggregation process which involved time grouping the data and creating derived features to calculate message frequency, sender to senderpseudo ratio and number of messages sent during the frequency period. Second, a custom hybrid approach was used for feature selection, using embedded and wrapper methods used to determine the most relevant features for identifying misbehavior. The combination of the data preprocessing and feature selection technique enabled the ability to detect abnormal patterns in the message data which are otherwise not seen.

- A dataset template for processing vehicle message communication data, which defines the optimum feature sets to use for the sybil and position falsification attack types. Using this dataset template with the Random Forest classifier yields an accurate classification of these misbehavior types, while also achieving low false rates.
- A machine learning model with improved hyperparameter specialization to accurately classify the VeReMi (Kamal et al., 2020) Grid Sybil, Random DoS Sybil and Random Position Offset datasets. Additionally, insight into the patterns and characteristics of these types of attacks is provided and can be used to help refine existing models.

5.5 Recommendations for Future Research

The evolving landscape of vehicular communications, coupled with the increasing sophistication of attackers, necessitates continued research in protecting the VANET infrastructure. More simplistic models were chosen for this praxis due to their ease of interpretability, computational efficiency, and lower proneness to overfitting. As such,

areas of focus for future solutions should include an expanded focus on anomaly detection and irregularities in communication patterns with the adoption of more complex models and solutions involving artificial intelligence. Deep learning methods such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have advanced capabilities that can address cybersecurity issues within the wider IoV infrastructure. RNNs, beneficial for sequential processing and handling tasks where the order of the data points is important, could be trained to model temporal dependencies between consecutive axis data points using vehicle position, speed and header axis data. The model would capture patterns that evolve over time, and be useful for detecting anomalies occurring in the positional data. CNNs known for their advanced capability in processing grid-like image data, would be effective for examining time-series formatted sensor data to capture temporal patterns and dependencies within that data. The trained CNN model could be used to predict future vehicle states or detect anomalies in the sensor data. Including these methods in a hybrid approach, utilizing vehicle imaging, sensor readings and message exchange data could greatly strengthen the approach to protection against external cybersecurity attacks within the IoV.

Another area for future research would be to prototype a detection solution using the vehicle misbehavior classification model where it could be embedded within a simulation environment where the vehicle message exchanges are detected and classified during the simulation run. Additional functionality could be added to determine the disposition of misbehaving messages within the simulation environment.

This research focused on using machine learning to classify the messages based on the position, speed and acceleration features, as well as the engineered features for the

number of pseudo identifiers per sender and number of messages sent during specified time gaps. Traditional models like Random Forest are well suited for the numerical data features of the datasets and provide a feasible balance between efficiency, accuracy and interpretability. The qualities are important in situations where computational resources can be limited, and real-time responses are still required, such as in VANETs .

References

Alladi, T., Agrawal, A., Gera, B., Chamola, V., Sikdar, B., & Guizani, M. (2021). Deep Neural Networks for Securing IoT Enabled Vehicular Ad-Hoc Networks. ICC 2021 - IEEE International Conference on Communications, 1–6. https://doi.org/10.1109/ICC42927.2021.9500823

Bangui, H., Ge, M., & Buhnova, B. (2022). A hybrid machine learning model for intrusion detection in VANET. Computing, 104(3), 503–531. https://doi.org/10.1007/s00607-021-01001-0

Bassiony, I., Hussein, S., & Salama, G. (2022). Detection Approaches for Position Falsification Attack in VANET. 2022 13th International Conference on Electrical Engineering (ICEENG), 86–90. https://doi.org/10.1109/ICEENG49683.2022.9781915

Baza, M., Nabil, M., Mahmoud, M. M. E. A., Bewermeier, N., Fidan, K., Alasmary, W., & Abdallah, M. (2022). Detecting Sybil Attacks Using Proofs of Work and Location in VANETs. IEEE Transactions on Dependable and Secure Computing, 19(1), 39–53. https://doi.org/10.1109/TDSC.2020.2993769

Chen, Y., Lai, Y., Zhang, Z., Li, H., & Wang, Y. (2022). Malicious attack detection based on traffic-flow information fusion. 2022 IFIP Networking Conference (IFIP Networking), 1–9. https://doi.org/10.23919/IFIPNetworking55013.2022.9829793

El-Rewini, Z., Sadatsharan, K., Selvaraj, D. F., Plathottam, S. J., & Ranganathan, P. (2020). Cybersecurity challenges in vehicular communications. Vehicular Communications, 23, 100214.

