6-18-2021

Position Falsification Detection in VANET with Consecutive BSM Approach using Machine Learning Algorithm

Aekta Sharma
University of Windsor

Position Falsification Detection in VANET with Consecutive BSM Approach using Machine Learning Algorithm

By

Aekta Sharma

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2021

Position Falsification Detection in VANET with Consecutive BSM Approach using Machine Learning Algorithm

by

Aekta Sharma

APPROVED BY:

B. Balasingam
Department of Electrical & Computer Engineering

S. Samet
School of Computer Science

A. Jaekel, Advisor
School of Computer Science

April 27, 2021

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship

I hereby declare that this thesis incorporates material that is the result of research conducted under the supervision of Dr. Arunita Jaekel. In all cases, the key ideas, primary contribution, experimental designs, data analysis, and interpretation were performed by the author, and the contribution of the co-author was primarily through providing feedback and the proofreading of the published manuscripts.

I am aware of the University of Windsor Senate Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my thesis and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis. I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my work.

II. Previous Publication

This thesis includes an original paper that has been previously submitted for publication as follows:

Section	**Publication title**	**Publication Status**
3.2, 3.3	A. Sharma and A. Jaekel, "Machine Learning Approach for Detecting Location Spoofing in VANET" International Conference on Computer Communications and Networks (2021)	Accepted

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution. I understand that my thesis may be made electronically available to the public.

ABSTRACT

Vehicular ad-hoc network (VANET) is an emerging technology for vehicle-to-vehicle communication vital for reducing road accidents and traffic congestion in an Intelligent Transportation System (ITS). VANET communication is vulnerable to various attacks and cryptographic techniques are used for message integrity and authentication of vehicles in order to ensure security and privacy for vehicular communications. However, if there is an inside attacker additional measures are necessary to ensure the correctness of the transmitted data. A basic safety message (BSM) is broadcasted by each vehicle in the network periodically to transmit its status. Position falsification is an attack where the attacker broadcasts a false BSM position, leading to congestion or even accidents. It becomes imperative to detect and identify the attacker to ensure safety in the network. Although many trust-based models are researched in the past, this research proposes a feasible and efficient data-centric approach to detect malicious behavior, using machine learning (ML) algorithms.

The proposed Machine Learning based misbehavior detection system utilizes labeled dataset called Vehicular Reference Misbehavior Dataset (VeReMi). VeReMi dataset offers five different types of position falsification attacks with different vehicle and attacker densities. This ML-based model uses two consecutive BSM approach to detect these attacks. Model classification on the Road-side Unit detects and could revoke malicious nodes from the network, reducing computational overhead on vehicles.

DEDICATION

I dedicate this thesis to my mom and dad, brother, bhabhi, my adorable niece Purvika, my friends for their support and encouragement and also my supervisor for her guidance throughout my research.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Dr. Arunita Jaekel and PhD student Muhammad Anwar Shahid, for guidance, support and encouragement throughout my research. I would moreover like to thank my friends Saiteja, Steffie and Atul who extended their care and support during my studies. Moreover, I would like to thank my internal reader Dr. Saeed Samet and my external reader Dr. Bala Balasingam for their support and feedback to improve my thesis.

Finally, I would like to thank my parents for providing the confidence and strength to complete my research and for being my pillar of strength.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

VANET	Vehicular ad-hoc network
RSU	Road-side Unit
OBU	On-board Unit
DSRC	Dedicated short range communication
C-V2X	Cellular Vehicle to everything
PKI	Public key infrastructure
BSM	Basic safety message
ITS	Intelligent Transportation system
WAVE	Wireless Access in vehicular Environment
VeReMi	Vehicular Reference Misbehaviour Dataset

CHAPTER 1

Introduction

1.1 Vehicular ad-hoc networks

Intelligent Transportation System [1] is an advanced technology that can improve road safety, traffic management and reduce traffic congestion in the transportation system. According to the 2018 Global status report on road safety by the World Health Organisation (WHO), road accidents and injuries have become the 8th leading cause of death with 1.35 million deaths annually. Moreover, road accidents are the 1st leading cause of death for children and young adults aged 5-29 years [2]. Vehicular ad hoc network (VANET) [3] is the emerging technology in the Intelligent Transportation System (ITS). VANET can make the transportation network more efficient, secure, and safe through information flow and communication. It is a highly dynamic wireless ad hoc network formed using vehicles, road-side units, and other infrastructures. As VANET has rapidly changing topology and high mobility, vehicles in the network can be stationary or continuously moving. Vehicles in the network are installed with On-Board Unit (OBU), which transmits a vehicle's status in the network to other nodes periodically. Road-side Unit (RSU) are infrastructures stationed on the road's side, which provides services and helps communication between the nodes in the network. There are other infrastructures, such as the Central Authority/ Authorization Party, which provides support such as registering a node in the network and revoking them in case of misbehaviour [4].

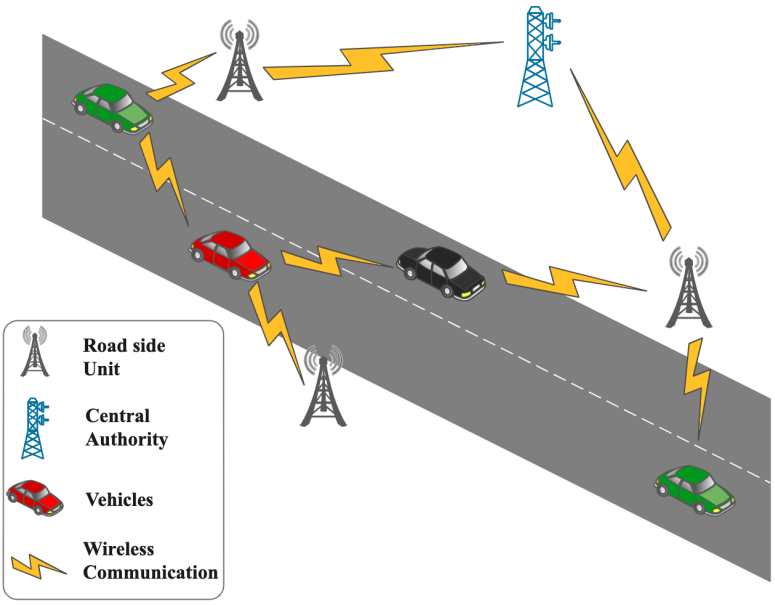


Figure 1.1: An example of Vehicular ad-hoc network

In 1999, the Federal Communication Commission (FCC) of the United States allocated Dedicated Short-Range Communication (DSRC), a licensed spectrum of 75MHz in 5.9 GHz frequency bandwidth for communication between vehicles and road-side units [5]. DSRC is a service used for short to medium-range communication that provides high data transfer with minimum latency. Wireless Access in Vehicular Environment (WAVE) is IEEE 1609 family standard protocol that uses the IEEE 802.11p standard to support communication in the vehicular network and provide standards for DSRC [6]. As DSRC has limitations in transferring a large amount of data and access the Internet of vehicles, a new standard is introduced, Cellular-V2X (C-V2X), which gives a better connectivity scope. C-V2X stands for the cellular vehicle to everything, and this cellular technology is designed to connect vehicles to other vehicles, road-side units, central authority and cloud-based services [7].

Communication in VANET is of different kinds such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Infrastructure-to-Infrastructure (I2I) and Vehicle-to-other devices (V2X). VANET supports two types of applications: Comfort application and Safety application. Comfort application includes comfort-based communication such as weather information, advertisement, pricing and details about nearest gas

stations or restaurants. However, the safety application includes safety-based communication between vehicles and infrastructures. Examples of safety applications are blind-spot warnings, emergency warnings, lane change assistance.

Wireless communication in the network can provide important information to the drivers or vehicles in time. However, wireless communication is vulnerable to various security and privacy attacks, which can cause misbehaviour in the network; hence, this information transmitted in the network must be verified and authenticated for correctness.

We can classify attackers in the network into the following [8]:

1. Insider vs. Outsider Attacker: Insider attackers are those who are authenticated members of the network, while outside attackers are those who are not authorized.

2. Active vs. Passive Attacker: Active attackers take part in the attack by directly interfering in the attack, such as altering the message or destroying the message packet in the network. Passive attackers listen to the conversation in the network without interfering directly and may use the information for malicious purposes.

3. Malicious vs. Rational Attacker: Attack that harms the network or causes extreme damage to the network by a malicious attacker. In comparison, rational attackers trigger the attack for personal gain.

Vehicles in the VANET network sends periodic status messages; such messages are called Basic Safety Messages (BSM). BSM contains the vehicle's current status, such as position coordinates, vehicle's speed, transmission time, which is broadcasted in the network periodically. These messages are digitally signed using cryptographic techniques [9] before broadcast, and only the authorized members of the network can access these BSMs.

VANET being a wireless network, is susceptible to attacks and detecting these attacks can be termed as misbehaviour detection. Misbehaviour detection can be

divided into *node-centric detection*, where the detection of misbehaviour depends on the credibility of the node and *data-centric detection*, where detection is based on data reliability.

This thesis aims to detect a Position falsification attack, where the attacker vehicle in the network sends a false position coordinate in the BSM. Position falsification attacks can lead to traffic congestion and even accidents and cause severe damage to the network.

Five types of position falsification attacks detected in this research are:

1. Constant attack: Attacker vehicle transmits fixed position in the network.
2. Constant offset attack: Attacker vehicle transmits a position with a fixed offset added to the actual position.
3. Random attack: Attacker vehicle transmits random position from the playground.
4. Random offset attack: Attacker vehicle transmits a uniformly random position from a pre-defined rectangle around the vehicle.
5. Eventual stop attack: Attacker vehicle behaves like a legitimate vehicle for some time and then transmits a current position repeatedly in the network.

Cryptographic techniques can provide message integrity but do not ensure message correctness; hence, cryptographic methods are insufficient to ensure network security. Additional detection methods are required to detect malicious vehicles sending false information in the network.

1.2 Motivation

VANET is highly dynamic as vehicles in the network are continuously moving and causing its topology to change every second. Communication between the VANET network is the crucial concept of VANET, where the information delay could be harmful to the network. Information authenticity, confidentiality and integrity is an

essential requirement for VANET to get implemented. Approaches like the Public key Infrastructure model (PKI) [10] provides authenticity in the network using a cryptographic technique such as digital signatures. PKI model only provides a secure infrastructure to manage identities and authenticate vehicles in the network, but PKI does not provide message integrity. PKI cannot alone detect if the information transmitted is correct; an additional misbehaviour detection model is vital to ensure message correctness.

VANET is prone to attacks [11], one such type of attack is position falsification attack, which this research aims to detect using a machine learning approach. In this attack, the sender vehicle transmits wrong information about position coordinate in the BSM and tries causing harm to the network by misguiding the legitimate vehicles. Malicious vehicles send false information in the BSM for their benefit, causing damage to the network. These malicious vehicles are the inside attackers with a rational or malicious motive to harm the network. These attackers are authenticated members of the network, and thus alone cryptographic techniques [12] [13] fail to identify such attackers. Position falsification attacks can cause severe damage to the network like traffic congestion or even accidents.

Researchers used machine learning algorithms to detect misbehaviour in the past but detecting misbehaviour in the network with a high correct detection rate is of the essence. This research aims to identify legitimate vehicles and attacker vehicles in the network with high accuracy using machine learning algorithms. Most approaches install the detection framework on the vehicles and expect the receiver vehicle to identify the attack. The consecutive BSM approach reduces the computational overhead on the vehicles and proposes an RSU based framework to identify position falsification attacks in VANET using machine learning algorithms.

1.3 Problem Statement

VANET can be of great benefit to Intelligent Transportation System to improve the road network. For VANET to function accurately, it needs to be safe and secure from

any type of attack. Position falsification is an attack that targets the integrity of the network. In this attack, the attacker can be an authenticated member who is dishonest to the network and somehow tampers with the vehicle's GPS and tries to send false position coordinates to the network. There could be another scenario where a legitimate vehicle has faulty GPS that transmits wrong position coordinates in the network. In both cases, the network can be harmed and can create an ambiguity in the network security. Previous approaches focus on vehicle-to-vehicle reliability to detect position falsification attacks; this research removes the vehicle's reliability on neighbouring vehicles and creates an RSU based approach. The research objective is to classify five different types of position falsification attacks for a different vehicle and attacker densities. The goal is to classify each attack with a high correct classification rate using machine learning algorithms with two consecutive BSM approach.

1.4 Solution Outline

The proposed solution to the problem is a generalized model to detect malicious nodes using a Data-centric approach with a low misclassification rate. As each vehicle will transmit BSM periodically in the network, two consecutive BSM from a vehicle is stored in a shared database by the RSU. A machine learning based classification model is installed at the RSU to classify vehicles into legitimate or attacker based on two consecutive BSM's from a vehicle. This model is trained on the attacker and legitimate vehicle dataset with different densities. The classification model is trained on all five types of position falsification attacks individually and combining them. Dataset used in this research is the first public extensible dataset available in the field of VANET: VeReMi Dataset (Vehicular Reference Misbehavior Dataset) [14]. The proposed solution consists of two main stages: the first stage is dataset preparation, followed by the second stage of classification. The first step includes extracting the Ground Truth file, including actual correct information and Log files containing false information and map both together. Extracted data is pre-processed, and two consecutive BSM dataset is generated. This generated dataset is passed on

to the second stage. In this stage, machine learning algorithms are implemented to classify the vehicles.

