Misbehavior Detection System on Vehicular Network based On 2-Step Prediction, Deep Learning Algorithm and Basic Safety Messages

Nur Cahyono Kushardianto

Thèse de doctorat

Pour obtenir le grade de Docteur de

l'UNIVERSITE POLYTECHNIQUE HAUTS-DE-FRANCE

et de l'INSA HAUTS-DE-FRANCE

Discipline, spécialité selon la liste des spécialités pour lesquelles l'Ecole Doctorale est accréditée :

Télécommunication

Présentée et soutenue par Nur Cahyono KUSHARDIANTO.

Le 31 janvier 2023, à Valenciennes

Ecole doctorale :

Ecole Doctorale Polytechnique Hauts-de-France (ED PHF n°635)

Unité de recherche :

Institut d'Electronique, de Micro-Electronique et de Nanotechnologie - Site de Valenciennes (IEMN - UMR CNRS 8520)

Système de Détection de Comportement Suspect sur le Réseau de Véhicules Basé sur la Prédiction en deux Étapes, l'Algorithme d'Apprentissage Profond et les Messages de Sécurité de Base

JURY

Président du jury

- FOUCHAL, Hacène. Professeur. Université de Reims Champagne-Ardenne.

Rapporteurs

- FOUCHAL, Hacène. Professeur. Université de Reims Champagne-Ardenne.
- ELASSALI, Raja. Professeure. Université Cadi Ayyad.

Examinateurs

- MOKDAD, Lynda. Professeure. Université Paris-Est Créteil.
- CHRIFI ALAOUI, Meriem. Maître de Conférences. Université Polytechnique Hauts-de-France.

Directeur de thèse

- ELHILLALI, Yassin. Professeur. Université Polytechnique Hauts-de-France.

Co-directeur de thèse

- TATKEU, Charles. Directeur de Recherche. Université Gustave Eiffel.

Co-encadrant

- RIBOUH, Soheyb. Maître de Conférences. Université de Rouen.
- GOUDALO, Wilson. Dr-Ingénieur de Recherche. ABERDI Inc.

PhD Thesis

Submitted for the degree of Doctor of Philosophy from

UNIVERSITE POLYTECHNIQUE HAUTS-DE-FRANCE

and INSA HAUTS-DE-FRANCE

Subject :

Telecommunication

Presented and defended by Nur Cahyono KUSHARDIANTO.

On 31 January 2023, Valenciennes

Doctoral school :

Doctoral School Polytechnique Hauts-de-France (ED PHF n°635)

Research unit :

Institute of Electronics Microelectronics and Nanotechnology – Valenciennes site (IEMN – UMR CNRS 8520)

Misbehavior Detection System on Vehicular Network based On 2-Step Prediction, Deep Learning Algorithm and Basic Safety Messages

JURY

President of jury

- FOUCHAL, Hacène. Professor. Université de Reims Champagne-Ardenne.

Reviewers

- FOUCHAL, Hacène. Professor. Université de Reims Champagne-Ardenne.
- ELASSALI, Raja. Professor. Université Cadi Ayyad.

Examiners

- MOKDAD, Lynda. Professor. Université Paris-Est Créteil.
- CHRIFI ALAOUI, Meriem. Associate Professor. Université Polytechnique Hauts-de-France.

Thesis director

- ELHILLALI, Yassin. Professor. Université Polytechnique Hauts-de-France.

Thesis co-director

- TATKEU, Charles. Research Director. Université Gustave Eiffel.

Co-supervisor

- RIBOUH, Soheyb. Associate Professor. Université de Rouen.
- GOUDALO, Wilson. PhD - Research Engineer. ABERDI Inc.

Abstract

Over the past decade, we have seen that a large and growing number of people use their own vehicles as their main mode of transportation. This situation has an impact on traffic conditions that are increasingly unpredictable. Traffic jams and accidents are increasing from year to year. To try to overcome these problems, we use Intelligent Transport System (ITS) which seems quite promising with the possibility of exchanging information through V2X – Vehicle to Ever Things communication links. Thus, with the modernization of transport infrastructures, the continuous improvement of safety devices within vehicles and the use of information and communication technologies to supplement the existing ADAS devices for driving assistance, intelligent transport systems attempt to significantly reduce accidents and make traffic flow more smoothly. For example, intelligent transportation systems can automatically provide early warning of hazards on the road or can even take control of the vehicle in the even of driver failure or inadequacies and prevent him from losing control of his vehicle. But this sophisticated system will bring harm if the security side is not noticed. Intelligent transportation systems depend on network technology and rely on wireless communications systems to establish V2V and V2I – Vehicle to Vehicle and Vehicle to Infrastructure links. And if the security of the links is not ensured, this sophisticated system can cause damage. Being open, network technology is quite vulnerable to interference and exposed to attacks by sending inappropriate messages, misbehavior messages which can be in the form of malfunction messages or attacks. To overcome this and protect communications, techniques can be used which implement the MisBehavior Detection System (MDS) which works like an Intrusion Detection System (IDS). Traditional IDS works by using database patterns of attacks. Along with the increasingly complex network technology, the MDS is finding it increasingly difficult to detect new attack patterns. For this reason, it is necessary and essential to implement a technology that can adapt to any pattern of attack. Thus, the research work we developed, proposes a security method using Machine Learning technique as a basis of an IDS. The method we proposed can predict the behavior of a vehicle, regardless of whether or not the vehicle is an attacker, based on the vehicle's position and speed information. With the help of this method, we can simplify the necessary information needed to recognize misbehavior on the vehicular network. In addition, we developed also provide a prediction system based on basic safety messages, which serve as the industry standard for vehicle communication in the Cooperative ITS (C-ITS) ecosystem. This system, which predicts whether or not a message comes from the attacker's vehicle, has the potential to serve as an alternative to IDS. Both approaches have been evaluated offline and online with very encouraging outcomes. They offer interesting prospects with potential development for the advancement of C-ITS security technology in general.

Keywords : ITS, MDS, machine learning, vehicular network, attacker, misbehavior.

Résumé

Durant cette dernière décennie, on constate qu'un nombre important et de plus en plus croissant de personnes utilisent leur véhicule personnel comme mode de transport principal. Cette situation a un impact sur les conditions de circulation qui sont de plus en plus imprévisibles, les embouteillages et les accidents qui augmentent d'année en année. Pour tenter de surmonter ces problèmes, on a recours à des systèmes de transports intelligents (STI) qui semblent assez prometteurs avec des possibilités d'échanges d'informations par les liens de communication V2X – Vehicle to Every Things. Ainsi, avec la modernisation des infrastructures de transports, l'amélioration continue des dispositifs de sécurité au sein des véhicules et l'utilisation des technologies de l'information et de la communication pour compléter les dispositifs ADAS existants d'aide à la conduite, les systèmes de transports intelligents tentent de réduire significativement les accidents et fluidifier le trafic. En l'exemple, les STI peuvent automatiquement fournir une alerte précoce des dangers sur la route ou peuvent même prendre contrôle du véhicule en cas de défaillance ou d'insuffisances du conducteur et lui éviter la perte de contrôle de son véhicule. Les systèmes de transport intelligents dépendent de la technologie de réseau et s'appuient sur les systèmes de communications sans fil pour établir des liens V2V et V2I – Véhicule à Véhicule et Véhicule à Infrastructure. Et si la sécurité des liens n'est pas assurée, ce système sophistiqué peut occasionner des dommages. Étant ouverte,la technologie de réseau est assez vulnérable aux interférences et exposée aux attaques par l'envoi de messages inappropriés, de mauvais comportement qui peuvent prendre la forme de messages de dysfonctionnement ou d'attaques. Pour surmonter cela et protéger les communications, on peut avoir recours aux techniques qui mettent en oeuvre le système de détection de comportement suspect (MDS) qui fonctionne comme un système de détection d'intrusion (IDS). L'IDS traditionnel fonctionne en utilisant des modèles d'attaques de base de données. Parallèlement à la technologie de réseau de plus en plus complexe, le MDS a de plus en plus de difficultés à détecter de nouveaux types d'attaque. Pour cette raison, il est nécessaire et indispensable de mettre en œuvre, une technologie capable de s'adapter à n'importe quel type d'attaque. Ainsi, les travaux de recherche que nous avons développés, proposent une méthode de sécurisation utilisant la technique de Machine Learning comme base d'un IDS. La méthode que nous avons proposée peut prédire le comportement d'un véhicule, que le véhicule soit ou non un attaquant, sur la base des informations de position et de vitesse du véhicule. À l'aide de cette méthode, nous pouvons simplifier les informations nécessaires pour reconnaître les comportements anormaux sur le réseau véhiculaire. De plus, nous avons développé également un système de prédiction basé sur des messages de sécurité de base, qui servent de norme à l'industrie pour la communication des véhicules dans l'écosystème Cooperative ITS (C-ITS). Ce système, qui prédit si un message provient ou non du véhicule de l'attaquant, a le mérite de servir d'alternative à l'IDS. Les deux approches ont été évaluées hors ligne et en ligne avec des résultats très encourageants. Ils offrent des perspectives intéressantes avec des potentialités de développement allant de la technologie de sécurité au C-ITS en général.

Mots Clés : STI, MDS, machine learning, réseau de véhicules, attaquant, compertement suspect.

Dedicate to my loving parents, Dr Mujianto and Dyah Kusmaniarti, my loving wife Ziah and my adorable children Fahd, Asha and Hanane ...

Acknowledgment

I am grateful to Allah, the Almighty, for His blessings and will, which enabled me to complete this doctoral thesis.

I would like to thank everyone who contributed to this work.

Firstly, I would like to express my deepest gratitude to **Mr. Yassin EL-HILLALI** for agreeing to be my primary advisor during my doctorate. His ideas and support have been precious and have helped me complete my thesis successfully. I am grateful for his guidance and dedication throughout this process.

I also would like to express my gratitude to my co-supervisors, **Mr. Charles TATKEU, Mr. Soheyb RIBOUH, and Mr. Wilson GOUDALO**, for their patient guidance, support, and valuable contributions to my doctoral thesis. Their ideas and technical assistance were crucial in achieving the best possible results.

I am grateful as well to the jury member: Mr. Hacene FOUCHAL, Mme. Raja ELASSALI, Mme. Lynda MOKDAD, and Mme. Meriem CHRIFI ALAOUI, for examining my work and for their attention and interest in it.

I am also very grateful to the Directorate General of Higher Education, Research, and Technology Republic of Indonesia and the committee of BPPLN 2019, who have provided me with financial and administrative support during my doctoral studies.

