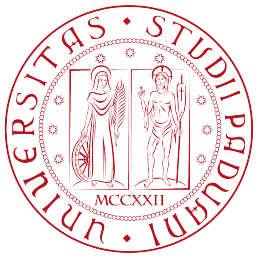


Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea Magistrale in Informatica

Location Claim Verification for Fast Alert Message Propagation in Vehicular Networks

Master's Degree Thesis

Relatore:
Claudio E. Palazzi

Laureando:
Davide Quaglio
1202490

Academic Year 2021/2022

One of the best programming skills you can have is knowing when to walk away for a while. **- *Oscar Godson***

Ringraziamenti

Desidero ringraziare Elena, che nonostante l'organizzazione del trasferimento in Svezia, lo studio della nuova lingua e delle attività lavorative ha saputo supportarmi e sopportarmi durante questo periodo di tesi.

Un sentito ringraziamento ai miei genitori dai quali ho imparato molto e che mi hanno sempre supportato in tutte le mie scelte e anche aiutato economicamente ad iniziare gli studi.

Infine desidero ringraziare il Prof. Palazzi per la grande disponibilità e professionalità dimostratemi, e per l'aiuto fornito durante la stesura di questa tesi.

Abstract

We have witnessed a tremendous progress in vehicle networking over the last few years, with a particular emphasis on safe mobility and intelligent navigation. One of the most crucial application domains is certainly represented by collision avoidance techniques aimed at mitigating chain accidents.

State-of-the-art solutions leverage on the fast propagation of *Alert* messages enabled by position based forwarding among vehicles. Unfortunately, due to the highly dynamic nature of vehicular ad-hoc networks (VANETs), needed information such as the position and the transmission range of vehicles is not only challenging to be collected precisely, but it is also easily affected by adversarial attacks. In this work, we discuss two possible attacks that malicious nodes could easily perform to jeopardize the performance of position based forwarding protocols, hence hindering road safety. We also propose and analyze a possible countermeasure: a validation system, based on machine learning (ML) techniques, able to detect malicious nodes, discard their false information, and protect against these attacks.

Introduzione

Negli ultimi anni abbiamo assistito a enormi progressi nelle reti veicolari, con un'enfasi particolare sulla mobilità sicura e sulla navigazione intelligente. Uno dei domini applicativi più cruciali è sicuramente rappresentato dalle tecniche di prevenzione delle collisioni volte a mitigare gli incidenti a catena.

Le soluzioni all'avanguardia sfruttano la rapida propagazione dei messaggi di allerta consentita dall'inoltro basato sulla posizione tra i veicoli. Sfortunatamente, a causa della natura altamente dinamica delle reti ad-hoc veicolari (VANET), le informazioni necessarie come la posizione e il raggio di trasmissione dei veicoli non solo sono difficili da raccogliere con precisione, ma sono anche facilmente influenzate da attacchi. In questo lavoro, si discutono due possibili attacchi che nodi malevoli potrebbero facilmente applicare per compromettere le prestazioni dei protocolli di inoltro basati sulla posizione, ostacolando quindi la sicurezza stradale. In aggiunta si propone anche una possibile contromisura: un sistema di validazione basato su tecniche di Machine Learning (ML), in grado di rilevare nodi dannosi, scartare le loro false informazioni e quindi proteggersi da questi attacchi.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the advancement of technology the smart vehicles industry is expanding rapidly, as it not only has the potential to lower prices by reducing the amount of gasoline required or the amount of time spent traveling, but it also has the potential to alleviate congestion on the roads and most importantly to save lives [1].

Data can now be exchanged between a wide range of everyday devices, improving the performance of many applications, thanks to recent advancements in the Internet of Things (IoT) field; given the many and interesting achievements obtained by this new field, and the many challenges that vehicular ad-hoc networks (VANET) pose, there have been a lot of studies being analyzed in order to bring such benefits to VANETs, causing the field of the Internet of Vehicles (IoV) to be born. The interactions between data from various types of sensors such as cameras, microphones, GPS, radars, and some more strictly related to vehicles such as speedometer, ABS, brake assist system (BAS), and electronic stability control (ESC) are studied in this new area, especially in the field of safety applications because it is aimed at reducing accidents and saving lives on the road [2].

Vehicles not only rely on their own devices, but they can also access data from external sensors such as other automobiles, pedestrians, roadside units (RSU), and application servers; all of these communication scenarios are referred to as Vehicle-to-Everything (V2X), as shown in Figure 1.1. V2X refers to a variety of communication scenarios that can collaborate by sharing data, resulting in a cooperative awareness system. All information sources must gain several perspectives on the environment in order to enable a wide range of use cases, such as improved safety, passenger infotainment, and traffic optimization.

One of the main challenges in a IoV scenario is given by the fact that vehicles

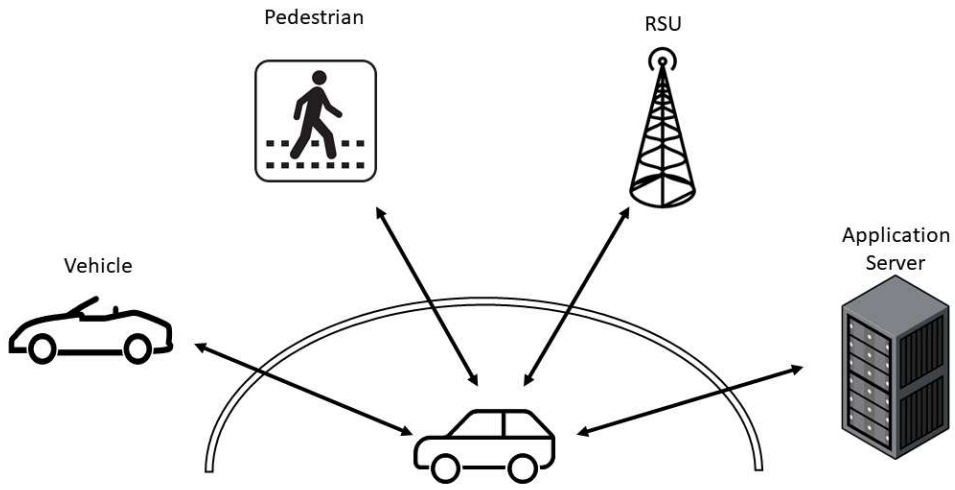


Figure 1.1: Example of main V2X communication scenarios.

move a lot, are fast and they need to handle and share data with many other vehicles and sensors. Add to that the requirement for high reliability, which is especially important in safety applications due to their time-critical nature, data must be exchanged without delay and packet loss. In the aforementioned type of scenario the information transmitted is of public interest, and typically benefits a group of vehicles rather than a single one, implying that a broadcast approach is preferable to a unicast one [3]. By examining the IEEE 1609.2 standard, it is possible to see that instead of relying on a cellular network for applications such as car safety, public service, and other similar operations, Dedicated Short Range Communications (DSRC) are to be deployed in order to reduce communication delays. Additionally, because safety applications are time-critical, the processing and bandwidth overhead due to security must be kept to a minimum to improve responsiveness and decrease the likelihood of packet loss.

Given the nature of VANETs, and also the requirements coming from the IoV field which requires the ability to communicate with a wide variety of devices, not only inside the car, but also with other "things" that can be found in the proximity of it, the sole source of communication of this network is represented by wireless links, which are susceptible to a variety of attacks. Security issues are extremely concerning for all the possible VANET scenarios because they have the potential to reduce, or completely eliminate, the benefits of implementing these systems and most importantly they have the potential to endanger human lives, particularly in safety applications. Given the aforementioned scenario definition, the first line of

defense is obviously implementing a security mechanism for authenticating these messages, but sometimes this is not enough and especially for safety applications it is important to be able to defend these networks against internal attackers too [4, 5, 6].

Clearly, one of the most crucial applications in IoV in general, and VANETs in particular, are those related to safety. These applications are based on the forwarding of warnings on unseen events in case of poor visibility or driver absent-mindedness. They hence embody an extremely useful aid for the driver or for an automated response system to avoid crashes; some representative examples are:

- Emergency vehicle warnings: alert other vehicles on the road about an incoming emergency vehicle, such as police car or ambulance, and that the road needs to be cleaned for its arrival;

- Lane change warnings: warn the drivers of possible unexpected lane changes of other vehicles;

- Sudden slowdown or brake of vehicle warnings: warn the drivers about any braking or slowing down in front of her/him in order to avoid rear-end collisions, this warning is especially useful in case of foggy environments;

- Collision warnings: whenever an accident happens, or is about to happen, this information is shared with interested cars in order to have them brake immediately (and not after a chain of drivers' reaction times) or dodge the problem.

For all these applications, a clear, detailed, and up-to-date perspective of the environment is required in order to make the best decisions to protect the lives of drivers and pedestrians. To ensure that the decisions made are based on the most detailed and up-to-date representation of the environment, the nodes in the network should exchange information periodically. However, by exchanging too many messages the network's likelihood of packet collision increases, resulting in the loss of essential information for the aforementioned choices and in increased delays before message delivery, which may significantly affect their effectiveness.

For most of the previously examined applications, alerts regarding what is going on must be provided not just to cars in close proximity to the issue, but also to farther ones as quickly as feasible; in order to do so, a multi-hop broadcast technique must be adopted. The main problems with such approaches are that

(i) the overhead generated by a simple broadcast routing mechanism (e.g., flooding) would be unbearable for the VANET, resulting in high packet loss, and (ii) each hop introduces a small delay in the transmission generated by the message handling procedure. Thus, if the number of hops increases too much, the delay increases as well, and the information being delivered loses significance [7, 8]. The ideal solution in order to solve problems (i) and (ii) would be to minimize or at least significantly reduce the number of hops required to deliver the message to other vehicles by having the farthest nodes in each hop being the solely forwarding the message; yet, this would require a perfect knowledge about the vehicles' position and their transmission range, which is not feasible in an everyday scenario. A more suitable solution is to try to estimate the transmission range of the vehicles by exploiting the exchange of *Hello* messages containing the vehicles' position and then to decide the forwarders based on the gathered information.

The main issue with the aforementioned mechanism is that it requires the knowledge and the exchange of the precise location of the vehicles to function properly, making the routing protocols based on this approach vulnerable to attacks that alter the vehicles' position. This is why we decided to investigate on the effects on VANET's position-based routing protocols caused by attacks changing the attacker's location and studied a possible countermeasure based on a position validation mechanism able to identify misbehaving nodes and false position statements.

The routing algorithm family that has been considered in this work is made up of some key characteristics that allow it to contain a wide variety of different protocols. These key characteristics are as follows:

- The routing method is position-based, adheres to the aforementioned criteria, and attempts to select as few forwarders as feasible along the multihop network to speed up the propagation of Alert messages.

- The IVC solely supports vehicle-to-vehicle (V2V) communication without the use of roadside units (RSU) or other infrastructures.

- Even if multihop forwarding is employed, messages should be delivered as rapidly as possible within a designated area of interest.

To assess the risks and effects of the selected attacks on the aforementioned family of routing protocols, a cutting-edge protocol representative of this class of algorithms, the Fast Broadcast multi-hop algorithm [9], has been considered,

which will allow the discussion to be clarified through a real-world case study while remaining general.

1.1 Main contributions

The main contributions of this thesis are (i) the assessment of the threats and impact of two possible attacks, i.e., position cheating and black hole, on state-of-the-art position based vehicular safety applications and (ii) the design and implementation of a validation system that can be used in the class of algorithms in exam to counteract the aforementioned attacks, this solution takes the name of Fast Broadcast with Location Validation (FBLV).

1.2 Organization

The rest of this thesis is organized as follows. The following Chapter introduces the background of the project and discusses research on VANET networks, with a special emphasis on tackling safety issues. Chapter 3 describes the Fast-Broadcast algorithm. Chapter 4 gives an introduction to k-NN the ML technique used to classify the nodes and Chapter 5 describes the implementation of the validation system inside Fast-Broadcast. The simulation assessment is detailed in Chapter 6. The position cheating attack and the black hole attack are described in Chapter 7. In Chapter 8 are described the metrics used to measure the performances of FBLV and in Chapter 9 the various simulation outcomes achieved under various scenarios are assessed. Finally, Chapter 10 draws the conclusions.

Chapter 2

Background

Although vehicular ad-hoc networks (VANET) are a subset of mobile ad-hoc networks (MANET), there are several features that make MANET's routing protocols unsuitable for usage in VANET. First and foremost, a MANET is a network that does not require any pre-existing infrastructure. Second, nodes in this sort of network can move as they like without following any particular rule or route, whereas in a VANET, cars perform predictable maneuvers and follow some roads, even though they move faster [10, 11].

As already mentioned, in a VANET safety application the messages are of public interest therefore the more vehicles receives it the higher the possibilities of preventing incidents are, and there should be as little delay as possible. Therefore, broadcast routing protocols are preferred to unicast ones. The broadcast routing protocols for these networks are divided into two main branches: single-hop and multi-hop broadcast routing protocols. The primary difference between these two types of protocols is the way that packets travel over the network; with a single-hop architecture, each node will periodically forward a fraction of the data he holds to neighboring vehicles, whereas with multi-hop protocols, each vehicle that receives the message will flood the network with data. When using multi-hop routing protocols, it is vital to reduce the number of hops used in order to reduce both the time that it takes for an Alert message to spread and the number of nodes that attempt to relay a message at the same time, since these are the primary sources of message delivery delays [8, 7].

Another way to differentiate between routing protocols is based on the route discovery method used, which divides them into two major classes: topology based and position based routing protocols.

2.1 Topology based routing protocols

The routing rules in this family of protocols are determined by a table that defines the connections with the neighboring nodes. When a packet needs to be delivered to a certain node, the optimal path in this table is picked. Depending on when this table is constructed, this family can be further divided into two categories: proactive and reactive topology-based routing protocols.

2.1.1 Proactive

The proactive routing protocols, also known as table-driven routing protocols, are distinguished by the fact that each node maintains a routing table including the path required to reach each node in the network. The tables are kept up to date by transmitting updates to the neighboring nodes on a regular basis, which helps to keep the tables consistent. When an unexpected change occurs in the network, the change is propagated quickly to all the nodes in the network. A large amount of control traffic must be generated in order to keep the routing table up to date; this keeps the channel busy and increases the likelihood of a packet collision. DSDV, OLSR, and WRP are examples of routing protocols that fall into this category.