1.4.1 Contributions

The contribution of this research is summarized as follows:

- Efficient misbehaviour detection model to classify position falsification attack.
- Reduce computational overhead on OBU of vehicle.
- Remove vehicle-to-vehicle reliability in the network.
- New misbehaviour detection model with vehicle-RSU pair-based approach.

1.5 Thesis Organization

The remaining outline of this thesis is as follows: chapter 2 includes an overview of fundamental concepts of VANET and position falsification attack along with a literature review of related work in misbehaviour detection using machine learning approaches. Chapter 3 contains an outline of the proposed methodology and a brief discussion of the VeReMi dataset, followed by chapter 4, including experimental setup and discussion of results. In the end, chapter 5 gives a conclusion followed by possible future work to the proposed methodology.

CHAPTER 2

Literature Survey

2.1 Overview of VANET

The modern world has advanced in communication and technologies to the extent where various networks are established. VANET is one such network that gives a great possibility to expand the road network with comfort, safety and security of drivers. VANET can help provide road safety, reduce fuel consumption, CO2 emission and traffic congestion, eco-friendly driving, and convenience on the road. VANET can also provide commercial advantages such as advertising nearby restaurants, hotels, locating nearby gas stations. VANET is known for its nodes moving freely in the network, leading to rapid changes in its topology as vehicles in the network travel with high speed. Every vehicle in the network is independent and can potentially communicate with any other node in the network. VANETs can cover a large geographical area and its nodes are not restricted by limited battery storage and power supply. VANET consists of uniform and non-uniform regions. The uniform region is when a vehicle shares speed, path and direction with other vehicles for a long time, such as on highways. In contrast, the non-uniform region includes streets where vehicles do not share the same path, direction or speed and interacts with many vehicles in their journey.

2.1.1 Types of Communication

Communication in VANET is categorized to have a transient, short-lived interaction with minimum latency. Vehicles registered in the network are equipped with an

On-board Unit (OBU) to communicate with Road-side units (RSU); OBU provides information regarding the current position in the network using a Global positioning system (GPS). RSUs are the backbone of the network that facilitates communication in the network [15]. There are five different types of communication in VANET, and some are demonstrated in Figure 2.1.

1. Vehicle-to-Vehicle (V2V): Each vehicle in the network can communicate with other vehicles. A vehicle can broadcast a message in the network to multiple vehicles in its range.

2. Vehicle-to-Infrastructure (V2I): Vehicles can communicate to nearby infrastructures such as RSU's or central authorities to request services or update their current status in the network.

3. Infrastructure-to-Infrastructure (I2I): Infrastructures can also communicate with each other in the network to provide updated services to the nodes in the backend.

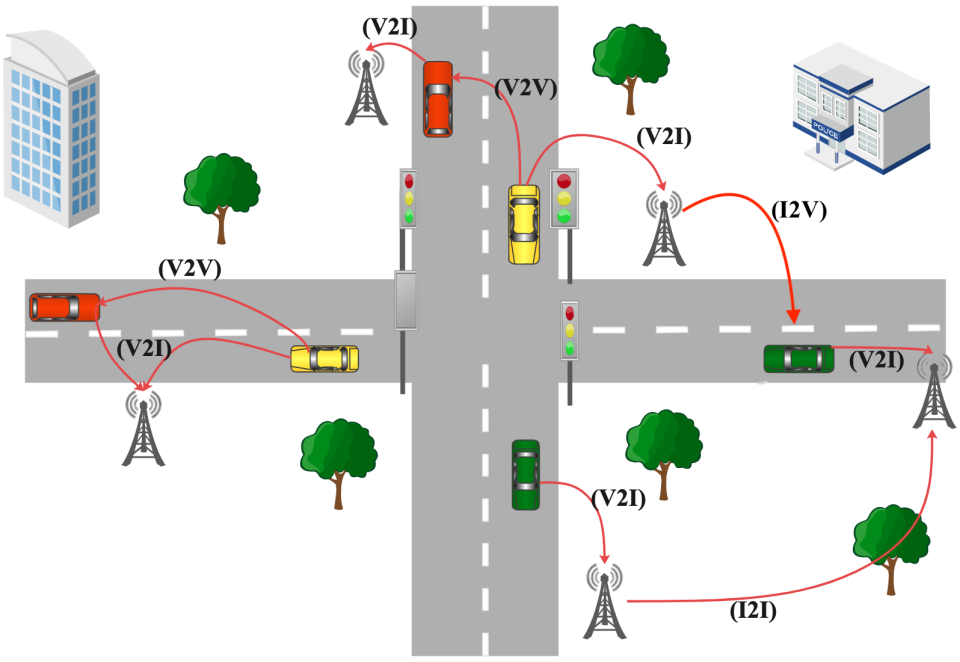


Figure 2.1: Types of communication in VANET

4. Infrastructure-to-vehicle (I2V): Infrastructure communicates with the vehicles to provide services to vehicles in the network. For example, RSUs broadcast the vehicles in their range with hazard warnings.

5. Vehicle-to-everything (V2X): Vehicles can also communicate with other devices such as mobile phones and internet-connected devices.

2.1.2 Security Requirements and Attacks in VANET

VANET offers facilities and services over the wireless channel, yet it has various drawbacks and is vulnerable to security and privacy threats and attacks. Some of the security attacks in VANET are shown in Figure 2.2. There are security requirements in VANET for the network to function correctly [16]. Susceptibility in the network can cause accidents and data loss. Wireless networks are prone to malicious attacks from attackers having different motives, as discussed in section 1.1. This section discusses the main security requirements in VANET as follows [17]:

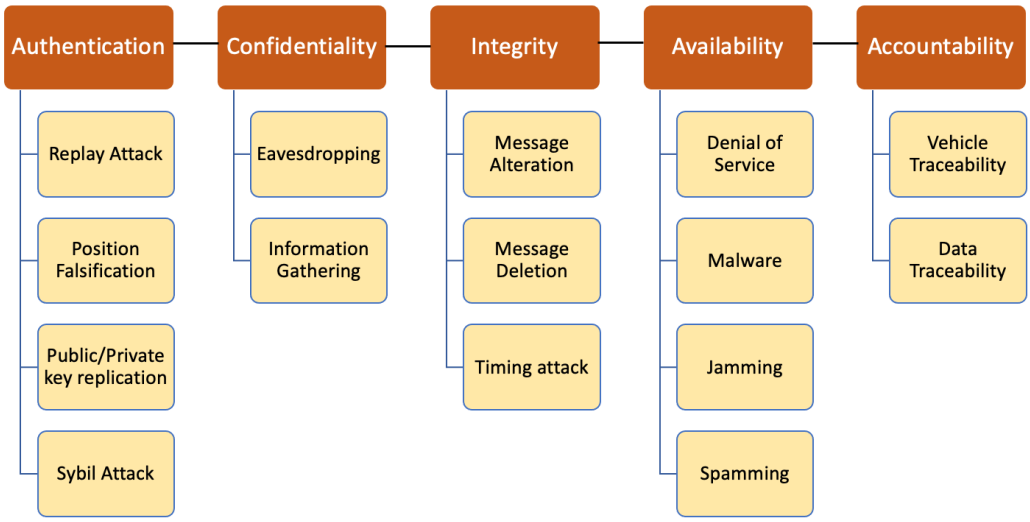


Figure 2.2: VANET attacks and threats

1. Authentication: Authentication is the process of verifying that members and the messages sent by them are legitimate [18]. The sender and receiver in the network should be authenticated member of the network. Information sent

and received must be authenticated to maintain the legitimacy of the network. Examples of authentication attacks are Replay attack, Position Falsification attack, Certificate Replication attack and Sybil attack. A Replay attack is where an attacker sends the same message with a different timestamp in the network [19]. In a Certification Replication attack, attackers have the replica of the public or the private key of the vehicle and try to send the false message by impersonating it as a legitimate vehicle [20]. In Sybil attack, attacker undertake multiple identities or create ghost vehicles in the network and mislead the legitimate vehicles by transmitting false messages [21].

2. Confidentiality: Information of any registered node in the network must be protected. The identity of the people registered in the network and their geographical information must not be exposed. Only authenticated members should be able to access the messages in the network. Confidentiality attacks include eavesdropping and information gathering attacks. In these attacks, attackers get private information about the network members and then may misuse the information. In an eavesdropping attack, the attacker silently listens to the communication in the network and gathers data [22].

3. Integrity: Information sent in the network must not be altered or deleted before reaching the receiver. Message exchange between sender and receiver must not be tampered with by an attacker. The attacker tries to alter the original message and send erroneous information to harm the network. Attacks on the integrity of the network are Message Deletion/Alteration and Timing attack. In the Message Deletion/Alteration attack, attackers either insert wrong information or delete the message before it reaches the receiver [23]. Timing attack inserts delay in the network causing emergency messages to be delayed deliberately [24].

4. Availability: Network should be available to provide services to the legitimate nodes without interruption. Services from the network should not be disrupted or unavailable for usage. Attack on availability of the network keeps legitimate

users from accessing the network. Attacks on availability include Denial of Service (DoS) attack, spamming attack, jamming and broadcast tampering. The spamming attack creates many requests to the network, due to which the network becomes unavailable [25]. DoS attack controls and makes the network unavailable for authenticated members [26]. A jamming attack is a type of DoS attack at the physical layer in which the attacker jams the signal and disrupts the network [27]. When the attacker inserts an erroneous message in the network, it results in a disturbance in the network is known as a Broadcast tampering attack [17].

5. Accountability: Vehicles should be able to account for their actions in the network. Any malicious activity by a vehicle should be able to trace out by the authorities. Data sent in the network should be able to trace back to its sender.

There are some other security requirements in VANET, such as:

- Scalability: The network should add and append additional nodes in the network without affecting its performance. For example, if the number of vehicles increases in a network, the network should assist them without any latency.
- Robustness: Network should be able to overcome any adverse conditions and provide services unaffected over time.
- Non-Repudiation: A vehicle should accept its activities in the network and should not deny being the origin of the information sent in the network.
- Revocability: For any misbehaviour in the network, a malicious vehicle should be identified and revoked from the network from causing more trouble.

2.1.3 Position Falsification Attack

VANET supports two types of applications, comfort and safety application. Comfort application includes services related to the comfort and convenience of the people in the network. These services consist of weather information, nearest gas station,

restaurants, and may involve commercial applications such as advertisements and entertainment streaming. In contrast, a safety application is associated with the security and safety of the network participants. These services provide situation awareness and warning messages on the road, for example, blind-spot warnings and hazard warnings. Vehicles in the network transmit their current status in the road network to the nearby nodes. All the vehicles and infrastructures in the sender vehicle's range will receive a BSM. These BSMs are transmitted periodically in the network. BSMs are digitally signed by the sender vehicle before transmitting into the network and contain the vehicle's current position, speed, direction and transmission time. Many models are researched, such as the PKI model, which uses cryptographic techniques such as digital signatures to encrypt the BSMs. Messages are encrypted before transmitting to the network, and only authenticated members can decrypt the BSMs received. Malicious vehicles send false position information in the BSM that can mislead the legitimate vehicles in the network and cause disastrous effects. These attackers can be insiders or outsiders trying to harm the network having rational or malicious intentions. This attack can also be triggered when GPS is faulty and transmitting incorrect position coordinates in the message. The attack caused by incorrect or false position information in the BSMs is known as the Position falsification attack, as depicted in Figure 2.3. Data integrity of the network is ensured when the attacker does not alter a message. A position falsification attack violates data integrity as the attacker alters the actual position of the vehicle.

Cryptographic techniques help us identify insider attackers of the network but do not ensure message correctness. Position falsification attacks cannot be detected using cryptographic methods, but an additional model must detect and ensure message correctness in the BSM. Detailed information about five Position falsification attacks detected in this research are listed below –

1. Constant attack: In this attack, the sender vehicle continuously broadcasts a fixed position coordinates in the BSM, pretending to be in the same network position. This attack could mislead the honest vehicles into thinking of it as a

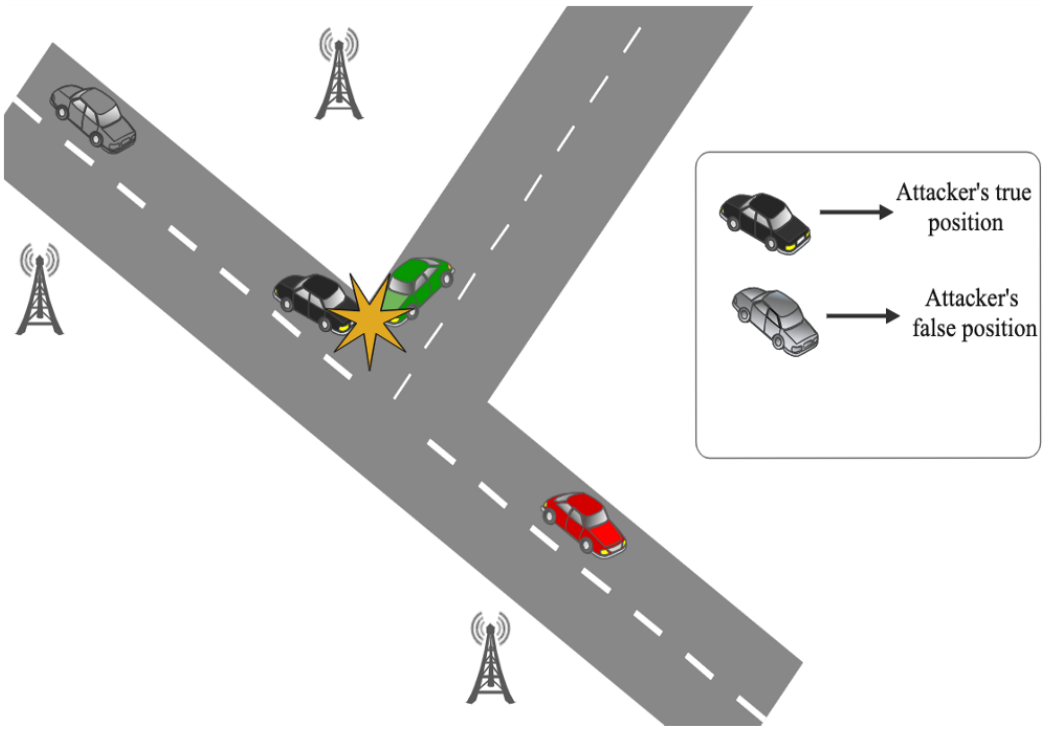


Figure 2.3: An example of Position Falsification Attack

hazard or traffic congestion on the road.