I would like to extend my heartfelt gratitude to my father, my mother and my brother, whose steadfast love and support gave me the strength to persevere and complete this thesis. To my wife, thank you for your unwavering support and patience during this journey. Your encouragement and belief in me were a constant source of my motivation. To my kids, your love and support were my driving force, and I am forever grateful for your presence in my life.

In addition, I would like to acknowledge my "comrades in arms", Mir'atul Khusna MUFIDA and Faisal Abdulrahman BUDIKASIH, who have supported and encouraged me during my doctorate.

Finally, I want to thank the members of the IEMN laboratory in Valenciennes, who provided invaluable assistance and support throughout this project.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Problematic

The number of vehicle growth in the world always increases year by year. Dargay et al have a projection that a total number of the vehicle over the world in 2002 until 2030 will increase 2,5 times greater, more than two billion vehicles will exist[17]. The surge in the number of vehicles has the effect of congestion on the highway and also the increasing number of accidents and traffic violations. According to WHO, traffic accidents are the number eight cause of death globally. It is even said that every year there are 1.35 million people who die due to accidents and more than 50 million people are injured [69]. Another example is that there were about 109.215 traffic accident cases in 2018 in Indonesia. This number is quite large compared to the number of traffic accidents in the previous ten years, which are 59.164 cases [2]. The negative impact of the increasing number of vehicles will also occur on the environment, social, and economic.

Moving along with the times, Information and communications technology can lessen the negative impacts. Nowadays, technology in the transportation sector has developed which is called the "Intelligent Transport System" (ITS). Although ITS was developed by advanced countries such as the USA, Japan and parts of Europe, for now developing countries have begun to use it to overcome traffic congestion problems in rapidly developing cities [53].

ITS are expected to provide safer travel, adaptive to road condition, less traffic congestion, and various entertainment service to the user [80]. In order to make it happen, ITS system will exchange data between different ITS entities, roadside units, and traffic management. Absolutely when it's happened, security and privacy will be much important. A system that is closely related to human safety must be considered security risks that can occur. Consequently, it is important to provide data integrity, authenticity, confidentiality and non-repudiation for all Intelligent Transport System. Some cyber-attack can be a threat to ITS such as DoS attack which floods the network with bogus messages until the legitimate user cannot be connected to the system, fake users can falsify information about traffic condition, malicious users will steal personal data from legitimate users, and so on [39].

Cyberattack mitigation is not a simple undertaking. Every possible assault should be identified. IDS (Intrusion Detection System) plays a crucial function in this section. Standard IDS can identify various attack types, but the rising network structure and complexity make it harder for the IDS to detect all of the attacks. IDS techniques include support vector machine, decision tree, genetic algorithm, data mining, Artificial Neural Networks (ANN), and others. This technique is extensively employed for IDS, although its training time is inadequate [23]. Machine Learning (ML) is another way for implementing IDS. Its method is frequently employed in the domain where it demonstrates its superiority over traditional rule-based algorithms. The method of ML is being standardized in cyber detection systems in order to replace the entry-level security analysts[7].

1.2 Challenges

Developing the need for security at ITS has opened several research challenges. As part of our work, we propose solutions to solve the following research challenges related to ITS:

- Misbehavior in vehicular networks is growing day by day. New misbehavior have sprung up along with the increasing use of C-ITS on the road.

- The more misbehavior in the vehicle network, the longer it will leave a digital trail that accumulates and will lead to big data. This will bring difficulty in analyzing it.

– Some misbehavior have been detected through several scientific studies. But the increasing network structure, complexity, high computation overhead and lack of pattern analysis will cause the Intrusion Detection System to have difficulty detecting all potential misbehavior/attacks.

1.3 Contributions

This research has resulted in the following contributions:

– In this study, we offer a technique for detecting attacks on the Vehicular Network based on attack database patterns and ML.

– Research has succeeded in making misbehavior detection models in the Vehicular Network in cases involving many types of attacks.

– Research has proven that the misbehavior detection model can be applied in real time implementation with good results.

1.4 Outline of the manuscript

This thesis is organized into three main chapters as follows:

– The first chapter explains intelligent transport systems, their characteristics, their safety factors, and recent C-ITS projects in Europe, the USA, and Asia. In this chapter, we also describe the studies that are our references, along with their comparisons as a form of state of the art research. The last is an explanation of misbehavior in the vehicular network.

– The second chapter explains ML-based misbehavior detection systems, primarily deep learning, as a research methodology. In addition, we also present the misbehavior detection system that we offer as a solution. In other words, we explain the offline phase of ML and its process results.

– In the third chapter, we explain the real-time simulation of misbehavior detection on the vehicular network. Begin with an explanation of the simulation framework we use, and then we explain how to implement the detection system model we propose. At the end of the chapter, we present the real-time implementation results in the form of measurable parameters.

Chapter 2

Intelligent Transport System

Contents

2.1 Introduction

An Intelligent Transportation System (ITS) is an advanced technology with a network infrastructure that facilitates the interaction between artificial intelligence elements such as sensors, actuators, databases, microprocessors, and others. Vehicles equipped with ITS will become more sophisticated and functional vehicles. The sensors in it will be able to consider internal and external factors. Meanwhile, a processor fitted with artificial intelligence will make the vehicle store information and plan actions. Especially in autonomous cars, vehicles can make their own decisions appropriately. ITS will be equipped with a modern network infrastructure that ensures the interaction between internal and external elements runs safely and efficiently. In addition to changing their state, intelligent vehicles use actuators and various signals that affect their environment, so they can adapt to external human commands or even independently [56].

2.2 Cooperative ITS (C-ITS)

The communication system between vehicles in C-ITS is usually called V2V communication. In this communication system, each vehicle is equipped with On Board Unit (OBU) on each vehicle that is used to broadcast its movements to neighbors or Road Side Unit (RSU) in the form of movement data, maneuvers, and so on. Of course, the goal is that the surrounding vehicles can anticipate if there are dangerous conditions based on the information received. Information from direct interactions between vehicles and between vehicles and road infrastructure forms the basis of C-ITS systems (V2I, V2V). New organizations have been established at the European level to provide ETSI-specific standards for these systems. They allow for drivers and traffic controllers to communicate and coordinate their activities. In order to help the driver make the best judgments and adapt to the traffic situation [**?**]. This cooperative element, made possible by digital connectivity between cars and between vehicles and infrastructure, promises to dramatically improve road safety, traffic efficiency, and driving comfort. These interconnected systems will lessen pollution and enhance air quality. Data from cellular networks and/or IEEE 802.11p Wi-Fi are used by C-ITS installations to communicate.

2.3 C-ITS Project

During the last years various initiatives providing policy rules for C-ITS deployment and a large number of projects demonstrating C-ITS implementation have taken place in Europe and USA.

2.3.1 European Project

2.3.1.1 I_HeERO Project

In 2015, International Road Transport Union (IRU) Projects opted to join the EU-funded I HeERO project, whose purpose is to prepare and increase the deployment of PSAPs in European nations. The primary objective of IRU Projects' participation is to feed inputs from road transport representatives into the planned activity, ensuring that eCall advancements take inputs from road transport operators into consideration. IRU Projects has now assumed the position of project task leader for future work on eCall for commercial vehicles. As a result of these experiments, the I HeERO project will provide a strategy for Member States to modernize their infrastructure to support eCall as a genuine pan-European idea. The primary objective is to aid the countries of Bulgaria, Cyprus, the Czech Republic, Finland, Germany, Greece, Ireland, Italy, Luxembourg, Portugal, Romania, and Slovenia in their efforts to set up eCall PSAP infrastructure. The second objective is to fund research into new technologies and standards that will allow eCall to be adapted to the demands of vehicles other than cars (such as buses, coaches, motorbikes, and trucks). The Action will also carry out studies on the expansion of eCall to other vehicle categories, such as powered two-wheelers, trucks, and dangerous goods carriers, which are not covered by the EU legislation on eCall. Additionally, it will look at the legislation's requirements for data integration and conformity evaluation for all PSAPs. [38]

2.3.1.2 NordicWay Project

The proposed activity, called NordicWay which was started in January 2015, is a pre-deployment pilot of Cooperative ITS (C-ITS) services in four countries, including Finland, Sweden, Norway, and Denmark. Wide-scale deployment will come after, with the possibility of scaling up to Europe. NordicWay offers the ability to integrate road transportation with other forms of transportation while enhancing mobility's comfort, efficiency, and safety. The first extensive C-ITS cellular communication (3G

and LTE/4G) pilot was conducted by NordicWay. It provides consumers cross-border roaming between various mobile networks and continuously interoperable services, providing C-ITS services in all member nations [36].

2.3.1.3 SCOOP Project

The SCOOP project is a test run for the eventual implementation of collaborative intelligent transportation systems in France. This initiative, which was kicked off by my ministry in the current calendar year, is predicated on the communication and exchange of information between cars, road infrastructure, and communication infrastructure. Through the SCOOP program, the French government has demonstrated its unwavering dedication to the growth of France's intelligent transportation industry as a whole. This is the largest test ever conducted in Europe, with 3,000 intelligent vehicles being deployed across over 2,000 kilometers of networked roadways. The SCOOP project should link various roadways and modes of transportation. As a result of the collaboration between public and commercial entities, the services can be evaluated in a diverse range of terrains and road profiles. In addition to the technological issues (feasibility on a national scale), the project will investigate the legal aspects that are associated with data exchanges (protection of private life, data ownership, etc.) and the safety of information systems. The objective of the SCOOP project is to outfit five pilot sites by the end of the year 2015, test and evaluate five sets of priority services on these sites throughout the years 2016–2018, and then prepare for the nationwide roll out of these services by the end of the year 2020 [60].

2.3.1.4 SolC-ITS Project

The SolC-ITS (SOLRED C-ITS Monitoring Network) Project started in March 2016. The overall objective of the project is to test a new Integrated Fuel and Fleet Management System, the so-called C-ITS Telemat, which enables the automatic real time calculation of the smartest route plan and the automatic estimation, authorisation and payment of the refueling needed to complete a planned route. Moreover, the system provides the heavy duty vehicles (HDV) drivers and fleet managers with useful notifications concerning maintenance scheduling, eco/safety driving, traffic issues as well as information on the estimated fuel consumption versus the real one.
The testing of this system will be done through a monitoring network which will involve approximately 53 Repsol service stations along the Spanish part of the Atlantic and Mediterranean core network Corridors [37].