It is not feasible to use this group of routing protocols in VANET networks because the nodes move frequently, and therefore many connections between them may be lost. This creates a major problem: the tables need to be updated very frequently to keep up with the nodes moving, resulting in significant overhead for the entire network [12].

2.1.2 Reactive

These protocols are also referred to as on-demand since the route to a specific node is only constructed when a node is actually required to transmit a message and it is immediately discarded once the message has been received and processed. Normally, these pathways are discovered by using a route discovery process, such as flood. A packet's path to the destination may be totally stored on the packet that is being forwarded (e.g., DSR), or every node in the forwarding chain may just have the information of the next node in the forwarding process (e.g., AODV).

2.1.3 Hybrid

There are also hybrid protocols, which attempt to blend proactive and reactive measures. The Zone Routing Protocol is an example of such a protocol (ZRP). ZRP divides the topology into zones and attempts to use multiple routing protocols within and between the zones based on the protocol's weaknesses and strengths. Because ZRP is completely modular, any routing protocol can be used within and between zones [13].

2.2 Position based routing protocols

This type of protocol needs that the nodes are equipped with a sensor capable of determining their spatial location (e.g., GPS), as this is the primary parameter utilized to determine the best path to take when forwarding a packet through the network. Apart from the fast-broadcast protocol, which has been utilized and examined in this thesis, there are other additional protocols in the same class, such as the Greedy Perimeter Stateless Routing protocol (GPSR). Based primarily on information about a router's network topology's immediate neighbors, this routing system makes greedy forwarding decisions that result in packet loss. When a packet enters a zone where greedy forwarding is no longer feasible, the algorithm recovers by routing the packet around the region's perimeter [14]. Another position based protocol built for networks with high node mobility is the Distance Routing Effect Algorithm for Mobility (DREAM) [15]. In this protocol, each node keeps track of the position of the neighboring nodes by utilizing a routing table in which the coordinates are stored; when a node needs to deliver a message to another specific node it sends the message to all of its one-hop neighbors in the direction of the target receiver. Then the process starts again until the message reaches the receiver.

FB, the algorithm used in this thesis as representative for the family of protocols in exam, is a multi-hop position based routing protocol.

2.3 Related work

One of the main problems when handling routing on a vehicular network with a multi-hop algorithm is that the ideal choice would be to have the farthest node in a hop forward the message to the rest of the network [8, 7], even though this would require every node to know the topology and transmission range of every

other node in the network, which is unrealistic in a real-world environment. In previous studies many routing solutions investigated how to avoid the need for global real-time knowledge by selecting the next forwarder probabilistically from one of the farthest away from the sender [16, 17, 18].

V2X systems allow interactions with various entities such as vehicles (V2V), pedestrians (V2P), and several other types of infrastructure (V2I). However, because of the many interactions, the security challenges in such a broad field are numerous. The authors of [19] in their survey discuss first of all the main security challenges for V2X e.g., (i) dynamic network topology, (ii) network scalability, (iii) communication latency, (iv) users trust, and many others. Moreover they categorized and explored the different attacks that can target vehicular networks and finally, they analyze the different defense techniques that can be implemented dividing them into three main categories: Symmetric Key Cryptography, Privacy Preservation, and Message Authentication.

The authors of [20], similarly deal with existing designs for securing V2X applications, focusing primarily on cryptographic solutions such as Public Key Infrastructure (PKI) and global efforts for standardizing V2X security from important organizations such as IEEE, SAE, ETSI, and many others. They also identified various threats that can harm V2X networks and investigate detection or mitigation mechanisms, with a particular focus on some of them, like DoS, Sybil, and false data injection.

With the rising popularity of self-driving cars and Intelligent Transportation Systems (ITS), the related security challenges are becoming increasingly important research areas. In their survey [21], Abdullahi Chowdhury et al. expressed the strict security requirements that must be abided to for the public adoption of such vehicles and analyze the most recent literature related to attacks reporting effects and possible countermeasures. Similar to the previous mentioned works, the authors of [22] examined the many threats to VANET networks, their logic, and potential solutions.

Some studies focus on the consequences of GPS spoofing in VANET networks by examining three separate outsider assaults that do not have access to important cryptographic key material, i.e., a replay attack, a sybil attack, and an attack that exploits old certificates. The authors identify two distinct families of potential solutions based on (i) the prevention of time stamp jumps, which were used by attackers to obtain valid future messages from other nodes in the network, and (ii) the introduction of short-lived pseudonym certificates, which

eliminates the possibility of creating messages with future time stamps [23].

Furthermore, recent researches have looked into benefits and the security challenges in VANETs using a Software Defined Networking (SDN) architecture, which has emerged as a promising solution for simplifying network management and enabling innovation through network programmability [24, 25].

Related to the main focus of this thesis, researchers already investigated the impact of position cheating-based attacks on VANETs on safety critical applications that use routing algorithms that rely heavily on the correct position of every node to function properly [6]. In [26] the authors investigated a mechanism to identify vehicles that lie about their position by exploiting directional antennas and by comparing their own information with the data coming from the neighboring nodes in a cooperative scenario. A similar solution can be also found in [27], where the authors proposed an authentication mechanism to guarantee the authenticity of the messages and a solution that detects malicious nodes that uses collaborative neighbors, after receiving enough information from the neighboring nodes a single car is able to detect misbehaving nodes. On the other hand, the authors of [28] employed a data-oriented trust model by defining a trust metric that combines vehicle speed and received power signal and is then used in a k-NN classifier to determine whether or not a vehicle is sending an inaccurate location.

In this context, our work investigates the effects of two unique forms of attacks on position based routing protocols designed for VANET networks i.e., (i) position cheating and (ii) black hole. The foundation of our validation system for preventing attackers from endangering the network is that the receiver can identify malicious nodes by assessing the SNR (Signal-to-Noise Ratio) of the messages and the distance from the sender.

Chapter 3

Fast Broadcast

Fast broadcast is a multi-hop broadcast protocol for IVC that has been designed to minimize the number of hops in order to reduce the time it takes for an *Alert* message to propagate and the number of nodes that try to relay a message at the same time, as these are the main causes of message delivery delays [8, 7]. Several routing algorithms assume that all cars are aware of their transmission range ahead of time. However this is an unrealistic assumption, particularly in VANET networks, due to the rapid mobility of the nodes and the constantly changing presence of obstructions in the transmission path.

Thanks to a distributed process, every vehicle is able to infer its maximum transmission range by taking advantage of the exchange of *Hello* messages in the network. This represents the most significant advantage of FB over other systems. Two primary phases comprise this algorithm i.e., (i) the estimation phase and (ii) the broadcast phase.

3.1 Estimation phase

This is FB's main phase and is always active during the whole car's activity. In order to keep the transmission range estimates up to date, the vehicles keep exchanging *Hello* messages, following the procedure shown in Algorithm 1. Every vehicle calculates a random waiting time (line 3 and 4), after which, if the channel is free (line 5), it will send the *Hello* message containing the sender GPS position and its estimation of the actual transmission range (line 6, 7, and 8). Time is divided into turns and to keep estimations fresh all the data collected during one turn are kept for the duration of the next turn, before being discarded. The creators of this protocol advise setting the time of a turn to one second; this value

could be decreased further to improve the freshness of the information used by nodes at the expense of the number of *Hello* messages exchanged.

Algorithm 1 Estimation phase - Hello message sending procedure

```
1:  function SendHelloMessage
2:      for each turn do
3:          sendingTime ← random(turnSize)
4:          wait(sendingTime)
5:          if ¬ (heardHelloMsg() ∨ heardCollision()) then
6:              helloMsg.declaredMaxRange ← max(LMFR, CMFR)
7:              helloMsg.senderPosition ← retrievePosition()
8:              transmit(helloMsg)
9:          end if
10:     end for
11: end function
```

Information for the current turn is represented by Current-turn Maximum Front Range (CMFR) and Current-turn Maximum Back Range (CMBR). The first one represents the estimate of the greatest frontward distance from which another car along the platoon can be heard by the one under consideration. The latter one calculates the greatest rearward distance at which the car under consideration can be heard; a visual representation of these two parameters can be seen in Figure 3.1.

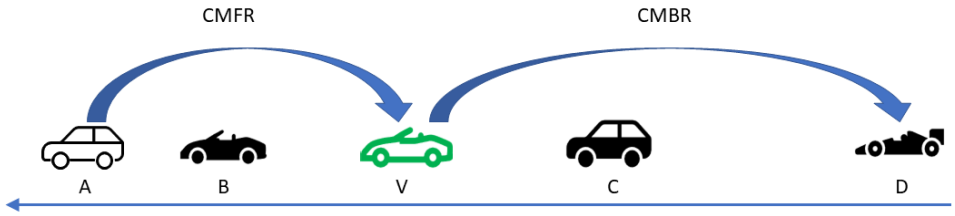


Figure 3.1: Graphical representation of CMFR and CMBR.

The receiving procedure is shown in Algorithm 2. When a vehicle receives a *Hello* message it must first calculates its distance from the sender and reads the sender's estimate of his maximum range. Following that, the vehicle must determines the direction from which the message is coming, which regulates which variable it must update between the *CMBR* and the *CMFR*. Using the message from the front as an example, the *CMFR* is updated between the maximum of the actual *CMFR*, the distance between the two nodes, and the MaxRange contained in the *Hello* message.

Algorithm 2 Estimation phase - Hello message receiving procedure

```
1: function ReceiveHelloMessage(helloMsg)
2:     myPosition ← retrievePosition()
3:     senderPosition ← helloMsg.senderPosition
4:     maxRange ← helloMsg.maxRange
5:     d ← distance(myPosition, senderPosition)
6:     if receivedFromFront(helloMsg) then
7:         CMFR ← max(CMFR, d, maxRange)
8:     else
9:         CMBR ← max(CMBR, d, maxRange)
10:    end if
11: end function
```

CMFR and *CMBR* are continuously updated during the course of a turn; when the turn ends, the values contained in these variables are stored in the Last-turn Maximum Front Range (*LMFR*) and the Last-turn Maximum Back Range (*LMBR*), respectively. The old values in *LMFR* and *LMBR* are discarded because they are considered obsolete, and *CMFR* and *CMBR* are cleaned up.

3.2 Broadcast phase

This is the phase that begins when a vehicle determines that it is important to notify preceding cars of a hazard or an issue that they should be aware of. In this phase the vehicle is therefore responsible for sending an *Alert* message to all nodes in the sender's zone of interest. When a vehicle has to send an Alert message, it will also include its *MaxRange* estimates, which is an estimate of how far the transmission is predicted to travel and is selected between the *CMBR* and the *LMBR*. This approach will make possible to exploit the estimated transmission range collected during the estimation phase to minimize redundancy and enable rapid message delivery.

Before attempting to convey the *Alert* message, each receiver must compute its waiting time in order to determine the vehicles' priority for forwarding the broadcast message. This period is measured in terms of a contention window (CW), which may be determined using (3.1).

$$CW = \left| \frac{MaxRange - d}{MaxRange} * (CWMax - CWMin) + CWMin \right| \tag{3.1}$$

Using (3.1) to calculate each vehicle's CW, we can see in Figure 3.2 that the

further a vehicle is from the sender, and thus its distance from the sender is closer to the *MaxRange* value, the more likely will be the next forwarder because his *CW* is smaller.

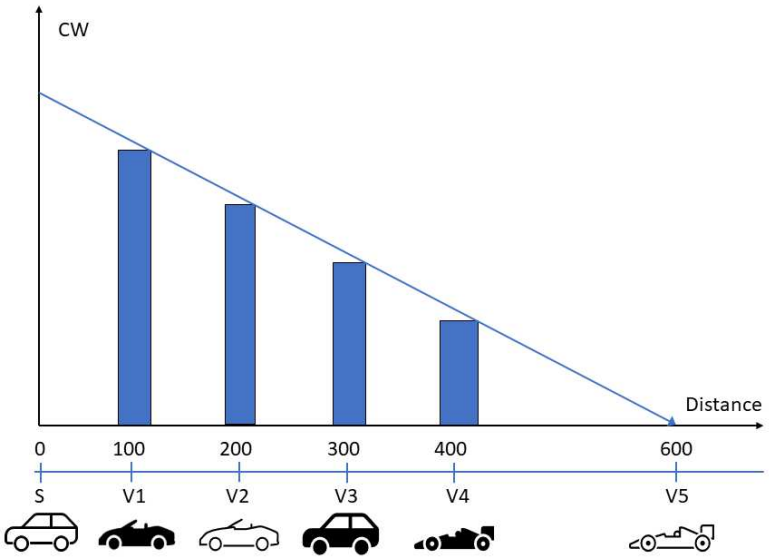


Figure 3.2: Image depicting the relationship between CW and distance.

Algorithm 3 Broadcast phase - Alert message handling procedure

```
1:  function HandleBroadcastMessage(alertMsg)
2:      cwnd ← computeCwnd()
3:      waitTime ← random(cwnd)
4:      wait(waitTime)
5:      if sameBroadcastHeardFromBack() then
6:          exit()
7:      else if sameBroadcastHeardFromFront() then
8:          restartBroadcastProcedure()
9:      else
10:         alertMsg.maxRange ← max(LMBR, CMBR)
11:         alertMsg.senderPosition ← retrievePosition()
12:         transmit(alertMsg)
13:     end if
14: end function
```

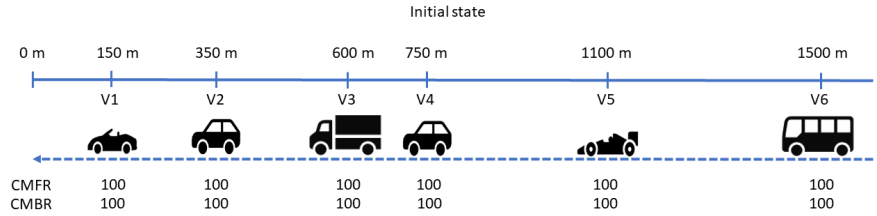
In Algorithm. 3 is shown the handling procedure for an *Alert* message. When a vehicle receives a broadcast message coming from the front it uses (3.1) to determine its *CW* (line 2) and then computes a random waiting time based on it (line 3). If the message is received from the back, it indicates that it has already been forwarded by another vehicle, and the node can stop attempting

to forward it (lines 5 and 6). On the other hand, if the message is received from the front during the *CW* expiration, the procedure must be restarted with the new parameters from the new *Alert* message (lines 7 and 8). If none of the aforementioned events have occurred while the node was waiting for the timer to expire, it can proceed to forward the *Alert* message, which includes its *MaxRange* and position (lines 10, 11 and 12).