2. Constant offset Attack: Attacker vehicle adds a constant offset/fixed value to the actual position and transmits the network's altered position. This attack is difficult to detect as the attacker is behaving normally by slightly altering the actual position in the BSM.

3. Random Position Attack: In a random position attack, the attacker sends a random position coordinate from the simulation area/playground in the network. It creates confusion in the network as every next BSM will have an entirely different and random value from the simulation.

4. Random Offset Position Attack: Attackers send a random value from a pre-configured area around their vehicle. This attack is very similar to a constant offset attack as both slightly alter the position information.

5. Eventual Stop Attack: The attacker tries to behave normally for some time in the network and then suddenly sends a fixed position repeatedly to depict an

eventual stopping of the vehicle. Attackers mislead the legitimate vehicles by gaining trust in the network for some time and then deceive them.

2.2 Overview of Machine Learning

Machine learning is the Artificial Intelligence branch that facilitates machines to perform specific jobs faster and skillfully using statistical learning [28]. It is extensively used in countless fields such as healthcare, e-commerce, law to detect diseases, perform facial recognition and provide a spam detection email system. Machine learning algorithms discover patterns in input data to make predictions, detect or categorize data, and solve real-world problems [29]. In VANET, machine learning algorithms can detect several attacks, intrusion and misbehaviour in the network. There are four main types of machine learning:

- Supervised Learning: Learning in which an algorithm is trained with labelled data is known as supervised learning. Supervised learning is useful in solving two types of problems: classification and regression.
- Unsupervised Learning: Unsupervised learning is where an algorithm is provided with the unlabelled data and uses its ability to find patterns and similarities to solve a problem. This type of learning is usually used for organizing data in clusters, anomaly detection and association learning.
- Semi-supervised Learning: Input data is mixed with both labelled and unlabelled data; it gives an advantage of both supervised and unsupervised learning.
- Reinforcement Learning: In this learning, the algorithm learns from its environment. It is given a reward for every success and gets nothing on failure.

2.2.1 Basic Machine Learning Concepts and Terminologies

Basic terminologies and processes of machine learning used in this thesis are defined below:

1. **Model**: A model is a machine learning algorithm trained to solve the problem.

2. **Dataset**: Input data used to train a machine learning model is known as the dataset.

3. **Training and Test set**: Dataset is sliced into training and test set where former trains the model, and latter tests the model.

4. **Feature:** Features are the data objects/columns in the dataset with essential characteristics to solve the problem.

5. **Data pre-processing**: Raw datasets have noise and duplicate data that cannot train the model as it will degrade the performance and not give accurate results. Data pre-processing [30] is a process to clean and organize the data before training the model.

6. **Cross-validation:** Dataset is divided into a set of data randomly split into groups. Each group has a train and test set, and the average of each group's result will be the model's performance. Cross-validation [31] is effective process to avoid overfitting of a model and to get accurate results.

2.2.2 Classification Algorithms

Classification is a category of supervised learning where input data is labelled dataset [32]. The classification problem is to categorize data points into different classes [33]. Classes are the target points or labels in the dataset. Algorithms to solve classification problems are known as classifiers. Classifiers train the model by finding the similarity to categorize the dataset. In VANET, machine learning can classify legitimate vehicles and misbehaving nodes. There are two main types of classification:

- Binary classification: Binary classification is predicting two classes from a dataset. An example of this type of classification is spam detection. In this thesis, two classes in binary classification are legitimate vehicles and attacker vehicles.

- Multiclass classification: Multiclass classification involves classifying/predicting more than two classes in a dataset. Five different position falsification attacks and legitimate vehicles are the classes for multiclass classification in this research.

Following is brief information related to classification algorithms used in this research. We also attempted to implement a Support Vector Machine (SVM) classifier, but the preliminary results were unconvincing, so we opted on K-nearest neighbour, Decision Tree, Random Forest, and Naïve Bayes Algorithm.

2.2.2.1 K-Nearest Neighbours

K-Nearest Neighbour algorithm [34] is widely used for solving classification problems. It is suitable for balanced as well as imbalanced datasets. K-Nearest Neighbour works by finding the distance between all the points and a query point and selects k nearest neighbours to a query point. Based on the labels of k nearest neighbours, it chooses the label based on popularity. This label is assigned to the query point by the majority vote of the neighbours.
Distance between the points can be calculated using Euclidean, Manhattan, Minkowski or Hamming distance functions.

2.2.2.2 Decision Tree Algorithm

Decision Tree algorithm [35] constructs a tree of a dataset with branches to perform classification. The top-most node known as the root node corresponds to the best feature in the dataset. It consists of two entities, the decision node and the leaf node. A decision node is the conditions on which a tree navigates, and leaf nodes are the outcomes of the decision node's conditions. The main advantage of this algorithm is it does not require any pre-processing of data and is faster. One major disadvantage is that it is more prone to overfitting.

2.2.2.3 Random Forest Algorithm

Random Forest algorithm [36] is an algorithm that solves classification and regression problems. As its name, the Random Forest model is a collection of decision trees. These decision trees predict the result based on the dataset. The best solution from the results is chosen through the ensemble method. This algorithm overcomes the disadvantage of the Decision Tree algorithm. Moreover, the Random Forest algorithm is robust and gives accurate results compared to the Decision Tree algorithm.

2.2.2.4 Naïve Bayes Algorithm

Naïve Bayes classification algorithm [37] depends on Bayes' theorem, a probabilistic approach to classify a problem. It is suitable for both binary and multi-class classification. This algorithm is firmly based on the assumption that features of a class are independent of each other. However, in real-world scenarios, features are dependent on each other. Nevertheless, the Naïve Bayes algorithm is considered highly scalable and fast for large datasets.

2.3 VeReMi Dataset

VANET is now actively being researched in modern vehicular networks. Much research has been in the past for misbehaviour detection, intrusion detection and various security attacks in VANET. However, the dataset and tools used in their research are mostly private and not shared in public, making it difficult to compare. Heijden et al. introduced the first public extensible dataset, namely Vehicular Reference Misbehavior Dataset (VeReMi) [38]. This dataset was introduced to create a baseline for detection mechanisms for position falsification attacks. It consists of 225 individual simulations with five different attacker types, three different attacker densities, three different traffic densities and five repetitions of each parameter set with random seeds. Dataset is created on Luxembourg SUMO traffic scenario (LuST) [39] using VEINS [40] and OMNET++ [41]. Simulation parameters used by the authors in the VeReMi dataset are shown in Table 2.1.

Table 2.1: Simulation parameters of VeReMi Dataset

Simulation Parameter	**Value**	**Description**
Duration	100s	Total duration of simulation
Vehicle Density	(3, 5, 7)h	3: Low density 5: Medium density 7: High density
Attacker density	0.1, 0.2, 0.3	10%, 20% and 30% Attacker density

Dataset consists of message logs files of each vehicle and ground truth files. Message log files are the received BSMs maintained by each vehicle, while the ground truth file consists of the actual values sent by a vehicle in the simulation. There is only one ground truth file in an individual simulation but has logs files equal to the number of vehicles in a simulation. Message logs at receiving vehicle consist of

Table 2.2: Messages transmitted per vehicle density

Vehicle Density	**No. of vehicles**	**Messages transmitted**
Low Density	35-39	908 to 1144
Medium Density	97-108	3996 to 4489
High density	491-519	20482 to 21878

a unique message ID, claimed position vector, position noise vector, claimed speed vector, speed noise vector, claimed transmission time, reception time and received signal strength value (RSSI). As mentioned in Table 2.2, the number of vehicles in low density is around 35-39, and medium density has vehicles between 97-108 while for high-density number increases to 491-519 vehicles. A subset of these vehicles is malicious in a simulation made using a uniform distribution. These malicious vehicles send false position coordinates in the BSMs. Message log files will record the false position sent by a malicious vehicle, but the ground truth file maintains the vehicle's

actual position coordinate. A vehicle can also receive 0 BSMs if it was not close to any other vehicle in the network.

Table 2.3: Attacker model in VeReMi Dataset

Attacker Type	**Attack name**	**Description**	**Example**
1	Constant	Vehicle transmits a fixed position.	x=5560, y=5820
2	Constant Offset	Offset added to vehicle's actual position	$\Delta x = 250, \Delta y = 150$
4	Random	Transmits random position from simulation area.	Random in Simulation area.
8	Random Offset	Random position from pre-configured rectangular area around the vehicle.	$\Delta x, \Delta y$ uniformly random from $[300, 300]$
16	Eventual Stop	Attacker behaves normally for some time and then transmits current position repeatedly.	Stop probability $+= 0.025$ with each position update

Attacker type value helps distinguish between legitimate vehicles and attacker vehicles. Attacker type for legitimate vehicles is set to 0, while it is 1,2,4,8,16 for different attacks, as shown in Table 2.3.
VeReMi dataset is built on an extensive traffic scenario, including highway, city and street regions. In this research, the VeReMi dataset provides a standard dataset which is further extended into two consecutive BSM datasets for misbehaviour classification of five different position falsification attacks.

2.4 Literature Review

Nowadays, many researchers are using a machine learning approach for misbehaviour detection or attack detection in VANET. Cryptographic frameworks such as the PKI model [10] provide authentication of the vehicle's identity in the network but do

not ensure message correctness. PKI model uses a digital signature to encrypt the messages from the registered vehicle in the network, and only registered legitimate vehicles can receive and decrypt the BSMs. It is assumed in the PKI model that the message contains the correct vehicle information. An additional model such as machine learning can help ensure message legitimacy. Machine learning helps identify the characteristics of a highly dynamic vehicular network [42]. It is a data-centric approach to optimize network performance by reducing the vulnerabilities of the network. Some of the machine learning approaches are discussed in this section. Comparative analysis of the literature review is addressed in Table 3.2.

2.4.1 Machine Learning in VANET

Grover et al. [43] introduced an ensembled learning-based approach for classifying honest and misbehaving vehicles. The authors extracted features of misbehaviour in VANET by performing different experiments. Malicious and honest data are trained and tested using classification algorithms supported by Waikato Environment for Knowledge Analysis (WEKA)[44], a data mining tool. The algorithms used by authors are Naïve Bayes, Instance-based learner, Random Forest, Decision Tree and AdaBoost. The authors combined five different classifiers to classify the attack individually. Individual results of all five classifiers are selected based on the majority, and a vehicle is classified. The authors claim to achieve better accuracy with their method. According to the authors, Random Forest and Decision Tree outperformed other classifiers.

Khot et al. [45] proposed a machine learning framework to predict the next position of the vehicle in the network. The authors used beacon messages from neighbouring vehicles and created features such as distance between sender and receiver. Machine learning algorithms were utilized for training and testing the model. The authors compared the predicted value with the actual value in the BSM and classified the vehicles based on the comparison. If the position is not equal to prediction, it is classified as an attacker vehicle. The authors claimed that Random Forest performs best among other algorithms.

In paper [46], authors use three features/predictors combination to detect position falsification attack. The first combination was the sender vehicle's position and speed; the second combination includes position, speed, and change in position coordinate of sender vehicle. The third combination was the position, speed and change in speed and position of the sender vehicle. Authors claim to analyze that speed does not contribute to the detection of position falsification attacks. They used SVM and Logistic Regression machine learning algorithms for detection.

Table 2.4: Comparison table of Literature Review

No.	**Paper**	**Machine Learning Model?**	**VeReMi Dataset Used?**	**Approach**
1	Xue et al.	No	No	Trust Table Method
2	Grover et al.	Yes	No	Ensemble method
3	Heijden et al.	No	Yes	Belief theory approach
4	Steven et al.	Yes	Yes	Additional plausibility checks
5	Gyawali et al.	Yes	Yes	sender-receiver pair approach
6	Khot et al.	Yes	Yes	Predicting new position
7	Singh et al.	Yes	Yes	Normalization of position features
8	**Proposed Method**	Yes	Yes	Vehicle-RSU pair approach

2.4.2 Detecting Position Falsification Attack

Xue et al. in paper [47] proposed a trusted neighbour table to detect position spoofing attacks. It is a location verification scheme in which they create a TNT at each vehicle to record its neighbouring vehicle's updated location. The TNT-based location verification requires every node in the network to maintain a TNT that contains its neighbouring nodes' latest location. The authors claimed their TNT is different from the neighbour table as TNT contents are authenticated contrary to the neighbour

table. Nodes create a trust value in the table, and the more the trust value, the more trustworthy is the neighbouring vehicle. The authors claim their approach to be secure and efficient if there is no infrastructure involved.

