FACULTY OF SCIENCE, TECHNOLOGY AND MEDICINE

Misbehavior Detection System for Position Falsification Attacks Detection in Vehicular Network

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Information and Computer Sciences

Author:
Faisal HAWLADER

Supervisor:
Prof. Dr. Thomas ENGEL

Reviewer:
Prof. Dr. Ulrich SORGER

Advisor:
Dr. Abdelwahab BOUALOUACHE

September 01, 2020

Abstract

Vehicular networks provide useful functionalities based on the data exchanged over the network that are not encrypted to reduce the computational complexity and time. Without a mechanism that guarantees authentication, and data integrity leaves the system open to endless security threats. A Public Key Infrastructure (PKI) is consistently used throughout the literature, which only works to avoid external attacks. However, the networks are also vulnerable to many types of internal attacks, including position falsification attacks. The position falsification attack can lead to hazardous situations for traffic efficiency management applications, and increase the possibility of collisions. Therefore, the vehicular network highly requires the deployment of additional security mechanisms. The misbehavior detection system is considered a possible solution to detect position falsification attacks. Numerous misbehavior detection systems have been proposed to thwart these attacks. However, the existing solutions suffer from a lack of accuracy and dynamicity and require extensive information to detect position falsification attacks.

To overcome these limitations, we propose a novel dynamic system that integrates a Machine Learning (ML) model and Software-defined Networking (SDN) c to detect attacks associated with position falsification. Our system leverages an efficient ML-model which only uses vehicles' positions as input to detect the attacks. In our system design, a central SDN controller is responsible for installing, updating, and deploying the ML-model on each vehicle. In addition, we design a report investigation system that enables collaboration between vehicles for increasing the accuracy of detecting attacks. Finally, we evaluate the performance of our system using several detection metrics to assess its ability for identifying different types of position falsification attacks precisely. Obtained results show that our proposed system is able to detect position falsification attacks with almost 100% of accuracy.

Keywords — Vehicular network, software define network, machine learning, security, misbehavior detection, position falsification attack.

Acknowledgements

This section is dedicated to everyone who has encouraged me to make my MICS studies one of the most rewarding periods of life. This thesis was conducted at the SECAN-Lab, Interdisciplinary Centre for Security Reliability and Trust (SnT), University of Luxembourg.

First and foremost, I would like to thank my advisor Dr. Abdelwahab Boualouache for letting me conduct my research, allowing me to freely explore, identify, and investigate my topics, while always being supportive. For taking the time to supervise my progress, review the manuscript on a continuous basis, and provide the much-needed criticism. Without his support, this dissertation would not have been possible.

I want to thank Dr. Riha SOUA for finding the advisor, and helping to decide the thesis. He has been very supportive since I have stated my student job with SECAN-Lab.

I would also like to thank Prof. Dr. Thomas Engle for agreed to be my supervisor and allowing me to work with SECAN-Lab, also for his help to find a reviewer. I want to thank Prof. Dr. Ulrich SORGER for being the reviewer, and his much-needed advice during the MICS journey, always advised me to make the right decision despite his business.

Finally, I would like to thank all my colleagues at SECAN-Lab, MICS, and the University of Luxembourg, who made works so enjoyable. Most of all, I want to thank my family and friends, who had always supported my efforts and understanding when things were not going as planned.

Acronyms

VANET Vehicular ad-hoc Network

ITS Intelligent Transportation Systems

MANET Mobile ad-hoc Network

DSRC Dedicated Short Range Communications

OBU On-Board Units

RSU Road Side Units

ISP Internet Service Providers

V2V Vehicle to Vehicle

V2I Vehicle to Infrastructure

ML Machine Learning

SDN Software Define Network

PKI Public Key Infrastructure

WHO World Health Organization

GPS Global Positioning System

MDS Misbehavior Detection System

AI Artificial Intelligent

CA Certificate Authority

MIB Management Information Base

C-V2X Cellular Vehicle to Everything

DoS Denial of Service Attacks

DDoS Distributed Denial of Service

AODV Ad Hoc on Demand Distance Vector

DSR Dynamic Source Routing

TSC Traffic Signal Control

ICT Information and Communication Technologies

US United States

DNN Deep Neural Networks

IoT	Internet of Things
BM	Bookmaker Informedness
LuST	Luxembourg SUMO traffic scenario
RSSI	received signal strength indicator
SVM	Support Vector Machine
DT	Decision Tree
LR	Logistic Regression
RF	Random Forest
KNN	K-nearest neighbour
NB	Naive Bayes

Contents

List of Figures

List of Tables

Introduction

The evolution of communication technologies, together with the variety of network access mediums and service providers, has led to the birth of Intelligent Transportation Systems (ITS). An ITS is an advanced application which aims to provide innovative services relating to transport and traffic management. The development of ITS has made a big step, which can enable users to be better informed and make their life safer on the road.

According to the World Health Organization (WHO) [1], road accidents annually cause around 1.2 million deaths worldwide. Another 20 to 50 million more people suffer non-fatal wounds, including many acquiring a disability due to the injury. If we do not take the preventive measures inside the account, street death is likely to become the third leading reason for death in 2022 from the ninth position in 1990 [2]. However, the total number of deaths produced by car collisions or other road incidents is avoidable in principle. According to the investigation of the U.S. Department of Transportation, roadway departures and intersection-related conflicts [3] cause 21,000 to 43,000 of the annual deaths barely in the U.S. This number of death can be significantly reduced by deploying local warning through the vehicular system.

On the other hand, the number of vehicles has risen dramatically in recent times. Vehicles, capable of achieving high speeds, are found jam-packed on roads during peak traffic hours in almost all big cities of the world. A small road maintenance task or accident can result in massive traffic jams and further accidents. Watchful information from surrounding vehicle positions could be vital in these cases to improve the driver's and passengers' safety on board. Meanwhile, vehicular technology has seen tremendous advancements in recent years, where the vehicles are equipped with multiple computing and sensing devices [4]. These devices collect vehicle statistics such as speed, GPS location [5] to assist the driver actively. It is expected that by 2026, most vehicles will be network-enabled and convert to a wireless node of a big dynamic network.

Vehicular networking is entirely based on data transmission to ensure road safety, which is also an open network. The safety messages exchanged through the network is not encrypted to reduce the computational complexity and overhead. This leaves the system open to endless security threats without

a system that guarantees sender authentication and data integrity. A public key infrastructure (PKI) is consistently used throughout the literature to solve the issue, PKI is a popular cryptographic technique, enables participants to communicate on an insecure public network securely, and reliably verify the identity of the participants via digital signatures, only works to avoid external attacks. However, the networks are also vulnerable to many types of internal attacks, including position falsification attack, which can lead the system to a hazardous situation, especially can destroy the road traffic management and efficiency of safety application. Therefore, securing communication against any position falsification attack has become a fundamental requirement. To address these issues, we propose a novel scheme based on machine learning and software-defined network (SDN) to detect position falsification attacks accurately. In our system design, a central SDN controller is considered responsible for installing, updating, and deploying the machine learning model on each vehicle, which can be controlled dynamically. The local exposure will be verified and enhanced by executing a collaborative report investigation system. We summarize the main contributions of the thesis below:

- We provide a synthesis of the exiting work associated with position falsification attack and identify their limitations.
- We propose a dynamic novel misbehavior detection system with the combination of SDN and machine learning model that overcomes the existing limitations.
- We propose a novel method for features extraction based on vehicles' positions and develop an accurate multi-class classifier for detecting different types of position falsification attacks.
- We assess the performance of our system for detecting position falsification attacks using several metrics.

The remainder of the thesis is as follows: chapter 1 describes the background literature, architecture, and communication standards, security requirements, and possible attacks on vehicular networks. Chapter 2 presents misbehavior detection systems and overviews existing mechanisms that have been proposed to detect position falsification attacks in vehicular networks. Chapter 3 describes our proposed misbehavior detection system. Moreover, chapter 4 validates our system through publicly available data sets and simulations as well. Finally, the thesis ends with a conclusion followed by some future directions.

CHAPTER 1

Background

1.1 Introduction

Today, the fleet is growing, and the roads are becoming more dangerous because of congestion, which increases the likelihood of collisions. Meanwhile, vehicular technology, usually developed as a part of ITS, has shown tremendous advantages in recent years. It is an emerging network that can effectively improve road safety, transportation efficiency, and passenger comfort, which are the main motivations of the vehicular networking systems. Because of the high mobility of vehicles, it exhibits several unique features and characteristics such as delay-tolerant network, dynamic topology due to the sensitive data exchanged, and the specific characteristics. The network could be subject to a variety of cyber-threats in principle; it could connect to an untrusted network. Hence, the security of vehicular networks has got paramount research importance to ensure road safety.

This chapter designs an illustration of the architecture with existing communication technology, characteristics, and applications of the vehicular network. Nevertheless, vehicular networks must maintain initial sets of standard requirements. We summarized the security specifications that must be considered and the most significant attacks that can be performed to violate these requirements. We also provide fundamental concepts of Artificial Intelligent (AI) and Software Define Network (SDN) since our future goal is to introduce a novel approach by using these two promising technologies.

The rest of this chapter organized as follows: Section 1.2 discusses the architecture, characteristics, communications technologies, and available services of vehicular networks. Section 1.3 presents the security requirements and attacker models. Section 1.4 presents machine learning and different kinds of learning processes. Section 1.5 discusses the architectures, benefits of software-defined networks for the vehicular network, and finally, section 1.6 concludes the chapter.

1.2 Vehicular Networks

In the vehicular communication system, vehicles and road side units are the primary communicating nodes, providing each other information, such as safety warnings, traffic updates, and accident notifications, where a vehicle is equipped with various communication technologies such as cellular network and IEEE 802.11p [6] to communicate with others.

1.2.1 Architecture

The architecture of vehicular networks involves various hardware and software components, as shown in Fig.-1.1. In principle, the entire vehicular system's design consists of three entities: the Certificate Authority (CA), the Road Side Units (RSU) and the vehicles illustrated below:

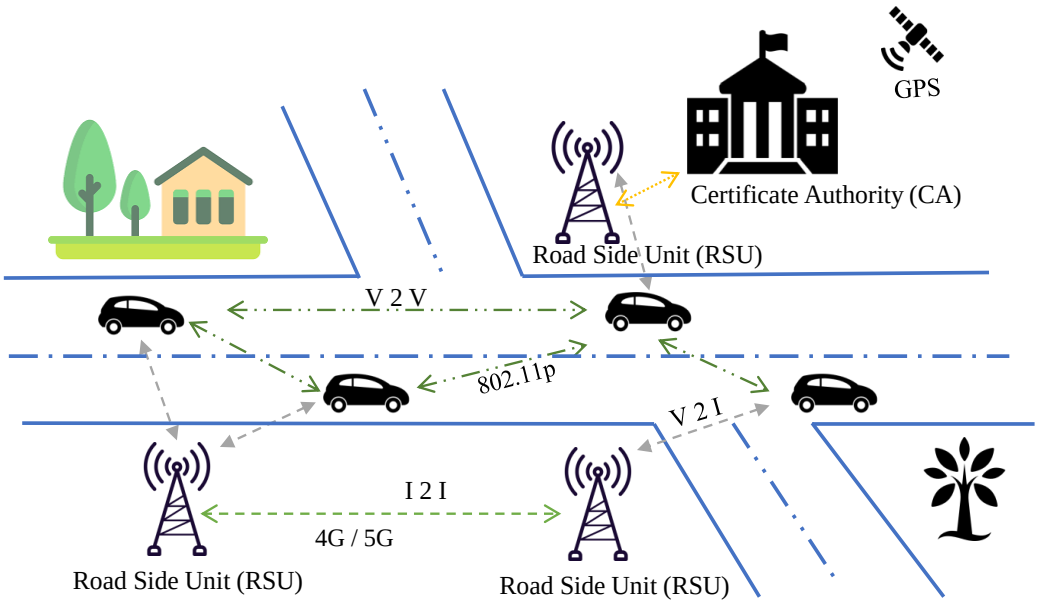

FIGURE 1.1. Different Components of Vehicular Network.

(1) **Certificate authority (CA):** is the only authentic credential verifier for the whole network consider as a control center, is also responsible for vehicle registration, key management, verification, and other significant operations [7].

(2) **On-board unit (OBU):** are intelligent functional units equipped with vehicles for interaction with RSUs or other vehicles [8] to communicate each others.

(3) **Road-Side Unit (RSU):** are computing infrastructure located on the roadside that provides connectivity support to passing vehicles.

In vehicular network, there are three types of communication: Vehicle to Vehicle (V2V) communication, Vehicle to Infrastructure (V2I) communication and Hybrid communications.

(A) **Vehicle to Vehicle Communication (V2V):** In today's vehicles, sensors carry out essential functions with the help of radar, and camera technologies allow vehicles to see and analyze their surroundings to make safe decisions while on board. A vehicle can communicate valuable data directly to other vehicles. Any relevant information gathered on a vehicle or transmitted to the vehicle can be sent to nearby vehicles, known as V2V communication. However, sensors have limited transmission range, particularly when it comes to hidden objects on-road and generally unexpected behavior from other vehicles, V2V communication aims to correct these limitations by letting cars speak with one another directly.

(B) **Vehicle to Infrastructure Communication (V2I):** The exchange of information between any onboard units (car, bus, trucks) and other roadside infrastructure (traffic signals, line markings) are known as V2I Communication. The vehicles use support from road site infrastructure to connect the service providers.

(C) **Hybrid communications:** this communications is a combination of V2V and V2I, and the primary motivation is to extend the existing services. The idea behind this technology is to make the vehicle able to communicate with its surroundings in real-time.

The main features of hybrid communication are followings:

- Informing the autonomous vehicles about sight vehicles.
- Warning distracted pedestrians of oncoming traffic.
- Delivering alerts for weather and road conditions to drivers

1.2.2 Characteristics

The communication on vehicular networks is wireless, where the high mobility vehicle is the central communicating node and implies unique individual characteristics [9]; these characteristics are:

- **Variable Network Density:** The network density varies depending on the location and the pick or off pick hours. Network density is a crucial parameter, it is highly variable in time and often deciding factor to consider while calculating the delay and overhead [10].
- **Dynamic Topology and Large Scale Network:** The architecture involves high mobility, the vehicle speed led the topology to change rapidly, and the connection times are also short,

especially between nodes moving in opposite directions. The highly dynamic network topology together the short connection times makes things challenging. However, the network scale depends on traffic density, with the tremendous growth of vehicle [11].

- **Energy storage and computing:** Since the vehicles can provide continuous power, and the system do not suffer energy problem, computing capacity, or storage failure like other mobile networks. However, the real-time operation of an outsized amount of information may be a challenge. Moreover, the nodes are often equipped with an adequate number of sensors and a Global Positioning System (GPS). They help to get reliable wireless communication and acquires accurate information regarding its current position, speed, and direction [12]; sometimes, these resources may move up to the computational capacity.
- **Scalability:** Especially in an extensive distributed network, scalability is a crucial aspect, can be defined as the ability to add additional participants without noticeable performance loss. In the vehicular networking scenario, scalability issues arise in several contexts, since the number of active nodes has ethical impacts on the network bandwidth [13].

1.2.3 Standards and communications technologies

The exchange of messages with information could be challenging due to the unique characteristics of the vehicular network, and we must need a set of standards to assure the best secure communication, and several standards have been proposed [14]. The standards vary country-wise, significant standards stacks are taking place in the United States (US), Europe, and Japan, because of the corresponding dominance in the automotive industry [15]. The summary of the most reliable standards is as in Fig.-1.2.

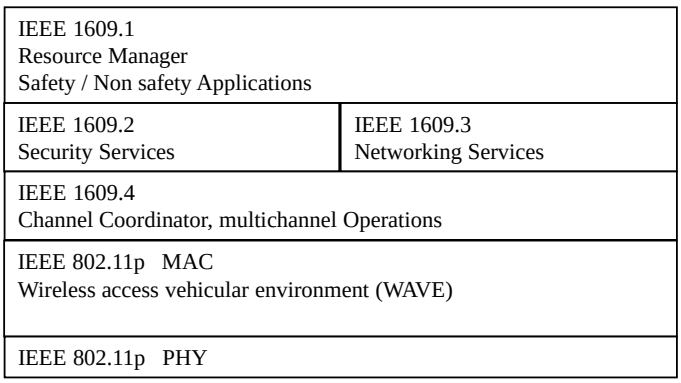

FIGURE 1.2. Wireless Access Standards for Vehicular Network.

- **IEEE 1609 Family:** According to the standards sheets published by IEEE [16], the 1609 family defines the communication model, network security, and physical access management to obtain low latency wireless communications in the vehicular network, together they provide the foundation of the networking environment with the following set of protocol:
  - **IEEE 1609.1 (Resource Manager):** This standard describes the basic components such as storage data formats and device types that can be supported by the on-board unit.
  - **IEEE 1609.2 (Security Services for Applications):** This standard describes some methods for securing application messages and the functions necessary to support the security of the vehicle's messages and privacy.
  - **IEEE 1609.3 (Networking Services):** This standard describes services for the network; these services include routing and addressing secure data exchange. Besides, IEEE 1609.3 defines the Management Information Base (MIB) (is a database used for managing the entities in a communication network) for wireless access.
  - **IEEE 1609.4 (Multi-Channel Operations):** It specifies interval timers, priority access parameters, control channels, and service channel operations. It also defines management services, channel routing, and switching parameters [17].
- **IEEE 802.11p:** is an approved amendment to the IEEE 802.11 [18] standard to add wireless access in vehicular environments is required to support ITS applications [19], including data exchange between high-speed vehicles and other communicating nodes. IEEE 80211p is also the basis of a European standard for vehicular communication known as ETSI ITS-G5 [20].
- **Dedicated Short Range Communication (DSRC):** is an extension of 802.11p-based wireless communication technology that enables vehicles to communicate with each other directly without involving any infrastructure [21]. A dedicated 75MHz spectrum in the band 5.9 GHz, was allocated for Dedicated Short Range Communications (DSRC) to provide low latency information exchange with maximal cybersecurity, can even handle a fast-changing environment at speeds as high as 500 km/h and in extreme weather conditions [22].