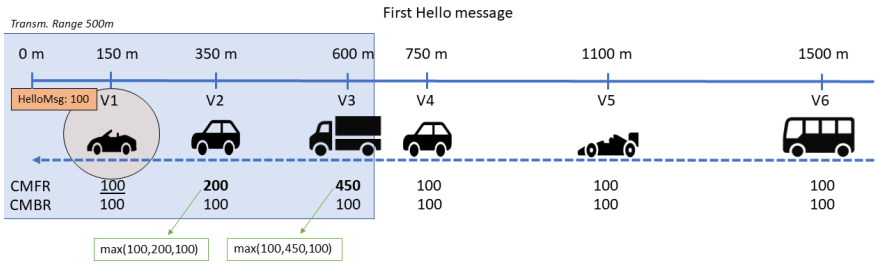
3.3 A pratical example

In order to better explain how Fast Broadcast works, in Figure 3.3 is provided a storyboard representing a FB's turn. For the sake of this example, the transmission range is to be considered equal to 500 meters and every node starts with an initial state in which CMBR = CMFR = 100 meters as shown in Figure 3.3a. The scenario represents a strip shaped road long 1500 meters, with cars at different distances.

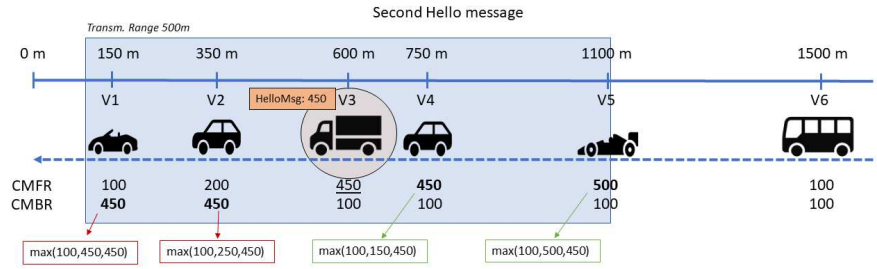
Vehicle V1 is the first node to initiate the *Hello* message sending procedure, as shown in Figure 3.3b. The *Hello* message includes the *MaxRange*, which is its *CMFR* actual estimation of 100 meters, as well as its position. Vehicles V2 and V3 calculate their distances from the sender after receiving the *Hello* message, and because they are receiving the message from the front, they will update their *CMFR* estimation with the maximum value between (i) their old value, (ii) the distance, and (iii) the declared MaxRange in the *Hello* message. Figure 3.3c and 3.3d depict the generation of two more *Hello* messages, and the vehicles receiving the message from behind change their *CMBR* based on their old value, and (ii), and (iii). After these three basic *Hello* messages, the estimation values results significantly altered, and when Vehicle V3 in Figure 3.3e transmits the *Alert* message, the estimation of the *MaxRange* has reached the value of 500 meters, which is indeed the value of the transmission range. It is important to highlight that with just three *Hello* messages a good amount of vehicles estimated a value of 500 meters and many others obtained closer values.



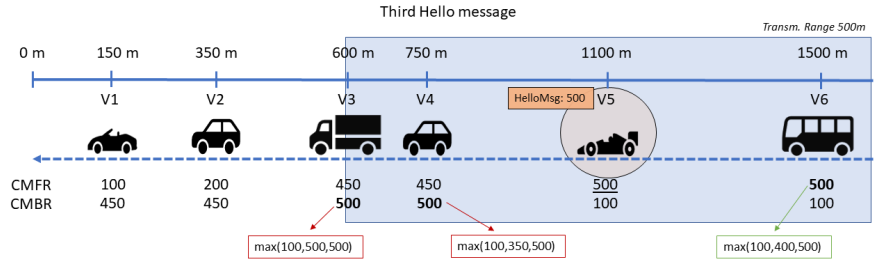
(a) Initial state.



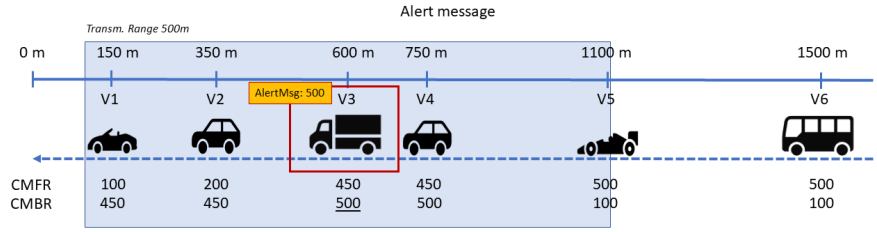
(b) First Hello message sending procedure.



(c) Second Hello message sending procedure.



(d) Third Hello message sending procedure.



(e) Alert message sending procedure.

Figure 3.3: FB simulation using Two Ray Ground and offsets 500, 1000 and 1500 m.

3.4 Fast-broadcast changes

To simplify the simulations some changes to Fast-broadcast protocol had been made. First of all the antennas are omni-directional and so all the messages, Alerts and Hellos, will be sent in both directions at the same time; the vehicles will still be able to discern the direction of the incoming message by comparing their position and the position contained in the message received.

Since all vehicles have the same distance between them, to speed up the convergence of the estimated transmission range value it has been decided to use only one parameter between *CMBR* and *CMFR* that will represent the *MaxRange* for both preceding and following cars. This way the value that represents the *MaxRange* will always be the maximum value estimated for every vehicle.

Chapter 4

K-Nearest Neighbors

K-Nearest Neighbors (k-NN) [29] is a simple and powerful supervised machine learning technique that can be used for classification or regression. The basic premise that this algorithm follows is that similar elements must be closer to each other, causing them to be similar. Each input is categorized in respect to the k nearest element, generally using a euclidean distance. In Figure 4.1, it is possible to see that if $K = 3$, the input star would be categorized as a green diamond, but if "k" is changed to a number of 7, the new membership class of the star would be yellow triangle.

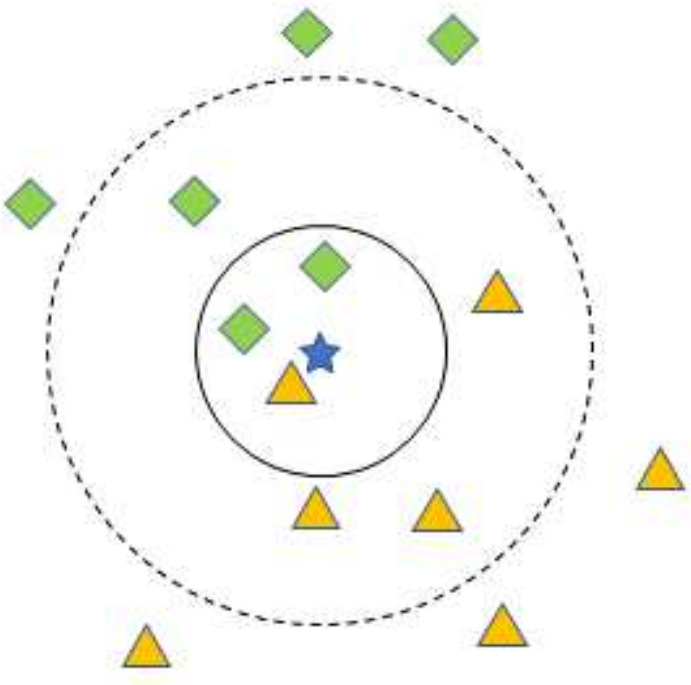


Figure 4.1: Classification example with k-NN.

The input in this work is a pair composed of the SNR of the received message and the distance from the sender. At the same time the expected output is the classification of the sender, which can be either a good or a misbehaving node.

4.1 Training dataset

The balanced training dataset has been built from a series of simulations run by having different nodes move with the RandomWalk2d mobility model in a box scenario in which the vehicles keep exchanging some messages; even if RandomWalk2d would not be the best to use in a VANET scenario, in this case it is very useful in order to acquire samples from very different distances. During the simulation for every message received the SNR and the distance between the nodes have been saved generating the correct samples of the dataset. The negative samples are generated from the correct samples using these steps:

1. Selection of one of the correct samples and creation of a copy of it changing the label to malicious;
2. Randomly selection of a offset value in the range [450, 600];
3. Addition or removal of the obtained value to the previously selected sample.

An example of the final dataset can be seen in Figure 4.2

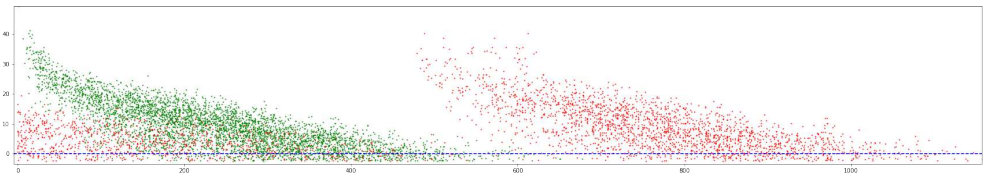


Figure 4.2: Example of dataset with offset in range [450, 600].

It is important to explain the step 2 since it represents the core behind a correct representation of a malicious behaviour in order to be able to correctly classify as much messages as possible. When a malicious node sends a message, he shift his position by a certain offset down the platoon. Afterwards, as we can see in Figure 4.3 when node M sends a message, what the other vehicles perceive as M position becomes M', meaning that vehicle V4 which should be the farthest from M is now the closest one, and the other way around vehicle V1 is now the farther. If step 2 instead only adds the offset, the only nodes that k-NN would be able to classify as misbehaving would be the farthest from the real position of the sender.

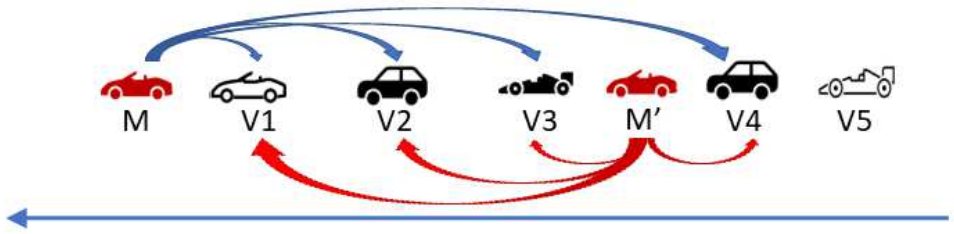


Figure 4.3: Example of malicious hop in the platoon.

Chapter 5

Fast broadcast with Location Validation

As stated in the previous chapter, k-NN can detect patterns in the elements of a population. The main purpose of the validation system is to correctly identify fraudulent nodes in order to ignore their messages. As shown in Figure 4.2, we can easily detect a trend in the data composed of the distance between the sender and the recipient of a message and the Signal to Noise Ratio (SNR) of the same message, and this indicates that these features may be deemed significant for use in distinguishing between well-behaving and malevolent nodes.

Algorithm 4 FBLV - Hello message handling procedure

```
1:  function ReceiveHelloMessage(helloMsg)
2:      myPosition ← retrievePosition()
3:      senderPosition ← helloMsg.senderPosition
4:      maxRange ← helloMsg.maxRange
5:      snr ← helloMsg.snr
6:      d ← distance(myPosition, senderPosition)
7:      class ← knn.predict(d, snr)
8:      if class == 1 then
9:          exit()
10:     end if
11:     if receivedFromFront(helloMsg) then
12:         CMFR ← max(CMFR, d, maxRange)
13:     else
14:         CMBR ← max(CMBR, d, maxRange)
15:     end if
16: end function
```

In Algorithm 4 and 5 it is possible to see how FBLV handles the reception

of a *Hello* or *Alert* message. The receiver must first computes both the distance between its position and the sender's position and acquire the SNR of the received message; these two variables are then utilized to perform the classification via k-NN. If the sender is deemed malicious, the entire message is ignored, otherwise the message is handled according to the standard FB procedure. For example, if the message is a *Hello*, the receiver will update its *CMFR* (line 12) or *CMBR* (line 14) if necessary. On the contrary, if the message is an *Alert*, the node will calculate its *CW* (line 10) using the MaxRange read in the message and will forward it in case of being the first one to wake up (lines 18, 19, 20).

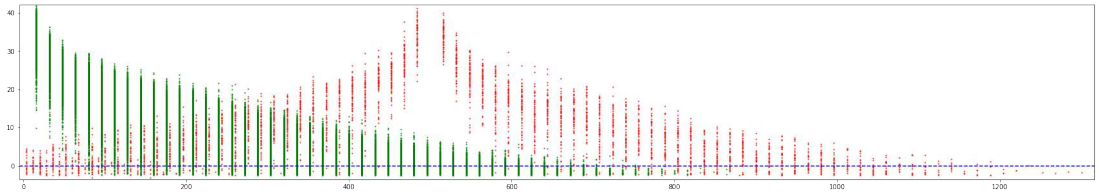
Algorithm 5 FBLV - Alert message handling procedure

```
 1: function HandleBroadcastMessage(alertMsg)
 2:     myPosition ← retrievePosition()
 3:     senderPosition ← helloMsg.senderPosition
 4:     d ← distance(myPosition, senderPosition)
 5:     snr ← helloMsg.snr
 6:     class ← knn.predict(d, snr)
 7:     if class == 1 then
 8:         exit()
 9:     end if
10:     cwnd ← computeCwnd()
11:     waitTime ← random(cwnd)
12:     wait(waitTime)
13:     if sameBroadcastHeardFromBack() then
14:         exit()
15:     else if sameBroadcastHeardFromFront() then
16:         restartBroadcastProcedure()
17:     else
18:         alertMsg.maxRange ← max(LMBR, CMBR)
19:         alertMsg.senderPosition ← retrievePosition()
20:         transmit(alertMsg)
21:     end if
22: end function
```

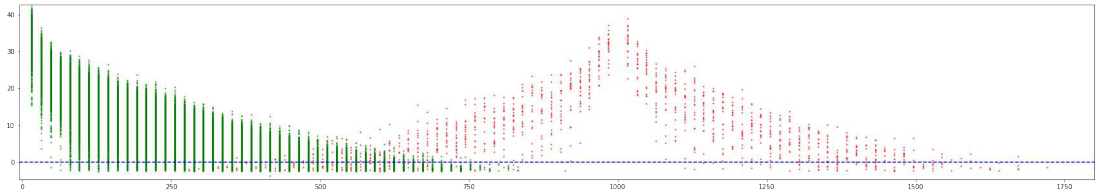
5.1 Technique limit

From Figure 4.2 it is possible to see that there are some areas in which malicious and well behaving nodes blend without a clear division between them, showing that here k-NN has problems to distinguish between misbehaving and correct nodes. In Figure 5.1 is easier to see how actual messages in the platoon's with

their correct labels are space-distributed. Figure 5.1a shows that if the vehicles distance is between 80 and 380 meters, which is a very wide range, the SNR between good and misbehaving nodes is very similar, worsening the performance of the detection mechanism.