Ercan, S., Ayaida, M., & Messai, N. (2021). New Features for Position Falsification Detection in VANETs using Machine Learning. ICC 2021 - IEEE International Conference on Communications, 1–6. https://doi.org/10.1109/ICC42927.2021.9500411

Faisal, S. M., Kumar Gupta, B., & Zaidi, T. (2022). A hybrid framework to prevent VANET from Sybil Attack. 2022 5th International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), 1–6. https://doi.org/10.1109/IMPACT55510.2022.10029103

Gu, P., Khatoun, R., Begriche, Y., & Serhrouchni, A. (2017). Support Vector Machine (SVM) Based Sybil Attack Detection in Vehicular Networks. 2017 IEEE Wireless Communications and Networking Conference (WCNC), 1–6. https://doi.org/10.1109/WCNC.2017.7925783

Guan, T., Han, Y., Kang, N., Tang, N., Chen, X., & Wang, S. (2022). An Overview of Vehicular Cybersecurity for Intelligent Connected Vehicles. Sustainability, 14(9), 5211. MDPI AG. Retrieved from http://dx.doi.org/10.3390/su14095211

Hamdi, M. M., Dhafer, M., Mustafa, A. S., Rashid, S. A., Ahmed, A. J., & Shantaf, A. M. (2022). Effect Sybil attack on security Authentication Service in VANET. 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 1–6. https://doi.org/10.1109/HORA55278.2022.9799810

Hashem Eiza, M., & Qiang Ni. (2017). Driving with Sharks: Rethinking Connected Vehicles with Vehicle Cybersecurity. In IEEE vehicular technology

magazine (Vol. 12, Issue 2, pp. 45–51). IEEE. https://doi.org/10.1109/MVT.2017.2669348

Ismath, I., Samarasinghe, T., Dias, D., Wimalarathna, M., Rasanga, W., Jayaweera, N., & Nugera, Y. (2019). Emergency Vehicle Traversal using DSRC/WAVE based Vehicular Communication. *2019 IEEE Intelligent Vehicles Symposium (IV)*, 1981–1986. https://doi.org/10.1109/IVS.2019.8813981

Iqbal, S., Ball, P., Kamarudin, M. H., & Bradley, A. (2022). Simulating Malicious Attacks on VANETs for Connected and Autonomous Vehicle Cybersecurity: A Machine Learning Dataset. https://doi.org/10.48550/arxiv.2202.07704

Iyengar, S., & Iyengar, S. S. (2010). Fundamentals of sensor network programming : applications and technology (1st edition). Wiley. https://doi.org/10.1002/9780470890158

Kamel, J. (2020). (https://github.com/josephkamel/VeReMi-Dataset/issues/7).

Kamel, J., Haidar, F., Jemaa, I. B., Kaiser, A., Lonc, B., & Urien, P. (2019). A Misbehavior Authority System for Sybil Attack Detection in C-ITS. 2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 1117–1123. https://doi.org/10.1109/UEMCON47517.2019.8993045

Kamel, J., Jemaa, I. B., Kaiser, A., Cantat, L., and Urien, P. (2019). Misbehavior detection in c-its: A comparative approach of local detection mechanisms", IEEE Vehicular Networking Conference (VNC), pp. 1-8, 2019.

Kamel, J., Wolf, M., Van Der Hei, R. W., Kaiser, A., Urien, P., & Kargl, F. (2020, June). Veremi extension: A dataset for comparable evaluation of misbehavior detection

in vanets. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-6). IEEE.

Kim, C., Chang, S.-Y., Lee, D., Kim, J., Park, K., & Kim, J. (2023). Reliable Detection of Location Spoofing and Variation Attacks. IEEE Access, 11, 10813–10825. https://doi.org/10.1109/ACCESS.2023.3241236

Kohli, V., Chougule, A., Chamola, V., & Yu, F. R. (2022). MbRE IDS: An AI and Edge Computing Empowered Framework for Securing Intelligent Transportation Systems. IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 1–6. https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9798390

Kostrzewski, M., Marczewska, M., & Uden, L. (2023). The Internet of Vehicles and Sustainability—Reflections on Environmental, Social, and Corporate Governance. *Energies*, *16*(7), 3208. https://doi.org/10.3390/en16073208

Krishna, K. V., & Reddy, K. G. (2022). VANET Vulnerabilities Classification and Countermeasures: A Review. Majlesi Journal of Electrical Engineering, 16(3), 63–83. https://doi.org/10.52547/mjee.16.3.63