In Europe, to replace the US promoted DSRC, Cellular Vehicle to Everything (C-V2X) was developed within the 3rd Generation Partnership Project (3GPP) [23].

C-V2X 5G communication: C-V2X wireless communication technology has been one of the notable services for 5G [24]. For C-V2X transmission mode, there is a newly defined communication channel (sidelink) that can support direct C-V2X communication. Direct C-V2X communication is a technology that allows vehicles to interact and share safety-related data with other participates instantly without going through infrastructure support [25].

1.2.4 Services and applications

The vehicular network supports a wide range of promising applications and services are typically classified in *(i)* active road **safety** applications *(ii)* traffic efficiency and **management** applications and *(iii)* **comfort** and **infotainment** applications [26].

i. Safety Applications: According to the World Health Organization (WHO) [1], road accidents annually cause approximately 1.2 million deaths worldwide, and it is expected that till 2022 road accidents will become the third cause of death. However, this number of death can be significantly reduced by deploying safety applications (local warnings) through the vehicular network. For example, the departing vehicles can inform other vehicles that they intend to depart the highway, and arriving cars at intersections can send warning messages to other vehicles traversing that intersection. Safety applications are always paramount to reduce the number of accidents significantly, and the main focus is to avoid accidents from happening in the first place [27].

ii. Traffic Efficiency and Management Applications: Traffic monitoring and management are essential to maximize road capacity and avoid traffic congestion. Crossing intersections in city streets can be tricky and dangerous at times. Traffic light scheduling can facilitate drivers to cross intersections. Allowing a smooth flow of traffic can significantly increase vehicle throughput and reduce travel time.

iii. Comfort and Infotainment Application: In general, comfort and infotainment applications are motivated to provide comforts of the driver and passengers, essentially provides services such as mobile internet access, messaging, a discussion between vehicles, collaborative network games, and various traveler information. For instance, the driver could receive local information regarding restaurants, hotels using vehicular network services.

1.3 Security for Vehicular Networks

Though vehicular networks are the most prominent enabling network technology for intelligent transportation systems that provide many new exciting facilities, still security features remain a significant concern in deployment to mitigate possible attack vectors. Standardization agencies have developed an initial standard for practical implementation that serves as the basis. This section aims to describe the security requirements, including security threats on each requirement.

1.3.1 Security requirements

The base security requirements for the vehicular network summarised in a several survey tutorial [28], these requirements include the following as shown in Fig.-1.3:

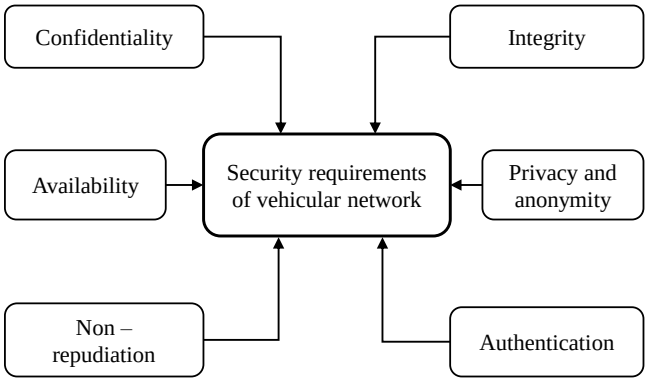

FIGURE 1.3. Security requirements of vehicular networks.

- **Authentication:** Authentication is the verification of the identity verifies the validity of the credentials (Public or private keys) to determine if it is cleared to use the resources. It ensures that all vehicles are the right vehicle to communicate within the network.
- **Confidentiality:** Confidentiality involves a set of rules usually executed through agreements that limit access to certain information and ensures the data can be accessed only by the designated user. These requirements ensure that outside users cannot access the confidential information of the designated user.
- **Integrity:** Data integrity ensures that the delivered message's content is not altered or modified during the transmission process; the receiving vehicle performs data verification to check whether the message contains the correct or corrupted data. The task can be done by the public key infrastructure and cryptography revocation mechanism [29].
- **Availability:** Availability plays a crucial role since it ensures that it remains functional even if it is under an attack without affecting its performance [30]. This concept is not different from other networks but not easy to ensure, mainly because of the high speed of vehicles.
- **Privacy and anonymity:** used to hide the identity of the vehicle and location information against nodes that are not authorized so that no one can trace the movement.
- **Non-repudiation:** This requirement used to ensure that a person who sends a message cannot deny later that he has not sent the message, which also used to find the person who performs a malicious activity; for example, drivers must be identified in case of accidents.

1.3.2 Attacker models

Before describing the attacks, it is essential to define the attacker, to classify the capacities of an attacker and to do so [31] defined four dimensions as in Fig.-1.4:

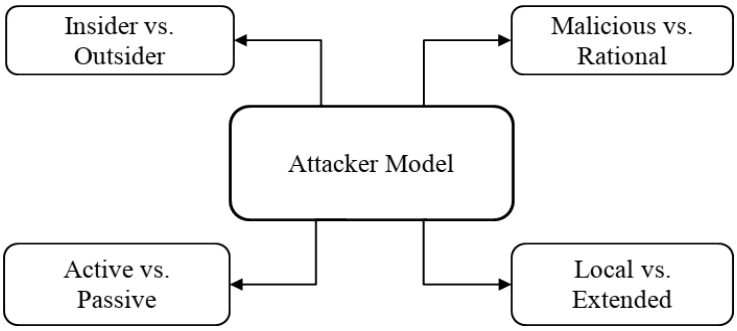

FIGURE 1.4. Attacker model in vehicular networks.

- **Insider vs. Outsider:** The origin of any attack is through an external or internal source. If the attacker is an authenticated member to communicate with other members of the network, it will be known as an insider [32]. Whereas an outsider not authenticated to communicate with other members directly, an outside attacker aims to obtain the credentials of an insider.
- **Malicious vs. Rational:** A malicious attacker uses methods to damage the network without looking for personal benefit. However, a rational attacker expects to benefit from the attacks.
- **Active vs. Passive:** An active attacker can modify the content of the messages, whereas a passive attacker attempts to learn by listing like eavesdropping or monitoring transmission to obtain information but does not affect system resources.
- **Local vs. Extended:** An attacker is considered local if it is limited in scope. An extended attacker expands its scope by controlling several entities that are scattered across the network.

1.3.3 Attacks and threats on Vehicular Networks

In this subsection, we describe the security threats facing vehicular networks. Since we cannot envision all the possible attacks available on the networks due to space limitations, we provide a general classification of the most critical attacks we have identified, and this classification was done according to the security requirements that they violate, respectively and Fig.-1.5 shows the classification.

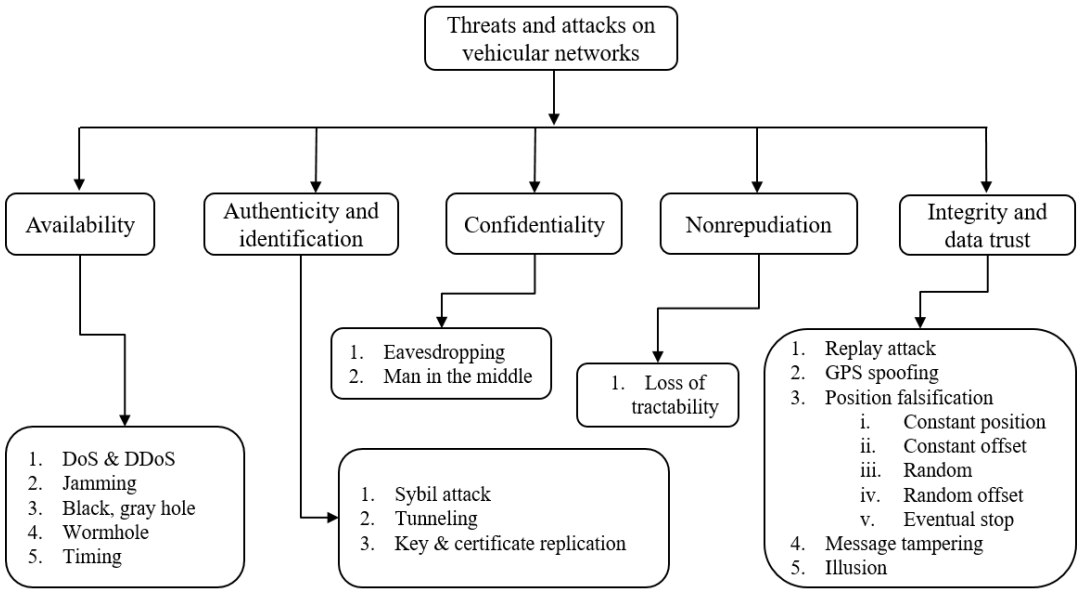

FIGURE 1.5. Classification of attacks by security requirements.

1.3.3.1 Attacks on availability

One of the crucial requirements for the vehicular network is the services available, and all use protocols should remain functional, even attack occurs. This requirement is known as availability, guarantees that the working network is functional, and all useful information is available throughout the networks at any functioning time. Several attacks belong in this category, but the most concerning and hurtful one is the Denial of Service Attacks (DoS) attacks, [28].

- **Denial of service attacks (DoS) and Distributed denial of service (DDoS):** is always one of the most severe attacks in every network, [32] not only in the vehicular network. That targets network service availability, which can have serious consequences, especially for safety applications. The main aim is to prevent authentic vehicles from accessing network services. In DoS attacks, attackers try to block the principal means of communication, transmit dummy messages to jam the channel and reduce the network performance. The Distributed Denial of Service (DDoS) is more severe than the DoS, where several malicious cars attack a legitimate vehicle in a distributed manner, which considers as a distributed DoS attack [33].
- **Jamming attacks:** The jamming attack is an attack particularly challenging to detect and consider as a physical level of DoS attack. A signal is transmitted by the attacker to distract the communication channel, which is mostly intentional, and it lowers the signal to noise ratio for the receiver [28]. The study [34] showed the impact of jamming attacks in vehicular

communications by creating a series of indoor and outdoor jamming scenarios under different jamming behaviors such as constant and reactive.

- **Black-hole attack:** in this attack, the attacker node refuses to forward or even drop the data packet [35] that makes a most dangerous attack on the vehicular network. As the name says, the black hole can be considered an area where the routing traffic is redirected, either there is no node, or the nodes reside in that area refuse to participate [36]. A malicious node always claims that it has the shortest path to the destination that the source node is looking for and cheats the routing protocol without having a look at the routing table firstly. Moreover, the attacker node wins the right to receive the packet to forward to the destination node, and the forged route is successfully established after it depends on the attacker whether to drop or forward the packets even can redirect to wherever it wants.
- **Gray-hole attack:** is mostly a variation of black hole attack [37], where an adversary vehicle first behaves as usual during the route discovery process. Then silently drops some or all of the received data packets sent to it without further forwarding to its neighbors. Detection of this kind of attack is more laborious than others because nodes can partially drop packets not only due to malicious behavior but also due to selfish nature or overload, such as node may unwilling to spend its battery consumption, unavailability of network bandwidth [38].
- **Wormhole attack:** A wormhole [39] is a security attack at the network layer, which can severely affect the unicast multihop communication based on popular routing protocols such as Ad Hoc on Demand Distance Vector (AODV) [40] and Dynamic Source Routing (DSR) [41]. This attack typically requires the presence of at least two malicious vehicles in the network. These two malicious vehicles need to be geographically separated and connected via a tunnel.
- **Timing attack:** One of the most essential and promising use cases of vehicular networks are safety applications. However, they are time-critical and require data transmissions from one vehicle to another vehicle at the right time. In timing attacks [42], when malicious vehicles receive a message, they do not forward it as usual but add some timeslots to the original message to create delay. Thus, neighboring vehicles of the attackers receive the message after the moment when they should receive that message.

1.3.3.2 Attacks on authenticity

In these types of attacks, the affected area is identification. Only a reliable vehicle is allowed to communicate after successfully identified or authenticated. The different types of attacks that aim to violates this requirement are as follows:

- **Sybil attack:** The Sybil attack is a well-known hurtful attack that was first described and formalized by Douceur [43] in the context of peer-to-peer networks. In this kind of attack, a vehicle declares to be several vehicles either at the same time or in succession. This attack is hazardous since a vehicle can claim in different positions at the same time, thereby creating chaos and substantial security threats. The Sybil attack may damage network topologies as well as bandwidth consumption.
- **Tunnelling attack:** In this attack, internal attackers establish a private connection (tunnel) to exchange packets to the partner. The Tunneling attack connects two distant parts of the vehicular network using an additional communication channel such as a tunnel [44].
- **Key and Certificate Replication attack:** The attack consists of duplicate keys and certificates to use as proof of identification. That makes it more difficult for authorities to identify a vehicle, especially in a conflict [45].

1.3.3.3 Attacks on confidentiality

confidentiality is one of the critical security requirements in vehicular communication, and it assures that the intended receiver will only read the message. Some threats aim to violate this requirement, such as:

- **Eavesdropping attack:** The eavesdropping attack is against confidentiality, without an imminent impact on the network only attacks by listening [46] the communication. Through this attack, several types of useful information can be collected, such as location data that can be used for tracking vehicles. These attacks represent a significant violation of privacy.
- **Man in the middle attack:** As the name indicates, the attacker comes between the sender and the receiver [47]. The attacker controls the communication between the two victims, while they believe that they are in direct communication.

1.3.3.4 Attacks on integrity

In a vehicular network, it is always expected to ensure the data integrity and trust to assure the accuracy and consistency of data broadcasting over the network. The following attacks address on data integrity and trust, which affects the whole system's reliability.

- **Replay attack:** This classic attack consists of replaying (broadcast) a message already sent to take the benefit of the message at the moment of its submission [48]. Therefore, the attacker injects it again in the network packets previously received. This attack can be used to replay beacons, to manipulate the location and routing tables.
- **GPS spoofing attack:** In the vehicular network, the position of a vehicle is critical should be very accurate. A log file maintains the location generated by the GPS satellite [49]. In this attack, the attacker used a trick to create false GPS location information and did not reveal the correct position to avoid the vehicles that may think it is available in another location.
- **Position falsification attack:** the attacker broadcast false position information in the safety messages, due to system error or to obtain personal benefits on the road traffic. There are five varieties of position falsification attack studied in the literature [50].
  - **Constant position attack:** the attacker transmits a fixed pre-configured position.
  - **Constant offset Position attack:** The attacker added a fixed offset to the actual position.
  - **Random position attack:** sends a position inside the simulation area, which is uniformly random in the playground.
  - **Random offset attack:** add a random offset bounded by a rectangle around the vehicle.
  - **Eventual stop attack:** usually behaves for some time and then attacks by repeatedly transmitting the same position (i.e., as if it had stopped).
- **Message Tampering/Fabrication attack:** this attack consists of modifying, constructing, or altering existing data. It can occur by modifying a specific part of the message to be sent. For example, the attacker falsifies received data indicating that the route is congested, and changes them to no congestion and traffic on the road is ordinary [28]. In this attack, the attacker alters or makes new messages to achieve the intended purpose of the attack.
- **Illusion attack:** A direct application of the fabrication of messages attack is the Illusion attack, which is an attack against integrity and data trust [51]. It consists of placing voluntary sensors that generate false data that can generally move in the network.

Attacks on non-repudiation: When two or more user shares the same key, then non-repudiation occurs, two users are not distinguished from each other, so their actions can be repudiated. That is entirely unexpected in the vehicular network case, and the following are the attacks that implemented to violate this requirement.

- **Loss of tractability:** The non- repudiation attack consists of exercising action, allowing an attacker to deny having made one or more actions. Some attacks can serve as preliminary to non-repudiation attacks such as Sybil attack [52] and duplication of keys and certificates.

1.3.4 Application vs Security

We have already seen vehicular networks suffer from different threats and attacks, the damage caused by attacks, and the negative impact that also influences the available service and application [29]. Most of the suitable security scheme comes with a high computational cost; therefore, it is considered an inappropriate approach for low latency safety applications (especially time-critical safety applications) which raises the security threat on the application itself [53].

- **Attack on Safety Applications**

  The available safety applications are vulnerable by different attacks related to the channel allocation such as DoS attacks, distributed denial of service attacks and jamming attack, the attack occurs when the dishonest vehicle sends multiple messages which block all possible way of communication [54].
- **Attack on Communication and Traffic management application**

  In traffic management, communication is time sensitives and requires message arrival on time for settlement. However, they are vulnerable by several attacks [55], such as the timing attack can create massive damage to the traffic management system of vehicular networks [42].
- **Attack on Comfort and Infotainment Application**

  The increasing demand for comfort and infotainment functionality in vehicular networks has also created new attack surfaces. They are easily vulnerable, increase the privacy, and anomaly threat [56] compare to others, overwhelming use of comfort and infotainment application also can disrupt the network performances [57].

1.4 Machine learning

Machine learning primarily consists of designing efficient and accurate prediction algorithms (learning algorithms), and the achievement of a learning algorithm depends on the data utilized for training, machine learning is inherently associated to data analysis and statistics, which can be broadly described as computational methods that use experience (historical data) to enhance performance (accurate predictions). Here, experience refers to past information, typically collected data; the quality and size of the data are crucial to the prediction's success. According to the working criteria of different algorithms, machine learning can be divided into four branches as shows in Fig.-1.6, with possible application.

Machine learning yields a ubiquitous set of practical applications, which include the following:

- **Internet of Things (IoT):** Rapid developments of technologies have facilitated the emergence of Internet-connected devices that provide observations and data measurements from the real world, known as Internet of Things (IoT) [58]. Machine Learning can eliminate errors and enable data to generate real-time insights and allow IoT devices to reach their full potential.
- **Autonomous Vehicle:** Significant progress in Machine Learning techniques like Deep Neural Networks (DNN) [59], has enabled the development of autonomous cars. Several major car manufacturers, including Tesla, BMW, and Waymo, are building and actively testing them.
- **Intelligent Transportation Systems (ITS):** The latest advances in autonomous technologies, transportation are evolving into more intelligent systems, called ITS. A combination of ITS and machine learning provides practical and effective solutions for optimal traffic Traffic Signal Control (TSC), autonomous vehicle control for 21st-century transportation, which is initial to provide safe, effective, and reliable trip on the road [60].