2.3.1.5 C-Roads Project

C-Roads is a platform which brings together road authorities and operators from the Member States: France, Austria, Belgium, Czech Republic, Denmark, Finland, Germany, Hungary, Ireland, Italy, Portugal, Slovenia, Spain, Sweden, - Netherlands, United Kingdom, Norway, Switzerland and Australia. The C-Roads Platform was officially launched on December 12th, 2016 with a Kick-Off event held in Brussels. To officially launch the C-Roads Cooperation, Commissioner Bulc and representatives from the C-Roads Member States will have a signing ceremony.
The objective of the C-Roads platform is to ensure road safety at European level by aligning the specifications of Cooperative ITS (C-ITS) to guarantee interoperability between European ITS. A rapid and EU-wide deployment of harmonized C-ITS services is key to this objective. C-Roads member states strive to ensure seamless operation of cross-border C-ITS services and as such contribute to laying the foundations for connected and automated driving.[63]

2.3.1.6 The InterCor Project

InterCor is a three-year (2017–2020) European initiative project that unites France, the Netherlands, Belgium, and the United Kingdom. It has a budget of about 30 million euros. It attempts to link up European road transportation. In fact, the goal of this project is to coordinate the strategic deployment of common specifications in the four Member States and the realization of C-ITS. For the purpose of operating and evaluating C-ITS services, C-ITS pilot sites—which are utilized to transmit data from cellular and/or ITS-G5 networks—will be built along roughly 1530 km. With the help of the interCor project, people and commodities will be moved more safely, effectively, and affordably throughout France, the Netherlands, Belgium, and the United Kingdom. By specifying, employing, and promoting a hybrid communication technique that combines cellular and ITS-G5 communication, it also seeks to provide C-ITS services on a greater scale [92].

2.3.1.7 CITRUS Project

The Belgian project CITRUS (C-ITS for Trucks) investigates the technological and financial potential of a truck driver companion app. At least 300 truck drivers from Colruyt Group will participate in a pilot rollout of the app on the Belgian highway network over the course of 36 months. For 21 months, the pilot will be in operation (January 2018 – September 2019). The associated app will offer some "Day 1 services," including warnings about traffic congestion, stationary vehicles, and road construction. Additionally, it will optimize green light cycles and approaching vehicle speeds at crucial junctions and offer suggestions on speed, routing, and other information. The application will improve driving conditions and lower CO2 emissions from truck traffic [78].

2.3.1.8 InDiD Project

One of the C-ITS projects that France is supporting is InDiD. It was chosen by the European Commission as part of the CEF -Coordinate Europe Facilities- call for projects with the goal of developing intelligent transportation systems. The European Union will contribute 50% of the project's funding over a five-year period (from 2019 until 2023). It is an extension of earlier C-ITS initiatives including SCOOP, C-ROADS, and InterCor. The Indid project involves creating new use cases for the urban environment as well as use cases of augmented perception for the autonomous car, in addition to ensuring improved traffic management and road safety. It also covers mapping of high-definition digital infrastructure. It also targets experiments of 5G-based vehicular communications for driverless vehicles. This project is supported by a powerful partnership that unites 24 partners from around France, including motorway firms, industrial players, interdepartmental road directorates, and academic partners (universities and research centers) [64]. InDiD plans to continue deploying C-ITS on new road test sites to expand infrastructure services. The pilot locations are in 4 major French basins: *the Mediterranean, South-west, Center, and North,* see Figure 2.1

2.3.2 USA Project

2.3.2.1 E-VII Project

The pilot project "Arizona Emergency Vehicle Infrastructure Integration (E-VII)" was financed in 2008 by the Arizona Transportation Research Center, Arizona State University, Maricopa County Department of Transportation (DOT), and Michigan DOT. The project was divided into two phases: Phase 1, which involved the evaluation and deployment of prototype applications, and Phase 2, which involved the demonstration of applications, equipment interfaces, and driver engagement with the

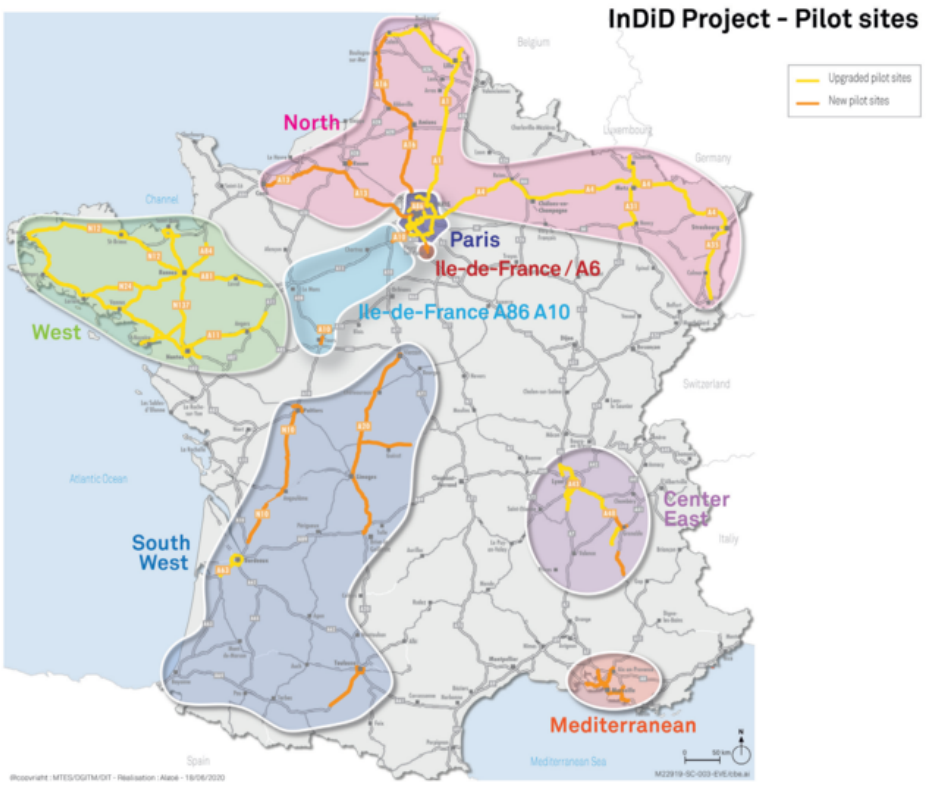


Figure 2.1: InDid Project Pilot Sites [64]

on-board systems in a "parking lot" site in Maricopa County. The goal was to better respond to traffic incidents by developing and testing cutting-edge technologies for emergency vehicles. The "Multipath Signal Phase and Timing (SpaT) Broadcast project" was started in 2009 by the Michigan DOT, the University of Michigan Transportation Research Institute (UMTRI), and the Institute for Information Industry. The project's goal was to give drivers speeding tips so they could safely navigate the green period of the next signalized intersection [83].

2.3.2.2 SPMD Project

2011 saw the launch of the "Connected Vehicle Safety Pilot Model Deployment (SPMD)" project by UMTRI and USDOT. Real-time data was gathered as part of the experiment in order to assess how well Connected Vehicle (CV) safety technologies work. Vehicle-To-Vehicle (V2V) and Vehicle-To-Infrastructure (V2I) communication devices were installed in more than 2800 volunteer vehicles from Ann Arbor so that they could communicate Basic Safety Messages (BSM) concerning speed, location, and direction on a 73 lane-miles-long stretch of road (US Department of Transportation, 2018a). The "Integrated Mobile Observations 2.0 (IMO)" project was funded by the USDOT, Michigan DOT, and the Federal Highway Administration (FHWA) in the same year. The project built a system receiving weather-road data from the I-94 corridor users (fleet of 60 vehicles) and transmitting it to weather experts [11].

2.3.2.3 Heavy Truck CACC Project

The "Heavy Truck Cooperative Adaptive Cruise Control" project, sponsored by the Federal Highway Administration of the United States Department of Transportation and Auburn University, addressed the implementation of Driver Assistive Truck Platooning (DATP), a type of Cooperative Adaptive Cruise Control (CACC) for heavy

trucks. Radars, Dedicated Short-Range Communication (DSRC) based on V2V communications, and satellite positioning technologies were all part of the apparatus. Using data from connected vehicles, the University of Washington financed and oversaw the "Enhancing Safe Traffic Operations" initiative in 2015. The project created a low-cost Communication Note (CN) gadget and an Android-based mobile application to alert Vulnerable Road Users (VRU) and inform drivers about dangerous situations (VRUs) [26].

2.3.2.4 CV Pilot Deployment Program

The USDOT introduced the "Connected Vehicle Pilot Deployment Program" in an effort to advance CV technology. The program's primary goal was to reduce environmental impacts while increasing traveler mobility and safety through the creative and economical fusion of CV technology and mobile applications.
Three corridors are part of the NYC pilot: Brooklyn Flatbush Avenue, Manhattan FDR Drive, and Manhattan Grid. The integrated applications promote NYC's Vision Zero campaign while focusing on safety. The implemented safety applications, which encompass 8000 CV and 300 RSU in the three corridors, are based on V2V, V2I, and Infrastructure-to-Pedestrian (IVP) communications (US Department of Transportation, 2018c). In order to lessen congestion and collisions, the Tampa pilot focuses on the implementation of V2V and V2I applications. The pilot program's other objectives include using CV technology to improve pedestrian safety, accelerate bus operations, and prevent conflicts between street cars, pedestrians, and passenger automobiles at sites with significant quantities of mixed traffic [62].

2.3.2.5 TPIMS Project

"Truck Park Information Management Systems," a TPIMS project, was financed in 2016 by a federal Tiger Grant. This project's objective was to give truck drivers access to real-time information to help them make wise and cost-effective parking selections (Mid America Association of State Transportation Officials, 2016). US Route 33 was fitted with fiber optic connections as part of the Ohio Smart Mobility Corridor project, allowing researchers and traffic monitors to link in real time with wireless road sensors (Smart Mobility Corridor).
In order to gather data on the road's weather, the "5.9 GHz Dedicated Short-Range Communication Vehicle-based Road and Weather Condition Application" project was started in 2017 [65]. TPIMS cover more than 150 monitored parking sites on nine high-volume freight corridors: Indiana, Jasper, White,Boone, Bartholomew and Clark counties, Delaware and DeKalb counties, Vigo, Hendricks, Hancock and Wayne counties.see Figure 2.2

2.3.3 Asian Project

2.3.3.1 Seoul TOPIS

Seoul TOPIS (Seoul Transport Operation and Information Service) is the Seoul Metropolitan Government's ITS brand. It was created in 1998 as the first service of its sort in Korea to address urban transportation issues. It is a sophisticated transportation information system that enables quick decisions and responses in times of emergency and forecasts and prevents transportation problems through the analysis of large amounts of data. For the effective operation of traffic management systems, TOPIS was formally introduced in 2005. The Seoul Metropolitan Government built and began operating the Seoul Integrated Safety Center in 2013 to deal with transit management, natural disasters, and national emergencies. In the course of its development, TOPIS is piloting an autonomous vehicle testbed, and in December 2020, Sangamdong will become the first autonomous vehicle pilot driving region [68].