Kushardianto, N. C., Hillali, Y. E., & Tatkeu, C. (2021). 2-Step Prediction for Detecting Attacker in Vehicle to Vehicle Communication. 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 1–5. https://doi.org/10.1109/VTC2021-Fall52928.2021.9625513

Liu, J., Khattak, A. (2016). Delivering Improved Alerts, Warnings, and Control Assistance Using Basic Safety Messages Transmitted between Connected

Vehicles. Transportation research. Part C, Emerging technologies 68: 83–100. Web

Liu, Y., Xue, H., Zhuang, W., Wang, F., Xu, L., & Yin, G. (2022). CT2-MDS: Cooperative trust-aware tolerant misbehaviour detection system for connected and automated vehicles. IET Intelligent Transport Systems, 16(2), 218–231. https://doi.org/10.1049/itr2.12139

Mohamed, T. M., Ahmed, I. Z., & Sadek, R. A. (2021). Efficient VANET safety message delivery and authenticity with privacy preservation. *PeerJ. Computer Science*, *7*, e519–e519. https://doi.org/10.7717/peerj-cs.519

Naresh, V. S., & Reddi, S. (2023). An identity-based secure VANET communication system. Security and Privacy. https://doi.org/10.1002/spy2.349

Noman, S. A., & Atkison, T. (2023, March). Techniques to Overcome Network Attacks (Sybil Attack, Jamming Attack, Timing Attack) in VANET. In Journal of The Colloquium for Information Systems Security Education (Vol. 10, No. 1, pp. 7-7).

Nilsson, D. K., Larson, U. E., Picasso, F., & Jonsson, E. (2009). A first simulation of attacks in the automotive network communications protocol flexray. In *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08* (pp. 84-91). Springer Berlin Heidelberg.

Pengwenlong Gu, Khatoun, R., Begriche, Y., & Serhrouchni, A. (2017). k-Nearest Neighbours classification based Sybil attack detection in Vehicular networks. 2017 Third International Conference on Mobile and Secure Services (MobiSecServ), 1–6. https://doi.org/10.1109/MOBISECSERV.2017.7886565

Reddy, D. S., Bapuji, V., Govardhan, A., & Sarma, S. S. V. N. (2017, February). Sybil attack detection technique using session key certificate in vehicular ad hoc networks. In 2017 international conference on algorithms, methodology, models and applications in emerging technologies (ICAMMAET) (pp. 1-5). IEEE.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN computer science, 2(3), 160.

Sangapu, S. C., Prasad, K. S. N., Kannan, R. J., Chen, T. M., & Sathiyanarayanan, M. (2023). Impact of class imbalance in VeReMi dataset for misbehavior detection in autonomous vehicles. Soft Computing (Berlin, Germany). https://doi.org/10.1007/s00500-023-08003-4

Sharma, A. (2021). Position Falsification Detection in VANET with Consecutive BSM Approach Using Machine Learning Algorithm. ProQuest Dissertations Publishing.

Sharma, A., & Jaekel, A. (2021). Machine learning based misbehaviour detection in VANET using consecutive BSM approach. IEEE Open Journal of Vehicular Technology, 3, 1-14.

Sharma, P., & Liu, H. (2021). A Machine-Learning-Based Data-Centric Misbehavior Detection Model for Internet of Vehicles. IEEE Internet of Things Journal, 8(6), 4991–4999. https://doi.org/10.1109/JIOT.2020.3035035

Slama, O., Alaya, B., & Zidi, S. (2022). Towards Misbehavior Intelligent Detection Using Guided Machine Learning in Vehicular Ad-hoc Networks (VANET). Inteligencia Artificial, 25(70), 138–154. https://doi.org/10.4114/intartif.vol25iss70pp138-154

Slama, O., Tarhouni, M., Zidi, S., & Alaya, B. (2023). One Versus All Binary Tree Method to Classify Misbehaviors in Imbalanced VeReMi Dataset. IEEE Access, 11, 135944–135958. https://doi.org/10.1109/ACCESS.2023.3337378

So, S., Petit, J., & Starobinski, D. (2019). Physical layer plausibility checks for misbehavior detection in V2X networks. Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks, 84–93. https://doi.org/10.1145/3317549.3323406

So, S., Sharma, P., & Petit, J. (2018). Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 564–571. https://doi.org/10.1109/ICMLA.2018.00091