The following signifies a list of terminology regularly used in machine learning.

Data Set: A collection of related sets of information can be manipulated as a unit by a computer, instances of data use for learning or evaluation.
Features: the feature, or column, represents a measurable piece of data that can be used for analysis. Sometimes features are also called attributes and the number of features are called dimensions.
Labels: Labels are the final output (misbehavior or normal), to be predicted by learning algorithms
Hyper parameters: Free parameters that are not determined by the learning algorithm, the user can specified as inputs to the learning algorithm.

Training Data: the data used to train a learning algorithm.

Validation: The process to find the best hyper parameters setting using validation sample.

Test Data: sample data use to evaluate the performance of a learning algorithm.

Loss function: a function that measures the difference between a predicted label and a true label.

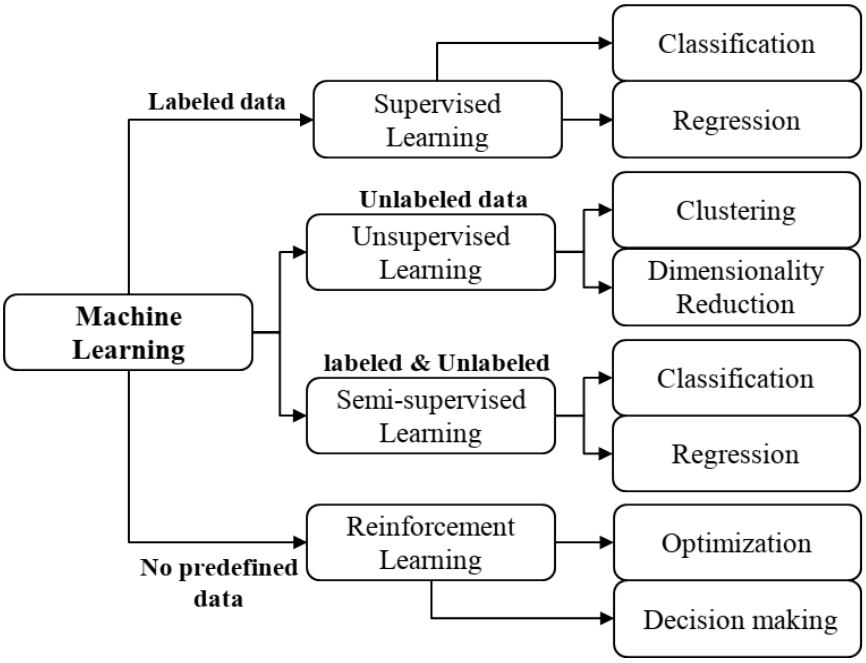

FIGURE 1.6. Branches of machine learning and possible application.

1.4.1 Supervised learning

Supervised learning is the machine learning task where each learning algorithm was learning a function that maps an input to an output based on features - labels pairs. In supervised learning, the operator provides a known data set that includes desired inputs (features), outputs (labels), and the algorithm must find a method to determine how to arrive at those inputs and outputs pair, the algorithm identifies patterns in data, learns from observations, and makes predictions.

The following are some standard machine learning tasks associated with supervised learning:

Classification: The task is assigning a category to each observation. For example, each underlying safety message consists of assigning a group such as malicious or normal.

Regression: this is the problem of predicting a real value for each observation. For example, predicting a trust value or threshold based on attacker movements.

1.4.2 Unsupervised learning

The learning algorithm exclusively receives unlabeled training data and makes predictions for incoming data. Unsupervised learning is a type of machine learning algorithm used to draw inferences from data sets consisting of input data without labeled responses. Since, in general, no labeled example is available, it can be challenging to evaluate the performance of a learning algorithm quantitatively.

The most common standard machine learning tasks associated with unsupervised learning:
Clustering: this is the problem of partitioning a set of observations into homogeneous subsets. Clustering is often used to analyze extensive data sets.

Dimensionality Reduction: this problem consists of transforming an initial representation of observation into a lower-dimensional representation while preserving some properties of the initial representation. Mostly use for data preprocessing in computer vision tasks.

1.4.3 Semi-supervised learning

Semi-supervised learning is similar to supervised learning, but the training sample consists of both labeled and unlabelled data to make predictions for incoming data. Semi-supervised learning is an approach that combines a small amount of labeled data with a large amount of unlabeled data during training, which falls between unsupervised learning and supervised learning.

The semi-supervised is commonly used where unlabeled data is easily accessible, but labels are expensive to obtain. Likely supervised learning semi-supervised learning also can be used for classification, regression, but unlabeled data accessible to the algorithm can help to archive a better performance [61].

1.4.4 Reinforcement learning

Reinforcement learning refers to goal-oriented learning, which learns how to attain a complex objective and to maximize a particular dimension over many steps. It focuses on controlled learning processes, where a learning algorithm takes a set of actions, parameters, and end values (reward) with associate rules and tries to explore different options and possibilities, monitoring and evaluating each reward to determine which one is optimal (best reward).

The algorithm must learn from past experiences and adapts its approach to the situation to achieve the best possible result, which differs from supervised learning in several ways. The essential difference is that there is no presentation of input (features) - output (labels) pairs. Instead, a software-defined agent must gather useful experience about the possible system states, actions, transitions, and rewards actively to act optimally.

Reinforcement learning has proved their efficiency in several applications, such as a self-driving car, vehicle route optimization, and predictive maintenance of automated systems.

1.5 Software Defined Networking (SDN)

Recently, software-defined networking SDN has become one of the most promising solutions to Information and Communication Technologies (ICT). However, being a new concept, it is kind of hard to reach on its exact definition, many different definitions have surfaced over the literature, each of which has its own merits. In this section, we first present a generally accepted definition of SDN, outline a set of essential benefits, and finally promising applications of SDN.

1.5.1 General SDN Architectures

The most explicit and well known definition of SDN as: is an emerging network architecture where control plane is decoupled from forwarding and is a directly programmable approach to network management that enables dynamic, programmatically efficient configuration in order to improve performance and monitoring, making it more like cloud computing than traditional network management [62].

As in the Fig.-1.7 SDN decouples the control plane from the network devices and becomes an external entity: the network operating system With SDN, management becomes simpler.

1.5.2 Application and Promising Benefits of SDN

Software-defined networking has applications in a wide variety of ICT environments and introduces promising benefits, especially for the enterprise network [63]. By decoupling the control and data planes, programmable networks enable customized control, eliminate middleboxes, and simplify the development and deployment of new network services and protocols [64]. Some of the promising applications are as below:

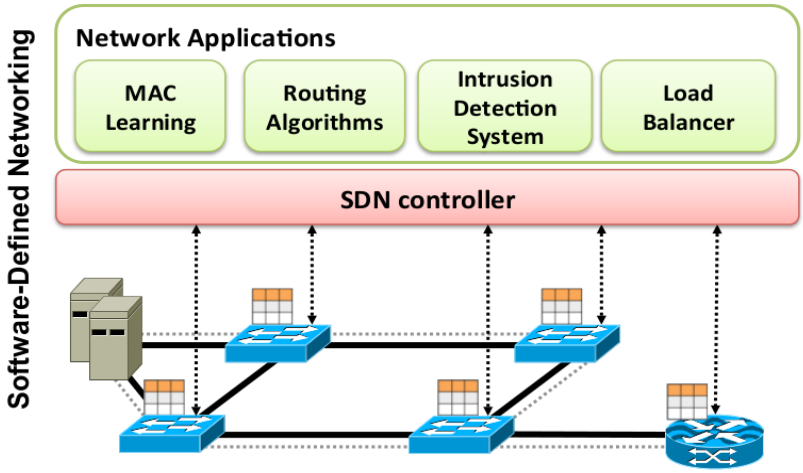

FIGURE 1.7. Software-Defined Networking (SDN)

- **Enterprise Networks:** Enterprises often run large networks, while also having strict security and performance requirements. Furthermore, different enterprise environments can have very different requirements, characteristics which can be very challenging for management. Adequate management is critically important in enterprise environments and SDN can be used to pro-grammatically enforce and adjust network policies as well as help monitor network activity and tune network performance.
- **Satellite communications:** Satellite systems have served humankind in several ways since its invention. Including the feature of global coverage, they have provided services such as TV broadcasting, digital messaging, GPS navigation, and worldwide telecommunications for decades [65]. In recent times, the tremendous growth in mobile devices and the desire for connectivity, Satellite systems enhance an essential complement to the next-generation cellular system (5G) [66]. The integrated dynamic network management and control in a programmable manner is the demand of time. The integration of SDN to satellite communication enhances those functionalities and reprograms the data plane at any time.
- **SDN for Vehicular Network:** The integration of SDN into vehicular networks is considered one of the promising future solutions for routing and resource allocations, which makes the system update and dynamically reconfigure without having challenging tasks [67].

1.6 Conclusion

In this chapter, we have reviewed the fundamental architecture of vehicular networks with various communication technology and standards. Few significant applications and the network characteristics were discussed. We have also summarized the security requirements and the corresponding feasible attacks. That can be executed to violate the particular requirements, including the promising application that is also vulnerable by several attacks. An attacker model was designed to classify the dimensions of an attacker. The significant portions of machine learning and SDN were also noted.

A secure and attack-free atmosphere is a prerequisite in vehicular networks for message distribution over the network. However, since the networks are vulnerable to many types of attacks, including the position falsification attacks, it can produce dangerous circumstances and defeat the purpose of vehicular networks. Therefore, the system extremely requires the deployment of additional mechanisms for more immediate attack exposure before any critical damage occurs. The next chapter intends to investigate the feasibility of existing literature linked with position falsification attacks.

CHAPTER 2

ML-based Misbehavior Detection systems for position falsification attacks

---

2.1 Introduction

Vehicular networks provide useful functionalities entirely based on the data exchanged over the network are not encrypted to reduce the local computational complexity and time. Without a system that guarantees sender authentication, and data integrity leaves the system open to endless security threats. A public key infrastructure (PKI) consistently used throughout the literature as a solution. That enables participants to securely communicate on an insecure public network after reliably verifying the identity via digital signatures, which only works to avoid external attacks. However, the networks are also vulnerable to many types of internal attacks, including position falsification attacks, which can lead to hazardous situations, especially for traffic efficiency and management application, and increase the possibility of traffic collisions. Therefore, the vehicular network highly requires the deployment of additional security mechanisms that can detect the misbehaving entities. Misbehavior Detection Systems (MDSs) are considered an efficient way to detect such data semantic levels of internal attacks.

This chapter intends to overview existing ML-based MDSs for position falsification attacks within the current ecosystem to secure vehicular networks. We describe detection mechanisms based on their feasibility and focus on every detection method's relevant details. The main contribution is to provide a novel comparison of the existing ML-based MDSs for false position attacks identifying their weaknesses.

The chapter's organization is as follows: Section 2.2 presents the definition and different types of MDSs. Section 2.3 provides the existing literature review associated with the position falsification attacks. Section 2.4 presents a synthesis of existing solutions. Section 2.5 concludes this chapter.

2.2 Misbehavior Detection Systems

In this section, we define misbehavior detection systems with a classification based on system development and requirements. There are fundamentally different approaches that can be used to categorize existing mechanisms. We focus on considering whether mechanisms based on data values contained in messages or on the node behavior that is sending the messages.

2.2.1 Definition

The misbehavior is a broad term, and it is commonly used for ad-hoc networks to discuss specific attacks that are executed by the participating nodes, as opposed to cyber-attacks or intrusions. Many authors are not using misbehavior to define attackers, but only malicious participants [68]. However, our definition covers not only malicious and attackers but also faulty nodes of the networks. Therefore misbehaving nodes are thus any node that transmits erroneous data, should not transmit in case of normal behavior. Misbehaving node is the type of node we should detect to ensure safety.

However, the literature often distinguishes [69], [70], between the malicious node and faulty node. Faulty nodes are those in the network that produce incorrect data without malicious intent. For example, a malfunctioning GPS sensor can produce incorrect data due to damage or other technical issues, which can cause to transmit an erroneous position and bring threats. On the other hand, malicious nodes or attacker nodes are those nodes that transmit erroneous messages with malicious intent. They get more research and development attention by researchers as they bring direct security threat to the network's safety-related application. Sometimes, they can actively attempt to avoid detection and control other nodes to transmit their erroneous messages, such as the denial of service attacks. In this study, we use misbehavior detection to refer to the detection of both faulty and malicious nodes, though throughout the literature are not consistently used these definitions [71].

2.2.2 Classification

According to the detection requirements and behavior, there are fundamentally different approaches to misbehavior detection that can be used to categorize, as shown in Fig. 2.1, the first distinction is whether mechanisms focus on the node sending the messages: **(i). node-centric detection** or on data values contained in messages: **(ii). data-centric detection** and combination of both: **(iii). hybrid detection**, which we are going to discuss over this subsection in detail.

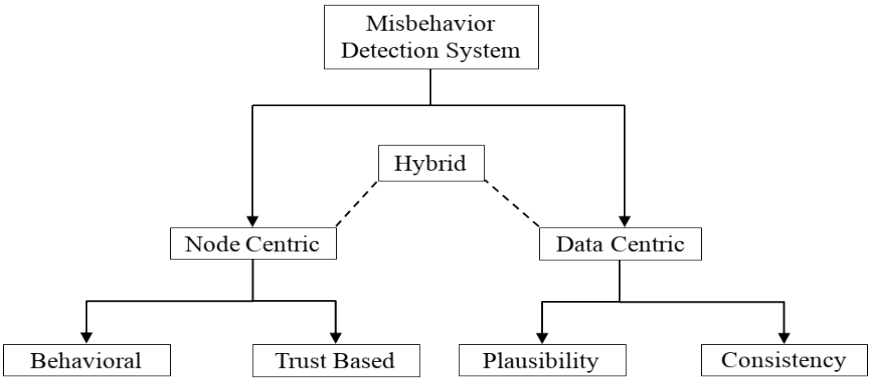

FIGURE 2.1. Classification of misbehavior detection system

2.2.2.1 Node-centric Misbehavior Detection

In the node-centric detection, a security model monitors authentication credentials, like a digital signature with PKI's help and use knowledge to detect malicious senders. The node centric detection mechanism is precisely concerned with a participating node behavior by analyzing whether the message frequency, headers, and content are in line with protocol specifications to find unusual patterns and create a history with a trust value based on the current behavior. Node centric detection can be divided into two categories, **(i). behavior-based detection**, and **(ii). trust-based detection**.

(A) **Behavior-based detection:** This mechanism focuses on finding messages that are not in the correct pattern (message format), identifying nodes that send messages too frequently or nodes that modify the content of the messages in a way that does not adhere to protocol criteria [72]. The main focus of these mechanisms is to monitor the behavior and find abnormal operation such as messages sending frequency, packet drop, or duplication rates are exceeding from standard rates, neighbor nodes in the vehicle neighborhood are responsible for monitoring.

(B) **Trust-based detection:** The main idea of trust-based detection mechanisms is to assign trust values according to the participating nodes' behavior. When a node reputation (trust) exceeds a predefined threshold [73], it will be considered as an attacker. Furthermore, the trusted authority is available to remove malicious nodes according to the trust values. Which nodes should remove from the network will depend on the past and present reputation history. The fundamental assumption is that a vehicle behaving correctly in the past is more likely to behave correctly. Essentially, this brings down to some form of reputation control where correct performance increases the reputation, and misbehavior reduces.

2.2.2.2 Data-centric Misbehavior Detection

The data-centric detection mechanism concentrates on analyzing transmitted data for plausible misbehavior. The broadcasted message data is compared to find implausibility and inconsistency. The node explores reasonable evidence to verify introduced message data locally or with neighbors' cooperation and use it to generate safety alerts. The disclosure of false safety alerts consists of local-based detection or cooperative-based detection, [74]. Based on the underlying data verification specifications, we can divide the data-centric detection mechanism into **(i). Plausibility check** and **(ii). consistency check**.

(A) **Plausibility check:** The plausibility checks utilize to validate the correctness of the communicated data and rapidly filters that are malicious. For example, the movement plausibility can be verified from two successive beacon messages: measuring the distance traveled and comparing it with speed in that direction. This detection technique operates by analyzing packets originating from unique senders, and the data in the packet is either examined against a predefined threshold or a confidence interval [75].

(B) **Consistency:** Consistency checking is a mechanism for monitoring whether messages do not contain semantically conflicting data. Consistency-based disclosure uses relations between packets to discover the trustworthiness of newly received packets. For example, a consistency-based detection mechanism may recognize previously computed average speed of vehicles on a highway to judge newly received speed messages. Messages that differ (departing from accepted thresholds) from the average speed are contradictory can be recognized as suspicious. However, honest historical majorities are usually expected to draw reliable judgments.

2.2.2.3 Hybrid Misbehavior Detection

In some states, neither a data-centric nor node centric strategy alone can efficiently disclose the misbehavior in vehicular networking. To this end, a combination of both appearances has been introduced [76] to enhance the detector's performance, which is known as a hybrid misbehavior detection system. In the hybrid approach, vehicles are judged based on their behavior, integrity, consistency, and plausibility inspection of the generated message. The hybrid method is more promising to adequately detect multiple types of illusion and context originated attacks since it merges numerous detection concepts in a single design and mounted on top of node centric and data-centric misbehavior detection system.

All misbehavior exposure tools studied in the literature rely on the fundamental concept of the above three varieties of strategies. Many scholars previously produced schemes based on these strategies that could be considered resolutions of attack associated with position falsification in the vehicular network, which we review in the next section.

2.3 Misbehavior Detection Systems for Position Falsification Attacks

Numerous approaches have been suggested to detect position falsification attacks over the past year through scholarly research. The well-known approach consists of recognizing the correctness and comparing of data broadcasted with the past received data. In this section, we evaluate the existing works associated with false position attacks and focusing on summarizing them.