Figure 2.2: TPIMS Deployment Corridors [65]

2.3.3.2 SAVI

The Singapore Autonomous Vehicle Initiative (SAVI) is a collaborative effort between the Land Transport Authority (LTA) and the Agency for Science, Technology, and Research (A*STAR) with the goal of creating a testing ground where Autonomous Vehicle (AV) technologies, applications, and solutions can undergo rigorous development and testing. The program expands on A*STAR's existing expertise in video and image analysis. With AVs, people in Singapore can reduce pollution and crime by choosing car sharing over individual vehicle ownership. This is the end goal of A*STAR's research and development efforts, and we are making significant progress toward it. The Institute for Infocomm Research (I2R) at A*STAR is developing an AV Vehicle at 2017, which is a driverless bus for a mass transport service that operates on defined routes and planned timings and can alleviate manpower restrictions for bus services. The first place where autonomous vehicles will be tested is in One-North.Autonomous Truck Platooning Trial and Autonomous Bus Trial are two examples of the output from this project [9].

2.3.3.3 Next-Gen C-ITS

The National Institute of Land and Infrastructure Management (NILIM) Japan has been conducting public private "Joint Research for Next Generation C-ITS in Japan" since January 2018 in order to realize next-generation Cooperative ITS (C-ITS) by combining a next generation of Vehicle-To-Infrastructure (V2I). This project is funded by the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) Japan with the cooperation of road administrators, major domestic automobile manufacturers, electronic equipment manufacturers, a map company, and other corporations. This research aims to share and utilize information collected by vehicles and traffic and other information possessed by road administrators. Next-generation C-ITS enables road administrators to utilize diverse information collected through vehicles, thereby improving the efficiency of road control, while providing road users with information about roads and traffic conditions based on data collected from specific vehicles. This

next-generation C-ITS will allow for safe and comfortable expressway travel [81].

2.3.3.4 SIP-adus

Japan Strategic Innovation Promotion Program Automated Driving for Universal Service (SIP-adus) is running in second phase in 2019. Cooperative regions have been prioritized for development projects. Utilizing traffic signal information from the transport infrastructure of arterial and general public highways, merging lane assistance information from expressways, etc. Field Operational Tests (FOT) of vehicle-infrastructure cooperative driving automation began in Tokyo's waterfront district in October 2019. In addition to its social relevance, this project, which intends to accomplish practical application of automated driving, has economic value, such as minimizing traffic accidents and congestion, providing mobility in underpopulated areas and other locations, and eliminating driver shortage. see Figure 2.3. The first phase of the SIP-adus began in 2014 and was crucial in encouraging collaborative automated driving research and development. In 2017, The project conducted large-scale FOT for various reasons, including validating the usefulness of dynamic maps and developing standardized standards. Among the specific accomplishments was constructing the essential structure for map improvement. Long-term objectives include establishing the necessary cooperative areas technology for deployment by 2023. [52].

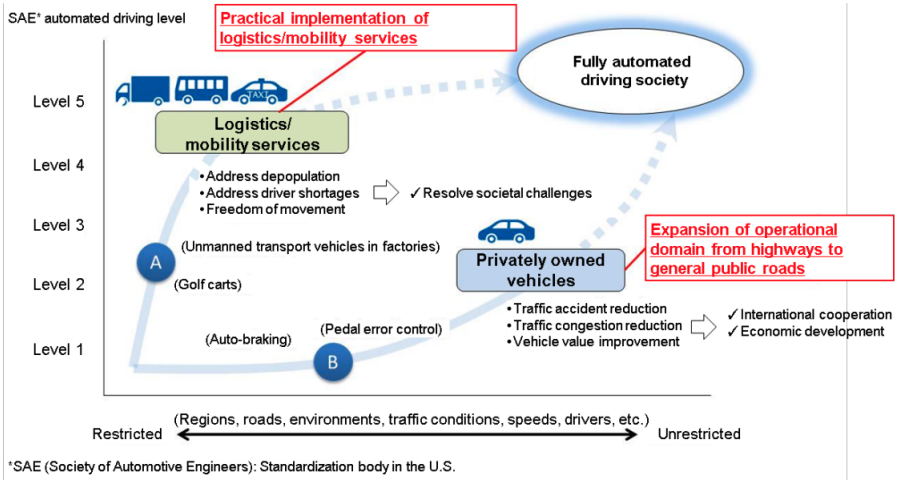


Figure 2.3: SIP-adus 2nd Phase Overview [52]

2.4 Vehicular Communication

Communication between vehicles is broadly included in Vehicle-To-Everything (V2X) communication, in which there are 2 more specific communication systems, namely Vehicle-To-Vehicle (V2V) and Vehicle-To-Infrastructure (V2I).

2.4.1 V2X

The V2X represents a generalization of the previously discussed V2V and V2I communication paradigms. The latter entails the data transfer from a vehicle to any entity that can influence it, or vice versa, and incorporates other, more specialized types of communication, such as Vehicle-to-Pedestrian (V2P) [89], Vehicle-to-Roadside (V2R) [96], Vehicle-to-Device (V2D) [41], and Vehicle-to-Grid (V2G) [24]. According to the assessment on the situation of road safety around the world [69], there are about 1.25 million people died because of road accidents every year around the world. Vulnerable Road Users, which include motorcyclists, cyclists, and pedestrians, made up over half of the victims (VRU) [58]. It is important to remember that poor road design and improper separation from traffic play a big role in creating a dangerous environment for both automobiles and pedestrians [71]. Another problem that shouldn't be overlooked, especially in metropolitan areas, is the distraction that comes from pedestrians using their smartphones and earbuds while walking along the street [67]. As a result, it is essential to create a warning system that includes pedestrians.
One of the main goals of V2X technology is to promote effective communication methods between automobiles and pedestrians in order to reduce accidents, which can sometimes be fatal.

2.4.1.1 V2V

Wireless data communications between moving vehicles make up V2V technology. This communication, which enables moving cars to share information about their location and speed inside an ad hoc mesh network, is primarily intended to prevent accidents [5] Connections between vehicles in V2V can be in the form of partial mesh topology or full mesh topology. In a partial mesh topology, vehicles exchange messages with neighboring vehicles by choosing different multihop paths. In a full mesh topology, only one hop is needed for a vehicle to exchange messages with neighboring connected vehicles. This topology also increases the robustness of the network structure. Even if there is damage to one of the nodes, the communication route will be redefined based on the forwarding table so that communication reaches its destination [8].
Suppose a vehicle is built to carry out safety intervention. In that case, it may independently perform preventive steps, such as emergency braking, without the driver's knowledge, depending on how the technology is developed [22]. Since the functionality of the onboard sensors, cameras, and radars now determines the safety of the vehicle, it is anticipated that V2V communications will be significantly more effective than the OEM's present embedded systems [91]. Based on particular criteria recognized by various gadgets installed on the car, the system responds to any risky situations. Usually, the travel speed, the distance from an obstruction, or the presence of a vehicle in the blind spot are the key factors that are assessed. Even though the technologies being employed are becoming more trustworthy, calculation errors should still be taken seriously. Instead, V2V communication protocols will enhance security performance by allowing all nearby vehicles to communicate with one another. This will enable a car that is in danger (due to a driver's lack of attention, a component failure, an obstacle in the lane, etc.) to make a more wise decision regarding how to handle the problem as it arises.

2.4.1.2 V2I

The V2I communication model enables vehicles in motion to communicate with the road system, in contrast to the V2V communication model, which only permits the transmission of information between vehicles, see Figure 2.4. These elements consist of RFID readers, parking meters, traffic signals, cameras, lane markings, street lamps, and signage [42]. V2I communications sometimes use DSRC frequencies to transmit data wirelessly in both directions, comparable to V2V communications [75]. This information is delivered from the elements of the infrastructure to the car, or the other way around, using an ad hoc network. In the ITS, V2I sensors can collect data on the infrastructure and provide real-time information to drivers regarding road conditions, traffic congestion, potential accidents, the presence of work sites, and parking availability. Similar to this, in order to reduce fuel consumption and enhance traffic flow, traffic monitoring and management systems can change the Signal Phase and Timing (SpaT) and set variable speed limits [86]. The development of autonomously driven cars must begin with the hardware, software, and firmware that enable adequate communication between vehicles and infrastructure. The FHWA received V2I recommendations from the US Department of Transportation in January 2017, with the goal of enhancing mobility and safety while expediting the use of communication systems [66]. The purpose of these guidelines is to assist state governments in setting up V2I projects and maintaining the data required to support them. As was already indicated, government funding and resource issues exist for the implementation of these projects. Because these expenses cannot be covered solely by the money the states receive from fuel taxes and tolls on the highways, a collaboration with the major automakers is required. These companies may profit from the use of big data in communications for their commercial interests.

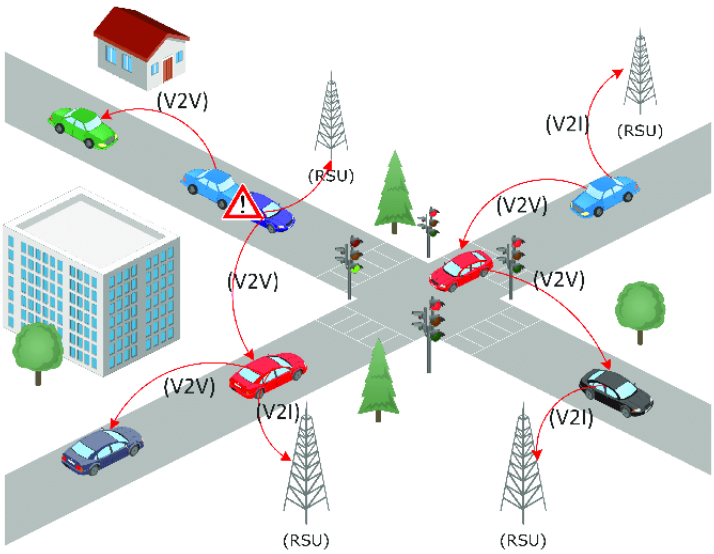


Figure 2.4: Vehicular Network Communication Architecture.[87]

2.4.2 Vehicular Communication Technologies

2.4.2.1 Dedicated Short-Range Communication (DSRC)

DSRC are ad hoc (decentralized) short- and medium-range data transmission systems that support public and private security operations in vehicle-to-infrastructure and vehicle-to-vehicle communications environments or vice versa. The DSRC are standardized to guarantee their interoperability independently of the manufacturer of the media access devices, following the protocol layer stack of ISO Model of Architecture for Open Systems Interconnection, comprising five layers (Physical, MAC and Link, GeoNetwork, Transport, and Application), where we can highlight three differentiating characteristics:the IEEE 802.11p (ITS G5 in Europe) specification is followed at the physical level and MAC, which allows the transmission of data in the dedicated 5.9 GHz channel through spread spectrum technique, and the sending of MAC-level broadcast packets. The network level includes the geographical location of the information handled by the communications device, enabling the so-called GeoNetworking. Finally, the transport level enables the multihop capability for the retransmission and routing of the packets of the vehicular network [40].
DSRC technical description:The communication modes of DSRC allow V2V and V2I communication.