In paper [38], the VeReMi dataset authors introduced a framework called Maat, which ensures the validity of received data. Maat is a framework based on subjective logic - a mathematical framework that enables uncertainty through objects called subjective opinions on data. Subjective opinions are the relationship between actors and objects that express their trust and confidence with a degree of uncertainty. It is based on belief theory same as Dempster-Shafer's theory. Maat applies this logic to build a fusion and data management system to determine the trustworthiness of data. For data management and storing detection results, Maat uses a directed graph. The authors used four comparison checks for performance evaluation of the model, namely Acceptance Range Threshold (ART), Sudden Appearance Warning (SAW), Simple Speed Check (SSC), and Distance Moved Verifier (DMV). ART determines if the beacons are received from the minimum transmission range as each vehicle has a fixed transmission range. SAW is a detector based on the fact that beacons are received in a regular interval and do not appear suddenly. DMV checks if a vehicle has moved the minimum distance from the previous position, and SSC is a simple speed check detector inspired by the Kalman filter [48].

In paper [49], the authors proposed integrating plausibility checks and a machine learning framework for misbehaviour detection using the sender-receiver pair approach in the VeReMi dataset. They added six features: 1-6, from which two are plausibility checks capable of detecting fake location and the remaining four are quantitative information used to describe vehicle's behaviour in the network. Two plausibility checks included by authors are location and movement plausibility checks. Location check verifies if the transmitted position lies in the range of plausible predicted locations, whereas movement check is for a constant position. Along with these two features, they added four quantitative information of vehicle behaviour as features. Feature 3 and 4 is the difference between calculated and predicted velocities, where feature 5 is the magnitude of features 3 and 4, and feature 6 is the difference

between actual and predicted displacement.

In recent research by Gyawali et al. [50], they introduced a misbehaviour detection model for both false alert verification scheme and position falsification attack. This framework is also based on the sender-receiver pair approach. The false alert is created when an attacker sends a false alert to its neighbouring vehicles. These alerts include hazard condition notification, emergency vehicle stopping warning or emergency braking of a vehicle. In the proposed framework, the authors equipped each vehicle with a misbehaviour detection model. Each vehicle broadcasts detected results to its neighbours, and these results are aggregated together to determine which vehicle must be evicted from the network. The authors use the Greenshield model [51], which assumes a linear speed-density relationship to estimate uninterrupted traffic. The receiver vehicle calculates the change in its speed, position and difference in sender vehicles speed, position, receiving distance and RSSI value. All these values are created as features in the dataset, and a machine learning algorithm is applied to it.

In this research, the proposed methodology uses the vehicle-RSU pair approach for position falsification detection. Machine learning algorithms are used to classify legitimate vehicles and attacker vehicles.

CHAPTER 3

Consecutive BSM Based Authentication

3.1 Introduction

Misbehaviour detection is a method to identify the attacks on the VANET using various techniques. In this research, the proposed methodology aims to detect position falsification attacks on VANET using machine learning algorithms. The vehicle transmits BSMs into the network, and all nearby vehicles and infrastructures can receive these BSMs. BSMs carry information related to the vehicle's current status in the network. This information includes sender id, position, speed, time, RSSI value and a unique message-id. Data in the BSM can help identify the characteristics and behaviour of an attacker vehicle in the network. For this proposed method, collecting data comprising BSMs from vehicles in a road network is necessary. As VANET is a very dynamic and vulnerable network, it is not practical to directly apply and test the proposed techniques in real-time hence, simulation environments are essential. In this proposed methodology, we use the VeReMi dataset, which consists of a collection of BSMs in a network under different traffic scenarios and attacker densities.

The proposed methodology aims to:

- Provide a new framework to detect position falsification attacks.
- Classify five types of position falsification attacks.
- Provide a machine learning-based approach for classifying vehicles in the

network.

- Provide a two-consecutive BSM dataset extended from the VeReMi dataset.
- Detect inside attackers with rational or malicious motives injecting false position information in the network.

3.2 Proposed Architecture

Registered vehicles transmit BSMs periodically in the network. All the neighbouring vehicles and infrastructures can receive the transmitted BSMs. Cryptographic methods such as encryption and decryption provide authentication of the BSM in the network. In these methods, the vehicle is assigned public and private keys by the Central authority at the time of registration. These registered vehicles then use these keys to sign the messages in the network using Digital Signature algorithms. These techniques ensure only authenticated vehicles in the network can send and receive BSMs.

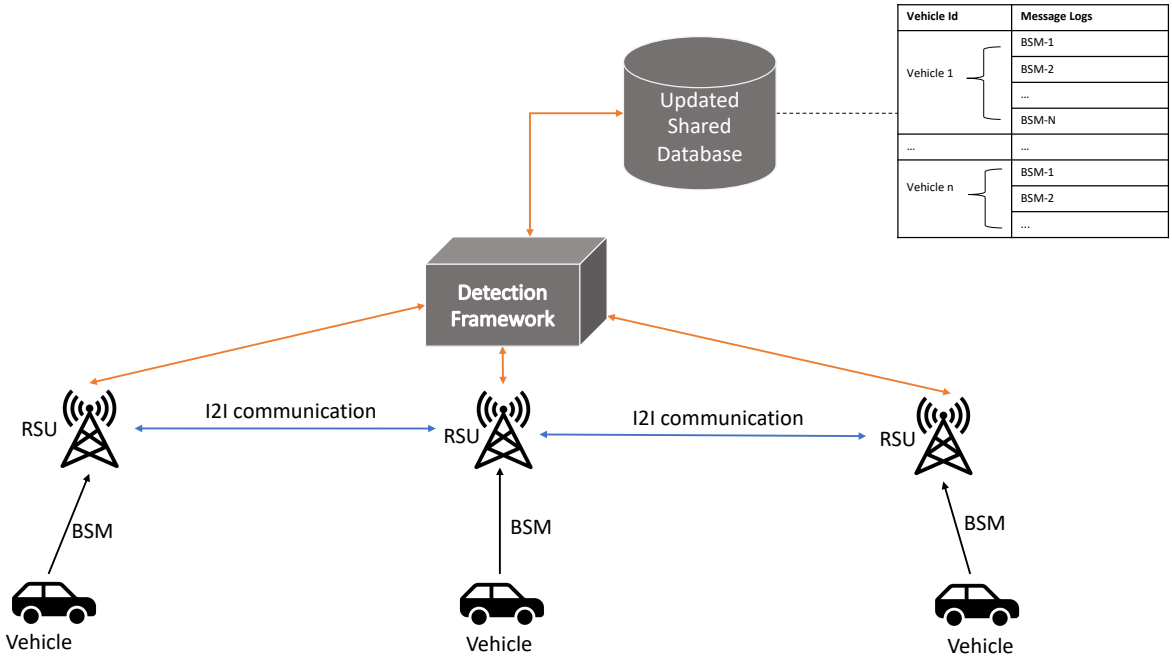


Figure 3.1: Proposed Architecture

This proposed architecture is an additional model for providing message integrity on top of existing cryptographic methods. Vehicles transmit periodic BSMs, and the nearby RSUs receive these BSMs in the network, as shown in Figure 3.1. RSUs in the network have I2I communication with each other. BSMs received by the RSUs get updated in the shared database. For each vehicle entry in the database, BSMs are updated in the order of their transmission time. The proposed detection framework is installed at the RSU. The detection framework can access the shared database. When a vehicle sends the BSM to the network, RSUs then verifies the message correctness in the BSM. On receiving BSMs from a vehicle, the proposed detection framework installed at the RSU retrieves the last received BSM from a vehicle from the shared database using a unique sender ID assigned to each vehicle during registration. The proposed model prepares the received data into two-consecutive BSM data format to apply the machine learning algorithm. The proposed model applies machine learning classification after retrieving recent two-consecutive BSM from a vehicle and classifies the vehicle as a legitimate or attacker vehicle. After classification, the latest BSM received by a vehicle is updated in the shared database. This database is shared and can be accessed by other RSUs in the network. When a vehicle is classified as an attacker vehicle, RSUs inform the nearby vehicles and infrastructures about the misbehaving vehicle.

3.2.1 Operations performed at Vehicles

- The vehicles get registered from the authorization party before entering the network.
- The vehicles generate the BSM from the OBU and encrypts the messages using Digital Signatures.
- The vehicle sends periodic BSMs in the network.
- The vehicle maintains a local database about logs of nearby vehicles being attacker or legitimate.

- The vehicles listen to the broadcast from RSU about the misbehaviour in the network.

3.2.2 Operations performed at RSU

- RSU decrypts the BSMs received from the vehicles.
- RSUs can access the shared database containing the collection of BSMs from a vehicle.
- The proposed detection framework is installed at the RSUs, which prepares two-consecutive BSM data from retrieved BSMs of a vehicle and then classifies the BSM into legitimate or attacker.
- RSU updates the shared database in real-time.
- When the detection framework classifies the vehicle into an attacker vehicle, RSU informs the other vehicles and infrastructures about the misbehaviour.
- In the case of classification as a legitimate vehicle, RSU only updates the shared database.

3.2.3 Assumptions

This proposed approach has few assumptions for the network on which this methodology will perform to its full potential. These assumptions are mentioned below:

- We assume that RSUs in the network and the storage database updated by them cannot be compromised with any attack type.
- We assume RSUs maintain a shared updated storage database in real-time, accessed by all the other RSUs in the network.
- We assume there is always an RSU in the vicinity of the vehicle in the network and stores the periodic BSMs generated by them.

3.3 High-level Outline of Proposed Approach

The proposed methodology has three main stages: dataset extraction, data preparation and classification as shown in Figure 3.2. A detailed discussion of these three stages is as follows:

3.3.1 Data Extraction

VeReMi dataset includes a total of 225 simulations with different traffic scenarios. Each simulation consists of two types of files: Ground truth file and log files. There is only one ground truth file in a simulation that includes a vehicle's actual behaviour in the network. Ground truth file also comprises an attacker type, which differentiates the legitimate vehicles from misbehaving vehicles. On the other hand, the number of log files in a simulation is equal to the number of vehicles in the network. Each vehicle creates a log file that includes all the received BSMs from other vehicles. As in position falsification attack, attacker vehicles transmit false information in the BSM; hence, log files contain incorrect information.

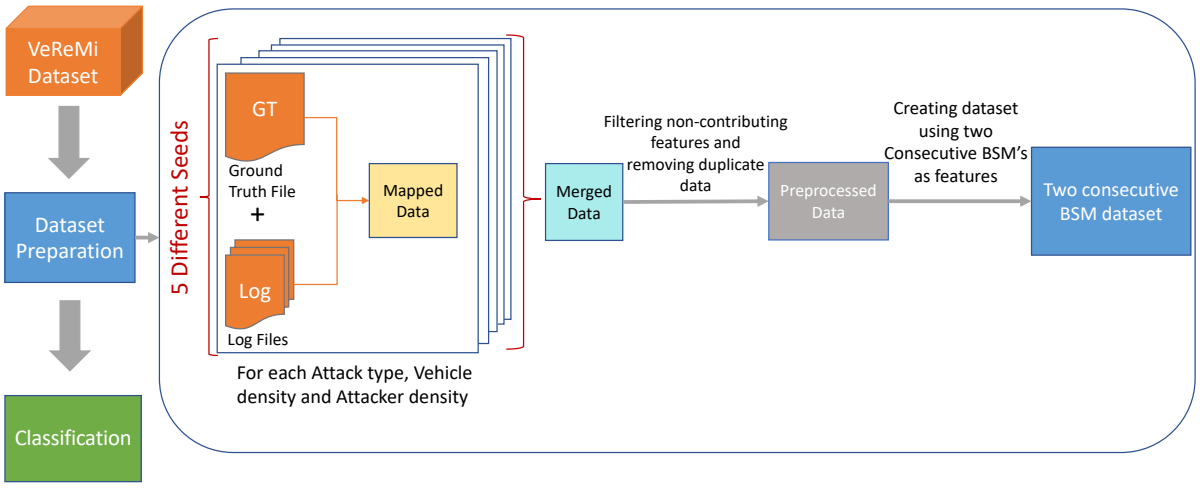


Figure 3.2: Proposed methodology

Ground truth files and log files must join to create a labelled dataset. In the data extraction stage, the ground truth file is mapped to log files for each simulation. For a single simulation, the number of log files is equal to the number of receivers; hence the first step is to combine these separate log files into a single file. Ground truth file and log files contain a unique id named messageID. To create a labelled dataset, the ground truth file's attacker type must be mapped to data in the combined log file, as shown in Figure 3.3. As there are five different seeds of the same scenario in the VeReMi dataset to create randomness in the network, this process is repeated for all five repetitions. All five repetition was combined in the end to create a merged dataset for a single scenario.