Grover et al. [77] proposed an approach using a machine learning technique to discover the features of the legitimate and misbehaving vehicle based on their behavior. The features were extracted based on validation checking, such as geographical position validation, acceptance field verification, speed variation verification, and received signal strength. To determine the pattern of onboard vehicle behavior, they practiced another set of features by calculating the total number of generated packets, and packets received, delivery ratio, drop ratio, capture ratio, and transmission error ratio, based on the neighbor observation. They examined packet suppression, packet replay, packet detention, identity spoofing, position falsification, and a combination of identity and forging attack to evaluate the proposed approach. A network simulator (NCTUns-5.0) [78] was used to produce these sets of features, and the generated features used as input for several machine learning classification algorithms contained in the WEKA [79] toolset to classify as malicious or legitimate. This work was improved in [80] by choosing a multi-class classification, and a majority decision was considered classification results. However, both of these works rely on the specific attack implementation depends on the extracted features, but no details provided regarding these. There are no base scenarios to link the generated data sets and features with associated attacks.

Joseph Kamel et al. [81] introduced a mechanism based on the fundamental of plausibility and consistency checks; these mechanisms can be divided into four steps: local misbehavior detection, misbehavior reporting, global misbehavior detection, and misbehavior reaction. The local detection mechanism was applied in the first place, and the results were published to the misbehavior authority after a global check was performed on the authorities' side. They proposed a comparative approach of

local detection mechanisms, which are as follows: **threshold Based:** consists of testing the result of every local plausibility and consistency checking against a predefined threshold, a message is considered malicious if any check fails. **A non-cooperative threshold-based:** This solution aims to evaluate a node centric trust by using data-centric plausibility factor considering the same logic as in [82], **Cooperative Trust-Based:** This solution aim to determine a shared level of trust between all participating node in the network by using the similar approach described in [83]. **Machine learning-based:** The aim is to detect misbehavior by using trained ML algorithms, using four popular algorithms and a grid search based on 5-fold cross-validation, to find the best hyperparameters of the algorithms, more technical details, and the implementation shared on GitHub [84]. For every message, two features sets were created: **Checks Feature:** based on historical plausibility and consistency check done and **Kinematic Feature Set:** based on the difference between position, speed, acceleration, heading and time of the last two beacons. To evaluate the solutions, they used an open-source framework known as F2MD [84], and the considered evaluation metrics are Recall, Precision, F_1-score, accuracy, bookmaker informedness (BM), markedness (MK) Matthews correlation coefficient (MCC) and cohen's kappa, these evaluation metrics are detailed in [85]. The results comparison showed machine learning solutions outperform some time but only by a small margin. This work relies entirely on the local and global plausibility, and consistency check depends on a predefined threshold even for the machine learning-based detector to generate features of the training data. A message was considered misbehaving if the global trust level falls below an absolute value. However, these solutions are vulnerable to sophisticated attacks [86] in which the attackers are aware of the predefined threshold context since the predefined static consistency and plausibility thresholds were kept and not replaced dynamically or updated in a timely fashion. Besides, to develop cooperative trust, they considered sharing the global trust levels between all the communication nodes, and no details found how sharing helped to improve the detector's performance.

Issam Mahmoudi et al. [87] proposed an ML-based global misbehavior detection system to analyze the reported misbehavior sent by vehicles and RSU. The aim was to judge local detection solutions; a set of algorithms was trained to assess the detection based on a few selected features. The first features are local detection functionality like [85] relay on plausibility and consistency check. The second set of features was considered from kinematic data from V2X communications and the final set of generic features such as binary features computing the number of checks that return a complete failure. In order to evaluate the solution, they used the same framework as [84].

Rens W. van der Heijden et al. [50] took the first steps by introducing the vehicular Reference Misbehavior Data set (VeReMi), which is extensible and publicly available and allow researchers to compare detection results. Five different attacks associated with position falsification called random, constant, constant offset, random offset, and event stop was implemented, and many researchers already used this data set to show the efficiency of their proposed mechanisms [88], [89]. Since this data set contains a list of message logs, it is easy to run against a misbehavior detector and compare it to others. They also introduced a detector system based on plausibility check (the acceptance range threshold and sudden appearance warning), as two detector speed checkers and distance moved versifier. However, the proposed detector working mechanisms depend on a pre define threshold (static).

So et al. (1), [90] also proposed mechanisms to detect position falsification attacks using a set of plausibility checks. They designed plausibility metrics with six features: **local plausibility check (feature-1):** The sender's reported location was compared with a predicted plausible location based on the previous velocity, location, and the distribution of average acceleration. To give a score [0,2] on the plausibility of the x and y coordinates, the sum of these two scores was considered the total score of the plausibility check. A confidence interval was calculated for the average acceleration of 95% and 99% confidence, respectively. Then a range of plausible location of the vehicle was estimated using the formula:

$$predicted_{(x,confidence)} = x_i + \Delta t(v_{(x,i)} + a_{(x,confidence)} * \Delta t) \tag{2.1}$$

where $predicted_{(x,confidence)}$ is the range of predicted coordinates values for 95% or 99% confidence accordingly, x refers previous GPS x- coordinate, Δt time difference between previous and current packet, $(v_{(x,i)}$ velocity in the x-direction and $a_{(x,confidence)}$ acceleration range in x- direction within the respective confidence. If the range of predicted coordinates in 95% confidence range then the plausibility score was set to 0, if outside 95% but within 99% then the score one otherwise the score is 2. **Movement plausibility check:** the feature was designed for constant position attack, the aim is to find average velocity during the entire trip and compare with total displacement. If the total displacement is 0, but the average velocity is not 0, then leveled as malicious. **Quantitative Information (features 3, 4, 5, 6):** these features are nothing but a numerical description of the vehicle behavior, feature 3 and 4 represent the difference between the calculated average velocities. Based on total displacement, time, and the predicted average velocity, feature 5 is the magnitude of features 3 and 4, where feature 6 is the total between the calculated distance and predicted displacement. Two machine learning algorithms were trained based on the output generated by the mention plausibility. The VeReMi dataset was considered to Check the efficiency of the system. The results analysis showed that SVM was able to obtain the most

significant precision.

Le and Maple [91], suggested machine learning approaches to detect misbehavior in vehicular networks based on n- sequence trajectory inspection where a sequence of messages was considered to form a trajectory [92]. Three features used to extract the data: Movement plausibility check, Minimum distance to trajectories, and minimum translation to the trajectories. Movement plausibility check: focus where the vehicle is moving but reported as uncharged, by observing a sequence of trajectories, aims to classify constant and eventual stop attacks. On the other hand, constant attack transmits the same location all the time and eventual stop attack, there should be at least one message that reflects the previous. This check only works for constant and eventual stop attacks and fails for other types of considered attacks since they do not transmit a fixed location. Minimum distance to trajectories: aims to find moving patterns of the vehicle in the legitimate set. According to their consideration, the minimum distance to the trajectories should be large in the case of constant offset, random, and random offset attack since they have small changes in the movement patters. The minimum distance to trajectories work well to find the random attacks but would not help very much to detect constant offset and random offset since they have a very close distance to the trajectories. To solve this issue, they proposed another feature. Minimum translation to the trajectories: The paper's analysis showed the value should be very close to 0 for constant offset attack since it is translated from legitimate trajectories, and the value remains large for random offset attack. They implemented two machine learning algorithms like [90] to detect and classify the malicious attack with the proposed features. The training data was extracted from the VeReMi data set, and 80% of the sample data were used to training the model but the remaining to test the performance. To evaluate the results, they did cross-validation with [90], and the analysis showed that the proposed approach provides better performance all most in every case. However, we found in case of constant offset and random offset attack, the distance comparison is not semantically accurate always since the added position to the normal can be tiny and will not be possible to differentiate. The attacker can be aware of the predefined confidence interval (threshold) and implement an attack by adding tiny little offset that lies between the confidence interval.

So et al. (2) [75], proposed three physical layer plausibility checks that leverage the received signal strength indicator (RSSI) of BSM. To execute the proposed plausibility check, VeReMi was used but with some source code modification mainly to record the location of the receiver, but the simulation was run only for 30% attack density. In their consideration, when a vehicle enters a new area, it must know the distance versus RSSI distribution, either by downloading from roadside units or predefined, every safety message will be classified using this distribution, and the output of these checks defined

the sender as usual or misbehaving. A threshold was generated in terms of the confidence interval. To determine the certainty interval, they measured the distance between the sender and receiver and group all the messages have 1-meter distance maximum. RSSI was calculated for each group of the messages to find a confidence interval. This confidence interval was used as a threshold for plausibility checks. When a message is outside the confidence interval, the transmitting vehicle is immediately classified as a misbehaving refers to the first plausibility check. The second plausibility check uses majority rules; if the majority of the incoming messages from the same source are classified as malicious, then the vehicle is classified as malicious. The third check is to assign a score and update this score for every new message. When a vehicle reaches a lower than 99.7% of the regular vehicle scores, it is classified as an attacker. However, these plausibility checks have multi-step mechanisms to improve the detection rate and decrease false positives. These checks run independently by each vehicle based on a predefined threshold.

Gyawali and Qian [93] proposed a cooperative MDS using machine learning algorithms, where each vehicle is equipped with MDS, always broadcast the disclosure results to its neighbors. To create a learning feature for ML algorithms, they generated labeled data sets through vehicular network simulations called the Luxembourg SUMO traffic scenario (LuST) [94], but this simulation was done only for false information messages. They used the VeReMi data set to get the learning features for position falsification attacks and compared each received beacon with the previous one to measure the distance between the sender and receiver. In order to evaluate system performance and show the effectiveness, they compared the results with VeReMi. However, this work focuses on the false alter attacks detection rather than position falsification attack.

2.4 Synthesis

In this section, we analyze the exiting literature associated with position falsification attacks, shown in Table-2.1, where each column describes a comparison criteria. The first column **" Solution"** signifies the referees of exiting work in alphabetical order accompanied by the author's name and publishing year. The second column, **" Category"**, indicates to which category the proposed solution belongs. For example, *Data-centric: plausibility* indicates the proposed solution is based on a data-centric plausibility check. The third column **"Collaborative"** specifies if proposed solution is collaborative or not. The collaboration in our context refers to the need for the ML-based MDSs of multiple vehicles' feedback before making a decision. The fourth column **" Dynamic"** exposes, all mechanisms are suggested based

on a predefined threshold, which narrows the detection system as static (model parameters are fixed and can not be updated in a timely fashion), one the other hand, a dynamic system is the one responsible for updating system parameters in a timely fashion. In a dynamic model, the predefined static consistency, plausibility, and behavioral thresholds were replaced by dynamic setting recommendations that are constructed online and updated in a timely fashion. The fifth column **" Model update"** determines if the ML-model model is updated after its deployment. The sixth column **"ML Features"** specifies which information is needed to build the ML-model. The seventh column **"Validation"** determines which tools are used to validate, and the eighth column **"Used dataset"** specifies which data set is used to verify the solution.

Solutions	Category	Collaborative	Dynamic	Model update	ML Features	Validation	Used dateset
Grover et al. 2011 [77]	Hybrid: Behavioral plausibility	No	No	No	Position Velocity, RSSI Transmitted packets	Analytical	Simulator generated
A Le, C Maple 2019 [91]	Data centric: Plausibility	No	No	No	Position Trajectories	Analytical	VeReMi
P Sharma, J Petit, 2018 [90]	Data centric: plausibility Numerical check	No	No	No	Position Velocity time difference	Analytical	VeReMi
S So, J Petit 2019 [75]	Data centric: Plausibility	No	No	No	Position RSSI	Analytical	VeReMi
S Gyawali, Y Qian 2019 [93], [95]	Data centric: Plausibility	No	No	No	Position Velocity RSSI	Simulation	VeReMi
Issam Ma. 2020 [87]	Data centric: Plausibility Consistency	No	No	No	Position Velocity Acceleration time difference	Simulation	F2MD [84]
J Kamel 2019 [81]	Hybrid: Plausibility Consistency Behavioral	No	No	No	Position Velocity Acceleration time difference	Analytical simulation	F2MD
PK Singh 2019 [96]	Data centric: Plausibility	No	No	No	Position (sender) position (receiver) Speed difference	Analytical	VeReMi
Our solution	Data centric: Numerical check	Yes	Yes	Yes	Position	Analytical Simulation	VeReMi Simulator generated

TABLE 2.1. Misbehavior detection systems overview associated with position falsification attacks

After reviewing the table -2.1, from the category column, we can conclude, all the works are practicing data-centric mechanisms, while only [77] and [81] are using a hybrid technique, which combines data-centric and node-centric mechanisms. We can also see from the collaborative column all proposed solutions are non-cooperative approaches, but our proposed solutions are collaborative in which each vehicle cooperates to provide accurate detection of the attacks. One of the most significant parameters is

the detection threshold that demands to be dynamic since the static threshold can be subject to security threats and vulnerable by attacks [97]. In our system, the threshold is considered as dynamic, and the SDN controller dynamically update. Additionally, the SDN controller is responsible for the ML-model installation and update on the dynamic context. It is also clear that our proposed solutions required less information (only position) than the exiting works. We propose using the VeReMi data set to validate our detection mechanisms but also introduce a simulator generated data set for cross-validation.

2.5 Conclusion

The vehicular network is a promising future to achieve safer, more efficient, and comfortable travel. However, this network is vulnerable by different variations of position falsification attacks requiring earlier detection before any critical damage occurs where the PKI already fails, leading to the development of a misbehavior detection system on top of PKI. In this chapter, we have first defined the misbehavior detection system, and a classification of different mechanisms proposed in the literature, specially designed for position falsification attacks variation to understand better the limitations of existing approaches. To solve the limitation of the existing solutions, we introduce a novel system which integrates ML and SDN to secure the system from different types of false position attacks. In the next chapter, We will discuss the methodology and design of our proposed system to solve the limitations of exiting works.

CHAPTER 3

SDN and ML for detecting false position attacks in vehicular networks

---

3.1 Introduction

The earlier chapter forces us to conclude, the existing literature to secure vehicular network from position falsification attacks relies on plausibility and consistency check, has significant disadvantages. A considerable drawback is that they required filtering based on a certain threshold, and a combination of unnecessary data having no link with the considering attack detection led to substantial privacy issues. We have also seen it is intuitively hard to build a mechanism that deals with such insider attacks since the PKI fails.

To address these issues, we propose a novel scheme based on machine learning and software-defined network (SDN) to detect earlier mentioned attacks accurately. In our system design, a global SDN controller is considered the brain and responsible for installing, updating, and deploying the machine learning model for local attack detection. The local exposure will be verified and enhanced by executing a collaborative report investigation system.

The remainder of the chapter is as follows: Section 1.2 describes the architecture of the proposed misbehavior system. Section 1.3 presents the details concept about the system attacker model and the variation of position falsification attack. Then section 1.4 presents the building model for ML-based local misbehavior system followed by section 1.5 describes the misbehavior report investigation. Finally, section 1.6 concludes this chapter and gives some future concepts for the next chapter.

3.2 System Architecture

As shown in Fig.-3.1, our proposed architecture mainly consists of three types of entities:

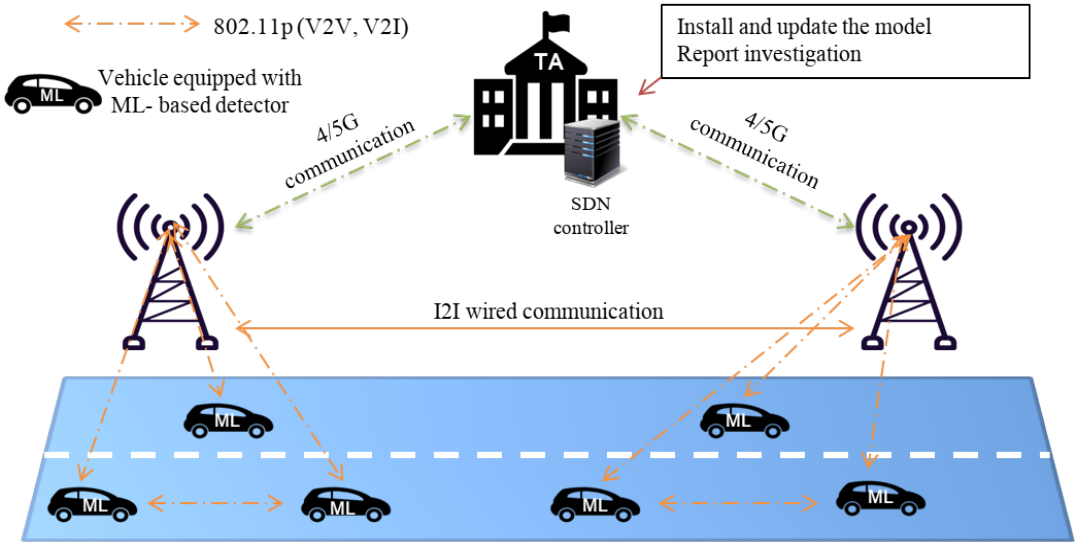

FIGURE 3.1. Software defined vehicular network architecture for MDS

(A) **The Trusted Authority (TA)**: is only trustworthy control center. All significant operations such as vehicle registration, key management, verification, and secret keys allocated to the vehicles are stored in TA since it has adequate storage and computation ability.

(B) **Road-Side Unit (RSU)**: In our architecture, RSU is one of the fundamental components, equipped with two network interfaces: wired link to communicate with the neighboring RSUs, and a 4/5G interface to communicate with the global SDN controller located at the TA and all communication links are secure, RSU performs as the intermediaries between TA and vehicles.

(C) **Vehicle**: The vehicles are service receivers, where each vehicle is equipped with an 802.11p network interface to communicate with RSUs (V2I) and other vehicles (V2V). Every t millisecond, each vehicle periodically broadcasts a safety message, where each message includes location, time, velocity etc. We consider each vehicle has the ML-based misbehavior detector installed locally, always activated, ready to classify every safety message upon arrival, and any detection should be reported to the TA immediately.