(a) Example with malicious offset of 500 meters.



(b) Example with malicious offset of 1000 meters.

Figure 5.1: Example of a message exchange phase. In red misbehaving messages coming from a spoofed position and in green good ones.

There are 2 main causes for the problem mentioned above:

- The SNR decline is significant only for the first 80 meters, and the values that the SNR can assume are very spread along the distance, meaning that the values are very similar even for a distance of 100 meters or more.
- When the offset is close to the transmission range, the vehicles that are in the middle between the real position of the malicious node and its spoofed position are still at the same distance, and they can't discern through the SNR if it is fake or not. This is visible from Figure 4.3 in which the vehicle V2 is at the same distance from M and M' and this means that the SNR will be probably recognized as valid.

Chapter 6

Simulation Assessment

6.1 Network Simulator 3

Network Simulator 3 (ns-3) is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use [30]. ns-3 is a free software, licensed under the GNU GPLv2 license, publicly available for research, development, and use. In this work ns-3 version 3.26 had been used to evaluate the effects of position cheating and black hole attacks on routing algorithms that heavily rely on accurate information about the vehicles' location to work properly.

6.1.1 Simulator structure

ns-3 is constructed on a modular architecture, with each module encapsulating a highly specialized business logic to minimize overall coupling between different modules. The main modules are *Core* and *Simulator*, which form the skeleton of the simulator and handle event creation and expiration, simulation time, and the callbacks API, which is used to connect the modules. There are numerous built-in modules available for usage with ns-3, ranging from routing techniques to mobility and propagation models, but new modules may also be constructed for any other lacking feature. The custom modules built for this thesis are the fast-broadcast and machine-learning modules.

When building a custom module, the best practice is to use the same structure as any other module which includes the following components:

- **bindings:** there is the possibility to create Python API bindings so that this module can work with Python;

- **doc:** the folder should contain all the documentation needed to understand the usage of the module;
- **examples:** here are contained some potential applications of the module;
- **helper:** these APIs should be the one used to implement the module in any simulation;
- **model:** the various classes that represents the business logic of the module;
- **test:** unit tests for the module to ensure its correctness.

6.1.2 Simulation setup

For any simulation, a file containing the main experiment and the simulation's scenario must be created. In this primary file, the network *nodes* must firstly be created, which are the fundamental pieces of any simulation. Following this it is necessary to specify which communication medium (wireless or cabled) will be used by these *nodes*, which protocols they will use (e.g. 802.11b), which propagation loss and propagation delay model the signal should follow, the position of the *nodes*, and whether or not these nodes should move or follow any particular mobility model, among other things. Following the design of the scenario, the duration of the simulation and the events that should occur can be determined and then the simulation can start.

6.2 Experiment setup

Overall, this thesis tested several different scenarios with 40 simulations each and averaged the results that have then been used to obtain the final results and charts. The environment in which the cars are placed is depicted as a strip-shaped road with an 8-kilometer length, in which every car follows the preceding one and the distance between all the vehicles is the same and equal to 16 meters.

6.2.1 Fast-broadcast setup

The FB's estimation phase for each experiment consists of four turns of *Hello* message exchange; 40% of the vehicles are randomly selected, and each vehicle transmits one *Hello* message in broadcast in both directions, 3 ms apart. In every simulation the malicious nodes are randomly chosen excluding the first and

the last nodes of the platoon. The node atop the platoon is always the first transmitter of the Alert message during each broadcast phase.

6.2.2 Parameters settings

The parameters utilized in the various simulations are listed in Table 6.1. The Nakagami propagation loss model [31] was employed in all scenarios since it is a stochastic fading model that can properly fit empirical data [32]. However, since Nakagami does not account for path loss because of the signal distance traveled, it needs to be combined with Two Ray Ground. The Two Ray Ground model alone was only utilized as a proof of concept for comparison purposes for the position cheating attack.

Table 6.1: Simulation settings

Parameter	Value
Road length	8Km
Number of nodes	500
Malicious nodes	3%
Transmission range	500m
Malicious offset	500m, 1000m, 1500m
MAC Protocol	802.11b
Bit Rate	11Mbps
Propagation loss model	Two Ray Ground, Nakagami
Propagation delay model	Constant Speed
CW size	[32, 1024]

The simulations employed the 802.11b protocol with a constant speed propagation delay model, and the transmission range was 500 m. The malicious nodes were chosen at a rate of 3%, and the positions claimed in their messages were changed using three different offset values: 500, 1000, and 1500 m.

Chapter 7

Threats

As mentioned in the introduction, the attacks that could be performed to a VANET are several [33]. The attackers that may attempt to attack a vehicular network can be mainly divided into (i) external and (ii) inside attackers. External attackers are nodes that cannot be validated by the authentication mechanism in use by the system. Moreover they can't read or participate in the communication between nodes in the network and they aim to attack the availability of the network for valid users. Even though external attackers are not part of the network, there are plenty of ways that they can exploit to jeopardize the communications between valid nodes. The most common one is Denial of Service Attack (DOS), which aims to disrupt the availability of the network by jamming the channel used for communication so that authenticated users will not be able to access the network services.

The inside attackers are recognized as valid nodes by the authentication mechanism and the attacks they can perform are wide. The attacks performed can either be aimed at disrupting the communication between vehicles or at stealing information. The most dangerous attacks in a safety application are those that aim to jeopardize the IVC since this could lead to severe injuries or even deaths. Some of these attacks are:

- **GPS spoofing:** this kind of attack let other nodes in the network think that the malicious node's position is different from the real one.
- **Black hole attack:** the attacker will not participate in the forwarding procedure and drops the packets.
- **Replay attack:** the fraudolent node can exploit an old message and its content to exploit the situation of the message at sending time.

- **Spamming:** the attackers can keep sending messages with the intent of creating delays in the communication.
- **Sybil attack:** different malicious nodes can send fake messages reporting problems on the road that would slow or even stop the other vehicles movements.

7.1 Position cheating attack

In this attack the aim is to add a significant delay to the forwarding procedure of the *Alert* message, by adding a significant amount of delay the vehicles will not be able to brake in time. The implementation of this attack in fast-broadcast affects both the estimation and the broadcast phases; the modified procedure to broadcast a *Hello* message is shown in Algorithm 6, when a malicious vehicle has to send a *Hello* message it will get its original position (line 2), and to this position it will add an offset value (line 3) and then broadcast the message (line 4). During the broadcast phase the malicious nodes will not participate in the forwarding procedure.

Algorithm 6 Position cheating attack example

```
1: function SendHello
2:     myPosition ← retrievePosition()
3:     myPosition.x ← myPosition.x + maliciousOffset;
4:     BroadcastMessage(myId, myPosition, myMaxRange);
5: end function
```

The fundamental result of this attack is that the nodes in the network estimate a higher maximum range, which is subsequently used in (3.1) to calculate the CW, which is likewise larger than the normal value. The consequence of this is demonstrated in Figure 7.1, where vehicle V3 is 300 meters away from vehicle S, but because an offset value of 500 meters is added, the perceived position of V3 by S is the one shown by the red car V3', leading the *MaxRange* to be evaluated at 800 meters resulting all the vehicles to consider the minimum CW in correspondence to the position of V3'. When S sends an *Alert* message during the broadcast procedure, all cars behind it calculate a CW based on 800 meters, resulting in the red dotted line rather than the blue continuous one. The consequences of this attack are twofold: firstly, there will be a lot of wasted time because vehicle V3' does not exist, it will not participate in the forwarding, and therefore the

farthest vehicle from the sender will be V5, which will have to wait a significant amount of time before transmitting since his *CW* has been enlarged as an effect of the attack; secondly, the difference between the *CW* is not as significant as in the normal case, increasing the likelihood that the actual forwarder will not be the farthest vehicle but another one closer to the sender increasing this way the number of hops required to deliver the *Alert* to the platoon's last node.

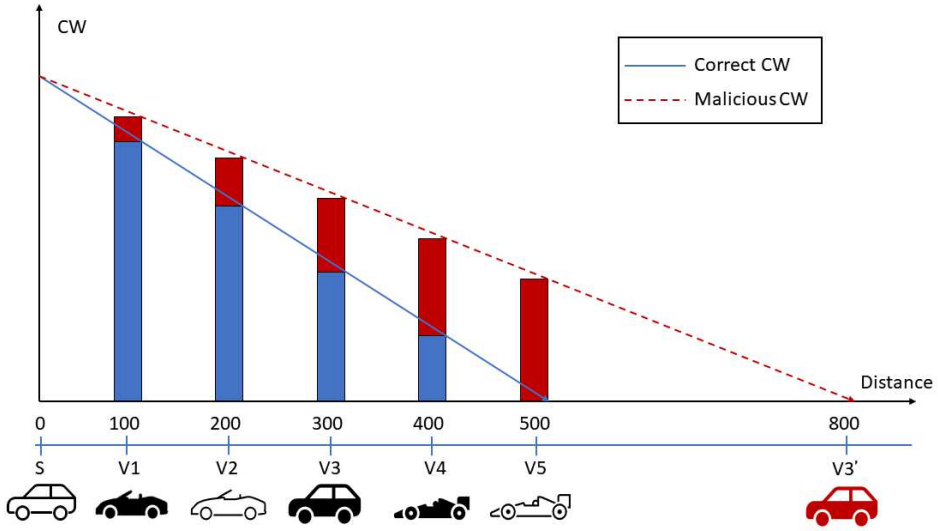


Figure 7.1: Impact of position cheating on the CW [27].

7.2 Black hole attack

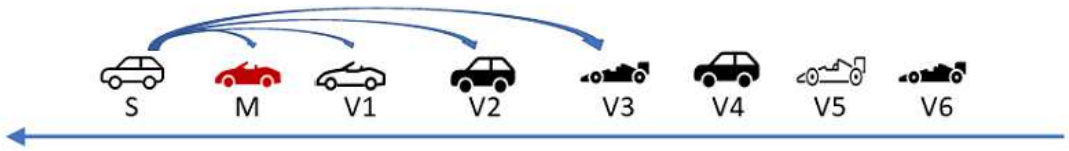
This attack tries to jeopardize the network by attempting to completely halt the forwarding mechanism; this attack only affects the broadcast phase and is depicted in Algorithm 7. As stated in Section 3.2, when a vehicle receives an Alert message to forward, it first calculates the distance between himself and the sender (line 2), then computes a *CW* value based on that distance, and then waits a random period of time depending on that value (lines 7 and 8). If a message is heard again from the front while waiting for the random amount of time to expire, the node must restart the forwarding procedure (lines 9 and 10); however, if the new *Alert* message is heard from behind, it means that the message has already been forwarded by another vehicle, and the actual node can stop the forwarding procedure (lines 11 and 12).

The attackers take advantage of the aforementioned mechanism, and as soon as they receive an Alert message, they forward it immediately and fake their position by adding the malicious offset to their actual position (lines 4 and 5), in this way, the receivers of this malicious *Alert* message, if the offset is large enough, believe the message is coming from behind and stop their forwarding procedure.

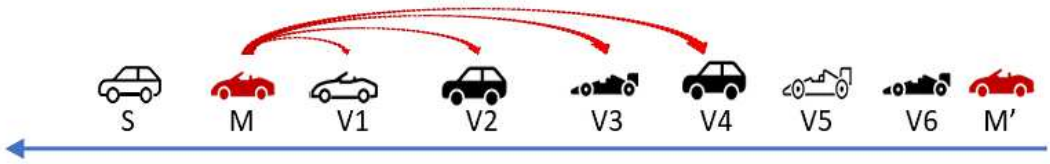
Algorithm 7 Black hole attack example

```
 1: function ReceiveAlert(alert)
 2:     distance ← calcDistance (alert.pos, myPos);
 3:     if Node is malicious then
 4:         CW ← 0;
 5:         myPos ← myPos + maliciousOffset;
 6:     else
 7:         CW ← getCW(distance, maxRange)
 8:         Wait CW;
 9:         if (sameBroadcastHeardFromFront) then
10:             restartBroadcastProcedure();
11:         else if sameBroadcastHeardFromBack() then
12:             exit();
13:         end if
14:     end if
15:     forwardAlert(myId, myPos, maxRange);
16: end function
```

Figure 7.2 is an example of a black hole attack. In Figure 7.2a, we observe a standard vehicle S delivering an *Alert* to the automobiles behind it until it reaches vehicle V3. Vehicles from V1 to V3 will calculate a CW and will begin waiting for the random amount of time to expire before forwarding it, whereas vehicle M immediately forwards the message after receiving it (Figure 7.2b), but changes its position by adding a malicious offset; all vehicles that hear this last message will believe that M's position is M', and because it is behind all of them, they will not try to forward the message, resulting in all vehicles after V4 never receiving the Alert message.



(a) Normal vehicle S sending the Alert message.



(b) Malicious vehicle M forwarding Alert message with faked position M'.

Figure 7.2: Black hole attack example.

Chapter 8

Metrics

The key metrics used to measure the impact of the attacks on the routing protocol, as well as the effects of the validation system, are examined in this section. As stated in 6.2.1, it is necessary to keep in mind that during every broadcast phase, the node that transmits the first *Alert* message is always the node atop the platoon, and that every metric is measured over the entire distance traveled from this node to the platoon's last node.

8.1 Number of hops (NOH)

As mentioned in Chapter 3, decreasing the number of hops not only is the major goal of fast-broadcast, but it is also the main objective to achieve in order to decrease message delivery delays. This metric represent the mean number of hops required in order to deliver the *Alert* message from the first to platoon's last node. Through analytical evaluation it is reported that on average for every hop the forwarder is located at 3/4 of the actual transmission range [34, 35].
The metric can be calculated as follows:

$$NOH = \sum_{n \in F} n \tag{8.1}$$

where F is the set of vehicles which have successfully forwarded the *Alert* message backward until reaching the last node of the platoon.