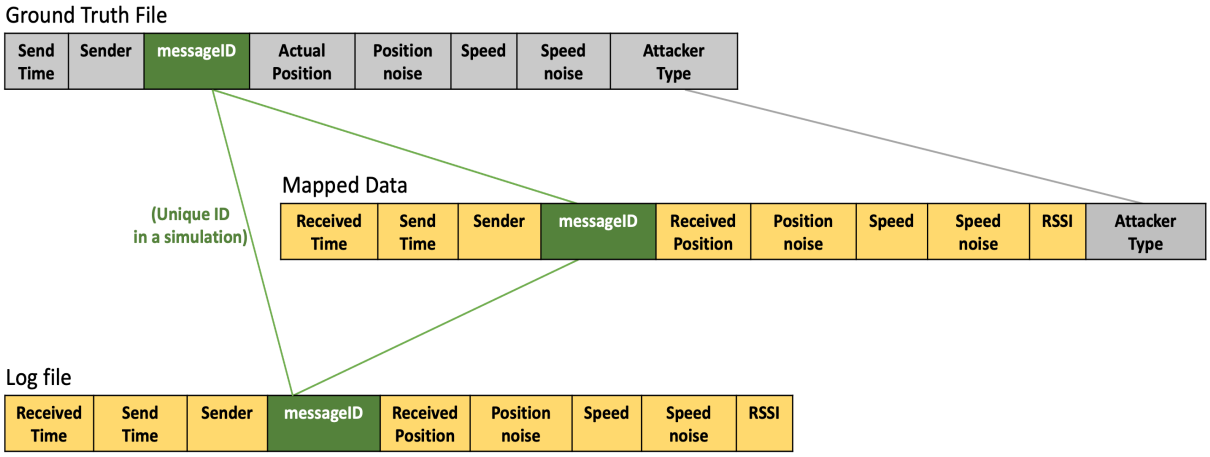
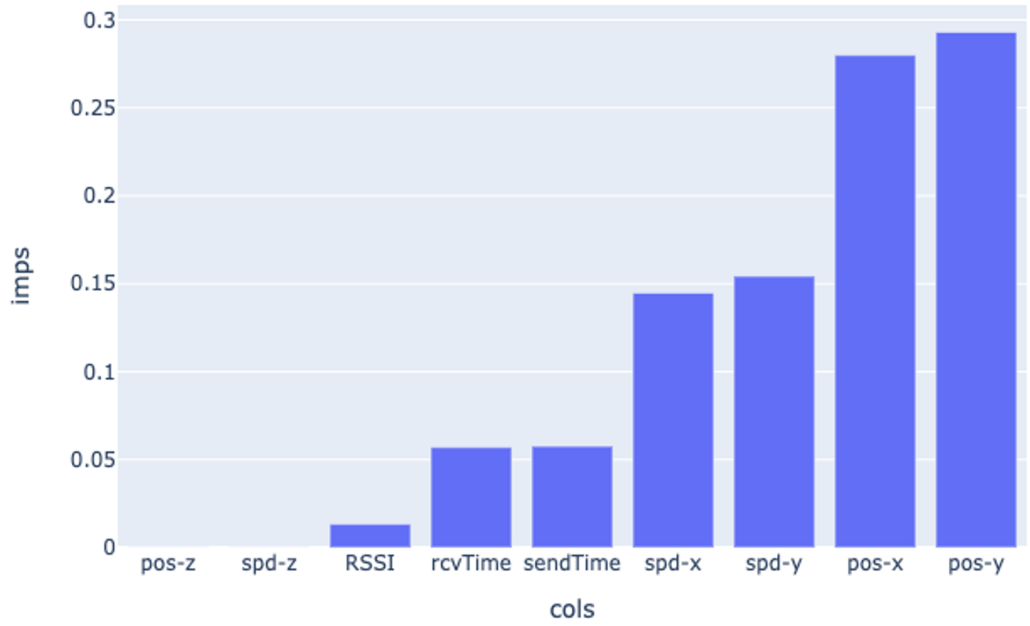


Figure 3.3: Data extraction of Ground truth file and Log files to create Labelled data

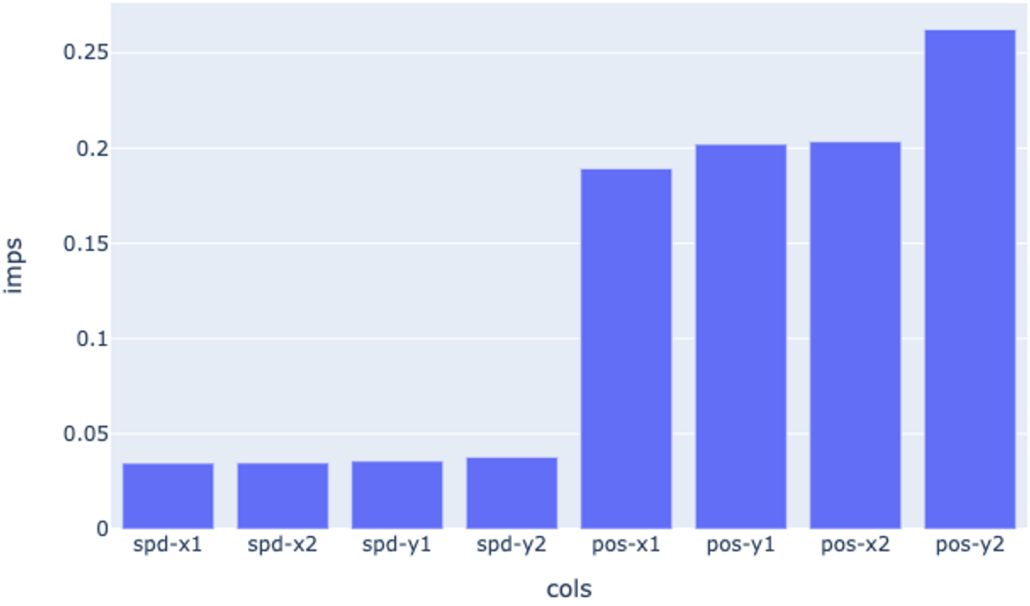
3.3.2 Data Preparation

In this stage, merged data is pre-processed by filtering non-contributing features and removing duplicate data. As each vehicle has a separate log file, a single BSM was recorded in multiple vehicles, creating duplicate data in the dataset. Duplicate data was removed during the data preparation process to prevent the algorithm from memorizing the data points. Non-contributing features are removed using the feature importance process. This process will provide information about which feature contributes more to detect attacker behaviour and which contributes the least. The

non-contributing features can decrease model accuracy and efficiency; hence it is best practice to remove such elements.



(a) Single BSM Dataset



(b) Two-consecutive BSM Dataset

Figure 3.4: Feature importance graph

As shown in Figure 3.4(a), position and speed features contribute the most, and others provide less important information for the model to train. The main idea behind the proposed approach is to find information on vehicle behaviour in the network. With a single BSM, it is unviable to gather information to detect misbehaviour. Figure 3.4(b) shows that two BSMs can yield more information of vehicle's behaviour than a single BSM as initial and final position coordinate along with both speeds can provide meaningful information for misbehaviour detection. As the vehicle's position and speed have more feature importance, position and speed coordinates from two consecutive BSM data from a vehicle are made as features to create a two-consecutive BSM dataset. A detailed discussion regarding the dataset is included in section 3.4. We also removed few features in the dataset that were not providing meaningful information before training the Machine learning model. These features include position noise vector, speed noise vector, message-id and sender-id.

3.3.3 Classification

Third and the last stage of this methodology is to perform classification on the dataset. In this step, machine learning algorithms are implemented to classify the legitimate vehicles from the network's attacker vehicles. In this thesis, we will implement both binary and multiclass classification. The binary classification will be performed on separate attacks, and on all five position falsification attacks combined in a single dataset, the multiclass classification will be performed. Machine learning algorithms used for classification are K-Nearest Neighbour, Random Forest, Decision Tree and Naïve Bayes algorithms. Classifiers giving a better correct classification rate out of the four algorithms will be used in a detection framework. These algorithms train the model using a training set and classify the future data as legitimate or attacker.

3.4 Modifications to the VeReMi dataset

VeReMi dataset is a baseline of datasets in VANET with only five position falsification attacks. We modify the dataset such that it is compatible with the proposed model.

We take two consecutive BSMs from a vehicle and create them as features. These features include x and y coordinates of position and speed for BSM1 and BSM2 and their attacker type. An example of a two-consecutive BSM dataset is as shown below:

Table 3.1: An example of a two-consecutive BSM dataset

Vehicle No.	**pos1_x**	**pos1_y**	**spd1_x**	**spd1_y**	**pos2_x**	**pos2_y**	**spd2_x**	**spd2_y**	**Label**
1	3609.39	5446.80	-3.53	30.62	3605.87	5477.34	-3.53	30.62	0
2	3586.20	5707.55	0.19	0.45	3816.45	5245.45	1.10	2.37	1
3	3815.61	5243.85	-5.72	36.70	3816.45	5245.45	-5.71	36.64	0

In Table 3.1, *pos1_x*, *pos1_y*, *spd1_x*, *spd1_y* are the position and speed coordinates of BSM 1 and *pos2_x*, *pos2_y*, *spd2_x*, *spd2_y* are the position and speed coordinates of BSM 2. Label 0 depicts a legitimate vehicle, and 1 depicts the attacker's vehicle. We removed other non-contributing features from the dataset, such as z-coordinate of position and speed, message-id, RSSI value, sent and received time. These features decrease the accuracy of the model and removing these improve the quality of results.

3.4.1 Modified Attack Type 16

In attack type 16 vehicle behaves normally for some time in the network and then transmits the same position repeatedly in the network as if it made an eventual stop. In this case, the VeReMi dataset labelled the vehicle as an attacker vehicle when it has not yet started behaving abnormally. For example, when an attacker vehicle behaves normally in the network, the machine learning algorithm will classify the BSM sent by the vehicle as "Legitimate" as no misbehaviour is detected. However, the label corresponding to this BSM is "Attacker"; therefore, it creates confusion in the network, and as a result, it affects the efficiency of the model.

So, we modified attack type 16 and created another attack, namely modified attack type 16. In this attack, the attacker vehicle is labelled as the attacker only when it starts misbehaving in the network.

3.4.2 Multiclass Classification

Classification of more than two classes/labels in a dataset is performed using multiclass classification. We create a dataset where all five types of position falsification attacks are combined into a single dataset for this classification type. It can help the model to train on all five attacks at once and classify them separately. Multiclass classification eliminates the need for the model to train individually for each attack instead of attempting to learn the pattern of all attacks together and classifying them accordingly. In a real-life scenario, every detection model must be versatile enough to identify various attacks. Multiclass classification ensures that the detection model is not limited to classifying a single attack but can detect multiple attacks.

3.5 How the Proposed Algorithm Differs from Existing Approaches

As discussed in section 2.4, many researchers have introduced a misbehaviour detection framework to detect position falsification attacks using VeReMi dataset. Some of the current work involves adding features of calculation such as a change in speed and position to train the model, whereas some use trust-based models to detect an attack. Most of the researchers have worked on sender-receiver pairs to identify misbehaviour in the network.

Table 3.2: Comparison of proposed method with existing approaches

No.	**Existing Approaches**	**Proposed Methodology**
1	Detection performed at OBU	Detection performed at RSU
2	Computation overhead on OBU	No computational overhead at OBU
3	There is vehicle-to-vehicle reliability in the network	No vehicle-to-vehicle dependency in the network
4	Sender-receiver pair approach	Vehicle - RSU pair approach

In the sender-receiver pair approach, a detection framework is installed on the OBU in vehicles. In this proposed methodology, instead of a single BSM calculation, two consecutive BSMs are considered features in a dataset. The detection framework is installed on the RSU rather than OBU, reducing computational overhead on the vehicles. In the case of multiple attacker vehicles in the network, the attack's detection becomes challenging as there are fewer honest vehicles to detect the attack.

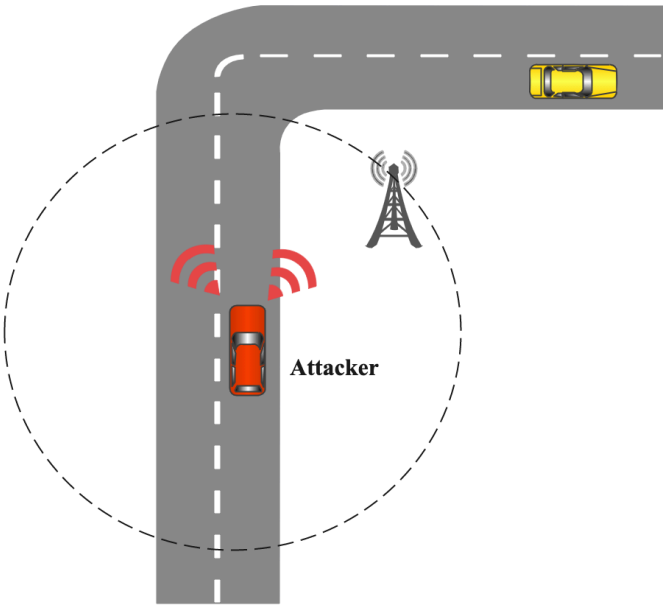


Figure 3.5: An example of attacker in the network

This proposed method removes the vehicle-to-vehicle dependency in the network as RSU gives a broader view of any misbehaviour in the network. As shown in Figure 3.5, there is no vehicle in the range of an attacker vehicle. The attacker vehicle needs to be in the range of other legitimate vehicles to get detected. In this scenario, the proposed methodology with the vehicle-RSU pair approach will detect the attack faster than the existing sender-receiver pair approach. Vehicle-RSU pair approach will detect the attacker vehicle before it misleads the legitimate vehicles. Existing approaches depend on the legitimate vehicles in the network to detect the attack. In the case of the majority of attackers in the network, the sender-receiver pair approach will not detect the attack until a legitimate vehicle comes in the range

of the attacker vehicle. In comparison, the vehicle-RSU pair approach will detect the attack irrespective of the number of attackers in the network.