The global SDN controller: The global SDN controller, which is installed at the TA, have comprehensive knowledge of the network and acts as a strategic control point; essentially, it is the brain of the networks and has the following two major functionalities:

- **Install/ Update the model:** The vehicle meets the eligibility criteria (registered member and has the secure keys) check by opening a bidirectional stream. The stream is used to verify

PKI, and after successful keys verification, the SDN controller (TA), send the ML-based detection model (binary) to the vehicle and installed it for local detection. The SDN controller periodically updates the model based on several criteria. A sequence diagram that shows install and model updating process in time sequence is in Fig.-3.2

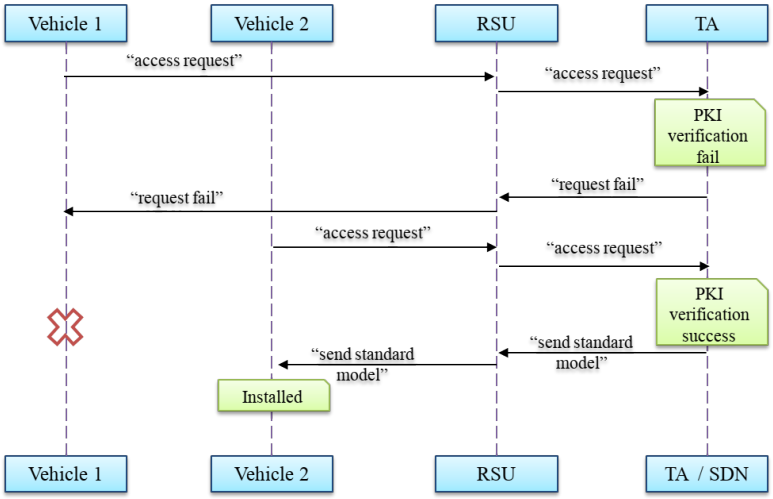

FIGURE 3.2. Sequence diagram of the system interactions to install the model

- **Report investigation:** After receiving a misbehavior detection reports from the onboard vehicle, the SDN controller always needs to verify whether the report is accurate or not that where report investigation comes from, which we are going to discuss in section 1.5 later. The SDN controller asks feedback from the vicinity of the vehicle before taking any action against the reported detection. It investigates all the feedback and comes up with a threshold. Based on the already generated threshold and neighbor response, the SDN controller decides whether it is an attack. The top level view of this interactions are in the sequence diagram Fig.-3.3.

3.3 System Attacker Model

We focus on a specific set of attacks associated with position falsification to show the efficiency of our approach.

It is one of the fundamental and, most crucial issue to consider the vehicle on broad will not be able to transmit a false location. Considerable different variations of position falsification attacks [50] are in Table-3.1:

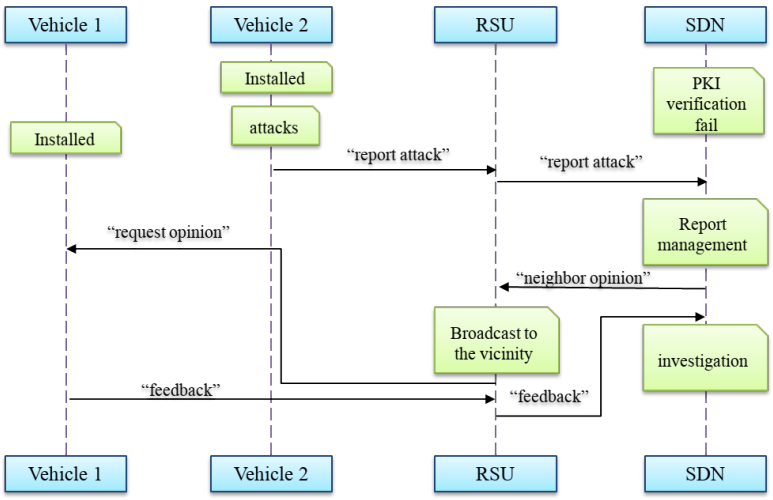

FIGURE 3.3. Sequence diagram of the reports investigation

ID: Attack	Detail	Parameters
1: Constant	Transmits a fixed location	(x, y)
2: Constant offset	a fixed offset added to the real location	$(\Delta x, \Delta y)$
4: Random	Transmits a random location	uniformly random constant in playground
8: Random Offset	Transmits a random bounded by a rectangle	$(\Delta x, \Delta y)$ bounded range $[-a, a]$
16: Eventual Stop	transmits the same location repeatedly	eventually stop (stop $probability = p$)

TABLE 3.1. Position falsification attack and associate parameters

(A) **Constant position attack:** In this type, the attacker transmits a fixed, pre-configured position. As in Fig.3.4, the attacker sends a constant pre-configured $(x = 5560, y = 5820)$ position.

(B) **Constant offset Position attack:** The attacker always generates a fixed offset, and add to the real locations, as in the Fig -3.4 a constant offset $(x = 250, y = -150)$ added; It is hard to detect by using a traditional misbehavior detection system since the offset can be a small number and does not make a big difference with a regular trip.

(C) **Random position attack:** The attacker transmits a random position from the simulation area, which is a newly generated random message. The transmitted position could be any values from the playground; as shown in Fig-3.4, it does not seem the road trip at all.

(D) **Random offset attack:** This case of attack generates a random offset from a pre-configured rectangular area around the vehicle, which could be considered a close variation of the random attacks. However, in this case, the vehicle chooses a random value that ranged over the rectangle region, as in Fig-3.4.

(E) **Eventual stop attack:** An attacker generally behaves for some time and then attacks by transmitting the same position repeatedly (i.e., as if it had stopped), Fig-3.4 represents a scenario, where two green triangles to show the start and endpoint of the trip. This attack can be very harmful as the attacker can pretend not to be on board.

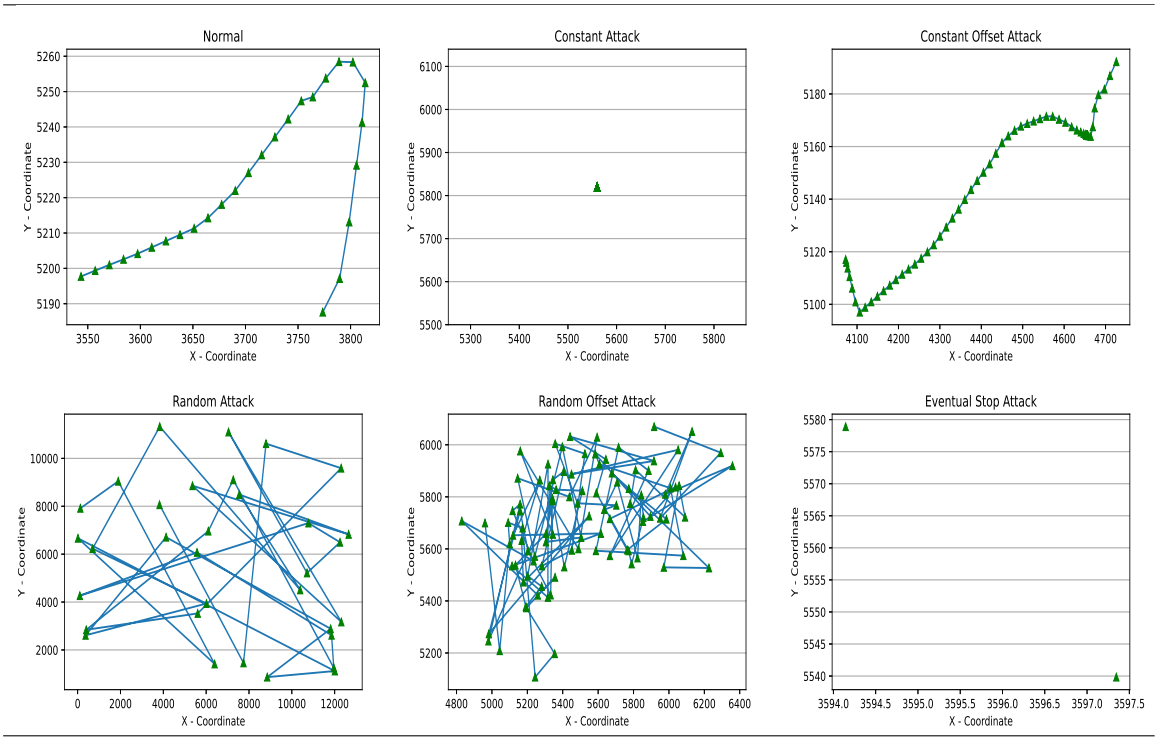

FIGURE 3.4. Different Attacker Scenarios

3.4 Building a model for ML-based Detection

Our detection mechanism has several stages based on the fundamental of supervised machine learning techniques, and these phases are as described in Fig- 3.5:

3.4.1 Data Set

Our approach's goal is to use machine learning techniques for misbehavior detection to come up with a smart model from finite training data and correctly classify future data as malicious or usual. As we all know, machine learning techniques entirely rely on the training dataset, so it is essential to select the right dataset. Different approaches could be taken to select the dataset, such as real-world scenario testing, analytical models, and a simulation-generated dataset. One of the biggest challenges is interpreting

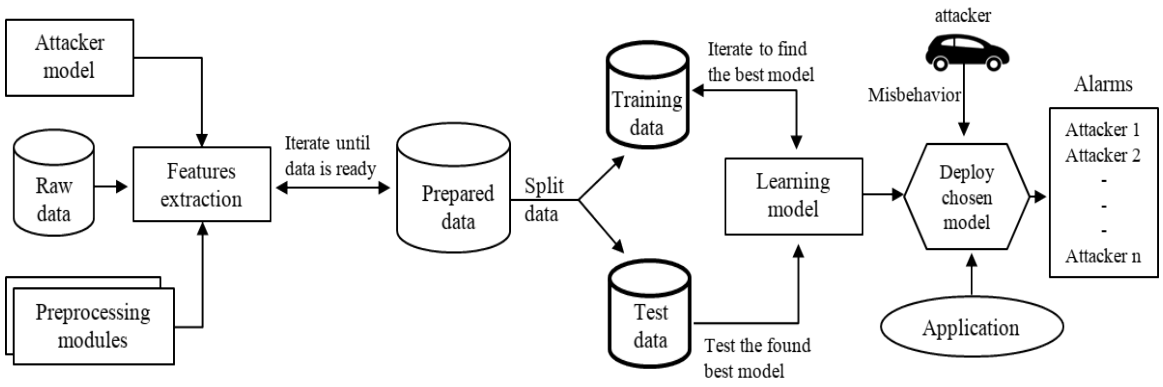

FIGURE 3.5. Machine learning based misbehavior system.

locally collected data to find patterns or merge results from a set of participating nodes into one coherent training data, and several schemes adapt ideas from machine learning [98], [99]. These approaches provide useful tools for analyzing data and deriving certain exciting features that might point at attacks, permanently the idea behind various machine learning approaches. A large volume of data is labeled and used to train a classifier algorithm, for this latter case, collecting labeled data that is sufficiently diverse is a considerable challenge [100].

It is obvious to say that one of the main issues of this research field is the luck of the dataset, only a few available datasets, but most of them are not an optimal choice due to the luck of the implemented attackers, BSMs broadcast rate, and the vehicle density as well. For that reason, we decided to use the VeReMi dataset [50], which has been published recently and publicly available for research purposes. As the dataset is already labeled, it is very convenient for our approach.

VeReMi Data Set:

The VeReMi dataset comprises of 5 position falsification attacks, three-vehicle densities (low, medium and high), three attacker densities (10, 20 and 30 percent), and each parameter set was repeated five times for randomization, has 225 individual simulations; the summary of the dataset is in the Table-3.2. The three density split into low density (corresponding to a run starting at 3:00) has 35 to 39 vehicles, while the medium density (a run at 5:00) has between 97 and 108 vehicles, and the high density (7:00) has between 491 and 519 vehicles. Out of these vehicles, a subset is malicious: this decision is made by sampling a uniform distribution $[0, 1]$ and comparing it to the attacker fraction parameter, essentially assigning each vehicle to be an attacker with that probability. For each scenario, a log file maintains the

record of BSMs received by a single-vehicle (same) from neighboring vehicles (300-meter range) during its entire journey. The implementation was done based one the principal of the sender-receiver pair, defining the complete trip for an individual sender. Each vehicle node keeps track of all the messages in a JSON file locally; one the other side, a Ground Truth file keeps track of the real messages which allow us to label the data as usual or an attacker.

VeReMi Data Set				
Time	Traffic Density	Number of Vehicles	Attacker Density	Number of Messages (sent)
3.00	Low	35 to 39	10%, 20%, 30%	908 to 1144
5.00	Medium	97 to 108	10%, 20%, 30%	3996 to 4489
7.00	High	491 to 519	10%, 20%, 30%	20482 to 21878

TABLE 3.2. Brief summary of VeReMi data-set

3.4.2 Feature extraction

For accurately detecting the position falsification attack, we need to extract representative features that characterize the different patterns of the attack. Figures 3.6 illustrates the variation of positions received from normal and suspicious vehicles. Each curve illustrates the variation in positions of a given vehicle. The blue curves represent the behaviors of normal vehicles, while the rest represent the behaviors of suspicious vehicles. As we can see, suspicious vehicles can have no variation in the consecutive positions or can a random variation in position every two consecutive positions, which can be interpreted as constant and random attackers, respectively.

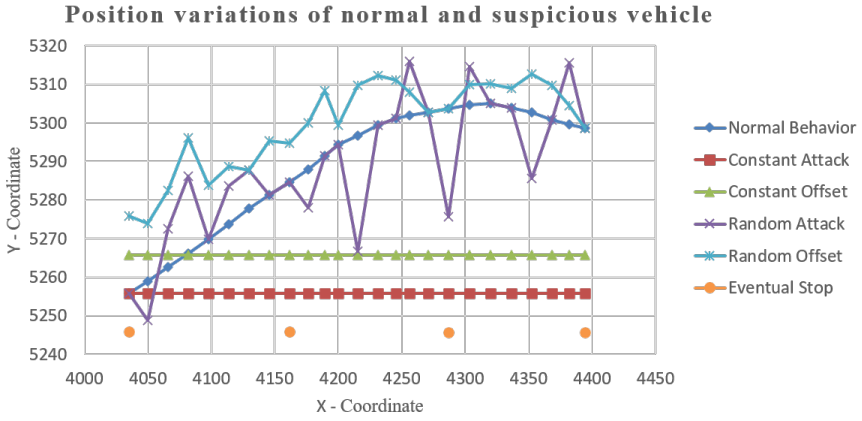

FIGURE 3.6. The variation in distance of normal vehicles and suspicious vehicles

Thus, the problem becomes how to select a set of features that can best represent the variation in position patterns. In our algorithm, we use the simple concept of sampling in signal processing to convert the variation of positions to a sequence of samples, and each sample will be used as one feature in the learning vector. Figure 3.7 shows an example of the process of feature extraction from the variation in positions trace. There are two main parameters, sampling length Δ and sampling interval δ, that determine the dimension and attribute value of the feature vector. Specifically, we split the curve into small segments of length δ and calculate the average distance x_i of each segment. A set of consecutive Δ samples constitute the final feature vector $X = [x_1, x_2, ..., x_\Delta]$.

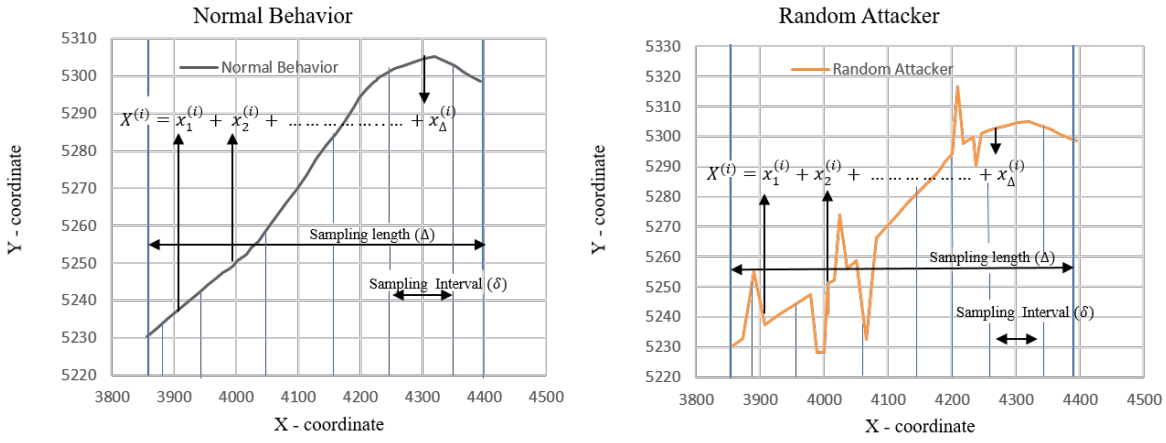

FIGURE 3.7. Sampling process

3.4.3 Training process

This stage aims to find an optimized model, a hyper-parameters set, and a training dataset. The labeling process aims to label an instance according to its actual class (e.g., misbehaving or not misbehaving) of the preprocessed dataset, and a sampling process divides the labeled dataset into multiple datasets. Then choose a common approach to divide the dataset into training and testing datasets. The first one is for training, while the second dataset is for testing the model. During the training process, the model learns to associate a given instance to its actual label and must learn to distinguish a misbehaving instance from a non-misbehaving one based on the actual label and learning features (each features column). The shuffling process aims to avoid having training data that contains only a dataset of misbehaving instances.

3.4.4 Testing

The testing stage evaluates the performance of a previously trained model on the testing data set and commonly has three classification, evaluation, and validation steps. The classification process aims to tests data set where each instance label is unknown; for example, in our case, this could be the learning features, the classifier model predicts the label of each encountered instance based on these learning features and classifies as an attack or normal. Evaluates the classifier performance on the testing data by doing cross-validation between the actual label and the predicted one, validation of the classifier model depends on several criteria.