If the message is not able to be delivered until the last node, the simulation will not be counted in this metric since nor adding the value of hops reached nor adding a fixed high value of hops are viable options.

8.2 Number of slots (NOS)

This metric is used to measure the number of slots required to deliver the *Alert* message to the platoon's last node. The slots are used to better measure the time variable; they can be quite small (e.g., the IEEE 802.11g's slot is 9 μs long), allowing for a fine-grained representation of time.
The metric can be calculated as follows:

$$NOS = \sum_{slot \in Wait_F} slot \tag{8.2}$$

Where $Wait_F$ is the set of waited slots for all vehicles that successfully forwarded the *Alert* message until it reached the platoon's final node.

Similarly to NOH, if the message for a simulation does not reach the last node, the simulation will not be counted towards this metric.

8.3 Average range

This metric analyses how, during the estimation phase, the estimation of the transmission range changes with respect to the attacks or the validation system.

$$AverageRange = \frac{\sum_{range \in R} range}{N^{o}\ of\ vehicles} \tag{8.3}$$

where R is the set of all the range estimations for all the vehicles after the estimation phase ends.

8.4 Coverage

The coverage is the metric that measures the number of unique vehicles that received the *Alert* message. Differently from the previous metrics, this is always valid for all the simulation even if there are unsuccessful deliveries.

8.5 Unsuccessful deliveries

When the platoon's last node is unable to receive the *Alert* message, the message propagation abruptly stops along the forwarding channel, resulting in an unsuccessful delivery. This metric is also required to avoid losing important information when examining other metrics in the event that an *Alert* message is unable to be

correctly delivered to the platoon's last node; as stated in all previous metrics, if an unsuccessful delivery occurs, the simulation will not count towards the metric, resulting in values that do not entirely express the scenario situation. For example, comparing the NOS values of a scenario with a high number of failed deliveries to another with a low number of failed deliveries may yield similar results; however, in a scenario with a high number of failed deliveries, an expert reader should be able to project these values over the NOS value and understand that the NOS value should be considered higher.

Analyzing the failed deliveries in conjunction with the coverage assists in determining where the message stopped and the severity of the problem, since having a high value of coverage suggests that the forwarding procedure stopped closer to the target vehicle and so the problem is less serious.

8.6 Precision

Precision is defined as the ratio of the results that properly predicted positive observations (True Positives) to the total predicted positive observations, both accurate (True Positives) and incorrect (False Positives), and is expressed by (8.4).

$$Precision = \frac{TP}{TP + FP} \tag{8.4}$$

8.7 Recall

The Recall is the proportion of the results obtained by the classificator that properly predicted misbehaving nodes (True Positives) to all observations in the actual misbehaving class and is expressed by (8.5).

$$Recall = \frac{TP}{TP + FN} \tag{8.5}$$

8.8 Accuracy

Accuracy is the proportion of successfully predicted categories (including True Positives and True Negatives) to the total results obtained by the classification and is shown as (8.6).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8.6}$$

8.9 F1 Score

The F1 score or F-measure represents the harmonic mean of the precision and recall and is expressed by (8.7).

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{8.7}$$

Chapter 9

Results

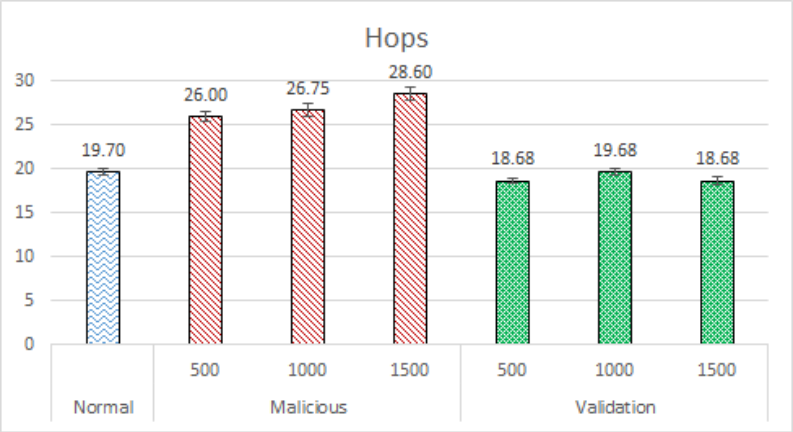
Following the analysis of the protocol and the attacks in the exam in Chapters 3 and 7, this chapter will report the results of the simulations performed on the various simulations done using the settings explained in Chapter 6 with FBLV analyzed in Chapter 5.

Position cheating and black hole attacks are firstly tested separately in order to better understand the impact of the singular attack as well as the achievements and limitations of the k-NN classificator. Following that, they are tested together to see how they affect the VANET system and whether the k-NN classifier can correctly identify misbehaving nodes and restore system performance to normal levels.

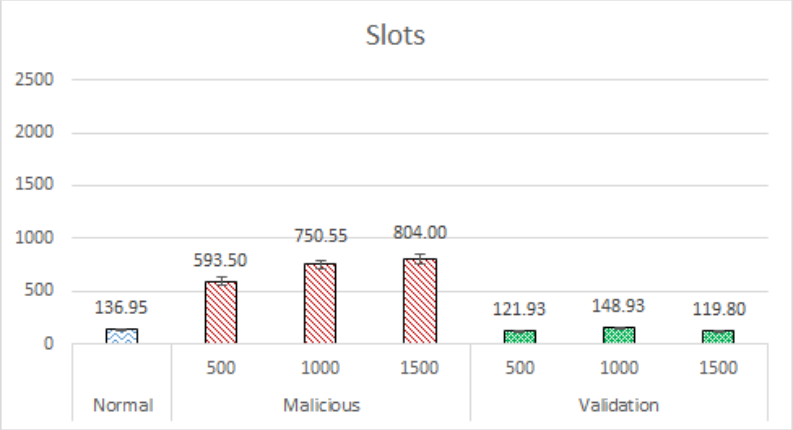
For each scenario, the simulations are first run without the presence of malicious nodes or validation system, then the impact of malicious nodes in the network is examined, and finally the validation system's performance in identifying misbehaving nodes and its impact on the routing protocol is tested.

9.1 Proof of concept

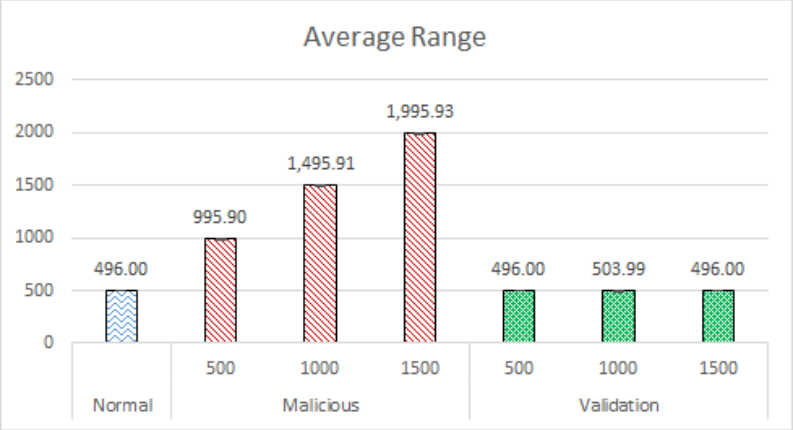
This section presents the first results achieved by testing the position cheating attack on a scenario that utilizes the Two Ray Ground propagation loss model. The key effect of solely employing the Two Ray Ground model is that the hearing communication range is symmetrical, which implies that if vehicle A hears vehicle B, vehicle B will hear vehicle A. This is a huge assumption that would not be met in a real-world application, but for the purpose of simplicity, it will help to study the effects of the position cheating attack and the results achieved by the validation system.



(a) Average number of hops required to deliver the message to the last node of the platoon.



(b) Average number of slots required to deliver the message to the last node of the platoon.



(c) Average range for the nodes at the end of estimation phase.

Figure 9.1: FB simulation results using Two Ray Ground and position cheating attack with offsets 500, 1000 and 1500 m.

The results of this attack are presented in Figure 9.1, where the blue bar represents a typical environment with no misbehaving nodes or validation system, the red ones indicate a system with 3% malicious nodes, and the green colored bars represent a scenario with both the validation system and misbehaving nodes. By examining the Hops graph in Figure 9.1a, it is possible to observe that the addition of malicious nodes boosted the average value of the hops from a minimum of 6 hops to a maximum of nearly 8, representing a 32% and 45% increase in value, respectively. Figure 9.1b shows that the slots necessary to transmit the *Alert* message to the platoon's final node increased by at least 330 percent with a 500 meter distance and up to 486 percent with a 1500 meter offset. Looking at Figure 9.1c it is important to notice how the average range estimated during the estimation phase varies, and because the scenario is using Two Ray Ground, the increment is always equal to the offset value, resulting in an estimated range four times larger than the conventional one at 1500 meters of offset.

The worsening of the metrics with the introduction of misbehaving nodes is consistent with the analysis in Chapter 7. By estimating larger transmission ranges, the vehicles compute larger CW values and on this value they choose similar waiting times, decreasing the probability that the first forwarder is the farthest node from the sender; the CW expansion directly impacts the number of slots because there is more time to wait for every hop, and because the CWs are very similar, the probability of choosing a forwarder that is not the last will increase, resulting in more hops required to reach the last node.

Scenario	Precision	Recall	Accuracy	F1 Score
Malicious 500	1	0.9359	0.9980	0.9669
Malicious 1000	1	0.9844	0.9995	0.9921
Malicious 1500	1	1	1	1
Macro average	1	0.9734	0.9991	0.9863

Table 9.1: Macro average metrics for k-NN classificator on position cheating attack with Two Ray Ground.

The introduction of the validation system results in a significant improvement in the performance of the routing algorithm, with values that are quite near to those of the normal scenario. Looking at Figure 9.1c, it is possible to see that the estimated range was returned to its correct value, causing all of the other values to return to normality as well. The performance of the k-NN classificator in this case is shown in Table 9.1, and thanks to it, it is possible to confirm that the classificator performs admirably in this environment that uses Two Ray Ground

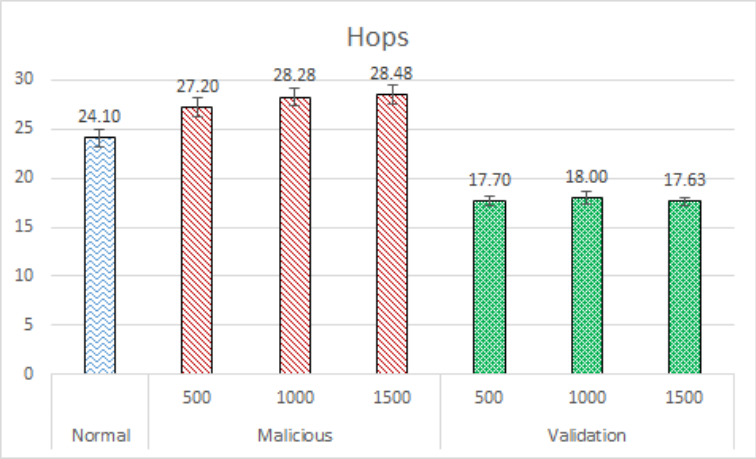
as propagation loss model, due to its deterministic nature and lack of significant SNR oscillations.

The results obtained with this experiment serve as the foundation for this thesis, indicating that the implementation of the proposed validation system may be able to act as a first defense against the position cheating attack, however the system must be tested and the results validated using a more realistic propagation loss model, as its impact on routing protocol performance is significant [36].

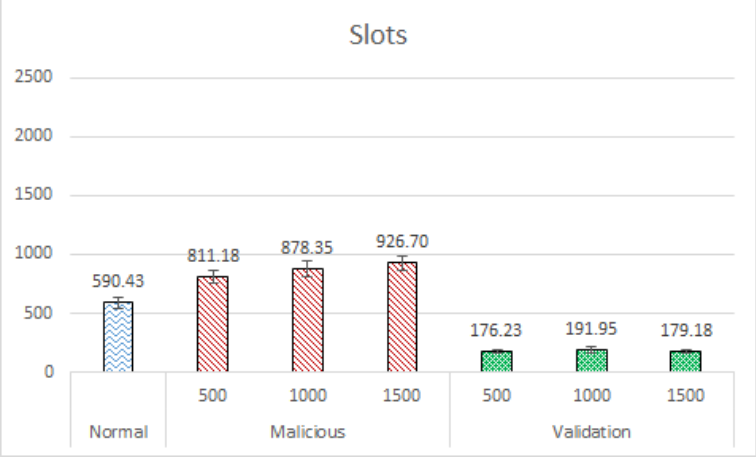
9.2 Position cheating attack

This section, like the previous one, will evaluate the impact of position cheating attacks and the validation system's performance. However, unlike the previous one, this scenario will use the Nakagami propagation loss model in conjunction with Two Ray Ground. The main difference between the scenarios is that the communication range in the previous one was symmetrical, whereas in this one it is not, and the SNR and transmission range is different for each transmission, and highly variable.

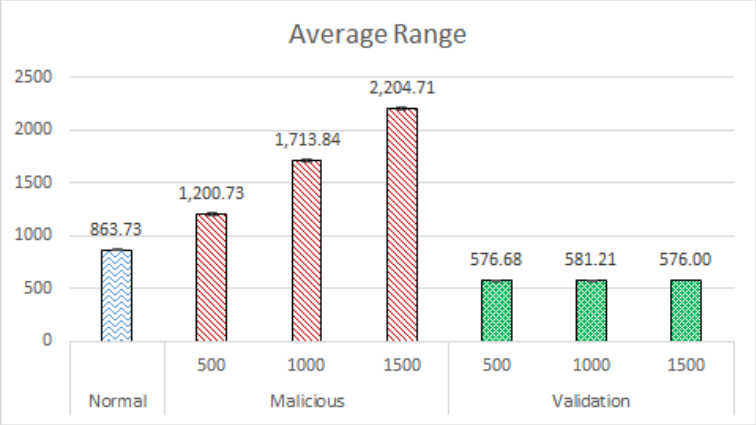
The results of this experiment are shown in Figure 9.2. Starting with the normal scenario, it is possible to notice a worsening of all the metrics used to gauge the performance of the routing protocol; the number of hops shown in Figure 9.2a shows a ≈22% increase over the previous results, while the number of slots shown in Figure 9.2b and the average range shown in Figure 9.2c both show significant value increases. The cause of this result can be found in the propagation loss model used; with Nakagami, there is a large variation in the transmission signal between all of the messages, resulting in highly variable transmission ranges; this means that the transmission range can sometimes be very different from the nominal 500 meters, even exceeding the 800 meters as shown in Figure 9.2c. As a consequence, even if only one vehicle estimates 800 or more meters once, this number will propagate to all other nodes in the network throughout the estimating phase, as discussed in Chapter 3.4.