1. V2V Communications: includes multihop geographic routing, using other vehicles as relays for the message delivery
    a. *GeoUnicast*: provides packet delivery from an emitting vehicle to a receiving vehicle that is located in a fixed geographic position, viamultiple hops.
    b. *GeoAnycast*: provides packet delivery to a vehicle (node) that is in aspecific geographic area as a function of set conditions (i.e.,nearer).
    c. *GeoBroadcast*: provides packet delivery in broadcast mode to all the vehicles that are in fixed geographic area.
    d. *Topollogically-Scoped Broadcast (TSB)*: provides packet delivery to every vehicle that is in a range of n-hops from the emitting vehicle.
2. V2I (uplink) and V2I (downlink) Communications: they have an equivalent behavior to V2V but involve DSRC modules installed in the roadside:
    a. One vehicle to beacon (Geounicast)
    b. Beacon to one vehicle (Geounicast)
    c. Beacon to many vehicles (GeoBroadcast, TSB)
    d. Beacon to selected vehicles (GeoAnycast)

In addition to Europe, DSRC 802.11p is widely used in Japan, Korea, Singapore and Australia to apply C-ITS technology [61].

2.4.2.2 3G/4G Mobile Telephony

In 2007, High Speed Downlink Packet Access (HSDPA) technology,corresponding to the 3.5G mobile phone, was available to users, allowing wireless broadband access over high speed UMTS to a maximum bandwidth of 14.4 Mbps. HSDPA technology was surpassed in 2010 by the Long Term Evolution (LTE), enabling the 4th generation of mobile telephony(4G). LTE is the standard for high-speed wireless data communications for mobile phones and data terminals, with transmission speeds of up to 75 Mbit/s for high mobility (200 km/h) and 300 Mbit/s for low mobility, with latencies between 50 and 150 ms. In 2014, the Long Term Evolution Advanced (LTE-A) technology, a 4G evolution, was developed, enabling transmission speeds up to 500 Mbit/s for high mobility (200 km/h) and 1 Gbit/s for low mobility, with

latencies between 10 and 20 ms. Mobile telephony technology applied to vehicular environments is currently in addition to the DSRC networks, the only one that is fully developed, operational, and available for all types of applications. While DSRC networks focus primarily on short/medium-range V2V communications, data exchange via mobile telephony allows operations with the infrastructure and even with other vehicles when DSRC networks are unavailable. Additionally, its implementation in road and automotive environments is much more deployed than any other technology and, in some cases, mobile telephony is used as the only system for all types of communications. However, there are two clear limitations regarding the use of mobile telephony-based communications in vehicular environments. On the one hand, given the characteristics of cell-based communications, a massive deployment in vehicles could lead to saturation of communications in areas with few nodes in the infrastructure. In C-ITS communication, this technology was used by the European Commission and the USA before being replaced by DSRC and Vehicle-To-Everything (V2X) (3GPP) [40].

2.4.2.3 5G Mobile Telephony

the European Commission defined the 5G Public Private Partnership (PPP) within the 2020 program for the purpose of developing 5G technology and the Internet of the future. 5G technology is expected to be a hybrid of 3G,4G, and WiFi-WLAN technology, which, when applied to the transport sector, unifies the advantages of mobile telephony and DSRC, including direct communication between multihop devices and device-to-device [79].The preliminary 5G technical capabilities are:

- Capacity: 50 to 100 times 4G.
- Quality of Service: Ultra reliable communication for many critical applications.
- Transmission time:50-100 times faster than 4G LTE.
- Latency: 1 ms.
- Bidirectional: Direct communications between devices (Device to Device (D2D)). In case of road transport, V2V.
- Broadcast: Enabled.

This 3GPP-based technology is better known as C-V2X. The countries that use the most technology in C-ITS are the USA and China. [61].

2.4.2.4 RFID

Radio-Frequency Identification (RFID) is a data storage and retrieval system that uses devices called tags, transponders, or RFID tags. The fundamental purpose of RFID technology is to transmit the identity of an object (similar to a unique serial number) using wireless data transmission. Depending on the frequencies used in RFID systems, cost, range, and applications are different. Systems employing low frequencies also have low costs, but also low usage distance. Those employing higher frequencies provide longer reading distances and faster read speeds. Thus, low frequency is commonly used for animal identification, goods tracking, car key for vehicles, pallet tracking and packaging, and tracking of trucks and trailers on shipments.Another important application of RFID in transport applications is the electronic toll collection. This technology has been used in many deployments in Spain, Mexico, USA, France, and Germany. In this case,a RFID tag installed in each vehicle connects and exchanges information with the infrastructure when the car enters onto the ramp of a highway,charging the costs of this access automatically. Another common application is the use of RFID in smart keys, avail-able in models from most car manufacturers. In this case, the key is replaced by a card with an active

RFID circuit that allows the car to recognize the presence of the key within 1 m of the sensor.Another proposed application is the use of RFID for road traffic signals (Road Beacon System). It is based on the use of floor-embedded RFID transponders (radio beacons) that are read by a vehicle-carrying unit (OBU) that filters the various traffic signals, warning the driver if necessary. Electronic toll collection is another important use of RFID in transportation. This technology has been used in many deployments in Spain, Mexico, the USA, France, and Germany. When a car drives onto a highway ramp, an RFID tag in the car connects to the infrastructure and shares information with it. The access fee is then automatically charged [40]. For now, several countries in North USA, Latin America, Asia Pacific, Middle East, and Africa have used this technology [73].

2.4.2.5 Bluetooth

Bluetooth (Bluetooth, 2016) is the specification for so-called Wireless Personal Area Network (WPAN) that enables data transmission between different devices through a radio frequency link in the 2.4 GHz band.The Bluetooth specification has been designed to enable the development of low-cost, low-power, and short-range communications devices (up to100 m).The reason for the creation of this specification is to obtain a single digital wireless protocol that is capable of interconnecting multiple devices Vehicular Communications very simply and solving classic problems such as the synchronization between them. Similar to WiFi networks, Bluetooth uses Frequency Hopping Spread Spectrum (FHSS) technology for data transmission, using the 2.4 GHz band. Bluetooth networks support up to 1 Mbps band rate in basic transfer mode and 3 Mbps in the enhanced data transfer mode.The normal operation of Bluetooth networks follows the master-slave scheme. One of the devices in the network, called master, provides the reference values for the connection, such as synchronization and frequency hopping sequence. The other devices in the network are called slaves and exchange data with the master. This network consisting of short-range devices is called a piconet (Net). One of the fundamental characteristics of this type of network is that the information can circulate between the master and any other device; however, different devices can change their roles among themselves and, in this way, a master can be transformed into a slave and vice versa, depending on the needs of applications that support communications.The Bluetooth specification also allows the interconnection of two or more piconets, thus forming a scatternet, in which some of the slave devices act as gateways between two networks, being master in one and slave in another.

2.5 Vehicular Communication Security

The fundamental security properties of the Vehicular Network are essentially identical to those of digital communication networks in general [97] [84]. These basic properties include: confidentiality, integrity, availability, authentication - identification, and non-repudiation.

2.5.1 Confidentiality

Confidentiality is paramount in maintaining data security in a communication network. In the field of ITS, confidentiality will ensure that essential vehicle data does not leak to unauthorized parties [25]. For example, two intelligent vehicles can exchange their position and speed information to maintain a safe distance. Confidentiality allows these two ITS components to exchange information through unsafe channels that are prone to eavesdropping by third parties. One example of security is steganography technology, which enables data to be disguised when transmitted. At the same time, there are malicious parties who have access to the same communication channel [19, 57]. Maintaining confidentiality in ITS is not easy. Because ITS involves a variety of different equipment, such as smart phones, sophisticated smart vehicles, ITS Stations, simple IoT devices, etc. So maintaining confidentiality across these different spectrum is a challenge.

2.5.2 Integrity

At ITS, The integrity of data communications, infrastructure, traffic controllers, etc., must always be maintained to ensure proper operation. If not, for example: When two intelligent vehicles send each other a message containing their respective positions, an attack vehicle performs a Man In The Middle Attack (MITM) attack and modifies the position message. Then the message becomes a reference for a legitimate vehicle and will result in an accident because this wrong message is used as a reference for making decisions. Another example is a GPS spoofing attack [98]; the attacker broadcasts the wrong GPS coordinates so that the victim's vehicle changes ITS travel route in the wrong direction. Another attack that affects integrity is the Sybil attack. Sybil attacks occur when an attacker uses a collection of available pseudonyms and uses them to disrupt the system. Periodically the attacker will broadcast V2X messages and sign them with different and valid pseudonyms making it difficult for Misbehavior Authority (MA) to detect them [46].

2.5.3 Availability

The availability of ITS devices that operate typically and are interconnected is essential to ensure passenger safety. The attack that is very influential on the availability aspect is Denial of Service (DoS) [59]. DoS attack is one of the attacks that cause congestion on ITS devices. Attacks in the availability aspect are dangerous because many ITS components require real-time operations. If these components do not work, the communication will be jammed, resulting in fatal traffic events.

2.5.4 Authentication - Identification

It is imperative for ITS to identify and verify the parties involved in communication and data exchange. To perform identification and authentication, use Message Authentication Codes (MACs) [25] or challenge-response protocols, which allow for verification of the sender. However, there is a weakness in its use, which is more or less causing additional computational overhead in the system, so the response time will increase and affect the effectiveness of communication [14, 54]. In addition to using MACs, using pseudonyms instead of vehicle identities to enhance privacy has

been an option in various studies to identify and authenticate in VNs [55]. However, this technique's problem of computational overhead still exists because pseudonyms still need to be authenticated by a trusted authority, which still requires additional time and costs to process safety messages on ITS.