CHAPTER 4

Results

Due to safety concerns, high infrastructure costs, facilities, and resource requirements, conducting experiments to test the efficiency of a detection system in a real-world scenario is hazardous and difficult. As a result, we run such experiments on a digital scale using simulation tools. This is a much more cost-effective and safe way of evaluating and analyzing algorithms. In this chapter, section 4.1 reviews setup discussion regarding simulation tools and parameters used in the VeReMi dataset, experimental setup toolkits, classification parameters, and evaluation metrics for measuring the proposed classification model's performance. Section 4.2 discusses the results obtained, followed by a comparison with existing approaches in section 4.3.

4.1 Setup Discussion

4.1.1 Simulation setup of VeReMi Dataset

In this research, we use the VeReMi dataset, which was extracted using simulation tools. These simulation tools are VEINS, SUMO and OMNET++. Simulation of Urban Mobility (SUMO) can generate highly portable traffic simulation, whereas OMNET++ is a framework for network simulation. VEINS provides communication between SUMO and OMNET++ to create a realistic simulation. VeReMi dataset uses Luxembourg traffic scenario (LuST) [39], which offers a wide-ranging scenario for evaluating VANET application. Few other simulation paramters used to generate VeReMi dataset is shown in Table 4.1.

Table 4.1: Simulation Parameters used in VeReMi dataset [38]

Parameters	**Value**	**Description**
Mobility	SUMO LuST	Luxembourg SUMO traffic
Simulation Area	2300, 5400–6300, 6300	Various road types
Simulation duration	100s	
Attacker probability	(0.1, 0.2, 0.3)	Attacker probability in the network
Simulation start	(3, 5, 7) h	Control density
Signal interference model	Two-ray interference	VEINS default
Obstacle shadowing	Simple	VEINS default
Shadowing	Log-normal	VEINS default
MAC implementation	802.11p	VEINS default
Thermal power	-110 dBm	VEINS default
Bit-rate	6 Mbps	VEINS default
Sensitivity	-89 dBm	VEINS default
Antenna model	Monopole on roof	VEINS default
Beaconing rate	1 Hz	VEINS default

4.1.2 Dataset Analysis and Classification parameters

In this research, three different traffic scenarios are combined to create three datasets, as shown in Table 4.2. A combination of a low, medium and high attacker and vehicle densities are created to evaluate the proposed model in all three cases. In this research, we will refer to the above mentioned dataset combinations as low, medium and high density datasets. All the attacks are evaluated in low, medium and high density to measure the impact of vehicle and attacker density on the proposed model's performance. In a single simulation in the VeReMi dataset, multiple JSON log files are merged into a single log file, and the "Attacker type" label from the Ground truth file is mapped to the log file to create a labelled dataset. This process is repeated for all five repetitions. Extraction of data by downloading the simulation scenarios

and generating mapped data from these files is performed using shell scripts. Pre-

Table 4.2: Dataset combinations for evaluation

S no.	Repetition	Attacker Type	Attacker Density	Vehicle Density	No. of Instances
1	0 to 4	1, 2, 4, 8, 16	Low (0.1)	Low	4100 - 4200
2	0 to 4	1, 2, 4, 8, 16	Medium (0.2)	Medium	18870 - 18890
3	0 to 4	1, 2, 4, 8, 16	High (0.3)	High	102460 - 102500

processing the data by filtering out non-contributing features and removing duplicate data is implemented using a Python script. After generating a clean, pre-processed two-consecutive BSM dataset, we perform classification. The classification includes the following:

Model selection

A model is selected to perform classification. There are different algorithms for classification, as discussed in section 2.2.2. In this research, four classifiers are used, K-Nearest Neighbour, Random-Forest, Decision tree, and Naïve Bayes.

Hyperparameter tuning

For a model to perform better, hyperparameters must be adjusted, this process is called hyperparameters tuning. Hyperparameters are the values that control the learning of the model. This step can improve the accuracy and optimize the performance of the model. In the K-Nearest Neighbour algorithm, the number of neighbours is tuned, and the value which performs the best was selected for classification. K-Nearest Neighbour classifier used in this research was tuned for values in range 3 to 20, it generated the best results with K=3 (K=number of nearest neighbours). For the Random Forest classifier, number of estimators used to generate the results was kept at 20. We tried increasing the estimators, no notable difference was seen in the results, but for high value of estimators, classifier took more time to train.

Cross-validation

K-fold cross-validation was performed on the dataset to prevent the model from overfitting aan efficiently measuring its accuracy. It splits the dataset into k folds of train

and test set where one split becomes validation set and remaining *k-1* split acts as a training set. Ideally, the value of k lies between 5-10, depending on the dataset. In this implementation, we use k=5, 10, and both generate similar results.

4.1.3 Evaluation Metrics

VeReMi Dataset consists of both legitimate vehicle and attacker vehicle data. VeReMi dataset is an imbalanced dataset, and an imbalance classification refers to classification where class distribution in the dataset is unequal [52]. Since accuracy alone is not considered a good performance metric for the imbalance dataset, we use precision, recall, and F1-score to measure the proposed model's performance. These metric values are obtained using a confusion matrix. A confusion matrix summarizes the model's performance by tabulating the correct and incorrect predictions, as shown below. In our dataset, positive denotes attacker, and negative indicates legitimate vehicle.

A confusion matrix summarizes the model's performance by tabulating the correct and incorrect predictions, as shown in Table 4.3:

Table 4.3: Confusion Matrix

	Predicted Negative	**Predicted Positive**
Actual Negative	True Negative	False Positive
Actual positive	False Negative	True Positive

Precision

Precision measures the proportion of positive classifications that are actually correct as shown in equation (1). Precision is also called positive predicted value.

$$Precision = \frac{Correct\ Positive\ Predictions}{Total\ Positive\ Predictions} = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

Recall

Recall measures the ratio of actually positive classifications which was classified as positive. Recall is also known as sensitivity.

$$Recall = \frac{Correct\ Positive\ Predictions}{Total\ Actual\ Positives} = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{2}$$

F1-score

F1-score is a harmonic mean of precision and recall. F1-score gives a trade-off between precision and recall such that a high F1-score denotes high precision and recall values.

$$F1\text{-}score = 2 * \frac{Precision\ *\ Recall}{Precision\ +\ Recall} \tag{3}$$

4.1.4 Implementation Environment and Toolkit

All the experiments in this research were conducted in the following environment and configuration:

- Operating system: MacBook Air - macOS Catalina
- Processor: 1.6 GHz Dual-Core Intel Core i5
- Memory: 8 GB

Tools and libraries used for the implementation of this research are:

- Programing language: Python 3.7
- Scripting language: Shell script
- Integrated Development Environment: Jupyter Notebook
- Libraries: Scikit-learn, matplotlib, NumPy, pandas

4.2 Classification Results

We implemented four algorithms in the proposed detection framework (K-Nearest Neighbour, Random Forest, Naïve Bayes and Decision Tree) on each attack type. Table 4.4, 4.5, 4.6 show tabular representation of the classification results of four algorithms, with precision, recall, and F1-score as evaluation metrics in low, medium and high density dataset as mentioned in Table 4.2. The performance of our proposed classification model for each attack type is discussed below:

Attack type 1:

K-Nearest Neighbour and Naïve Bayes algorithms showed successful detection for attack type 1 in all three densities, whereas Random Forest and Decision Tree identified all the attacker vehicles, but 0.01% of honest vehicles were misclassified in the high-density dataset. This may be because a vehicle constantly transmits a fixed location but not a fixed velocity, making it easily observable.

Attack type 2:

Constant offset attack is not detected easily as the attacker modifies the position by adding a fixed offset to it, making it harder to identify by a single BSM. Two consecutive BSM are created as features in the proposed method, allowing machine learning algorithms to detect patterns and recognize this attack type. The attack was classified with more than 99 percent precision and recall using K-Nearest Neighbour, Random Forest, and Decision Tree in low and high-density data and similar results with more than 98 percent classification in medium density. The Naïve Bayes algorithm, on the other hand, did not perform well in detecting the attacker's behaviour and showed improvement in classification results from low to high-density.

Attack type 4:

Attack type 4 is detected with high precision and recall by all four algorithms in all three densities. In this attack, the vehicle sends the random position from the simulation playground. With a two-consecutive BSM approach, ML models could

detect the attack as there was a range gap between the two position coordinates from a vehicle.

Attack type 8:

Similar to attack type 4, this attack transmits random positions from a fixed area near the vehicle. Since the range distance between two positions is small, detecting this attack is difficult. However, our proposed model performed well with Random Forest classifiers and Decision Tree classifiers in low and medium density. Although K-Nearest Neighbour has only 90% and 92% recall for low and medium density, it significantly improved the performance for high-density data and gave more than 99% precision and recall values. Naïve Bayes classifier did not perform well in classifying this attack.

Table 4.4: Classification results of Proposed model-LOW

Algorithm	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
	ATTACK 1			**ATTACK 2**		
K-N Neighbour	100	100	100	100	100	100
Random Forest	100	100	100	100	99.7	99.8
Naïve Bayes	100	100	100	22	16.6	20
Decision Tree	100	100	100	99.7	99.5	99.6
	ATTACK 4			**ATTACK 8**		
K-N Neighbour	100	97.9	98.9	100	90.2	94.6
Random Forest	100	99.4	99.7	99.2	96.7	97.9
Naïve Bayes	92.2	100	95.7	48.9	9.1	15.3
Decision Tree	99.7	96.5	98	97.7	96.7	97.5
	ATTACK 16			**MODIFIED ATTACK 16**		
K-N Neighbour	96.7	94.2	95	100	99.6	99.8
Random Forest	97.1	93.4	95.2	98.3	95.6	96.9
Naïve Bayes	11.4	100	20.5	10.6	99.3	19.4
Decision Tree	95.3	92.4	94.1	97.1	96.7	96.9

Attack type 16:

In this attack, the attacker acts normally for a brief period of time before repeatedly transmitting the same location in the BSMs. In contrast to the other four attack

types, the classification of attack type 16 yielded slightly lower precision and recall values in all three densities. The model showed no improvement in performance with an increase in the data density. One reason may be that the vehicle is labelled as an attacker even though it is acting normally, confusing the machine learning model.

Modified Attack type 16:

In this research, modifications were made to attack type 16 to improve the model's performance for classifying this attack. When the vehicle is sending normal behaviour BSMs in the network, the corresponding label of that instance was changed from "attacker" to "legitimate", and once the vehicle starts sending false information in the BSM, the label corresponding to it will be "attacker". The classification results on this modified dataset using K-Nearest Neighbour, Random Forest, and Decision Tree notably improved. Naïve Bayes classifier tries to classify all the BSMs into "attacker", giving almost 100% recall value but extremely low precision in low-density data. Naïve Bayes tried to improve the precision value for high and medium density, but recall value dropped, hence less F1-score in all three densities.