(a) Average number of hops required to deliver the message to the last node of the platoon.



(b) Average number of slots required to deliver the message to the last node of the platoon.



(c) Average range for the nodes at the end of estimation phase.

Figure 9.2: FB simulation results using Nakagami and position cheating attack with different offsets.

After including the misbehaving nodes in the network, the estimated range during estimation phase increases significantly accordingly to the offset used and, following what has been observed in the previous experiment, hops and slots values increased, degrading the performance of the routing protocol, which became less trustworthy. It is possible to notice that these values for offset 1000 and 1500 meters are very similar because, according to (3.1), the CW values grow rapidly at first for MaxRange values greater than the distance, but after a bit they start growing slower, as seen in Figure 9.3 where it is possible to see that the CW grows by 500 slots for the first 1000 meters, but only by 300 slots for the next 1250 meters.

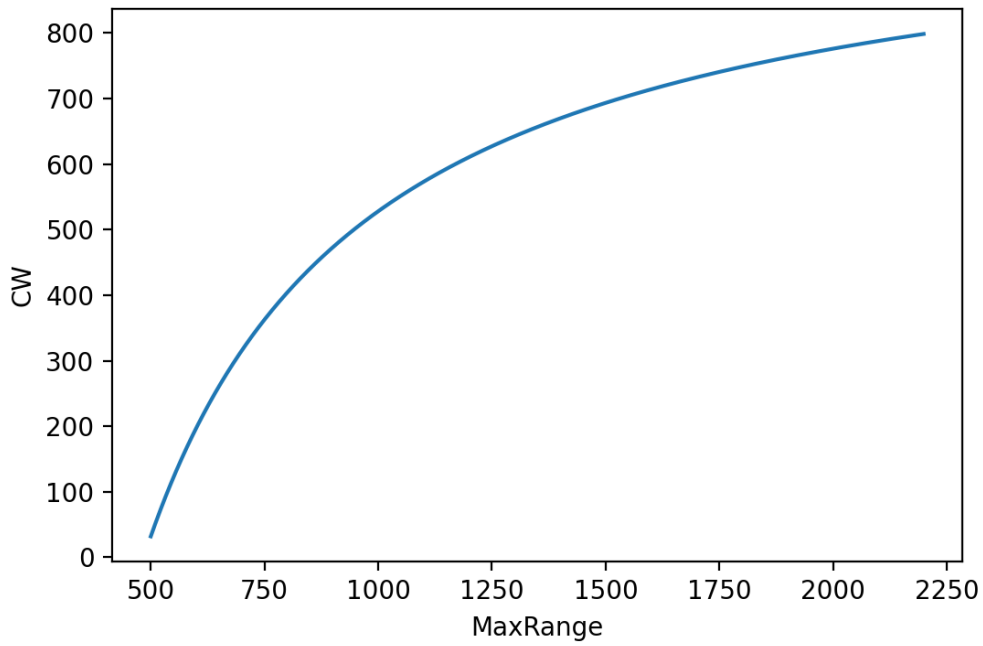


Figure 9.3: Example of CW expansion with different values of MaxRange and distance equal to 500 meters.

The inclusion of the validation system is capable of not only mitigating the negative impacts of the attackers, but also of improving the routing protocol's performance by reducing the number of hops and slots required to reach the destination node compared to the normal case, as illustrated in Figs 9.2a and 9.2b, respectively. When defending against a position cheating attack, by detecting and ignoring messages coming from malicious nodes the validation system's main effect is to control the estimated range during the estimation phase so that it is not influenced by the attack.

For this set of experiments, thanks to the validation system, the transmission range was able to settle down with values very similar to the nominal one (500 meters), not only resulting in a successful defense of the VANET, but also acting as a regulator of the estimated range by ignoring outlier messages that could travel for more than 500 metres due to Nakagami, which results in an overestimation of the transmission range as seen in Figure 9.2c. This gain in performance is accompanied with a deterioration in the classificator metrics, as the exclusion of these outliers results in a high number of false positives, and results in a low precision score and F1 score, as seen in Table 9.2. The high value in recall and accuracy highlights that the validation system is able to correctly identify most of the malicious messages except for those in the limit zone described in Chapter 5.1.

Scenario	Precision	Recall	Accuracy	F1 Score
Malicious 500	0.2635	0.6440	0.9371	0.3740
Malicious 1000	0.3453	0.9253	0.9447	0.5030
Malicious 1500	0.3741	1	0.9476	0.5445
Macro average	0.3276	0.8564	0.9431	0.4738

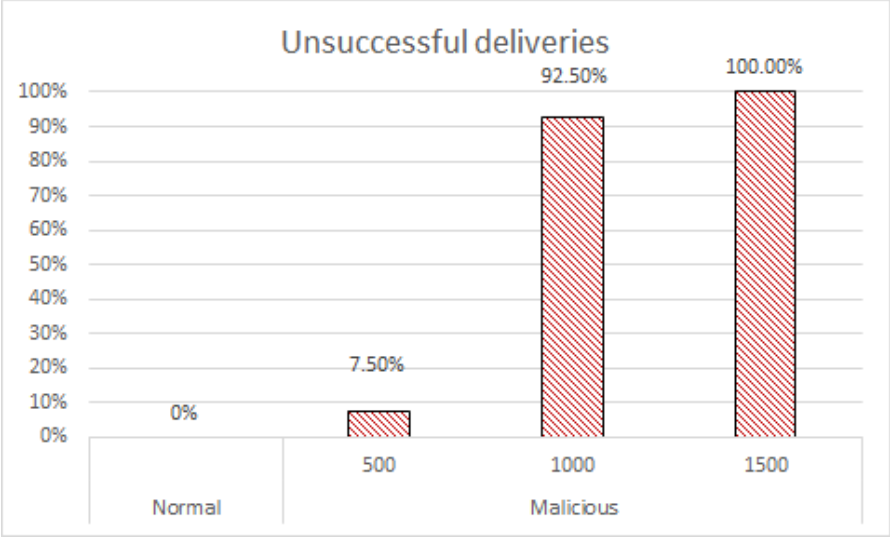
Table 9.2: Macro average metrics for k-NN classificator on position cheating attack with Nakagami.

9.3 Black hole/Sink attack

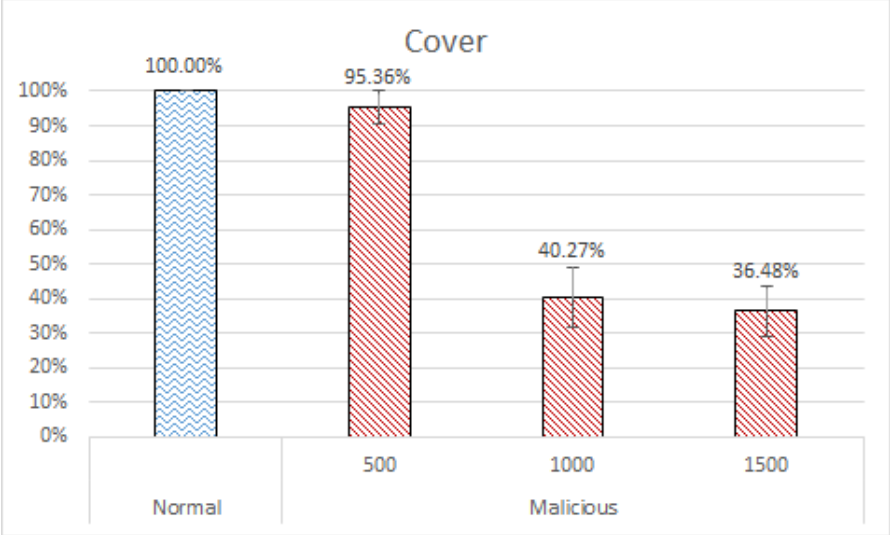
This section is going to summarize the results obtained from the Black hole attack. Instead of influencing the estimation phase, this attack influences only the broadcast phase as explained in Chapter 7.2, and tries to completely halt the propagation of the *Alert* messages. Figure 9.4 shows the effects of the malicious nodes in the network. Figure 9.4a shows the percentage of unsuccessful deliveries over the 40 simulations, which represents all the times that the Alert message was not able to be received by the platoon's last node. Figure 9.4b displays the average coverage obtained during the simulations.

It is possible to see in Figure 9.4a that with an offset value of 500 meters, the number of unsuccessful deliveries reaches only 7.5 percent, but as soon as the offset is set to 1000 meters this value reaches 92.5 percent. The difference between these values can be explained using Figure 9.5. Normally, when a malicious node, M, sends an *Alert* message, his faked position M' should be at least as large as the transmission range; this way, all nodes receiving the message will be fooled into thinking M is behind them, and they will stop forwarding the message, as shown

by the yellow arrow. However, because of Nakagami, the message can sometimes reach beyond the nominal transmission range, as seen by the red arrow, but all vehicles (V4 and V5) following the faked position will still be able to forward the *Alert* because it comes from the front, just like all other correct Alerts. This explains why the value of unsuccessful deliveries is so different between 500 meters and the other two offsets, since the message will never be able to travel further than these last two values.



(a) Percentage of unsuccessful deliveries during broadcast phase.



(b) Percentage of coverage during broadcast phase.

Figure 9.4: Black hole attack, results with different malicious offsets.

Thanks to Figure 9.4b it is possible to measure the severity of the black hole attack. We can notice that with 500 meters more than 95% of the vehicles are

still able to receive the Alert message, at the same time with offsets 1000 and 1500 meters this value decrease significantly reaching ≈40% and ≈36%, respectively.

Considering all of the results achieved by this assault, it is reasonable to conclude that the attacker will have to lie about his position by a significant amount in order to significantly harm the network's communication.

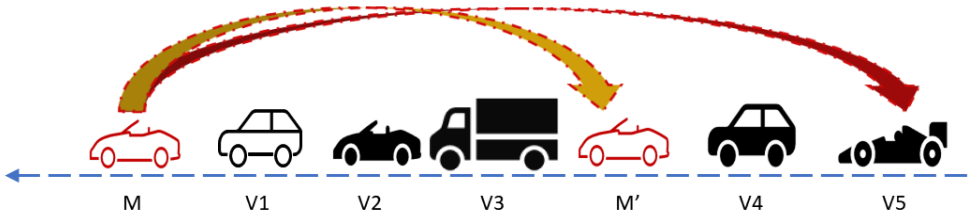


Figure 9.5: Example of failed black hole attack due to Nakagami propagation loss model.

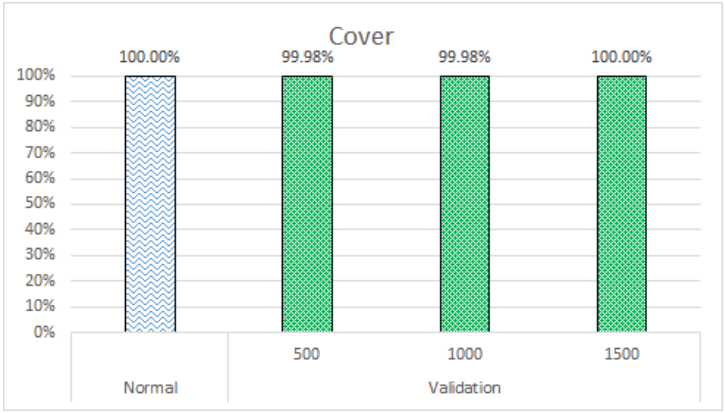
The introduction of the validation mechanism only in the broadcast phase is capable of providing an incredible defence against this type of attack, but it does so at the expense of some minor side effects, as illustrated in Figure 9.6. First of all, there have been no case of unsuccessful delivery over all the simulation, which is for sure the best possible outcome for the validation system, and the coverage reached all values very close to 100% as seen in Figure 9.6a.

Even though the validation system was able to negate the negative effects of the black hole attack, there were some minor side effects that can be seen in Figure 9.6b and Figure 9.6c; it is possible to see from Figure 9.6b that with the introduction of the validation system, the number of hops increased from 24 to 27 when compared to the normal scenario, and as shown in Figure 9.6c, the slots required to deliver the *Alert* message to the platoon's last node increased significantly and reached a peak of 1500 with the 500 meters offset.

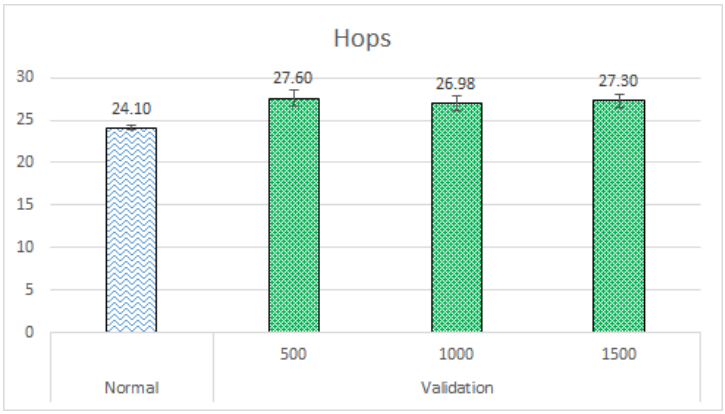
The main reason for these results is that, while it was beneficial to identify a few outliers as malicious nodes during the estimation phase in order to get a good estimation of the transmission range as shown in the previous section, filtering out these outliers during the broadcast phase will be detrimental to the routing protocol's performance in three ways:

- these outliers are messages that travel a great distance, which implies that only messages that travel shorter distances are accepted, increasing the number of hops required to reach the destination node;
- if an outlier is ignored, it is likely that it was one of the first forwarders;

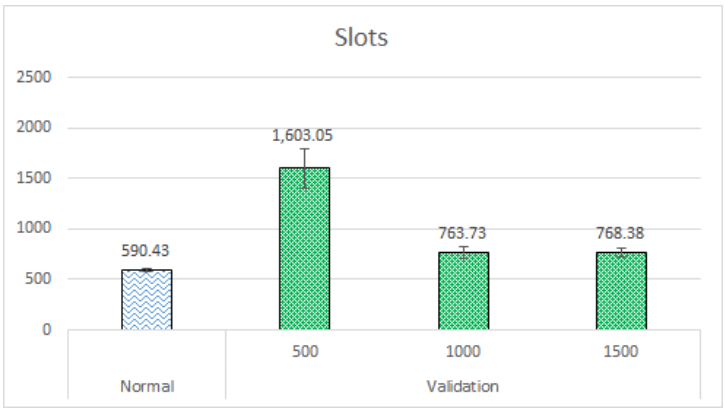
otherwise, it would have received another *Alert* from a node behind him and stopped the forwarding mechanism; by ignoring this quicker forwarder, the system increases the slots and therefore adds delay to the transmission;



(a) The broadcast phase's average percentage of coverage.