2.5.5 Non-Repudiation

Non-repudiation is a service that ensures that the sender cannot deny that his message has been sent and the integrity of the message remains intact and maintained. Or in other words, non-repudiation is the principle of indisputability of a transaction. Non-repudiaton is an important service in ITS related to communication within VN. In practice, an attacker will not be able to deny that the detected malicious message came from him. Most research in non-repudiation deals with pseudonym verification by third parties. Regional third parties can be in the form of physical infrastructure or groupings such as commanding authorities [20, 25, 59]. Setting the balance between non-repudiation and privacy is a challenge in realizing ITS security itself [29].

The fulfillment of the three principles: confidentiality, integrity, and availability, results in the achievement of system security goals, while authentication - identification and non-repudiation are the basis in this context, as illustrated in the figure 2.5.

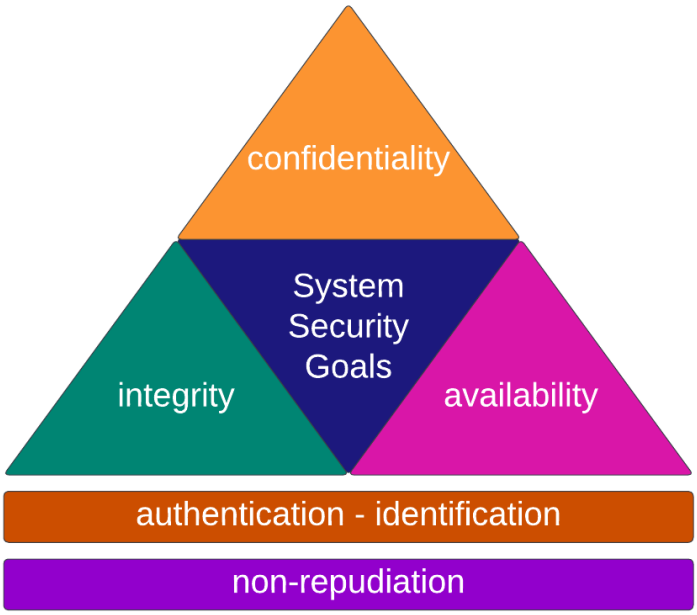


Figure 2.5: Vehicular Network Security Properties

2.5.6 Vehicular Network Security Issues

VN confront many of the same issues as wireless network systems; hence, assaults like as denial-of-service, Sybil, and replay attacks are also prevalent in VN [59, 82]. This is because VN can only work with wireless network technology. VN's worries about data protection are also relevant to other parts of ITS. When it comes to ITS security, it's important to work from the top down, since flaws in one part of the system can easily spread to other parts and pose a serious threat to the whole system's integrity.

The VN has also been evaluated in terms of maintaining the confidentiality of the participants' personal information and preventing unauthorized third-party access to the participants' important credentials. The majority of study on the problem of privacy in VN has focused on the use of pseudonyms, also known as fictitious names or pseudo IDs, to safeguard the privacy of travelers while maintaining solid non-repudiation procedures for ITS [14, 54, 55, 70].

The exchange of information between vehicles in VN is by exchanging a BSM, which contains information on the vehicle ID number, position based on GPS, speed, acceleration, direction, and so on. BSM is sent to each other between vehicles. However, any digital communication system will have security threats or loopholes, which irresponsible users can exploit for personal gain or cause chaos. Generally speaking, in the ITS system, there are two types of attackers,i.e., external attackers and internal attackers. External attackers are vehicles/users who do not have credentials in a V2V communication network. Meanwhile, the attacker from within is the vehicle/user that already has credentials in the V2V communication network and is still trying to carry out attacks. Attackers from outside can be overcome with Public Key Infrastructurs (PKI), while attackers from within can be overcome by misbehavior detection, seen figure 2.6.

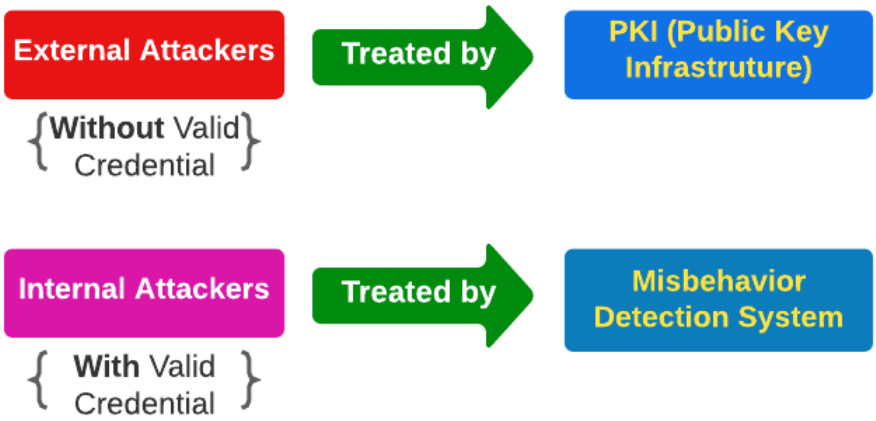


Figure 2.6: Type of Attacker in Vehicular Network

2.6 Related Works

Every research in detecting cyber attacks on the Vehicular Network has added to the scientific treasures that are useful for developing C-ITS in the future. The following sub-chapter is some of the projects/research the authors refer to. The basis for determining referrals is the similarity of case studies/research environment, methodology, problem, and tools used.

2.6.1 Project

2.6.1.1 D2H-IDS

D2H-IDS, is a hybrid attack/misbehavior detection system proposed by Aloqaily et al. [4]. It is based on the use of DBN and Iterative Dichotomiser 3 Decision Tree. The ID3 feature was used to select and attack categorization, while the Deep-Belief function was utilized to reduce data dimensionality. They illustrated the efficiency of their approach through ten simulations based on genuine cyber-security attack situations on Smart Vehicle (SV). In this study, 3 different DBN strategies were compared: DBN1-IDS, which is a standard of DBN, DBN2-IDS, that was developed and adapted from [94], and DBN3-IDS, which is sourced from [99]. The normal vehicle movement data was generated using the NS-3 application and NSL-KDD attack data-set during the preprocessing step. Then The data was passed through DBN IDS and ID3 algorithms using Matlab. The built data-set includes DoS, Remote to Local (R2L), User to Root (U2R), and Probing (Probe) attacks.

2.6.1.2 Spoof Attack Detection on Electric Vehicle (EV)

This research conducted by Kosmanos et al [50], focuses on detecting spoofing attacks on Dynamic Wireless Charging(DWC) Electric Vehicles with Mobile Energy Disseminators (MEDs). They use a supervised Machine Learning algorithm (KNN and Random Forest (RF)) as the basis for the IDS. In addition to ML, additional features of Position Verification using Relative Speed (PVRS (PVRS) are also used in the detection system. PVRS is a novel statistic used by the IDS, and it appears to impact categorization outcomes substantially. The Physical (PHY) layer is where signals are exchanged, and PVRS compares the observed distance between two communicating nodes with the estimated distance using the relative speed value. Using both supervised Machine Learning (ML) algorithms, the effect of this new PVRS metric on the performance of the suggested probabilistic IDS was a 6% improvement in accuracy.

2.6.1.3 Deep Learning LSTM-GAN

Rasheed et al. [76] pay more attention to attacks that involve the injection of fake data on Connected Autonomous Vehicles (CAV) which will result in a vehicle mistaking its distance from another vehicle. Based on the LSTM-Generative Adversarial Network (GAN), they propose a Deep Neural Network (DNN) attack detection approach. They name it the New Deep Reinforcement Learning (NDRL) algorithm structure, which would result in a safe dynamic system for Autonomous Vehicle (AVh) control. The focus of this system is on superior autonomous vehicle control, which allows it to keep a safe distance from other vehicles while regulating its speed. AVh sensor data and AVh beacon signals are the most significant infrastructure requirements.

2.6.1.4 Machine Learning and Dempster-Shafer

Gyawali et al suggested an established misbehavior detection framework formulated on a hybrid collaborative ML and reputation misbehavior disclosure methods [27].

In their research, they developed a data-set based on practical vehicular network circumstances to test false alerts and position falsification and then evaluate it using various ML techniques. KNN, Logistic Regression Model, Decision Tree Classifier, Bootstrap Aggregation, and Random Forest were the models they used. Bootstrap Aggregation and Random Forest provided the greatest outcomes concerning *Precision*, *Recall*, and F_1-*Score* based on their simulation results. They employ the initial version of the Vehicular Reference Misbehavior (VeReMi) in addition to the data-set [93] as a benchmark for location verification systems. They find that their technique was better than the VeReMi data-set for 30% attacker density for Eventual Stop, Random, Random Offset, and Constant Position forms of attacks. The simulation was running on the VEINS 4.7 framework, which includes SUMO with the LuST scenario and OMNET++.

2.6.1.5 F^{2}MD

Framework For Misbehavior Detection (F^{2}MD) is an additional framework for VEINS (Vehicle in Network Simulation) the preexisting and widely used Vehicular Network simulation applications. It created by Kamel et al [45] for simulating real-time misbehavior detection in vehicular networks. F^{2}MD also has ML modeling which consists of an offline phase and an online phase. The offline phase is used at the training model stage, while the online phase runs on the HTTP Server by calling the ML model classifier resulting from the offline phase to be used as a misbehavior detector. The ML classifier models used are SVM, Mutli-Layer Perceptron (MLP), and Long Short-Term Memory (LSTM). Its dataset is a development of the first version of the VeReMi dataset [93]. In this framework, the ML model that provides the best accuracy is LSTM and it should be noted, in this research, Kamel et al did not focus on comparing ML models but focused on the functionality of the framework itself. Further explanation about F^{2}MD will be explained in chapter 4 Real Time Implementation.

2.6.1.6 SerIoT

The SerIoT project is one of the projects that focus on C-ITS communication security. The project aims to protect the main network on IoT devices, provide solutions to detect misbehavior, mitigate them through the creation of alternative routes involving specialized devices such as honeypots, and reduce the impact of automated attacks on autonomous vehicles [31]. In this study Hidalgo et al. evaluated the system using a real vehicle in the Tecnalia Lab, to facilitate experiments and obtain a realistic simulation environment. They show that the system can detect and mitigate misbehavior quite quickly. The detection system used is based on a Graph Neural Network (GNN) consisting of MLP and Node DNN. This research takes into account only one type of attack which is the DoS attack. From the experimental results, the system can accurately detect and deliver early warnings of DoS attacks at an average time of 3.27 seconds and a standard deviation of fewer than 3 seconds.