Table 4.5: Classification results of Proposed model-MEDIUM

Algorithm	Precision	Recall	F1-score	Precision	Recall	F1-score
	ATTACK 1			**ATTACK 2**		
K-N Neighbour	100	100	100	99.7	98.3	99
Random Forest	100	100	100	99.8	99	99.4
Naïve Bayes	100	100	100	73.2	12.7	21.6
Decision Tree	99.9	100	99.9	98.9	98.6	98.7
	ATTACK 4			**ATTACK 8**		
K-N Neighbour	100	99.5	99.8	99.8	92.5	95.9
Random Forest	100	99.8	99.9	99.1	95.2	97.1
Naïve Bayes	100	99.2	99.6	47.6	7.6	13.1
Decision Tree	100	99.8	99.9	97.8	95.9	96.8
	ATTACK 16			**MODIFIED ATTACK 16**		
K-N Neighbour	96.5	95.4	95.7	98.1	98.7	98.5
Random Forest	96.5	95.5	96	97.7	98	97.9
Naïve Bayes	40.8	21.2	27.9	36.2	20	25.7
Decision Tree	94.1	96.1	95.1	96.6	97.8	97.2

Table 4.6: Classification results of Proposed model- HIGH

Algorithm	Precision	Recall	F1-score	Precision	Recall	F1-score
	ATTACK 1			ATTACK 2		
K-N Neighbour	100	100	100	99.8	99.7	99.8
Random Forest	99.9	100	99.9	99.8	99.6	99.7
Naïve Bayes	100	100	100	54.8	41.8	47.4
Decision Tree	99.9	100	99.9	99.4	99.3	99.4
	ATTACK 4			ATTACK 8		
K-N Neighbour	100	99.8	99.9	99.9	97.2	98.5
Random Forest	99.9	99.9	99.9	99.4	97.5	98.6
Naïve Bayes	100	99.5	99.7	68.6	14.9	24.4
Decision Tree	99.9	99.9	99.9	98.8	97.7	98.3
	ATTACK 16			MODIFIED ATTACK 16		
K-N Neighbour	96.8	95.2	96	98.7	98.5	98.7
Random Forest	96.7	94.6	95.5	98.4	97.3	97.9
Naïve Bayes	53.1	11	18.3	58	9	15.3
Decision Tree	94.1	94.6	94.3	97.6	97.1	97.3

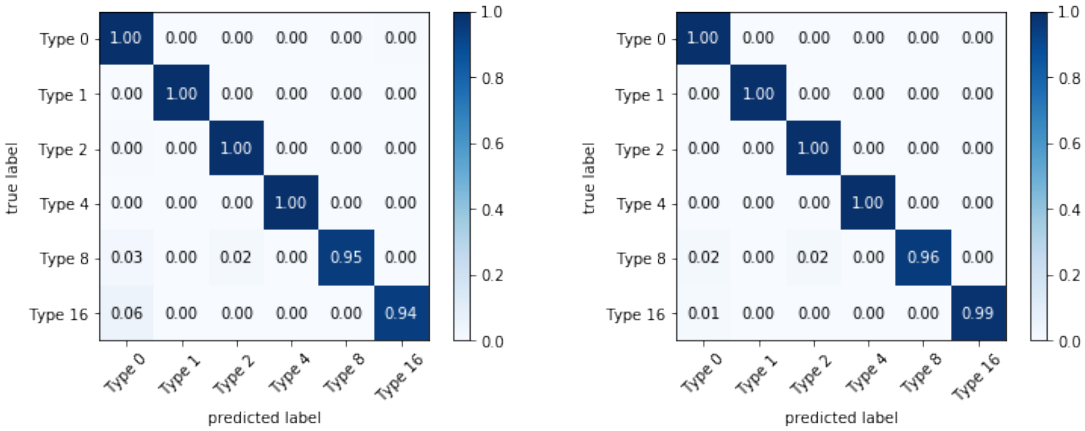
4.2.1 Multiclass Classification

Multiclass classification is used to classify a dataset of more than two classes/labels. For this classification category, we create a dataset that includes all five attack types in a single dataset. Two datasets are made with different attack type 16. It can help the model to train on all five attacks at once and classify them separately. Table 4.7 depicts classification results obtained using the proposed two-consecutive BSM approach on a multi-class dataset. The table contains two results, one with attack type 16 and the second with modified attack type 16. Compared with the other three classifiers, the K-Nearest Neighbour classifier achieved better results in both the datasets, while the Nave Bayes classifier showed unsatisfactory performance. A slight improvement in classification results is seen with modified attack type 16. Figure 4.1 also shows a normalized confusion matrix to depict which attack type was misclassified the most. It also indicates instances were misclassified as which other type in the dataset. In this confusion matrix, the "Type 0" denotes legitimate BSMs from a vehicle. These results were generated using a K-Nearest Neighbour

classifier. The results show that only attack types 8 and 16 were misclassified among the other attacks, while the other attacks were correctly identified. Attack type 8 was misclassified as "Type 0" and "Type 2". Figure 4.1 (a) shows that 94% of attack type 16 was classified correctly and 6% was misclassified as "Type 0". But from Figure 4.1 (b), misclassification reduced to only 1% in modified attack type 16.

Table 4.7: Classification results of Multi-class classification

Classification Algorithm	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
With:	**ATTACK 16**			**MODIFIED ATTACK 16**		
K Nearest-Neighbour	98.8	98.1	98.5	99.2	98.8	99
Random Forest	98.7	97.8	98.3	99	98.3	98.7
Naïve Bayes	64.5	54.9	59.1	64.4	54.8	59.2
Decision Tree	97.9	97.8	97.8	98.6	98.4	98.5



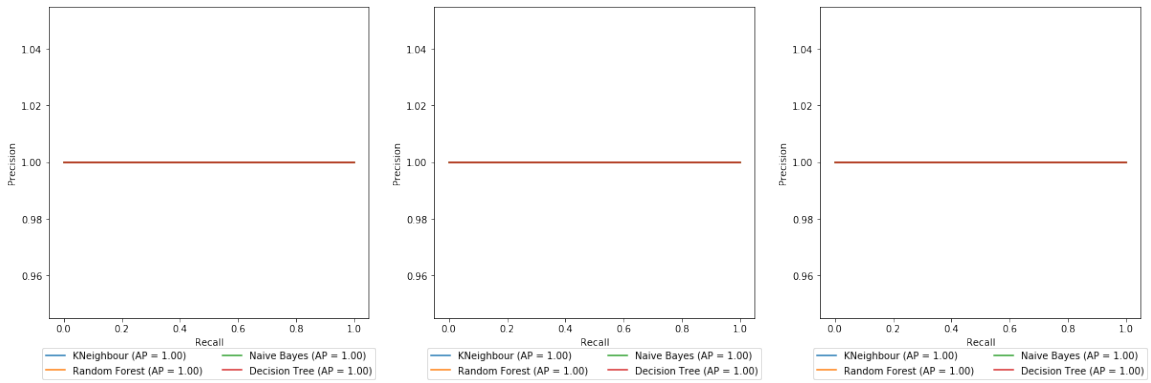
(a) With Attack type 16 (b) With Modified Attack type 16

Figure 4.1: Confusion matrix of Multi-class classification

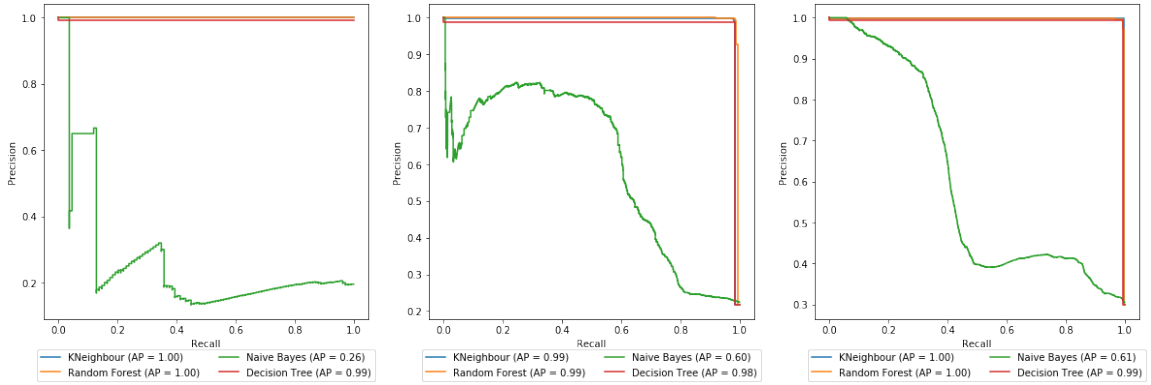
4.2.2 Visualizing the results

For visualizing the results obtained, we used a precision-recall curve [53]. The precision-recall curve is most commonly used for situations involving imbalanced datasets, and it is used for evaluating the performance of binary classification. The precision-recall curve demonstrates the trade-off between precision value and recall value. A larger

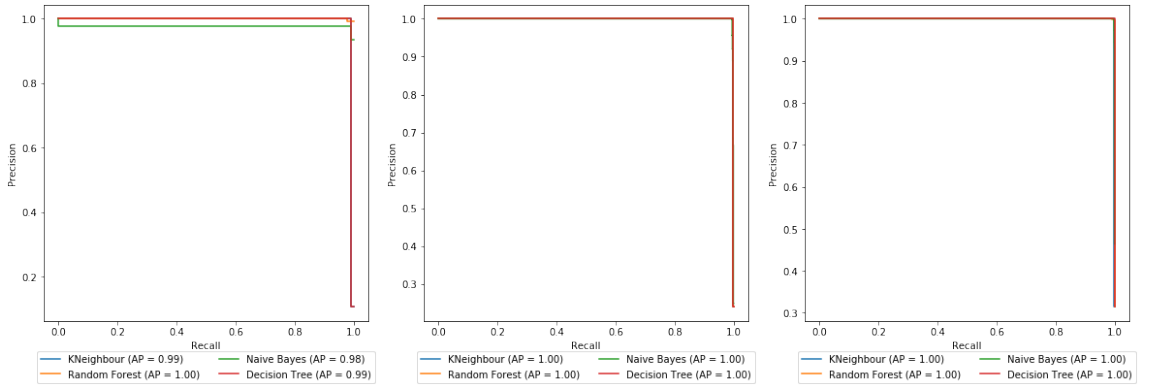
area under the curve implies both recall and precision have a high value. High precision denotes a low false position rate, and high recall means a low false-negative rate.



(a) Attack 1: Low density (b) Attack 1: Medium density (c) Attack 1: High density

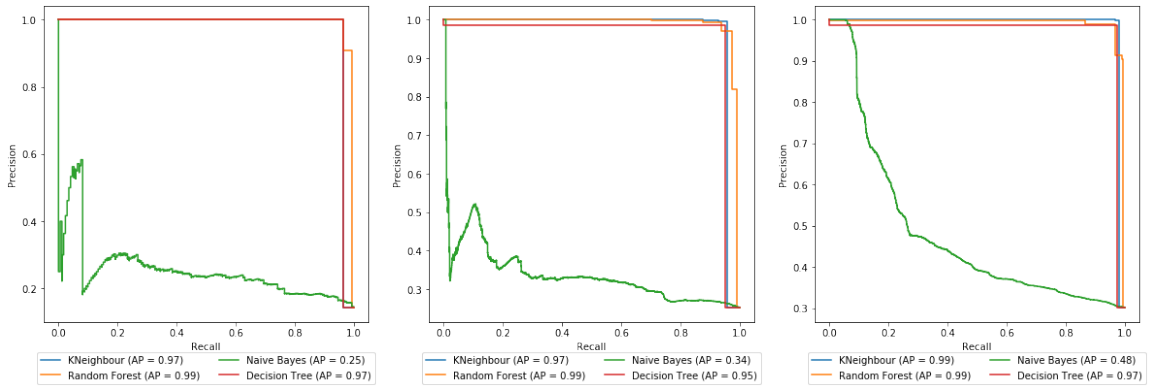


(d) Attack 2: Low density (e) Attack 2: Medium density (f) Attack 2: High density

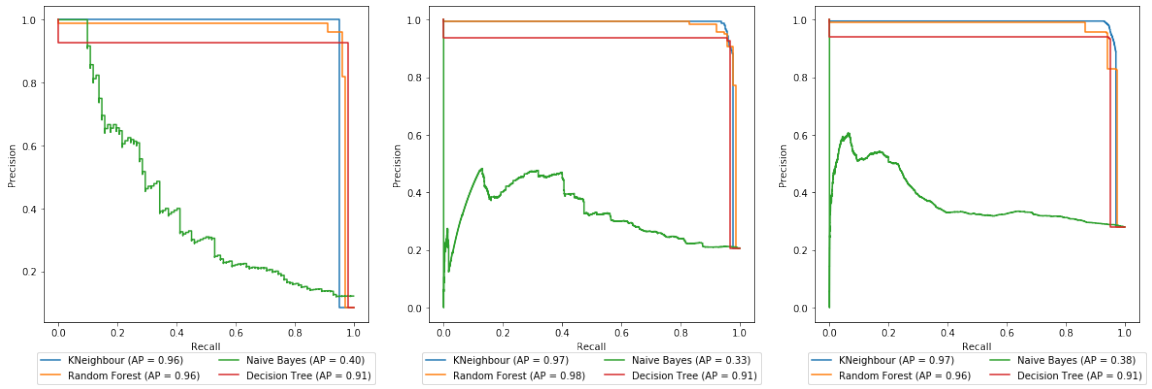


(g) Attack 4: Low density (h) Attack 4: Medium density (i) Attack 4: High density

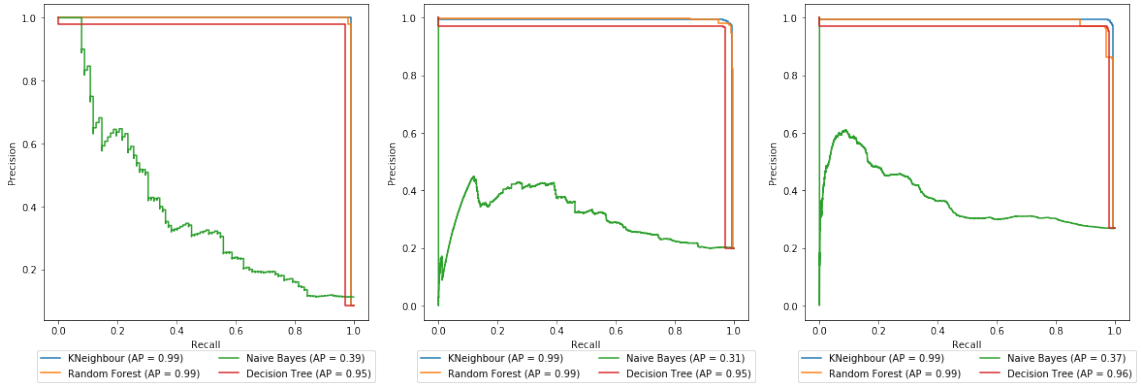
Figure 4.2: Precision-recall curve of attack types 1,2 and 4 in low, medium and high density



(a) Attack 8: Low density (b) Attack 8: Medium density (c) Attack 8: High density



(d) Attack 16: Low density (e) Attack 16: Medium density (f) Attack 16: High density



(g) Modified Attack 16: Low (h) Modified Attack 16: Medium (i) Modified Attack 16: High

Figure 4.3: Precision-recall curve of attack types 8, 16 and modified attack type 16 in low, medium and high density

The attack types are visualized in all three dataset combinations- low, medium and high density dataset. A classifier is said to be performing accurately if both metrics score is higher. Attack type 1 perfectly separates the area into two areas. It is evident

that attack type 1 has zero false positives. It shows the model completely classified the problem. K-Nearest Neighbour, Random Forest, and Decision Tree classifiers perform well for all the types of attacks, with Decision Tree performing slightly less than the other two classifiers. In contrast, Naïve Bayes showed poor results with a noisy graph (zig-zag curve). A noisy graph shows there are small recall values during classification.