(b) Average number of hops required to deliver the message to the last node of the platoon.



(c) Average number of slots required to deliver the message to the last node of the platoon.

Figure 9.6: FB simulation results of message validation against black hole attack with Nakagami model.

- because there is no validation mechanism in the estimation phase, the estimated range will be larger than 500 meters, but the messages that will be able to met the estimated range will be considered misbehaving, therefore resulting in useless larger CW generating delay to the transmission.

Scenario	Precision	Recall	Accuracy	F1 Score
Malicious 500	0.7146	0.6404	0.8925	0.6755
Malicious 1000	0.7574	0.9181	0.9406	0.8301
Malicious 1500	0.7796	1	0.9542	0.8761
Macro average	0.7506	0.8528	0.9291	0.7939

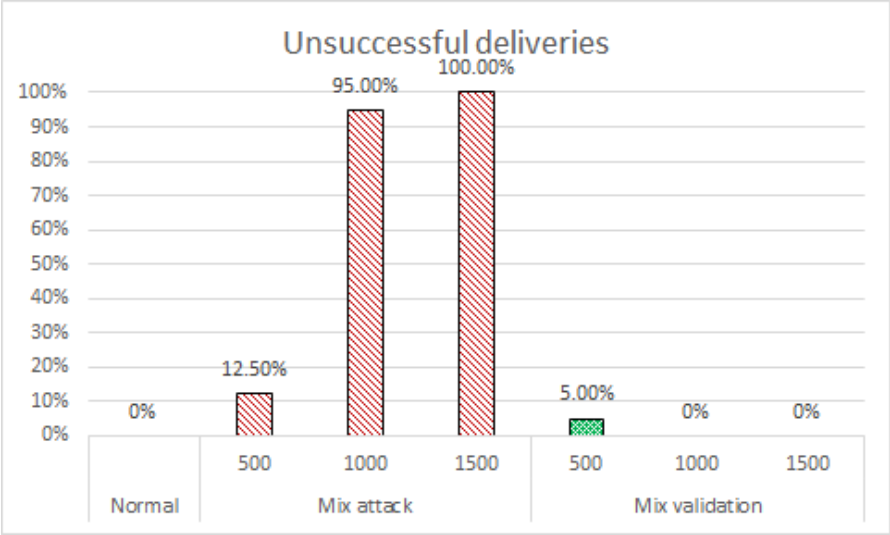
Table 9.3: Macro average metrics for k-NN classificator on black hole attack.

Table 9.3 shows that the precision values indicate a significant amount of false positives, which are attributable to the outliers detection mentioned above; it is also noticeable that for the scenario with offset 500 meters, there is a low value of recall, indicating a high value of false negatives, which is consistent with the analysis in Chapter 5.1. These false negatives compromised the routing protocol's performance by blocking the best forwarders estimated during estimation phase, resulting in a high number of slots required to reach the destination vehicle, as seen in Figure 9.6c. The high accuracy values prove that the validation system is capable of stopping the attack successfully.

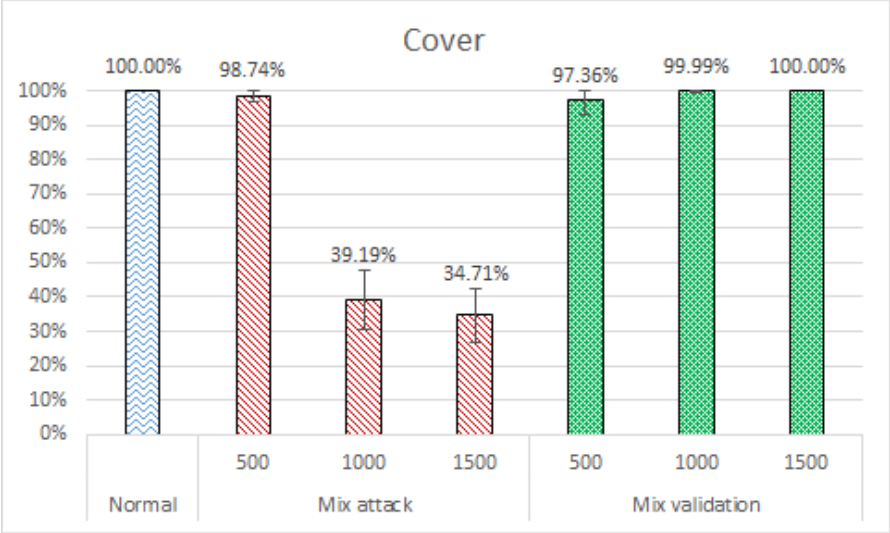
9.4 Mix attack

In this final experiment, the scenario implements both attacks at the same time, meaning that the 3% of malicious nodes chosen at random every time will perform both attacks, and the validation system will be implemented in both estimation and broadcast phases. Figures 9.7 and 9.8 report the results of these simulations; it is important to highlight that only Figure 9.7 accounts all the simulations, meanwhile the data shown in Figure 9.8 do not account for the simulations that were not able to deliver the message until the last node.

By analyzing the impact of both attacks on the system, it is possible to see in Figure 9.7a that the number of unsuccessful deliveries is slightly higher than those shown in Figure 9.4a; this is due to the CW expansion caused by the position cheating attack indeed, because normal vehicles will have to wait a longer time before forwarding the message, they will be less likely to try to forward the *Alert* at the same time as a malicious node, improving the chances of the attackers



(a) Percentage of deliveries that were not able to reach the last node in the platoon.

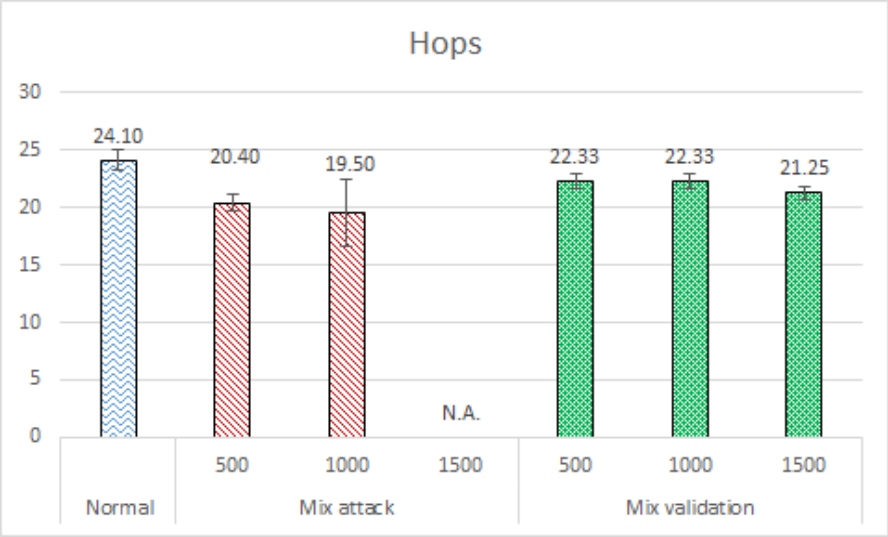


(b) The broadcast phase's average percentage of coverage.

Figure 9.7: Results related to mixed attack with 3% of nodes performing both attacks and the validation system active in all phases.

to successfully perform the black hole attack. Figure 9.4b shows the coverage obtained during the broadcast phase. The results are in line with those seen in the scenario with only the black hole attack active.

By looking at the effects of the mixed attack in Figure 9.8a it may seems that the attacks improved the performance of the routing algorithm since the number of hops decreased compared to the normal case, but as previously stated, these values represents only those Alerts that were able to be delivered to the platoon's last node. By looking at Figure 9.8b it is possible to notice that even tough there



(a) Average number of hops required to deliver the message to the last node of the platoon.

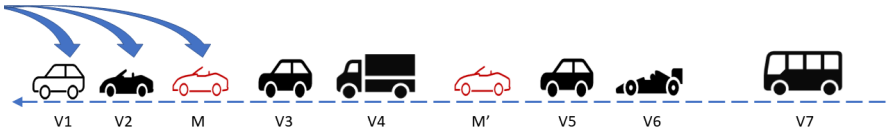
(b) Average number of slots required to deliver the message to the platoon's last node.

Figure 9.8: Results related to mixed attack with 3% of nodes performing both attacks and the validation system active in all phases.

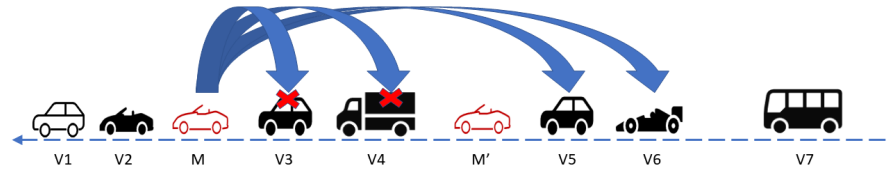
is a low value in hops, there is a massive increase in the slots required to complete the forwarding procedure jeopardizing the whole VANET.

The previously described results can be explained using Figure 9.9. In a scenario with both attacks active, when a new *Alert* message is received from different vehicles, as shown in Figure 9.9a, these vehicles will compute a CW and based on this value, choose a waiting time that needs to expire before forwarding the *Alert* message; if there is a malicious vehicle between these receivers, depicted as M in the figure, this vehicle will forward the *Alert* as soon as he receives it,

as shown in Figure 9.9b. In the same figure, V3, V4, V5, and V6 would have comparable *CW* values due to the position cheating attack impact, providing vehicles V3 and V4 a higher probability than normal to execute the forwarding and thus increasing the number of hops required to complete the delivery of the Alert message. However, because to the black hole attack, vehicles V3 and V4 are unable to participate in the contention period because they misinterpret the message from M as a message from behind, causing their forwarding mechanism to be halted. As a result, anytime the black hole attack fails to completely interrupt the forwarding mechanism, it actually reduces the number of useless forwarders by stopping the closest ones and allowing the farther vehicles to compete for forwarding. The final result, as shown in Figure 9.8 is that with the mixed attack there will be a lower values of hops than normal, but with a significant increase in slots.



(a) New Alert message incoming from the front received from different vehicles.



(b) Vehicle M, implementing black hole attack, forward as soon as he receives the message.

Figure 9.9: Example of forwarding hop in the mixed scenario. Vehicle M spoofing his position at M'.

The introduction of the validation mechanism is able to mostly stop the side effects of the attacks and to even improve the performance of the routing protocol when compared to the scenario without active attackers. It is possible to see from Figs. 9.7a and 9.7b that the unsuccessful deliveries have been reset and the coverage has nearly achieved total values, respectively, with the exception of the 500 meters offset. In this last scenario, there were a few failed deliveries (5%), as well as coverage that was somewhat less than complete (87%).

Similarly to the improvements mentioned above, hops and slots achieved good values showing that the validation system is able to defend well against the attacks examined in this thesis. In Figure 9.8a the number of hops for the validation

Scenario	Precision	Recall	Accuracy	F1 Score
Malicious 500	0.3252	0.6430	0.9335	0.4320
Malicious 1000	0.4080	0.9237	0.9448	0.5660
Malicious 1500	0.4240	1	0.9478	0.5955
Macro average	0.3857	0.8556	0.9420	0.5312

Table 9.4: Macro average metrics for k-NN classificator on mixed attack.

system are very similar and they are all lower than those of the normal environment, indicating an improvement in performances. Particularly important are the results obtained for the number of slots shown in Figure 9.8b that shows no significant delay introduced in the forwarding procedure, but instead a significant reduction of slots to reach the platoon's last node with offset 1000 and 1500.

Similarly to what examined in Chapter 9.2, the performance achieved in some cases shows better results than those of the scenario without misbehaving nodes. By looking at Table 9.4, the first thing that we notice is a very low value of precision in all scenarios, but this is in line with the expectations, given that there are many false positives being identified in the estimation phase and even some in the broadcast phase. Nevertheless, as already mentioned, the many outliers identified during estimation phase benefit the routing protocol with a better estimation of the transmission range and therefore CW values. The recall and accuracy values are also in line with the expectations and highlight a very good performance of the k-NN classificator, except for the 500 meters offset scenario.

Chapter 10

Conclusion

It is possible to notice that in Figure 9.1a and 9.2a the hops values for the normal scenarios identify the average length of a hop very close to the value estimated by the mathematical model of 3/4 of the transmission range [34]. This means that the representation of the scenario used in this thesis are in line with the analytical studies.

Thanks to the results reported in Chapter 9, it is possible to affirm that any IVC system that heavily depends on a position based routing protocol would suffer serious consequences in the case of a position cheating and/or black hole attacks, to a point that it could not be considered reliable anymore since it could cause serious injuries or even deaths. As seen in the previous chapters, the position cheating attack makes all the vehicles in the network overestimate their transmission range, and therefore will cause the useless computation of larger CWs, adding this way a lot of delay to the delivery process. At the same time the black hole attack is able to completely stop the broadcast phase from forwarding the alert message to the platoon's last node and to decrease the number of vehicles that receives these *Alert* messages.

The effects mentioned above are very serious and must be addressed to ensure everyone's safety. However, the tests conducted with FBLV are reassuring and show that it is possible to stop the position cheating and black hole attacks from jeopardizing the network because, by exploiting the close relationship between the SNR of a received message and the distance between the sender and the receivers, it is possible to distinguish well-behaving nodes from malicious ones. These results are supported by the overall high accuracy values, even though they need to be improved because both precision and F1 score are sometimes low due to a high number of false positives, even though do not have a significant impact

on the defense quality of the defense mechanism implemented and sometimes helps with a higher estimation phase quality.