2.6.1.7 Invariant State Detection

Another study by Zhou et al. focused on security in platoon vehicle communication systems, which are inherently vulnerable to cyber-attacks. In this research, they offer a new detection system using invariant set state using physical properties model and system control strategy [100].The invariant state based on the Distributed Information-weighted Set-membership Filter (DWSMF) and weighted Minkowski sum (WMS). The use of Software-Defined Networking (SDN) in the platoon vehicle communication network is the main approach. The types of attacks that were considered in this research were Message Falsification Attacks (MFAs) and

sensor spoofing attacks that were simulated using SUMO and OMNET++ applications. In this work, two detection methods were compared: (i) attributed to the set of constant states from ISWSM, and (ii) SMF-based attack detection (IEDCM).

2.6.2 Project Comparison

No research is ever going to be 100% accurate, of course,there are various aspects that the research can still develop. We can review the comparison of research projects by looking at table 2.1. These research projects have something in common: the use of the ITS-G5 802.11p protocol in the simulations carried out. All of these project research use ML as the basis for IDS, except for research no. 7, which uses an invariant state. Interestingly, almost every study focuses on a few types of attacks, except the project no.5. It means something in common; almost all projects have not been tested with various types of attacks at one time. Examples of various other types of attacks will be described in sub-chapter **??**. Of course, this is an excellent opportunity to be developed to complement C-ITS, which is guaranteed safe.

Table 2.1: Comparison of Related Work Research

No	Project	Case Study	Problematic	Solution Framework	Accomplishment	Limitation
1.	D2H-IDS	SV Communication	Detection of attacker	IDS based on DBN	The system can detect *DoS, R2L, U2R), and Probe attacks*	The system is proven to detect only four types of attacks.
2.	Spoof Attack Detection on EV	EV Communication	Spoof attack detection	IDS based on , RF,and PVRS	The system can detect *Spoof attacks.*	The system is proven to detect only one type of attack and has not yet been proven in a larger Vehicular Network environment
3.	Deep Learning LSTM-GAN	CAV	Faulty data attack detection	NDRL based on LSTM-GAN	The system can identify attacks based on *faulty data (position and speed).*	The system is proven to detect only attacks on vehicle position and speed and has not yet been proven in a larger Vehicular Network environment.
4.	ML and Dempster-Shafer	V2X communication	Internal attack detection	KNN, LR, DT, RF, Bagging, and Dempster-Shafer	The system can detect *false alarm generation and position falsification attackers.*	The system is proven to detect only two types of attacks.
5.	F^2MD	V2X communication	Internal attack detection	SVM, MLP, LSTM, and plausibility check	The system can detect all misbehavior in the *VeRemi extension* Dataset, based on plausibility and consistency checks. The ML system can detect *Delayed Messages, Data Replay, Grid Sybil, Constant Position Offset, Constant Speed, and Disruptive* Attacks.	The system depends on the plausibility check and The ML system is proven to detect only six types of attacks.
6.	SerIoT	CAV	Multiple cyber-threats detection	GNN and MLP	The system can identify *DoS, Port scans, and SSL attacks.*	The system is proven to detect only three types of attacks and has not yet been proven in a larger Vehicular Network environment.
7.	Invariant State Detection	Vehicular Platoon	Detection of attacker	Invariant state based on DWSMF and WMS	The system is capable of detecting *MFAs and spoofing attacks.*	The system is proven to detect only two types of attacks and has not yet been proven in a larger Vehicular Network environment.

2.7 Misbehavior on Vehicular Network

Currently, the Veremi Extension dataset has the most comprehensive information on misbehavior on Vehicular Network. Veremi is a dataset for the evaluation of misbehavior detection mechanisms for VANETs. The initial dataset contains a number of simple attacks: the idea of this dataset release is not just to provide a baseline for the comparison of detection mechanisms, but also to serve as a starting point for more complex attacks. Rens van der Heijden of the Institute of Distributed Systems at Ulm University, Germany, mostly compiled the data-set in 2018 [93]. In 2020 Joseph Kamel et al. published the VeReMi data-set to become a *VeReMi Extension* [48] which is referred also as VeReMi, from now to the rest of paper.

Misbehavior, as an intrusion in vehicle-to-vehicle communication, can be classified into **malfunctions** and **attackers**. Malfunctions can be caused by damage to equipment on the vehicle such as sensors, OBU, etc. so that the message sent by the vehicle becomes incorrect. In training and detection tasks, malfunctions is considered as an attacker, since the disruption they cause is almost similar. Meanwhile, an attack is the intentional act of an attacker vehicle to manipulate the message sent [47]. The list below provides a more detailed explanation [48].

Malfunctions

1. **Constant Position**
   This is one of the errors of the positioning system, for example, in GPS data. This error causes position data sent to other vehicles to show the same value occasionally, even though the vehicle has changed positions.

Lon_t	$\triangleq$	*Longitude at time t*
Lat_t	$\triangleq$	*Latitude at time t*

   The constant position of the malfunctioned vehicle can determine as :
   $Lon_t = Lon_c$
   $Lat_t = Lat_c$

2. **Constant Position Offset**
   In this case, the Constant offset position is added every time the vehicle sends the factual position information.

ΔLon_c	$\triangleq$	*Constant offset Longitude*
ΔLat_c	$\triangleq$	*Constant offset Latitude*

   The constant offset position can determine as :
   $Lon_t = Lon_t + \Delta Lon_c$
   $Lat_t = Lat_t + \Delta Lat_c$

3. **Constant Speed**
   In sending vehicle speed information, errors can occur due to an OBU error or physical sensor damage.

Vx_t	$\triangleq$	*Speed X component at time t*
Vy_t	$\triangleq$	*Speed Y component at time t*

   In the case of Constant Speed, it can be described by the following formula:
   $Vx_t = Vx_c$
   $Vy_t = Vy_c$

4. **Constant Speed Offset**
   The Constant speed offset is added every time the vehicle sends the factual

ΔVx_c	$\triangleq$	*Constant offset Speed X*
ΔVy_c	$\triangleq$	*Constant offset Speed Y*

speed information.
The constant offset position can determine as :
$Vx_t = Vx_t + \Delta Vx_c$
$Vy_t = Vy_t + \Delta Vy_c$

5. **Delayed Messages**
   It can be the result of a high network overhead or an inexpensive or sluggish on-board processor. These signals are issued with a delay Δt from reality while having all the necessary facts and information.

6. **Random Position**
   Random Position occurs if the position info will show a random value every time step.
   $Lon_t = U([Lon_{min}, Lon_{max}])$
   $Lat_t = U([Lat_{min}, Lat_{max}])$
   The simulation playground's size determines the minimum and maximum values.

7. **Random Position Offset**
   The genuine position in this instance will be supplemented with a random position offset.
   $Lon_t = Lon_t + U([-Lon_c, Lon_c])$
   $Lat_t = Lat_t + U([-Lat_c, Lat_c])$

8. **Random Speed**
   In this instance, the vehicle's speed information will display a random value at each time step.
   $Vx_t = U([Vx_{min}, Vx_{max}])$
   $Vy_t = U([Vy_{min}, Vy_{max}])$

9. **Random Speed Offset**
   In this case, a random speed offset will be added to the actual speed.
   $Vx_t = Vx_t + U([-Vx_c, Vx_c])$
   $Vy_t = Vy_t + U([-Vy_c, Vy_c])$

Attacker

10. **Data Replay**
    An attacker vehicle is sending information previously received from a specific target neighbor. The replayed data is signed using the attacker's certificate. The target vehicle will feel that the data received is from a legitimate vehicle when it comes from the attacker.
11. **Data Replay Sybil**
    This is the same technique, i.e. data replay, but done in Sybil mode. That is, the attacker changes the identity of each subsequent target to prevent detection. This will result in legitimate vehicles receiving incorrect messages regarding the condition of other vehicles in the vicinity. At the same time, the attacker's vehicle will be difficult to detect.
12. **Disruptive**
    An attacker vehicle is sending information replay previously received from random neighbors. In this case, the attacker's vehicle transmits a replay of information using random fake data. This will result in the target vehicle. This technique allows the attacker's vehicle to wreak havoc on the vehicular network.

13. **Denial of Service (DoS)**
    DoS attacks involve a vehicle transmitting messages at a rate that exceeds the ceiling established by the relevant IEEE or ETSI standards.
14. **DoS Disruptive**
    This is a combined attack from DoS and Disruptive, with the same goals as a DoS attack. The attack vehicle will send as much false information as possible to the legitimate vehicle.
15. **DoS Disruptive Sybil**
    This is the same attack as DoS Disruptive, but the real identity of the attacker is hidden so that the attacker will be challenging to detect.
16. **DoS Random**
    DoS attacks such as DoS Random use messages fields with all values set to random numbers. There's a chance that it's a plan to overburden the network and block the transmission of sincere messages.
17. **DoS Random Sybil**
    DoS Random, which is carried out in Sybil mode, and the attacker changes its identity with each message sent in order to evade detection.
18. **Eventual Stop**
    Attacks known as eventual stops include a vehicle simulating a sudden stop by setting the speed numbers to zero while freezing the location values.
19. **Grid Sybil**
    The goal of the attack known as Grid Sybil is to simulate heavy traffic. By keeping a new identity and the proper message frequency for each fake vehicle, the attacker creates a grid of false vehicles at the desired location.

2.8 Conclusion

Vehicular Communication consists of V2V, V2I, and V2X. V2V communication protocols will enhance security performance by allowing all nearby vehicles to communicate. The V2I communication model enables vehicles in motion to communicate with the road system. V2I sensors can collect data on the infrastructure and provide real-time information to drivers regarding road conditions, traffic congestion, potential accidents, the presence of work sites, and parking availability. The V2X represents a generalization of the previously discussed V2V and V2I communication paradigms. One of the main goals of V2X technology is to promote effective communication methods between automobiles and pedestrians in order to reduce accidents, which can sometimes be fatal.

C-ITS messages will be transmitted for a wide range of services, in different transport situations. End-users do not care about the specific communication technology used to transmit C-ITS messages, but will expect to receive all information on traffic and safety conditions seamlessly. This can only be achieved through Intelligent Vehicles a so called hybrid communication approach, i.e., by combining complementary communication technologies. Currently, the best option for the hybrid communication mix is a combination of IEEE802.11p/ETSI ITS-G5 and next-generation cellular networks (5G). This ensures the best possible support for deployment of all Day 1 C-ITS services. It combines low latency of ETSI ITS-G5 for time critical safety-related C-ITS messages with wide geographical coverage and access to large user groups of existing cellular networks.