3.5 Report investigation

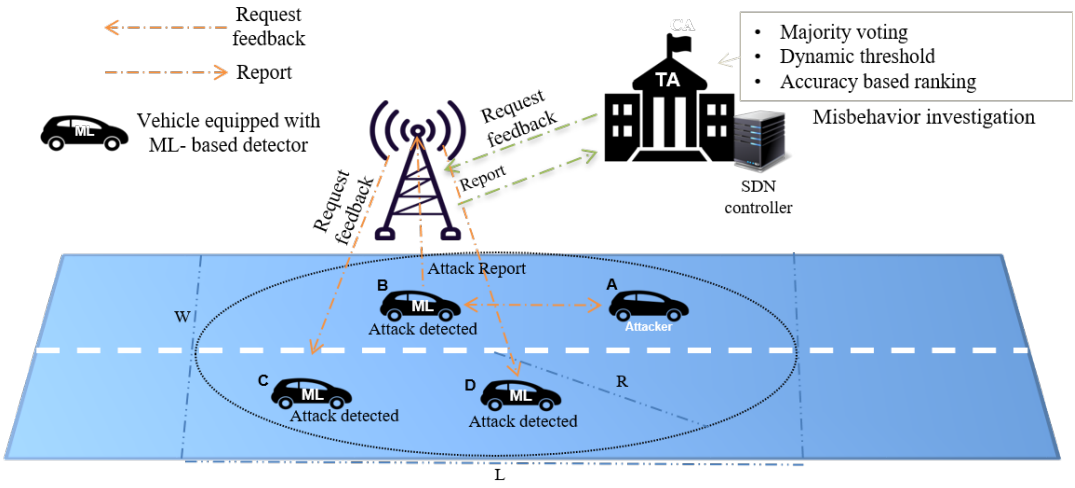

FIGURE 3.8. Report investigation

In order to increase the detection accuracy, we propose a report investigation system. When the vehicle detects an attack, it immediately sends a report about the detection to the controller. If the SDN controller receives a report, it requests all vehicles' opinions in the vicinity. The controller makes the final decision whether the vehicle is malicious or not based on a dynamic threshold. The threshold is basically the minimum number of attack reporting vehicles to confirm the attack. This threshold is dynamic .i.e. it is updated from time to time according to the number of attacks on the networks. The Fig.-3.8 shows the visual explanation of the process. As we can see, there is an attacker A, and the vehicle B can detect immediately. B sends reports to the controller with the help of RSU and the controller request

feedback of the neighborhood. Algorithm (1) represents a pseudo code that the SDN controller executes to determine the attacks. The attacks is added to the Attack Set (AS) only if the number of positive feedbacks are greater or equal to the dynamic threshold.

Algorithm 1: Position falsification attack detection

Data: Attack Report (TR)
Result: Attacker set (SA)
$nb\_report \leftarrow nb\_report + 1$
if $(nb\_report >= threshold)$ **then**
  $SA \leftarrow SA \cup ID_v$;

3.6 Conclusion

The vehicular communication system offers many exciting safety and comfort applications such as accident notification, traffic cognition alarm, mostly based on the vehicles' real-time position. The transmission of false positions can lead to the passenger's and drivers' life in danger and destroy the benefits of the system. Hence the system needs to be protected from false position attacks, needs to be detected immediately before something wrong happens. In this study, we proposed a misbehavior detection system to detect position falsification attacks based on the integration of ML and SDN, a novel approach from the best of our knowledge. Evaluation of the system by showing the accuracy of detection and other relevant matrices will be presented in later chapter 4 through an experimental simulation.

CHAPTER 4

Performance Evaluation

4.1 Introduction

It is essential to evaluate our proposed ML-based MDS to assess its efficiency of detecting position falsification attacks. This can be done through the building and testing of the ML-model using a realistic data set implementing position falsification attacks. To this end, this chapter focuses on describing the evaluation procedure and discussing the obtained results.

This chapter is organized as follows: Section 4.2 is designed to describe our machine learning methodology, which includes prepossessing, feature selection, evaluation metrics. The end of this section presents our obtained classification results. Section 4.3 presents a use case evaluation of the proposed architecture through simulation and compare the results with exiting works. Finally, section 4.4 concludes the chapter.

4.2 Machine Leaning Model Evaluation

Our proposed machine learning models should be able to provide reliable predictions to detect the attacks on actual use case circumstances. While training a model is a key, how the model concludes on unseen attacks is an equivalently significant aspect that should be examined. We need to recognize whether it operates and, consequently, if we can consider the model predictions. This subsection will illustrate the procedures used to provide the data pipeline, accompanied by evaluating how well the proposed model concludes to new, previously unseen position falsification attacks.

4.2.1 Experimental Setup

As discussed in the previous chapter, we will continue using the VeReMi dataset to estimate our system performances. We have also noticed that the complete dataset comprises 225 subdirectories for the modified parameter settings; we exercised a small subset from the random seed and combine them in a single CSV file to train our model and test it with the newly generated messages, Table - 4.1 represent the top-level view of the process.

Data Pipeline			
Repetition	Traffic Density	Attack Type	Attacker Density
2	Medium: 5	Constant: 1	20%
2	Medium: 5	Constant Offset: 2	20%
2	Medium: 5	Random: 4	20%
2	Medium: 5	Random Offset: 8	20%
2	Medium: 5	Eventual Stop: 16	20%

TABLE 4.1. Considered partial VeReMi data set as Pipeline

4.2.2 Prepossessing

To prepare the experimental instruments ready, we possessed to go through several preprocessing stages as follows:

(A) **Preparation:** We have downloaded the subset of the dataset, as specified beforehand, and received five different zip files for distinct kinds of attacks. The VeReMi is a simulated dataset was generated utilizing LuST (Version 2) and VEINS (with adjustments, based on version 4.6). Each simulation log holds a ground truth file (JSON) for all messages and a collection of message logs (JSON) for each vehicle. All JSON file is contacting both GPS data regarding the vehicle location and BSM messages received from other vehicles through DSRC.

The filename of a message log identifies the receiver by vehicle number and OMNeT++ module number, e.g., JSONlog-0-7-A0.json refers to the 0th vehicle with OMNeT++ module ID 7. A0 refers to the fact that this vehicle is not an attacker. In the case where A1 shows the fact that the vehicle is an attacker.

(B) **Cleaning & Integration:** This stage focuses on generating the data to train and test the ML model, cleans, and formats each source data into the preprocessed data set. Indeed, source

data contains duplicated, noisy, and unnecessary numbers of features. Therefore, we need to select the required features without compromising accuracy. To do so, we followed a few preprocessing steps:

The directory comprises 97-108 JSON log files and one ground truth for each particular attack. The number of log files depends on the number of vehicles engaging during the simulations and, we had to combine all the log files except the ground truth to a single CSV file. Furthermore, we prepared it for five separate attacks to come up with only one single CSV file for each kind of attack. The bottom line is to formulate a pandas data frame practicing Jupyter notebook, making our cleaning and filtering task more comfortable. We removed all the GPS (type: 2) messages and duplicated messages from our data frame and considered unique BSM (type: 3). The raw dataset has 16 features, but in our case, we only need the position coordinates, and the sending time of the messages, after removing all unnecessary features and cleaning, our Dataframe as in Fig.-4.2.

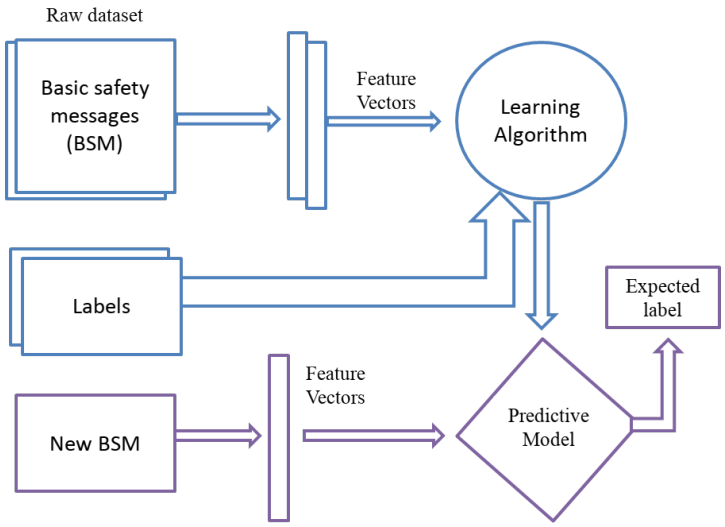

FIGURE 4.1. Overview of training & testing process.

(C) **Labeling, Shuffling** & **Sampling:** The labeling process intends to label an instance according to its actual class (e.g., misbehaving or not misbehaving). Our classifier must learn to recognize a misbehaving instance from a non-misbehaving one. Therefore, we label each occurrence non-misbehaving or misbehaving within the preprocessed dataset by assigning 0 or 1.

The Shuffling process sets a random situation order addressed in preprocessed data. This process avoids owning testing data that contains only a dataset of misbehaving cases; we did shuffle the dataset before spilled into training and testing.

The Sampling process distributes the preprocessed dataset into multiple datasets. A common approach is to divide the dataset into training data and testing data. The first data is for training the classifier model, while the second dataset is for testing the model performance. In this study, we used 70% of the sample occurrences from the preprocessed dataset for training, whereas the rest for testing.

	type	rcvTime	sendTime	senderID	messageID	x_pos	y_pos	distance
0	3	18002.18637	18002.18626	400007	5291	3616.163958	5388.181234	0.000000
1	3	18010.18637	18010.18626	400007	30963	3593.565431	5626.343766	239.232283
2	3	18017.18637	18017.18626	400007	57443	3587.657026	5835.470586	209.210268
3	3	18021.18637	18021.18626	400007	73483	3591.421790	5954.945337	119.534052
4	3	18004.18638	18004.18626	400007	10454	3609.306716	5447.598511	507.661967
...	...	...	...	...	...	...	...	...
4033	3	18095.17297	18095.17285	400637	295874	3610.060107	6053.176872	81.956315
4034	3	18099.17297	18099.17285	400637	309247	3600.592503	5944.315787	109.272006
4035	3	18097.17297	18097.17285	400637	302669	3604.210629	5998.849651	54.653757
4036	3	18098.21936	18098.21925	400643	306151	6149.505766	5824.058193	2551.289750
4037	3	18099.21936	18099.21925	400643	309475	6148.142454	5823.920847	1.370213

4038 rows × 8 columns

FIGURE 4.2. Screen shoot of clean and integrated data Frame

4.2.3 Features Selection

It is one of the most important steps to choose the right number of features. We tried with different numbers of features ranging from 5 to 300. These features were generated from the exchanged messages between each vehicle, as discussed earlier. We considered the distance between two consecutive position by using the following formula:

$$d_n = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{4.1}$$

where d_n represent distance between two positions (x_i, y_i) and (x_{i+1}, y_{i+1}) and n, i are integers. The Fig. - 4.3 shows a screed shoot our generated features data frame.

d1	d2	...	d12	d13	d14	d15	d16	d17	d18	d19	d20	level
598.732284	239.727566	...	59.955581	29.971150	209.826113	419.487503	389.548644	509.283938	509.283938	269.624906	479.197739	0
59.980148	89.930324	...	628.604140	29.976379	389.336714	359.597272	658.596399	329.574412	329.574412	598.807562	509.134521	0
0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1
5818.807194	4605.504671	...	3282.133912	3585.289400	6409.298319	8503.645410	5296.821681	3267.897435	3267.897435	2924.163661	1722.094213	4
8596.896165	8139.176925	...	5386.251686	3214.118004	1558.446807	7640.571001	2670.074594	3236.600555	3236.600555	3377.269423	5599.201353	4
...	...	...	...	...	...	...	...	...	...	...	...	...
363.542176	441.698906	...	387.591372	304.874437	21.511682	707.807572	224.163789	146.985494	146.985494	1026.348373	835.046192	8
207.325919	310.365322	...	426.293758	450.681722	547.706139	303.372414	390.306474	448.480283	448.480283	302.051942	255.017104	8
0.000000	0.000000	...	3.733083	3.733083	0.000000	0.000000	0.000000	8.418572	8.418572	0.148510	0.616897	16
0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	29.355369	29.355369	0.000000	0.000000	16
0.000000	0.000000	...	0.000000	0.000000	0.000000	0.000000	0.000000	88.535908	88.535908	0.040072	0.000000	16

FIGURE 4.3. Screen shoot of our generated features data

4.2.4 Metrics

Detection achievement is a complex task, varies across publications to publications, depending mostly on the detector's purpose. In misbehavior disclosure, it is intelligent to use false positive/negative valuations or comparable metrics to determine how well attacks are detected. The metrics we are going to examine for our model evaluation are accuracy, precision, recall, and F-1 score, see the Table.-4.3 for the definition of each term.

Confusion Matrix		
	Predicted	
Actual	Normal	Misbehaving
Normal	TP	FN
Misbehaving	FP	TN

TABLE 4.2. Binary classification Matrix

Confusion matrix: The Confusion matrix is one of the most intuitive and most comfortable metrics used for evaluation of the classification model, see Table-4.2. The terms associated with confussion metrics is shown in the Table-4.3 Using different partitions of the confusion metric, we can calculate the following detection evaluation metrics: Accuracy, Precision, Recall and F-1 score.

Terms associated with confusion matrix		
Terminology	Notation	Short definition
True Positive	TP	instance predicted as non misbehaving and truly non misbehaving
False Positive	FP	instance predicted as non misbehaving but is labeled as misbehaving
True negative	TN	instance predicted as misbehaving and truly misbehaving
False negative	FN	instance predicted as misbehaving but is labeled as non misbehaving

TABLE 4.3. Terms related to the confusion matrix

To evaluate the detection quality, the metric we use based on the well-established confusion matrix. There are many options to choose from, but we only choose accuracy, precision, recall, and F-1 score, and the formulas to calculate them is in the Table-4.4. The accuracy is the ratio of all correctly detected over all the considered detection. The precision indicates the classifier's ability to distinguish between misbehaving and palpable nodes; for example, a low precision means the system is yielding a lot of false positives. The Recall mark the classifier's ability to detect a misbehaving node, i.e., a low recall means an attack is difficult to detect. The F-1 score is the harmonic mean between the Recall and Precision. In our case, it could be considered as a measure of the overall detection quality.

Evaluation matrix Formulas	
Metric	Equation
Accuracy:	$\frac{TP+TN}{TP+FP+TN+FN}$
Precision:	$\frac{TP}{TP+FP}$
Recall:	$\frac{TP}{TP+FN}$
F1- score:	$2 * \frac{Precision * Recall}{Precision + Recall}$

TABLE 4.4. Formulas related to the evaluation matrix

4.2.5 Classifier Results for VeReMi data set

In order to better analysis of the pipeline data and the problem we are solving, we perform binary classification (one vs. one) and multi classification on the VeReMi data set using the six most efficient ML algorithms. The most relevant results are discussed in this subsection:

Binary Classification: The actual output of binary classification algorithms is a prediction level. The level indicates whether the observation should be classified as normal or misbehaving. We interpret the level by assigning 0 for normal vehicles an 1 for attacks and the experimental results of all the trained algorithms listed in Table-4.5. To make the results more visual, we have plotted in a bar diagram, see Fig.-4.4, showing in some cases do not constitute good results. Since the data is unbalanced and few rows have inconsistency due to the limitation of the simulation. To evaluate the detection results we consider the following detection metrics: $Accuracy$, $Precision$, $Recall$ and $F_1 score$, as shown in the Table- 4.5. The first thing we notice the detection quality largely depends on the type of misbehavior.misbehavior.

Evaluation Metrics						
	Accuracy					
Misbehavior Type	SVM	DT	RF	KNN	NaiveBayes	LR
Constant	0.916	**1.000**	0.960	0.965	0.965	0.833
ConstPosOffset	**0.833**	0.583	0.416	0.416	0.416	0.583
RandomPos	**1.000**	**1.000**	**1.000**	0.930	**1.000**	**1.000**
RandomPosOffset	0.708	0.750	0.708	0.708	**0.783**	0.654
EventualStop	**1.000**	**1.000**	0.916	0.916	0.910	0.675
	Precision					
Misbehavior Type	SVM	DT	RF	KNN	NaiveBayes	LR
Constant	0.888	**1.000**	**1.000**	0.900	**1.000**	**1.000**
ConstPosOffset	0.454	0.547	0.666	**1.000**	**1.000**	0.530
RandomPos	**1.000**	**1.000**	**1.000**	0.830	**1.000**	**1.000**
RandomPosOffset	0.500	0.571	0.500	0.500	0.500	**0.764**
EventualStop	**1.000**	**1.000**	0.714	0.714	0.714	0.745
	Recall					
Misbehavior Type	SVM	DT	RF	KNN	NaiveBayes	LR
Constant	0.888	**1.000**	0.900	0.900	0.900	0.712
ConstPosOffset	0.833	0.706	**1.000**	0.666	**1.000**	0.730
RandomPos	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
RandomPosOffset	**0.857**	0.571	0.485	0.425	0.428	0.540
EventualStop	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.670
	F1 - score					
Misbehavior Type	SVM	DT	RF	KNN	NaiveBayes	LR
Constant	0.964	**1.000**	0.960	0.965	0.960	0.833
ConstPosOffset	0.871	0.583	0.542	0.542	0.544	0.534
RandomPos	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
RandomPosOffset	0.708	0.750	0.701	0.701	**0.712**	0.710
EventualStop	**1.000**	**1.000**	0.921	0.921	0.9212	0.923

TABLE 4.5. Experimental results achieved by different classifiers, best results are in bold.