10.1 Future work and improvements

Thanks to the comparison of the results obtained with Two Ray Ground and those with Nakagami, it is clear that the propagation loss model used in the simulations highly affects the accuracy of the validation system. To better estimate the reliability of FBLV, our solution should be tested also in realistic urban or rural scenarios implementing an obstacle model [37, 38].

Many strategies can be used to increase the performance and reliability of the suggested validation process, one of which is the incorporation of an on-board visual position detector (e.g., radar, cameras, infrared sensors). Although the detection range may be limited due to physical constraints that limit the identification of malicious vehicles to only those in the immediate vicinity, a vehicle can improve its local security in this manner. However, through a collaboration mechanism, the information collected by a vehicle can also be shared with the network and used by other vehicles to determine malicious vehicles [39, 40].

Another technique to improve the detection mechanism's performance is to find other features to employ beside distance and SNR in order to exploit more parameters to distinguish between well-behaving and malicious nodes. One example could be the computing of a trust metric between vehicles, but this could also be done with the assistance of RSUs or other infrastructure [41, 42]. Other parameters to examine include the vehicle's speed and other sensors information [28]. With a growing number of parameters, it would also be possible to test approaches other than ML, such as deep learning ones [43].

Finally, given the numerous successful examples of position cheating mechanisms obtained through the collaboration of neighboring nodes [26, 27, 39, 40], the actual FBLV can be further improved through the implementation of this mechanism. One of the benefits of a collaboration system is that it should be able to overcome the limitation discussed in Chapter 5.1, since a false positive identified by one node may be correctly classified by the surrounding ones.

Bibliography

[1] NISSAN Motor Corporation. *Businesses need smarter tech in their fleets to survive e-commerce boom*. 2019. URL: https://global.nissannews.com/ja-JP/releases/release-41181d20da4d17b77ed88d700817c540-businesses-need-smarter-tech-in-their-fleets-to-survive-e-commerce-boom (visited on 01/23/2022) (cit. on p. 1).

[2] Nishant Sharma, Naveen Chauhan, and Narottam Chand. "Security challenges in Internet of Vehicles (IoV) environment". In: *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. 2018, pp. 203–207. DOI: 10.1109/ICSCCC.2018.8703272 (cit. on p. 1).

[3] Sooksan Panichpapiboon and Wasan Pattara-atikom. "A Review of Information Dissemination Protocols for Vehicular Ad Hoc Networks". In: *IEEE Communications Surveys Tutorials* 14.3 (2012), pp. 784–798. DOI: 10.1109/SURV.2011.070711.00131 (cit. on p. 2).

[4] Ruqayah Al-ani et al. "A Survey on Secure Safety Applications in VANET". In: *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 2018, pp. 1485–1490. DOI: 10.1109/HPCC/SmartCity/DSS.2018.00245 (cit. on p. 3).

[5] M. Aljunaid et al. "Classification of Security Attacks in VANET: A Review of Requirements and Perspectives". In: vol. 150. Jan. 2018, p. 06038. DOI: 10.1051/matecconf/201815006038 (cit. on p. 3).

[6] Wafa Ben Jaballah et al. "Impact of security threats in vehicular alert messaging systems". In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. 2015, pp. 2627–2632. DOI: 10.1109/ICCW.2015.7247575 (cit. on pp. 3, 11).

[7] Xiaolin Jiang et al. “Low-Latency Networking: Where Latency Lurks and How to Tame It”. In: *Proceedings of the IEEE* PP (Aug. 2018), pp. 1–27. DOI: 10.1109/JPROC.2018.2863960 (cit. on pp. 4, 7, 9, 13).

[8] Elena Fasolo, Roberto Furiato, and Andrea Zanella. “Smart Broadcast algorithm for inter-vehicular communications”. In: (Jan. 2005) (cit. on pp. 4, 7, 9, 13).

[9] Claudio E. Palazzi et al. “How Do You Quickly Choreograph Inter-Vehicular Communications? A Fast Vehicle-to-Vehicle Multi-Hop Broadcast Algorithm, Explained”. In: *2007 4th IEEE Consumer Communications and Networking Conference*. 2007, pp. 960–964. DOI: 10.1109/CCNC.2007.194 (cit. on p. 4).

[10] Surmukh Singh and Sunil Agrawal. “VANET routing protocols: Issues and challenges”. In: *2014 Recent Advances in Engineering and Computational Sciences (RAECS)*. 2014, pp. 1–5. DOI: 10.1109/RAECS.2014.6799625 (cit. on p. 7).

[11] Arohi Gupta et al. “Comparison of various routing algorithms for VANETS”. In: *2016 International Conference System Modeling Advancement in Research Trends (SMART)*. 2016, pp. 153–157. DOI: 10.1109/SYSMART.2016.7894509 (cit. on p. 7).

[12] Juan Angel Ferreiro-Lage et al. “Analysis of Unicast Routing Protocols for VANETs”. In: *2009 Fifth International Conference on Networking and Services*. 2009, pp. 518–521. DOI: 10.1109/ICNS.2009.96 (cit. on p. 8).

[13] Zygmunt Haas, Marc R. Pearlman, and Prince Samar. *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*. Internet-Draft draft-ietf-manet-zone-zrp-04. Work in Progress. Internet Engineering Task Force, Aug. 2002. 11 pp. URL: https://datatracker.ietf.org/doc/html/draft-ietf-manet-zone-zrp-04 (cit. on p. 9).

[14] Brad Karp and H. T. Kung. “GPSR: Greedy Perimeter Stateless Routing for Wireless Networks”. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. MobiCom ’00. Boston, Massachusetts, USA: Association for Computing Machinery, 2000, pp. 243–254. ISBN: 1581131976. DOI: 10.1145/345910.345953. URL: https://doi.org/10.1145/345910.345953 (cit. on p. 9).

[15] Stefano Basagni et al. “A Distance Routing Effect Algorithm for Mobility (DREAM)”. In: *Proc. of ACM/IEEE MobiCom*. Dallas, Texas, USA: ACM, 1998, pp. 76–84 (cit. on p. 9).

[16] Tony G and Dhandapani S. “Optimal Routing in VANET using Improved Meta-heuristic Approach: A Variant of JAYA”. In: *IET Communications* 14 (June 2020). DOI: 10.1049/iet-com.2018.6214 (cit. on p. 10).

[17] J. Fukuyama. “A Probabilistic Protocol for Multi-Hop Routing in VANETs”. In: *2009 IEEE International Conference on Communications Workshops*. 2009, pp. 1–6. DOI: 10.1109/ICCW.2009.5208064 (cit. on p. 10).

[18] Hongseok Yoo and Dongkyun Kim. “ROFF: RObust and Fast Forwarding in Vehicular Ad-Hoc Networks”. In: *IEEE Transactions on Mobile Computing* 14.7 (2015), pp. 1490–1502. DOI: 10.1109/TMC.2014.2359664 (cit. on p. 10).

[19] Amrita Ghosal and Mauro Conti. “Security issues and challenges in V2X: A Survey”. In: *Computer Networks* 169 (2020), p. 107093 (cit. on p. 10).

[20] Monowar Hasan et al. *Securing Vehicle-to-Everything (V2X) Communication Platforms*. 2020. arXiv: 2003.07191 [cs.NI] (cit. on p. 10).

[21] Abdullahi Chowdhury et al. “Attacks on Self-Driving Cars and Their Countermeasures: A Survey”. In: *IEEE Access* 8 (2020), pp. 207308–207342. DOI: 10.1109/ACCESS.2020.3037705 (cit. on p. 10).

[22] Richard Gilles Engoulou et al. “VANET security surveys”. In: *Computer Communications* 44 (2014), pp. 1–13. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2014.02.020. URL: https://www.sciencedirect.com/science/article/pii/S0140366414000863 (cit. on p. 10).

[23] Sebastian Bittl et al. “Emerging attacks on VANET security based on GPS Time Spoofing”. In: *Proc. of IEEE CNS*. 2015, pp. 344–352 (cit. on p. 11).

[24] Wafa Ben Jaballah, Mauro Conti, and Chhagan Lal. “A Survey on Software-Defined VANETs: Benefits, Challenges, and Future Directions”. In: (2019). arXiv: 1904.04577 [cs.NI] (cit. on p. 11).

[25] Othman S. Al-Heety et al. “A Comprehensive Survey: Benefits, Services, Recent Works, Challenges, Security, and Use Cases for SDN-VANET”. In: *IEEE Access* 8 (2020), pp. 91028–91047. DOI: 10.1109/ACCESS.2020.2992580 (cit. on p. 11).

[26] Ziwei Ren, Wenfan Li, and Qing Yang. “Location Verification for VANETs Routing”. In: Oct. 2009, pp. 141–146. DOI: 10.1109/WiMob.2009.33 (cit. on pp. 11, 60).

[27] Wafa Ben Jaballah et al. “Secure Verification of Location Claims on a Vehicular Safety Application”. In: *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*. 2013, pp. 1–7. DOI: 10.1109/ICCCN.2013.6614193 (cit. on pp. 11, 35, 60).

[28] Jordan Montenegro, Cristhian Iza, and Mónica Aguilar Igartua. “Detection of Position Falsification Attacks in VANETs Applying Trust Model and Machine Learning”. In: *Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*. PE-WASUN ’20. Alicante, Spain: Association for Computing Machinery, 2020, pp. 9–16. ISBN: 9781450381185. DOI: 10.1145/3416011.3424757 (cit. on pp. 11, 60).

[29] B. W. Silverman and M. C. Jones. “E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)”. In: *International Statistical Review / Revue Internationale de Statistique* 57.3 (1989), pp. 233–238. ISSN: 03067734, 17515823. URL: http://www.jstor.org/stable/1403796 (cit. on p. 21).

[30] URL: https://www.nsnam.org/ (cit. on p. 29).

[31] MINORU NAKAGAMI. “The m-Distribution—A General Formula of Intensity Distribution of Rapid Fading”. In: *Statistical Methods in Radio Wave Propagation*. Ed. by W.C. HOFFMAN. Pergamon, 1960, pp. 3–36. ISBN: 978-0-08-009306-2. DOI: https://doi.org/10.1016/B978-0-08-009306-2.50005-4. URL: https://www.sciencedirect.com/science/article/pii/B9780080093062500054 (cit. on p. 31).

[32] E. Pajala et al. “An improved simulation model for Nakagami-m fading channels for satellite positioning applications”. English. In: *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication (WPNC’06), Hannover, Germany, March 16, 2006*. Ed. by K. Kyamakya. ISBN 3-8322-4862-5 Contribution: organisation=tlt,FACT1=1. 2006, pp. 81–89. ISBN: 3-8322-4862-5 (cit. on p. 31).

[33] Muhammad Arif et al. “A survey on security attacks in VANETs: Communication, applications and challenges”. In: *Vehicular Communications* 19 (2019), p. 100179. ISSN: 2214-2096. DOI: https://doi.org/10.1016/j.vehcom.2019.100179. URL: https://www.sciencedirect.com/science/article/pii/S2214209619302268 (cit. on p. 33).

[34] Carla De Francesco, Claudio E. Palazzi, and Daniele Ronzani. “Fast Message Broadcasting in Vehicular Networks: Model Analysis and Performance Evaluation”. In: *IEEE Communications Letters* 24.8 (2020), pp. 1669–1672. DOI: 10.1109/LCOMM.2020.2993006 (cit. on pp. 39, 59).

[35] Armir Bujari et al. “Fast multi-hop broadcast of alert messages in VANETs: An analytical model”. In: *Ad Hoc Networks* 82 (2019), pp. 126–133. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2018.07.024. URL: https://www.sciencedirect.com/science/article/pii/S1570870518305225 (cit. on p. 39).

[36] Rhattoy ab and Abdelkarim Zatni. “The Impact Of Propagation Environment And Traffic Load On The Performance Of Routing Protocols In Ad Hoc Networks”. In: *CoRR* abs/1202.1677 (Feb. 2012). DOI: 10.5121/ijdps.2012.3106 (cit. on p. 46).

[37] Scott E. Carpenter and Mihail L. Sichitiu. “An Obstacle Model Implementation for Evaluating Radio Shadowing with Ns-3”. In: *Proceedings of the 2015 Workshop on Ns-3*. WNS3 ’15. Barcelona, Spain: Association for Computing Machinery, 2015, pp. 17–24. ISBN: 9781450333757. DOI: 10.1145/2756509.2756512. URL: https://doi.org/10.1145/2756509.2756512 (cit. on p. 60).

[38] Mohammad A. R. Abdeen et al. “Performance Evaluation of VANET Routing Protocols in Madinah City”. In: *Electronics* 11.5 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11050777. URL: https://www.mdpi.com/2079-9292/11/5/777 (cit. on p. 60).

[39] Gongjun Yan et al. “General Active Position Detectors Protect VANET Security”. In: *2011 International Conference on Broadband and Wireless Computing, Communication and Applications*. 2011, pp. 11–17. DOI: 10.1109/BWCCA.2011.12 (cit. on p. 60).

[40] Gongjun Yan Yan et al. “Providing VANET Security through Active Position Detection”. In: *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks*. VANET ’07. Montreal, Quebec, Canada:

Association for Computing Machinery, 2007, pp. 73–74. ISBN: 9781595937391. DOI: 10.1145/1287748.1287762. URL: https://doi.org/10.1145/1287748.1287762 (cit. on p. 60).

[41] Rasheed Hussain, Jooyoung Lee, and Sherali Zeadally. "Trust in VANET: A Survey of Current Solutions and Future Research Opportunities". In: *IEEE Transactions on Intelligent Transportation Systems* 22.5 (2021), pp. 2553–2571. DOI: 10.1109/TITS.2020.2973715 (cit. on p. 60).

[42] Xuanxia Yao et al. "Using trust model to ensure reliable data acquisition in VANETs". In: *Ad Hoc Networks* 55 (2017). Self-organizing and Smart Protocols for Heterogeneous Ad hoc Networks, pp. 107–118. ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2016.10.011. URL: https://www.sciencedirect.com/science/article/pii/S1570870516302943 (cit. on p. 60).

[43] Yi Zeng et al. "DeepVCM: A Deep Learning Based Intrusion Detection Method in VANET". In: *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. 2019, pp. 288–293. DOI: 10.1109/BigDataSecurity-HPSC-IDS.2019.00060 (cit. on p. 60).