Sommer, C., German, R., & Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. IEEE Transactions on Mobile Computing, 10(1), 3–15. https://doi.org/10.1109/TMC.2010.133

Soni, M., & Jain, A. (2018). Secure Communication and Implementation Technique for Sybil Attack in Vehicular Ad-Hoc Networks. 2018 Second International Conference on Computing Methodologies and Communication (ICCMC), 539–543. https://doi.org/10.1109/ICCMC.2018.8487887

Sultana, R., Grover, J., Meghwal, J., & Tripathi, M. (2022). Exploiting machine learning and deep learning models for misbehavior detection in VANET. International Journal of Computers & Applications, 44(11), 1024–1038. https://doi.org/10.1080/1206212X.2022.2099122

U.S. Department of Transportation. (2020). How Connected Vehicles Work, https://www.transportation.gov/research-and-technology/how-connected-vehicles-work, Thursday, February 27, 2020)

Van Der Heijden, R. W., Lukaseder, T., & Kargl, F. (2018). Veremi: A dataset for comparable evaluation of misbehavior detection in vanets. In Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, Singapore, Singapore, August 8-10, 2018, Proceedings, Part I (pp. 318-337). Springer International Publishing.

Wang, J., Liu, J., and Kato, N. (2019). Networking and Communications in Autonomous Driving: A Survey. *IEEE Communications surveys and tutorials* 21.2: 1243–1274. Web.

Xu, X., Wang, Y., & Wang, P. (2022). Comprehensive review on misbehavior detection for vehicular ad hoc networks. Journal of Advanced Transportation, 2022.

Zang, M., & Yan, Y. (2021, April). Machine learning-based intrusion detection system for big data analytics in VANET. In 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring) (pp. 1-5). IEEE.

Appendix A

Table A-1: ML Output - 70/30 Train/Test Split

Grid Sybil Dataset	DoS Random Sybil Dataset	Random Position Offset Table
Random Forest Accuracy: 0.9997 TN: 91010 FP: 47 FN: 0 TP: 91065 Processing Time: 2.372	**Random Forest** Accuracy: 0.9995 TN: 83945 FP: 11 FN: 71 TP: 83625 Processing Time: 2.497	**Random Forest** Accuracy: 0.9964 TN: 82903 FP: 435 FN: 172 TP: 83089 Processing Time: 3.793
- - - - - - - - - - - - -	- - - - - - - - - - -	- - - - - - - - - - - - -
KNN Accuracy: 0.9286 TN: 83854 FP: 7203 FN: 5798 TP: 85267 Processing Time:0.367	**KNN** Accuracy: 0.9526 TN: 81435 FP: 2521 FN: 5425 TP: 78271 Processing Time: 1.291	**KNN** Accuracy: 0.8510 TN: 79210 FP: 4128 FN: 20695 TP: 62566 Processing Time: 0.432
- - - - - - - - - - - - -	- - - - - - - - - - -	- - - - - - - - - - - - -
Decision Tree Accuracy: 0.9991 TN: 90964 FP: 93 FN: 79 TP: 90986 Processing Time: 0.137	**Decision Tree** Accuracy: 0.9988 TN: 83876 FP: 80 FN: 115 TP: 83581 Processing Time: 0.111	Decision Tree Accuracy: 0.9925 TN: 82850 FP: 646 FN: 646 TP: 82764 Processing Time: 0.213
- - - - - - - - - - - - -	- - - - - - - - - - -	- - - - - - - - - - - - -
MLP Accuracy: 0.9985 TN: 90908 FP: 149 FN: 127 TP: 90938 Processing Time: 2.70	**MLP** Accuracy: 0.9988 TN: 83852 FP: 104 FN: 93 TP: 83603 Processing Time: 6.69	**MLP** Accuracy: 0.9866 TN: 82454 FP: 772 FN: 1464 TP: 81909 Processing Time: 2.717

Table A- 2: List of Selected Dataset Features

Feature Name	Feature Type/Origin		
	Engineered Feature	Receiver Sensor Data	Sender Message Data
timeGap	X		
cntSenderPseudo	X		
pseudoRatio	X		
cntMsgSent	X		
avgSensPosX		X	
avgSensSpdX		X	
avgSensSpdY		X	
avgSensHedY		X	
avgPosX			X
avgPosY			X
avgSpdX			X
avgSpdY			X
avgAclX			X
avgAclY			X
avgHedX			X
avgHedY			X