However, Attack type 4 represents horizontal line perfectly splits the area and then goes vertically for all the classifiers with almost no noise. It shows the model is performing well for this attack.

4.3 Comparison with Existing Approaches

Based on the performance of the different ML algorithms, we selected K-Nearest Neighbour with the proposed consecutive BSM model to compare with existing techniques. The proposed model was also compared to a raw dataset consisting of single BSM data from a vehicle. As expected, the raw dataset performed poorly for almost all attacks. The only exception was a high precision score for attack type 4. For performance comparison, we selected three recent papers that also used the VeReMi dataset for detecting position falsification. A detailed review of these existing approaches can be found under the Literature survey chapter.

Table 4.8: Comparison of proposed model with existing approaches

Results	**Attack 1**		**Attack 2**		**Attack 4**		**Attack 8**		**Attack 16**	
from:	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
Raw Dataset	57.4	67.2	34.9	18.8	99.8	68.9	29	14.7	31	16
Paper 1: [38]	100	100	40	100	100	99	70	95	80	90
Paper 2: [49]	95.2	83.2	56.1	19.3	95	83.6	96.2	82.5	71.4	42.5
Paper 3: [50]	100	99	94	80	100	99	97	95	98	93
Proposed Model	100	100	99.8	99.7	100	99.8	99.9	97.2	96.8	95.2

The Table 4.8 shows a comparison of the proposed approach with existing techniques. Paper 1 and 3 performed similarly to the proposed model for attack types 1

and 4 generating high precision and recall values; however, Paper 2 showed comparatively less precision and recall value.

Attack type 2 was classified with more than 99% precision and recall using the proposed model, whereas the existing approaches showed varied results. Paper 1 obtained a 100% recall value, but the precision value was very low. Paper 3 showed the highest results out of the three existing approaches, with Paper 2 not showing satisfactory results.

Attack types 2 and 8 are more challenging to detect, and our proposed model achieved higher precision and recall than the other existing techniques. In the case of attack type 16, our model showed promising results in maintaining a balance between precision and recall. Although by modifying attack type 16, we achieved much improved results.

To the best of my knowledge, the proposed classification model outperforms the existing methods in classifying position falsification attacks using the VeReMi dataset.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

This thesis proposes a novel Machine Learning-based approach for classifying position falsification attacks in VANET. Unlike existing techniques that consider individual BSMs, we have used the two-consecutive BSMs from vehicles to create an augmented dataset. This augmented dataset consists of selected features from the individual BSMs based on feature importance and is used to train the proposed model using different machine learning algorithms. The performance of four different machine learning classification algorithms was compared with each other, and it was found that K-Nearest Neighbour and Random Forest classifiers yield the best results. The results obtained from the proposed model were compared with the recent existing techniques using the VeReMi dataset, discussed in the Literature Survey. The obtained results indicate that the proposed approach outperforms the existing methods for classifying all the attacks in terms of precision and recall. This research also designed the modified attack type 16, which shows improved performance for detecting attack type 16. The proposed model is based on the notion of sender and RSU pair approach. This approach aims to reduce the computational overhead from vehicles (OBUs) by designing a detection model to be built on RSU to detect the attack and provide a broader view for detecting the position falsification attack. It also aims to remove the vehicle-to-vehicle dependency in the network for detecting misbehaviour.

5.2 Future Work

The VeReMi dataset is limited to five forms of position falsification attacks and does not fully represent all the possible attacks in VANETs. This proposed model is bound to only the data given in the VeReMi dataset, but additional information can be added in the future. Other information such as sensor data records any obstacles and other information around a vehicle, but this information is not always reliable. Sensors used to store the data might be faulty, tampered with by the attacker, or covered using obstacles. Hence, the sensor alone might not be a stand-alone detector for attacks but can be combined with the proposed model to mark more accurate predictions. Other approaches, such as Deep learning and Neural Networks could be implemented as an extension for this approach.

REFERENCES

[1] Sheng-hai An, Byung-Hyug Lee, and Dong-Ryeol Shin. "A survey of intelligent transportation systems". In: *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE. 2011, pp. 332–337.

[2] Social Determinants of Health. *Global status report on road safety 2018*. `https://www.who.int/publications/i/item/9789241565684`. [Online; accessed 17 June 2018]. 2018.

[3] Saif Al-Sultan et al. "A comprehensive survey on vehicular ad hoc network". In: *Journal of network and computer applications* 37 (2014), pp. 380–392.

[4] Sherali Zeadally et al. "Vehicular ad hoc networks (VANETS): status, results, and challenges". In: *Telecommunication Systems* 50.4 (2012), pp. 217–241.

[5] John B Kenney. "Dedicated short-range communications (DSRC) standards in the United States". In: *Proceedings of the IEEE* 99.7 (2011), pp. 1162–1182.

[6] Ghassan MT Abdalla, Mosa Ali Abu-Rgheff, and Sidi Mohammed Senouci. "Current trends in vehicular ad hoc networks". In: *Ubiquitous Computing and Communication Journal* (2007), pp. 1–9.

[7] Federico Poli. "Vehicular communications: from DSRC to Cellular V2X". PhD thesis. Politecnico di Torino, 2018.

[8] Parul Tyagi and Deepak Dembla. "A taxonomy of security attacks and issues in vehicular ad-hoc networks (vanets)". In: *International Journal of Computer Applications* 91.7 (2014).

[9] Sunilkumar S Manvi and Shrikant Tangade. "A survey on authentication schemes in VANETs for secured communication". In: *Vehicular Communications* 9 (2017), pp. 19–30.

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

[11] 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.

[12] Sonia Alice George, Arunita Jaekel, and Ikjot Saini. "Secure Identity Management Framework for Vehicular Ad-hoc Network using Blockchain". In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE. 2020, pp. 1–6.

[13] Rasha Al-Mutiri, Mznah Al-Rodhaan, and Yuan Tian. "Improving vehicular authentication in VANET using cryptography". In: *International Journal of Communication Networks and Information Security* 10.1 (2018), pp. 248–255.

[14] *VeReMi dataset — VeReMi-dataset.github.io.*

[15] Gagan Deep Singh et al. "A review on VANET routing protocols and wireless standards". In: *Smart computing and informatics*. Springer, 2018, pp. 329–340.

[16] Richard Gilles Engoulou et al. "VANET security surveys". In: *Computer Communications* 44 (2014), pp. 1–13.

[17] Mohammed Ali Hezam et al. "Classification of security attacks in VANET: A review of requirements and perspectives". In: (2018).

[18] Ram Shringar Raw, Manish Kumar, and Nanhay Singh. "Security challenges, issues and their solutions for VANET". In: *International journal of network security & its applications* 5.5 (2013), p. 95.

[19] Mahmood A Al-shareeda et al. "Review of Prevention schemes for Replay Attack in Vehicular Ad hoc Networks (VANETs)". In: *2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP)*. IEEE. 2020, pp. 394–398.

[20] Muhammad Sameer Sheikh and Jun Liang. "A comprehensive survey on VANET security services in traffic management system". In: *Wireless Communications and Mobile Computing* 2019 (2019).

[21] Deepak Kushwaha, Piyush Kumar Shukla, and Raju Baraskar. "A survey on Sybil attack in vehicular ad-hoc network". In: *International Journal of Computer Applications* 98.15 (2014).

[22] Ajay N Upadhyaya and JS Shah. "Attacks on vanet security". In: *Int J Comp Eng Tech* 9.1 (2018), pp. 8–19.

[23] Irshad Ahmed Sumra, Halabi Bin Hasbullah, and Jamalul-lail Bin AbManan. "Attacks on security goals (confidentiality, integrity, availability) in VANET: a survey". In: *Vehicular Ad-Hoc Networks for Smart Cities*. Springer, 2015, pp. 51–61.

[24] 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.

[25] Muhammad Rizwan Ghori et al. "Vehicular ad-hoc network (VANET)". In: *2018 IEEE international conference on innovative research and development (ICIRD)*. IEEE. 2018, pp. 1–6.

[26] Halabi Hasbullah, Irshad Ahmed Soomro, et al. "Denial of service (DOS) attack and its possible solutions in VANET". In: *International Journal of Electronics and Communication Engineering* 4.5 (2010), pp. 813–817.

[27] S Balasubramani, SK Rani, and K Suja Rajeswari. "Review on Security Attacks and Mechanism in VANET and MANET". In: *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, 2016, pp. 655–666.

[28] Mohssen Mohammed, Muhammad Badruddin Khan, and Eihab Bashier Mohammed Bashier. *Machine learning: algorithms and applications*. Crc Press, 2016.

[29] Vineet Chaoji, Rajeev Rastogi, and Gourav Roy. "Machine learning in the real world". In: *Proceedings of the VLDB Endowment* 9.13 (2016), pp. 1597–1600.

[30] Carlos Vladimiro González Zelaya. "Towards explaining the effects of data preprocessing on machine learning". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. 2019, pp. 2086–2090.

[31] Michael W Browne. "Cross-validation methods". In: *Journal of mathematical psychology* 44.1 (2000), pp. 108–132.

[32] Pratap Chandra Sen, Mahimarnab Hajra, and Mitadru Ghosh. "Supervised classification algorithms in machine learning: A survey and review". In: *Emerging technology in modelling and graphics*. Springer, 2020, pp. 99–111.

[33] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. "Supervised machine learning: A review of classification techniques". In: *Emerging artificial intelligence applications in computer engineering* 160.1 (2007), pp. 3–24.

[34] Antonio Mucherino, Petraq J Papajorgji, and Panos M Pardalos. "K-nearest neighbor classification". In: *Data mining in agriculture*. Springer, 2009, pp. 83–106.

[35] Anuja Priyam et al. "Comparative analysis of decision tree classification algorithms". In: *International Journal of current engineering and technology* 3.2 (2013), pp. 334–337.

[36] Andy Liaw, Matthew Wiener, et al. "Classification and regression by randomForest". In: *R news* 2.3 (2002), pp. 18–22.

[37] K Ming Leung. "Naive bayesian classifier". In: *Polytechnic University Department of Computer Science/Finance and Risk Engineering* 2007 (2007), pp. 123–156.

[38] 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.

[39] Lara Codecá et al. "Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation". In: *IEEE Intelligent Transportation Systems Magazine* 9.2 (2017), pp. 52–63.

[40] Christoph Sommer et al. "Veins: The open source vehicular network simulation framework". In: *Recent Advances in Network Simulation*. Springer, 2019, pp. 215–252.

[41] Andras Varga. "OMNeT++". In: *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

[42] Le Liang, Hao Ye, and Geoffrey Ye Li. "Toward intelligent vehicular networks: A machine learning framework". In: *IEEE Internet of Things Journal* 6.1 (2018), pp. 124–135.

[43] 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.

[44] *Weka 3 - Data Mining with Open Source Machine Learning Software in Java*. `http://old-www.cms.waikato.ac.nz/ml/weka/`. (Accessed on 03/18/2021).

[45] Ankita Khot and Mayank Dave. "Position Falsification Misbehavior Detection in VANETs". In: *Mobile Radio Communications and 5G Networks*. Springer, 2020, pp. 487–499.

[46] Pranav Kumar Singh et al. "Machine learning based approach to detect position falsification attack in vanets". In: *International Conference on Security & Privacy*. Springer. 2019, pp. 166–178.

[47] Xiaoping Xue et al. "A trusted neighbor table based location verification for VANET Routing". In: (2010).

[48] Greg Welch, Gary Bishop, et al. "An introduction to the Kalman filter". In: (1995).

[49] 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.

[50] 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.

[51] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Mosa Ali Abu-Rgheff. "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks". In: *IEEE Internet of things journal* 1.6 (2014), pp. 570–577.

[52] J Brownlee. "A Gentle Introduction to Imbalanced Classification". In: *Machine Learning Mastery* 22 (2019).

[53] Kendrick Boyd, Kevin H Eng, and C David Page. "Area under the precision-recall curve: point estimates and confidence intervals". In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2013, pp. 451–466.

VITA AUCTORIS

NAME:	Aekta Sharma
PLACE OF BIRTH:	Faridabad, Haryana, India
EDUCATION:	B.Tech in computer science, Maharshi Dayanand University, Rohtak, Haryana, India, (2017)
	M.Sc. Computer Science, University of Windsor, Windsor, Ontario, Canada, 2021