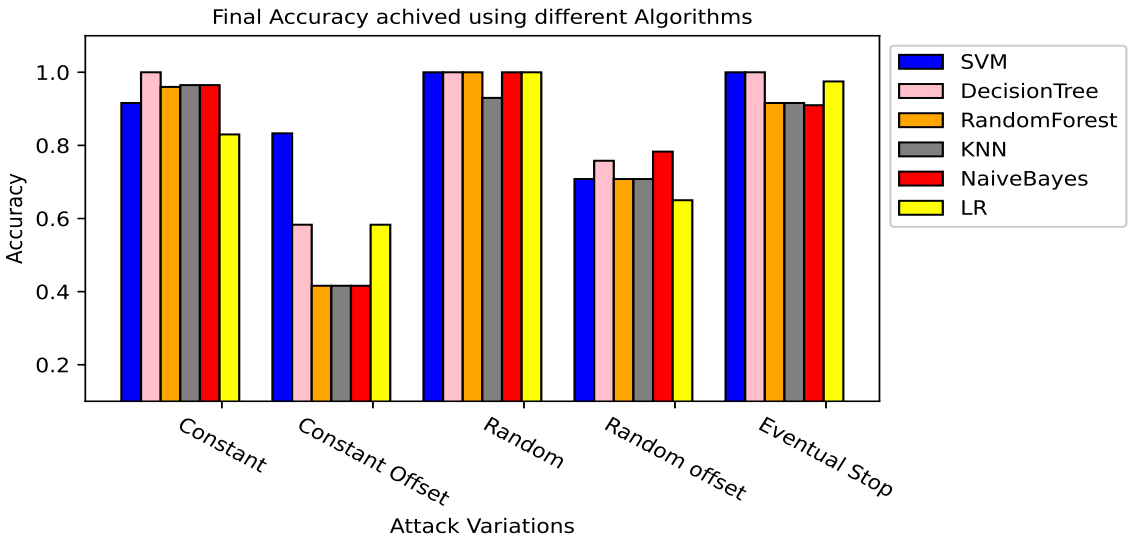

FIGURE 4.4. Accuray vs attack variations using different algorithms

(A) **Constant Attack:** The Decision Tree (DT) algorithm perform well to detect constant attack and was able to get 100% accuracy with same percents of precision, recall and F_1- score. However, rest of the algorithms were able to get 0.965% accuracy with different variations of the rest of the considered metrics.

(B) **Constant offset Attack:** Since constant offset is considered one of the most difficult attack to detect as mentioned through the literature, but in our case Support Vector Machine (SVM) was able to get 0.833% accuracy which is already higher than [93] and [75]. K-nearest neighbour (KNN) and Naive Bayes (NB) were able to get 100% precision, Random Forest (RF) is showing 100%. Though rest of the algorithms is showing significant decrease of the metrics.

(C) **Random Position Attack:** The results showing all detector algorithms were able to detect accurately the random position attacker except KNN algorithms has 0.830% accuracy.

(D) **Random Position offset Attack:** As mentioned earlier because of the limitation of the VeReMi data set and the offset attacks implementation our classifier was not able to show efficiency bout still we are able to show the better performance than the exiting work since we got 0.783% accuracy from NB algorithm.

(E) **Eventual stop Attack:** For eventual stop attack the SVM and DT were able to get the best results with efficient precision, recall and F_1-score.

The binary classification allow us to find misbehaving vehicle without knowing the types of misbehavior. However, in real use case it's also expected to know the type of the attacks in order to take action to

against the attacker. This is why, we also train a multi-class classifier which can help us to distinguish between different types position falsification attacks.

Multi-class classification: We perform a multi-class classification which takes into the account each class of position falsification attacks. Normal vehicles are labelled as 0, whereas attackers are labeled with their corresponding attacks ids. The obtained results are shown in the Table-4.6, which summarized the Logistic Regression (LR) has the better performance (74% scores) compared to the other algorithms. The multi-class classification provides more realistic use case scenario since it helps us to analysis

Evaluation Metrics (multi-classification)						
	ML- model					
Evaluation matrix	SVM	DT	RF	KNN	NaiveBayes	LR
Accuracy	0.633	0.733	0.696	0.422	0.400	**0.744**
Precision	0.633	0.707	0.696	0.416	0.410	**0.744**
Recall	0.630	0.734	0.696	0.430	0.410	**0.744**
F1-score	0.633	0.733	0.696	0.400	0.400	**0.744**

TABLE 4.6. Experimental results achieved by different classifiers for multi class classification, best results are in bold.

the corresponding types of attacks. However, as we can see from the table -4.6, our ML model is not able to get good accuracy using VeReMi data set. This is because that VeReMi data set presents some inconsistencies. For example, in many cases, eventual stop attacks were simulated as the same way as constant position attacks but they are considered as different attacks on the ground truth file. To well evaluate our multi-class classifier, we decided to generate our own data set through simulations, which is consistent and makes a clear difference between position falsification attacks.

4.3 Simulation

We have carried out a set of simulations to validate the performance of our proposed ML-based MDS. These simulations are conducted using the Veins Simulation Framework [101]. The main foundations of Veins are based on two well-established simulators OMNet++and SUMO [102]. These simulation tools are bi-directionally coupled and communicate through a Transmission Control Protocol (TCP) socket during the simulation run-time. Table 4.7 summarizes the simulation parameters.

4.3.1 Parameters settings

As illustrated in Figure 4.5, the considered scenario models the traffic of the city of Manhattan New York, USA using SUMO. We focused on a region of interest of dimensions 2km x 2km. The vehicles were generated using SUMO to take trips of 5 min duration over the city. We considered 100 vehicles, 30% of them are malicious. In our simulation, we have considered more advanced position falsification called attack-and-stop attack, which is an extension of eventual-stop attack. In attack-and-stop attack, the attack periodically switches between the attack behavior were null positions are sent and the normal behavior. The Table -4.8 describe the parameters of consider attackers.

Parameter	**Value**
Simulation duration	300 s
Transmission Range	500 m
Number of vehicles	100
Ratio of misbehaving vehicles	{30%}
Number of constant attackers	6
Number of constant offset attackers	6
Number of random attackers	6
Number of random offset attackers	6
Number of attack-and-stop attackers	6

TABLE 4.7. Simulation Parameters

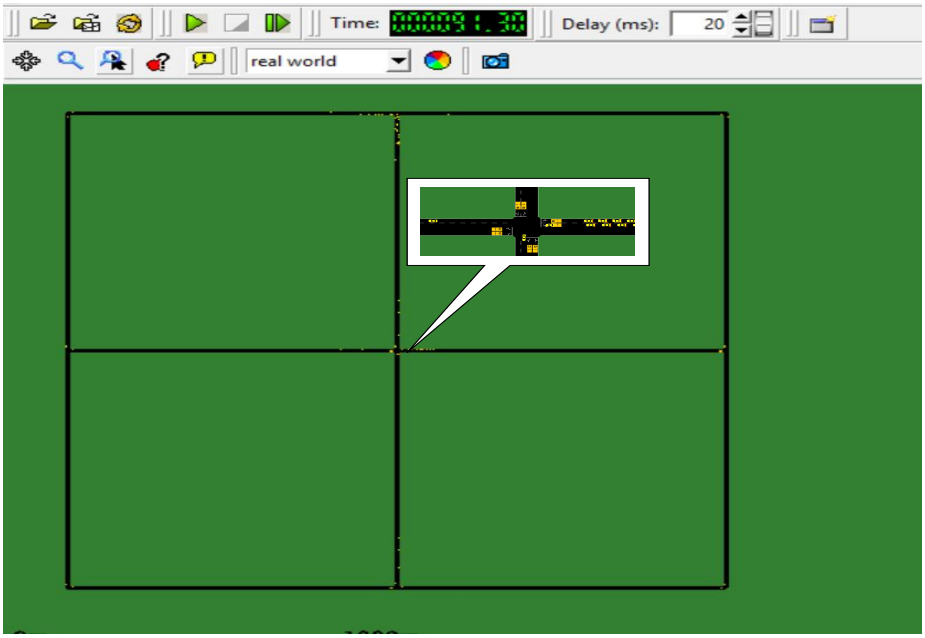

FIGURE 4.5. The scenario of simulation

In our consideration the constant attacks transmits a pre configured constant position $(x = 500, y = 500)$, where the constant offset always transmits position after adding a offset $(x = 200, y = 200)$.

Attack	**Description**
Constant	(x = 500 ; y = 500)
ConstantPosOffest	(x = X_position+200 ; y = Y_position+200)
Random	(x= random(100,1800) ; y= random(100,1800))
RandomPosOffest	(x = X_position+random(1,200) ; y = Y_position+random(1,200)
Attack-and-stop	periodically (attack (x=0 ;y=0) for 5s and stop (x = X_position ; y = Y_position) for 5s)

TABLE 4.8. Simulation Parameters

4.3.2 Simulation Results

We performed the same experiments as we did with the VeReMi data set to show the consistency of our data set and the efficiency of the proposed ML- based detection system. The results obtained is in the Table - 4.9, and the DT algorithm showing the efficiency with 100% percent scores. Where rest of the algorithms have the scores higher than 94% percent, which shows the efficiency of the built multi-class classifier.

Evaluation Metrics (multi-classification)						
	ML- model					
Evaluation matrix	SVM	DT	RF	KNN	NaiveBayes	LR
Accuracy	0.983	**1.000**	0.940	0.947	0.982	0.992
Precision	**0.965**	1.000	0.919	0.948	0.771	0.948
Recall	**0.965**	**1.000**	0.947	0.947	0.822	0.991
F1-score	0.965	1.000	0.947	0.944	0.944	0.983

TABLE 4.9. Experimental results achieved by different classifiers for multi class classification using our simulator generated data set, best results are in bold.

In vehicular network, the earlier detection of any attacks is always very important. In that case the system require less messages to efficient detection grantee the earlier detection. In that case we need to find a minimum set of messages that required to identify the misbehaving vehicle in the network. To this end, we have added another experiment to find the optimal number of features required to achieved the desired accuracy. The Fig. - 4.6 shows the our proposed ML- based MDS is able to achieved 100 % accuracy with 22 features which implies that we only need 23 messages from a certain vehicles to classify as normal or attacker with the types of attack.

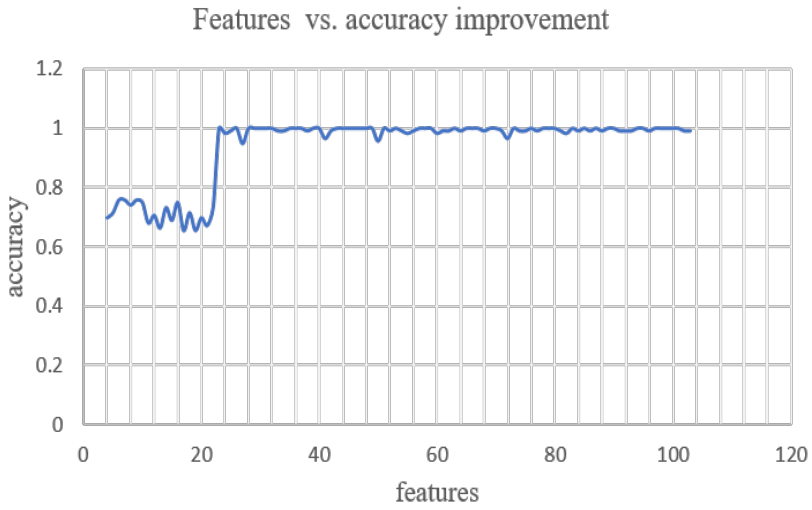

FIGURE 4.6. Accuracy improvement with respect to the number of features.

4.3.3 Results comparison

Figure 4.7 shows the accuracy of our built multi-class classifier on VeReMi data set and on the generated data set. We can see that the classification results obtained using the generated data set are better than the results obtained using VeReMi. Indeed, we get 100% of accuracy using DT. As we already, mentioned the low accuracy values obtained using VeReMi are due to the inconsistency of this data set. The results obtained using our generated data set demonstrate the effectiveness of our proposed classifier to detect position falsification attacks.

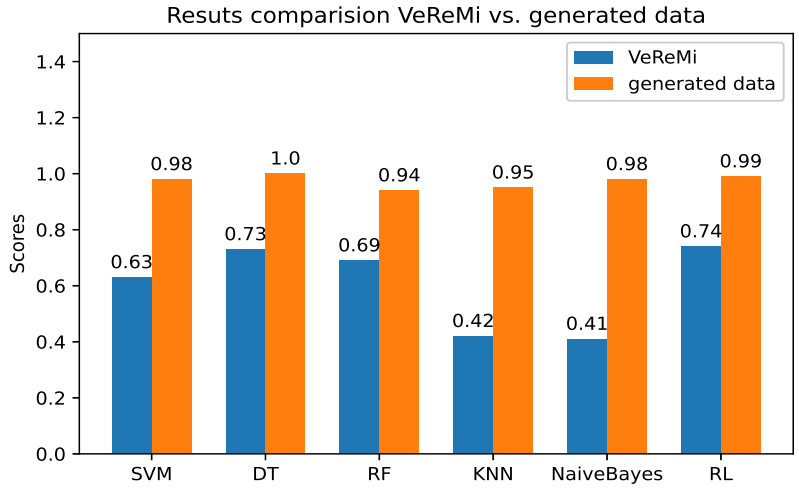

FIGURE 4.7. The final score comparison VeReMi vs our generated data set

4.4 Conclusion

This chapter has validated our proposed ML - base MDS using the VeReMi data set. Additionally, we have introduced a friendly simulator generated data set that includes five different position falsification attacks that can be considered as an extension of the VeReMi data set. We have also compared our results with the results obtained using the VeReMi data set. Our experiment also showed the optimal number of features need to detect position falsification attacks accurately. We have also provided a discussion on what should be preferred metrics for results comparison.

Conclusion

Vehicular networks offer exciting applications ranging from safety to comfort. However, the networks are vulnerable to many types of internal and external attacks, as summarized, such as position falsification attacks. This can lead to hazardous situations, especially road traffic management systems, increase the road accident's likelihood. The position falsification attacks need to be detected immediately before critical damage happens. Since the network's internal participant implements the position falsification attacks that can not be detected using popular cryptographic solutions only works for external attack prevention. Alternatively, MDS is considered an efficient way to detect position falsification attacks using different plausibility and consistency check to evaluate transmitted messages' correctness.

In this thesis, we inscribed the limitations of the existing solutions used to detect position falsification attacks. We proposed to use the machine learning models with the integration of SDN, which is unique and novel from the best of our knowledge. Our system design provides dynamic functionalities to update the ML models' time to time and improve detection performance without having a challenging task.

To facilitate the spread of our work, we implemented six different popular ML algorithms and evaluated them on the VeReMi dataset firstly. We have also conducted an inclination of simulations to validate the performance of our proposed ML-based MDS using the Veins Simulation Framework (OMNet++and SUMO). The results analysis showed the higher efficiency of our proposed work and produced dataset shows better performances for multi-classification, which is more realistic.

As future perspectives, we plan to develop and implement the report investigation for efficient attack detection. We also plan to carry out extensive simulations using multiple scenarios to validate the accuracy of our system.

Bibliography

[1] E Reinhard. “World report on road traffic injuries prevention”. In: *UN chronicle. June-August* (2004).

[2] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Tarek Bouali. “Predict and prevent from misbehaving intruders in heterogeneous vehicular networks”. In: *Vehicular Communications* 10 (2017), pp. 74–83.

[3] RELEASED TO A THIRD PARTY. “US Department of Transportation”. In: (2000).

[4] Markus Schütz and Klaus Dietmayer. “A flexible environment perception framework for advanced driver assistance systems”. In: *Advanced Microsystems for Automotive Applications 2013*. Springer, 2013, pp. 21–29.

[5] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[6] Kashif Dar, Mohamed Bakhouya, Jaafar Gaber, et al. “Wireless communication technologies for ITS applications [Topics in Automotive Networking]”. In: *IEEE Communications Magazine* 48.5 (2010), pp. 156–162.

[7] Jie Cui, Xiaoyu Zhang, Hong Zhong, et al. “Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment”. In: *IEEE Transactions on Information Forensics and Security* 15 (2019), pp. 1654–1667.

[8] Sateesh K Addepalli, Lillian Lei Dai, Raghuram S Sudhaakar, et al. *System and method for establishing communication channels between on-board unit of vehicle and plurality of nodes*. US Patent 8,718,797. 2014.

[9] Manu Sood and Shivani Kanwar. “Clustering in MANET and VANET: A survey”. In: *2014 international conference on circuits, systems, communication and information technology applications (CSCITA)*. IEEE. 2014, pp. 375–380.

[10] Raju Barskar and Meenu Chawla. “Vehicular ad hoc networks and its applications in diversified fields”. In: *International Journal of Computer Applications* 123.10 (2015).

[11] Nicholas Loulloudes, George Pallis, and Marios D Dikaiakos. “The dynamics of vehicular networks in large-scale urban environments”. In: *2015 IEEE Conference on Collaboration and Internet Computing (CIC)*. IEEE. 2015, pp. 192–199.

[12] Mohamed Nidhal Mejri, Jalel Ben-Othman, and Mohamed Hamdi. “Survey on VANET security challenges and possible cryptographic solutions”. In: *Vehicular Communications* 1.2 (2014), pp. 53–66.

[13] Wei Quan, Yana Liu, Hongke Zhang, et al. “Enhancing crowd collaborations for software defined vehicular networks”. In: *IEEE Communications Magazine* 55.8 (2017), pp. 80–86.

[14] Hassnaa Moustafa and Yan Zhang. *Vehicular networks: techniques, standards, and applications*. Auerbach publications, 2009.

[15] Georgios Karagiannis, Onur Altintas, Eylem Ekici, et al. “Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions”. In: *IEEE communications surveys & tutorials* 13.4 (2011), pp. 584–616.

[16] IEEE 1609 Working Group et al. “IEEE Standard for Wireless Access in Vehicular Environments (WAVE)-Multi-Channel Operation”. In: *IEEE Std* (2016), pp. 1609–4.

[17] Martijn van Eenennaam, Anne van de Venis, and Georgios Karagiannis. “Impact of IEEE 1609.4 channel switching on the IEEE 802.11 p beaconing performance”. In: *2012 IFIP Wireless Days*. IEEE. 2012, pp. 1–8.

[18] Fabio Arena, Giovanni Pau, et al. “A Review on IEEE 802.11 p for Intelligent Transportation Systems”. In: *Journal of Sensor and Actuator Networks* 9.2 (2020), p. 22.

[19] Abdullahi Chowdhury, Gour Chandra Karmakar, Joarder Kamruzzaman, et al. “Trustworthiness of Self-Driving Vehicles for Intelligent Transportation Systems in Industry Applications”. In: *IEEE Transactions on Industrial Informatics* (2020).

[20] David Eckhoff, Nikoletta Sofra, and Reinhard German. “A performance study of cooperative awareness in ETSI ITS G5 and IEEE WAVE”. In: *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. IEEE. 2013, pp. 196–200.

[21] Christian Cseh. “Architecture of the dedicated short-range communications (DSRC) protocol”. In: *VTC’98. 48th IEEE Vehicular Technology Conference. Pathway to Global Wireless Revolution (Cat. No. 98CH36151)*. Vol. 3. IEEE. 1998, pp. 2095–2099.

[22] Yunxin Jeff Li. “An overview of the DSRC/WAVE technology”. In: *International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*. Springer. 2010, pp. 544–558.

[23] Apostolos Papathanassiou and Alexey Khoryaev. “Cellular V2X as the essential enabler of superior global connected transportation services”. In: *IEEE 5G Tech Focus* 1.2 (2017), pp. 1–2.

[24] Donglin Wang, Raja R Sattiraju, Anjie Qiu, et al. “Effect of Retransmissions on the Performance of C-V2X Communication for 5G”. In: *arXiv preprint arXiv:2007.08822* (2020).

[25] Sherali Zeadally, Muhammad Awais Javed, and Elyes Ben Hamida. “Vehicular communications for its: Standardization and challenges”. In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 11–17.

[26] Anna Maria Vegni, Mauro Biagi, Roberto Cusani, et al. “Smart vehicles, technologies and main applications in vehicular ad hoc networks”. In: *Vehicular technologies-deployment and applications* (2013), pp. 3–20.

[27] Sandor Dornbush and Anupam Joshi. “StreetSmart traffic: Discovering and disseminating automobile congestion using VANET’s”. In: *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*. IEEE. 2007, pp. 11–15.

[28] Zehra Afzal and Manoj Kumar. “Security of Vehicular Ad-Hoc Networks (VANET): A survey”. In: *Journal of Physics: Conference Series*. Vol. 1427. 1. IOP Publishing. 2020, p. 012015.

[29] Chaker Abdelaziz Kerrache, Carlos T Calafate, Juan-Carlos Cano, et al. “Trust management for vehicular networks: An adversary-oriented overview”. In: *IEEE Access* 4 (2016), pp. 9293–9307.

[30] Yi Qian and Nader Moayeri. “Design of secure and application-oriented VANETs”. In: *VTC Spring 2008-IEEE Vehicular Technology Conference*. IEEE. 2008, pp. 2794–2799.

[31] Maxim Raya and Jean-Pierre Hubaux. “Securing vehicular ad hoc networks”. In: *Journal of computer security* 15.1 (2007), pp. 39–68.

[32] Irshad Ahmed Sumra, Iftikhar Ahmad, Halabi Hasbullah, et al. “Classes of attacks in VANET”. In: *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*. IEEE. 2011, pp. 1–5.

[33] Bryan Parno and Adrian Perrig. “Challenges in securing vehicular networks”. In: *Workshop on hot topics in networks (HotNets-IV)*. Maryland, USA. 2005, pp. 1–6.

[34] Oscar Puñal, Ana Aguiar, and James Gross. “In VANETs we trust? Characterizing RF jamming in vehicular networks”. In: *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*. 2012, pp. 83–92.

[35] Hu Yih-Chun and Adrian Perrig. “A survey of secure wireless ad hoc routing”. In: *IEEE Security & Privacy* 2.3 (2004), pp. 28–39.

[36] Vimal Bibhu, Roshan Kumar, Balwant Singh Kumar, et al. “Performance analysis of black hole attack in VANET”. In: *International Journal Of Computer Network and Information Security* 4.11 (2012), p. 47.

[37] Jingsha He, Muhammad Salman Pathan, Nafei Zhu, et al. “An efficient scheme for detecting and isolating gray-hole attacks in AODV-based mobile ad hoc networks”. In: *Int. J. Communication Networks and Distributed Systems* 24.4 (2020), p. 339.

[38] Sukla Banerjee. “Detection/removal of cooperative black and gray hole attack in mobile ad-hoc networks”. In: *proceedings of the world congress on engineering and computer science*. 2008, pp. 22–24.

[39] Yih-Chun Hu, Adrian Perrig, and David B Johnson. “Wormhole attacks in wireless networks”. In: *IEEE journal on selected areas in communications* 24.2 (2006), pp. 370–380.

[40] Charles E Perkins and Elizabeth M Royer. “Ad-hoc on-demand distance vector routing”. In: *Proceedings WMCSA’99. Second IEEE Workshop on Mobile Computing Systems and Applications*. IEEE. 1999, pp. 90–100.

[41] David B Johnson, David A Maltz, Josh Broch, et al. “DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks”. In: *Ad hoc networking* 5.1 (2001), pp. 139–172.

[42] Irshad Ahmed Sumra, Jamalul-Lail Ab Manan, and Halabi Hasbullah. “Timing attack in vehicular network”. In: *Proceedings of the 15th WSEAS International Conference on Computers, World Scientific and Engineering Academy and Society (WSEAS), Corfu Island, Greece*. 2011, pp. 151–155.

[43] John R Douceur. *The Sybil Attack. Peer-To-Peer Systems: First International Workshop, Iptps 2002, Cambridge, Ma, USA, March 7-8, 2002, Revised Papers*. 2002.

[44] Muhammad Rizwan Ghori, Kamal Z Zamli, Nik Quosthoni, et al. “Vehicular ad-hoc network (VANET)”. In: *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*. IEEE. 2018, pp. 1–6.

[45] Chakib Bekara and Maryline Laurent-Maknavicius. “A new protocol for securing wireless sensor networks against nodes replication attacks”. In: *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMOB 2007)*. IEEE. 2007, pp. 59–59.

[46] Zhaojun Lu, Gang Qu, and Zhenglin Liu. “A survey on recent advances in vehicular network security, trust, and privacy”. In: *IEEE Transactions on Intelligent Transportation Systems* 20.2 (2018), pp. 760–776.

[47] Farhan Ahmad, Asma Adnane, Virginia NL Franqueira, et al. “Man-in-the-middle attacks in vehicular ad-hoc networks: evaluating the impact of attackers’ strategies”. In: *Sensors* 18.11 (2018), p. 4040.

[48] Francesco Malandrino, Carlo Borgiattino, Claudio Casetti, et al. “Verification and inference of positions in vehicular networks through anonymous beaconing”. In: *IEEE transactions on mobile computing* 13.10 (2014), pp. 2415–2428.

[49] Anouar Boudhir, Mohammed Benahmed, Abderrahim Ghadi, et al. “Vehicular navigation spoofing detection based on V2I calibration”. In: *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*. IEEE. 2016, pp. 847–849.

[50] Rens W van der Heijden, Thomas Lukaseder, and Frank Kargl. “Veremi: A dataset for comparable evaluation of misbehavior detection in vanets”. In: *International Conference on Security and Privacy in Communication Systems*. Springer. 2018, pp. 318–337.

[51] Fatih Kurugollu, Syed Hassan Ahmed, Rasheed Hussain, et al. *Vehicular Sensor Networks: Applications, Advances and Challenges*. 2020.

[52] Syed Mohd Faisal and Taskeen Zaidi. “Timestamp Based Detection of Sybil Attack in VANET.” In: *IJ Network Security* 22.3 (2020), pp. 397–408.

[53] Asif Ali Wagan and Low Tang Jung. “Security framework for low latency vanet applications”. In: *2014 International Conference on Computer and Information Sciences (ICCOINS)*. IEEE. 2014, pp. 1–6.

[54] Yeongkwun Kim and Injoo Kim. “Security issues in vehicular networks”. In: *The International Conference on Information Networking 2013 (ICOIN)*. IEEE. 2013, pp. 468–472.

[55] Cezar Reinbrecht, Altamiro Susin, Lilian Bossuet, et al. “Gossip noc–avoiding timing side-channel attacks through traffic management”. In: *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE. 2016, pp. 601–606.

[56] Björn Wiedersheim, Zhendong Ma, Frank Kargl, et al. “Privacy in inter-vehicular networks: Why simple pseudonym change is not enough”. In: *2010 Seventh international conference on wireless on-demand network systems and services (WONS)*. IEEE. 2010, pp. 176–183.

[57] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. “Enhancing IEEE 802.11 p/WAVE to provide infotainment applications in VANETs”. In: *Ad Hoc Networks* 10.2 (2012), pp. 253–269.

[58] Mohammad Saeid Mahdavinejad, Mohammadreza Rezvan, Mohammadamin Barekatain, et al. "Machine learning for Internet of Things data analysis: A survey". In: *Digital Communications and Networks* 4.3 (2018), pp. 161–175.

[59] Daojian Zeng, Kang Liu, Siwei Lai, et al. "Relation classification via convolutional deep neural network". In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 2014, pp. 2335–2344.

[60] Ammar Haydari and Yasin Yilmaz. "Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey". In: *arXiv preprint arXiv:2005.00935* (2020).

[61] Avital Oliver, Augustus Odena, Colin A Raffel, et al. "Realistic evaluation of deep semi-supervised learning algorithms". In: *Advances in neural information processing systems*. 2018, pp. 3235–3246.

[62] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, et al. "Software-defined networking: A comprehensive survey". In: *Proceedings of the IEEE* 103.1 (2014), pp. 14–76.

[63] Soufian Toufga, Slim Abdellatif, Hamza Tarik Assouane, et al. "Towards Dynamic Controller Placement in Software Defined Vehicular Networks". In: *Sensors* 20.6 (2020), p. 1701.

[64] Bruno Astuto A Nunes, Marc Mendonca, Xuan-Nam Nguyen, et al. "A survey of software-defined networking: Past, present, and future of programmable networks". In: *IEEE Communications surveys & tutorials* 16.3 (2014), pp. 1617–1634.

[65] Pengyuan Du, Sobhan Nazari, Jorge Mena, et al. "Multipath TCP in SDN-enabled LEO satellite networks". In: *MILCOM 2016-2016 IEEE Military Communications Conference*. IEEE. 2016, pp. 354–359.

[66] Ramon Ferrús, Harilaos Koumaras, Oriol Sallent, et al. "SDN/NFV-enabled satellite communications networks: Opportunities, scenarios and challenges". In: *Physical Communication* 18 (2016), pp. 95–112.

[67] Wafa Ben Jaballah, Mauro Conti, and Chhagan Lal. "A survey on software-defined VANETs: benefits, challenges, and future directions". In: *arXiv preprint arXiv:1904.04577* (2019).

[68] David Antolino Rivas, José M Barceló-Ordinas, Manel Guerrero Zapata, et al. "Security on VANETs: Privacy, misbehaving nodes, false information and secure data aggregation". In: *Journal of Network and Computer Applications* 34.6 (2011), pp. 1942–1955.

[69] Philippe Golle, Dan Greene, and Jessica Staddon. "Detecting and correcting malicious data in VANETs". In: *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. 2004, pp. 29–37.

[70] Frank Kargl, Panagiotis Papadimitratos, Levente Buttyan, et al. “Secure vehicular communication systems: implementation, performance, and research challenges”. In: *IEEE Communications magazine* 46.11 (2008), pp. 110–118.

[71] Rens Wouter van der Heijden, Stefan Dietzel, Tim Leinmüller, et al. “Survey on misbehavior detection in cooperative intelligent transportation systems”. In: *IEEE Communications Surveys & Tutorials* 21.1 (2018), pp. 779–811.

[72] Ameneh Daeinabi and Akbar Ghaffarpour Rahbar. “Detection of malicious vehicles (DMV) through monitoring in Vehicular Ad-Hoc Networks”. In: *Multimedia tools and applications* 66.2 (2013), pp. 325–338.

[73] Rens van der Heijden, Stefan Dietzel, and Frank Kargl. “Misbehavior detection in vehicular ad-hoc networks”. In: *1st GI/ITG KuVS Fachgespräch Inter-Vehicle Communication. University of Innsbruck* (2013), pp. 23–25.

[74] Uzma Khan, Shikha Agrawal, and Sanjay Silakari. “A detailed survey on misbehavior node detection techniques in vehicular ad hoc networks”. In: *Information systems design and intelligent applications*. Springer, 2015, pp. 11–19.

[75] Steven So, Jonathan Petit, and David Starobinski. “Physical layer plausibility checks for misbehavior detection in V2X networks”. In: *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 2019, pp. 84–93.

[76] Fuad A Ghaleb, Mohd Aizaini Maarof, Anazida Zainal, et al. “Ensemble-Based Hybrid Context-Aware Misbehavior Detection Model for Vehicular Ad Hoc Network”. In: *Remote Sensing* 11.23 (2019), p. 2852.

[77] Jyoti Grover, Nitesh Kumar Prajapati, Vijay Laxmi, et al. “Machine learning approach for multiple misbehavior detection in VANET”. In: *International Conference on Advances in Computing and Communications*. Springer. 2011, pp. 644–653.

[78] Shie-Yuan Wang and Chih-Che Lin. “NCTUns 5.0: A network simulator for IEEE 802.11 (p) and 1609 wireless vehicular network researches”. In: *2008 IEEE 68th Vehicular Technology Conference*. IEEE. 2008, pp. 1–2.

[79] Mark Hall, Eibe Frank, Geoffrey Holmes, et al. “The WEKA data mining software: an update”. In: *ACM SIGKDD explorations newsletter* 11.1 (2009), pp. 10–18.

[80] Jyoti Grover, Vijay Laxmi, and Manoj Singh Gaur. “Misbehavior detection based on ensemble learning in VANET”. In: *International Conference on Advanced Computing, Networking and Security*. Springer. 2011, pp. 602–611.

[81] Joseph Kamel, Ines Ben Jemaa, Arnaud Kaiser, et al. “Misbehavior Detection in C-ITS: A comparative approach of local detection mechanisms”. In: *2019 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2019, pp. 1–8.

[82] Norbert Bißmeyer, Christian Stresing, and Kpatcha M Bayarou. “Intrusion detection in VANETs through verification of vehicle movement data”. In: *2010 IEEE Vehicular Networking Conference*. IEEE. 2010, pp. 166–173.

[83] Tim Leinmüller, Robert K Schmidt, and Albert Held. “Cooperative position verification-defending against roadside attackers 2.0”. In: *Proceedings of 17th ITS World Congress*. 2010, pp. 1–8.

[84] J. Kamel, M. R. Ansari, J. Petit, et al. “Simulation Framework for Misbehavior Detection in Vehicular Networks”. In: *IEEE Transactions on Vehicular Technology* (2020).

[85] Joseph Kamel, Arnaud Kaiser, Ines ben Jemaa, et al. “CaTch: a confidence range tolerant misbehavior detection approach”. In: *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2019, pp. 1–8.

[86] Fuad A Ghaleb, Anazida Zainal, Murad A Rassam, et al. “An effective misbehavior detection model using artificial neural network for vehicular ad hoc network applications”. In: *2017 IEEE Conference on Application, Information and Network Security (AINS)*. IEEE. 2017, pp. 13–18.

[87] Issam Mahmoudi, Joseph Kamel, Ines Ben-Jemaa, et al. “Towards a Reliable Machine Learning-Based Global Misbehavior Detection in C–ITS: Model Evaluation Approach”. In: *Vehicular Ad-hoc Networks for Smart Cities*. Springer, 2020, pp. 73–86.

[88] Dajiang Suo and Sanjay E Sarma. “Real-time Trust-Building Schemes for Mitigating Malicious Behaviors in Connected and Automated Vehicles”. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, pp. 1142–1149.

[89] Prinkle Sharma, David Austin, and Hong Liu. “Attacks on Machine Learning: Adversarial Examples in Connected and Autonomous Vehicles”. In: *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE. 2019, pp. 1–7.

[90] Steven So, Prinkle Sharma, and Jonathan Petit. “Integrating plausibility checks and machine learning for misbehavior detection in vanet”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 564–571.

[91] Anhtuan Le and Carsten Maple. “Shadows Don’t Lie: n-Sequence Trajectory Inspection for Misbehaviour Detection and Classification in VANETs”. In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE. 2019, pp. 1–6.

[92] Seong Hyeon Park, ByeongDo Kim, Chang Mook Kang, et al. “Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2018, pp. 1672–1678.

[93] Sohan Gyawali and Yi Qian. “Misbehavior detection using machine learning in vehicular communication networks”. In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–6.

[94] Lara Codeca, Raphaël Frank, and Thomas Engel. “Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research”. In: *2015 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2015, pp. 1–8.

[95] Sohan Gyawali, Yi Qian, and Rose Qingyang Hu. “Machine Learning and Reputation based Misbehavior Detection in Vehicular Communication Networks”. In: *IEEE Transactions on Vehicular Technology* (2020).

[96] Pranav Kumar Singh, Shivam Gupta, Ritveeka Vashistha, et al. “Machine Learning Based Approach to Detect Position Falsification Attack in VANETs”. In: *International Conference on Security & Privacy*. Springer. 2019, pp. 166–178.

[97] GGM Nawaz Ali, Md Noor-A-Rahim, Md A Rahman, et al. “An Efficient Cross-layer Coding-assisted Heterogeneous Data Access in Vehicular Networks”. In: *2018 IEEE International Conference on Communications (ICC)*. IEEE. 2018, pp. 1–7.

[98] Jihene Rezgui and Soumaya Cherkaoui. “Detecting faulty and malicious vehicles using rule-based communications data mining”. In: *2011 IEEE 36th Conference on Local Computer Networks*. IEEE. 2011, pp. 827–834.

[99] Maxim Raya, Panagiotis Papadimitratos, Virgil D Gligor, et al. “On data-centric trust establishment in ephemeral ad hoc networks”. In: *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE. 2008, pp. 1238–1246.

[100] Joseph Kamel, Michael Wolf, Rens Wouter van Der Heijden, et al. “VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs”. In: *IEEE International Conference on Communications (ICC)*. 2020.

[101] Christoph Sommer, David Eckhoff, Alexander Brummer, et al. “Veins: The open source vehicular network simulation framework”. In: *Recent Advances in Network Simulation*. Springer, 2019, pp. 215–252.

[102] SUMO. *Simulation of Urban MObility*. http://sumo.sourceforge.net/. June 2019.