Basic properties to achieve system security goals must include confidentiality, integrity, availability, authentication - identification, and non-repudiation. A flaw in any of these characteristics can make the system susceptible to attack.

There are two different kinds of attackers in the ITS system: external attackers and internal attackers. Vehicles or users without credentials in a V2V communication network are considered external attackers. The vehicle or user that already has access to the V2V communication network and is attempting to launch attacks is the attacker from within.

There have been many studies on security issues in Vehicular Network. Research focus varies such as on survival analysis, the injection of fake data, IoT devices, platoon vehicle communication, hybrid collaborative ML, hybrid misbehavior detection system, etc.
A complete study reviewing the types of attacks on VN is the Veremi Extension study which is supported by the use of the F^2MD application. Several misbehavior variations are reviewed in this study, as well as techniques for detecting them. However, the detection technique is not the main focus of this research.

Comparison of each project in related work more clearly illustrates the potential we can develop. Most projects focus on dealing with only a few types of attacks. Meanwhile, the types of anomalies in Vecuhilar Network communication will continue to grow. Several types of attacks or misbehaviors that are rarely handled by those projects are Delayed Messages, Random Position, Random Position Offset, Random Speed, Random Speed Offset, Data Replay Sybil, Disruptive, DoS Disruptive, DoS Disruptive Sybil, DoS Random, DoS Random Sybil, Eventual Stop, and the Sybil Grid.

Given the security issues that occur in the vehicular network, it is necessary to have an attack detection system that can identify potential attacks early on.

Chapter 3

Misbehavior Detection System

Contents

3.1 Introduction

The system to detect misbehavior on the communication network at C-ITS is basically an Intrusion Detection System (IDS), as in computer networks. Network traffic is analyzed by an IDS to spot any malicious traffic types. IDS are frequently categorized based on how they identify assaults [1]. They can be broadly split into two groups:

1. Signature-based detection
2. Anomaly-based detection

When new traffic is compared to known threats by *Signature-based detection* (another reference call it misuse-based [85]), a warning is raised. The majority of antivirus programs and Signature-based IDS operate pretty similarly [6]. They keep a database of the signatures that might indicate a specific kind of attack, and they compare incoming traffic to those signatures. This strategy generally works well, but occasionally we come across attacks that are either brand-new or have been designed purposefully to not match known attack signatures. One of the major limitations of this approach is how many signature-based systems only use their signature database to find attacks. The attack might not be detected at all if we don't have a signature for it. By searching the new traffic for any divergence from the usual,
Anomaly-based detection can identify it as such malicious and flag it as abnormal. A massive amount of data must be used to create a model for what is typical and abnormal in order to successfully detect new assaults. Usually, this detection begins by establishing a baseline of the ordinary network activity and traffic. In order to identify patterns that are not present in the traffic regularly, they can compare the current status of the network's traffic to this baseline. Such techniques can be quite effective when trying to find new assaults or attacks that have been purposefully put together to bypass IDS. On the other hand, compared to IDS based on signatures, anomaly-based IDS may also produce a higher amount of false positives. As with legal activity that results in odd traffic patterns, the IDS may interpret variations in network traffic from what was present when we collected our baseline as signs of an attack [1, 6]. MisBehavior Detection System (MDS) is the same as IDS. We use the term misbehavior because traffic data anomalies in VN are not always in the form of attacks but can also malfunction, as explained in sub-chapter 2.7.

Regarding to definition of MDS and and the current work in the field of C-ITS Security, we consider a wireless communication Vehicular Network (VN) (Figure 3.1), where each mobile node (vehicle) is assembled with an OBU to exchange messages with neighbors mobile nodes or with fixed nodes (Road Side Unit (RSU)). The sent messages are in the form of a **Basic Safety Message (BSM)**, which contains information on the vehicle ID, position, speed, acceleration, direction, and so on.

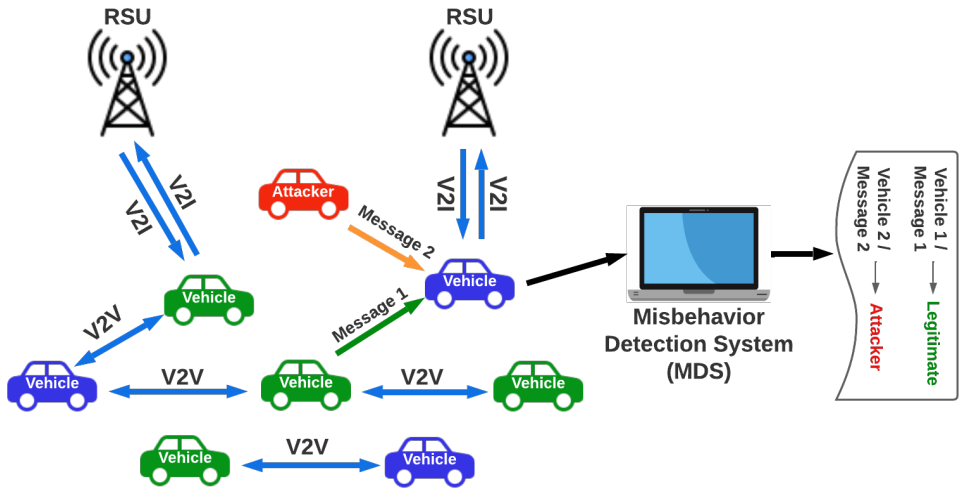


Figure 3.1: MDS System Model on Vehicular Network

3.2 Machine Learning

The primary purpose of an IDS in VN is to distinguish between normal behavior and abnormal behavior of a vehicle and sound an alarm if an attack is detected. Misbehavior detection methods in VN generally revolve around three techniques: *signature, specification, and anomaly* detection system. *The signature detection system* stores attack behavior and the normal behavior of a vehicle in a database. New behavior will be compared with the behavior in the database to determine whether the behavior is normal or an attack. *The specification detection system* defines a set of conditions every vehicle in VN must follow as a protocol. If a new behavior is not in accordance with the protocol, then the behavior can be categorized as an attack. *The Anomaly detection systems* create a model for normal and attacker behavior, which is where ML comes in handy [28].
In the cyber security world, a huge amount of information data is obtained from network sensors, logs, agent endpoints, and others. The data obtained is extensive in volume, speed, and variation, so it is included in Big Data. Big Data has its challenges in analysis. Classical techniques in attack detection systems can't keep up. On the other hand, Machine Learning (ML), as part of Artificial Intelligent (AI), is considered capable enough to overcome this problem, according to a survey by Vinayakumar et al [77].

Our proposed ML models were picked after carefully reviewing the literature on the subject of how best to employ ML in the detection of cyber-attacks on V2V networks. Therefore, we decided to use the following ML model to implement our strategy.

3.2.1 Decision Tree Model

3.2.1.1 Random Forest (RF)

RF is one of the methods in the Decision Tree, which is a tree-shaped flow chart that has a root node that is used to collect data. At the root node, there is an inner node that contains questions about data and a leaf node that is used to make decisions. Basic algorithm of the RF introduced by Ho et al [33]. Like its name suggests, a random forest is made up of numerous independent decision trees that work together as a set. Every tree in the random forest spits out a class forecast, and the classification that receives the most votes becomes the prediction made by

the model [13]. This algorithm is beneficial in classifying data, especially if the data is large, also easy to use and flexible. In research by Gyawali et al [28], it was found that RF has the best performance detecting attacks or misbehavior on ITS.
How the RF algorithm works are described in the following steps, according to figure 3.2:

(1) The algorithm selects a random sample from the provided dataset.

(2) Make a decision tree for each selected example. Then the prediction results will obtain from each made decision tree.

(3) The voting process is carried out for each prediction result. For classification problems, use the mode (the value that occurs most often), while for regression problems, will use the mean (average value).

(4) The algorithm will choose the prediction result with the most votes as the final prediction.

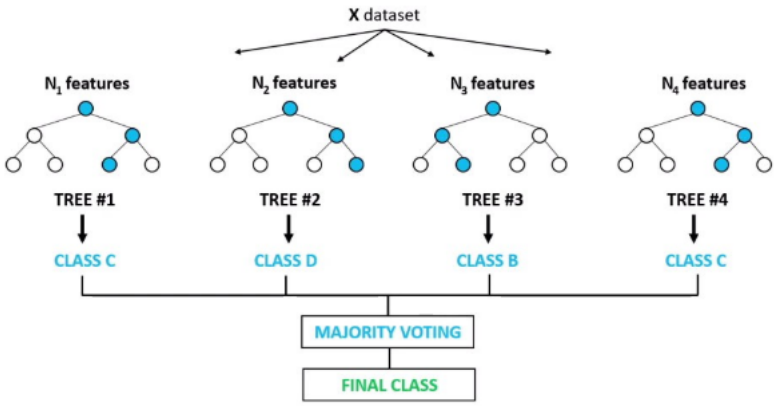


Figure 3.2: Random Forest Diagram [18]

3.2.2 Deep Learning Model

3.2.2.1 Deep Belief Network (DBN)

Geoffrey Hinton published a paper that introduced a neural network variant called DBN in 2006 [32]. This paper is the beginning of the emergence of the term deep learning, to distinguish conventional (single layer) neural network architecture from multi-layer neural network architecture. DBN is a ML that uses the Deep Learning method. DBN has undirected connections between hidden layers composed of stacked Restricted Boltzmann Machines (RBM). DBN will use labeled data because the dataset used for the training process has clearly included the label / data type, and also to synchronize it with other ML model inputs. The following describes the operational pipeline for DBN, see figure 3.3 [43]:

(1) The Greedy learning algorithm will be used by DBN to pre-train. Using a layer-by-layer methodology, the greedy learning method is used to learn the top-down generating weights. The correlation between variables in one layer and variables in the layer above is determined by these generative weights.

(2) The top two hidden layers will be subjected to many Gibbs sampling iterations by DBN. The RBM is defined by the top two hidden layers. Consequently, this step is actually taking a sample from it.

(3) After that, run a single ancestral sampling pass through the rest of the model to create a sample from the units that are visible.

(4) In order to infer the values of the latent variables in each layer, DBN will employ a single bottom-up pass. Greedy pretraining starts with an observed data vector in the lowest layer. It then adjusts the generative weights in the other direction.

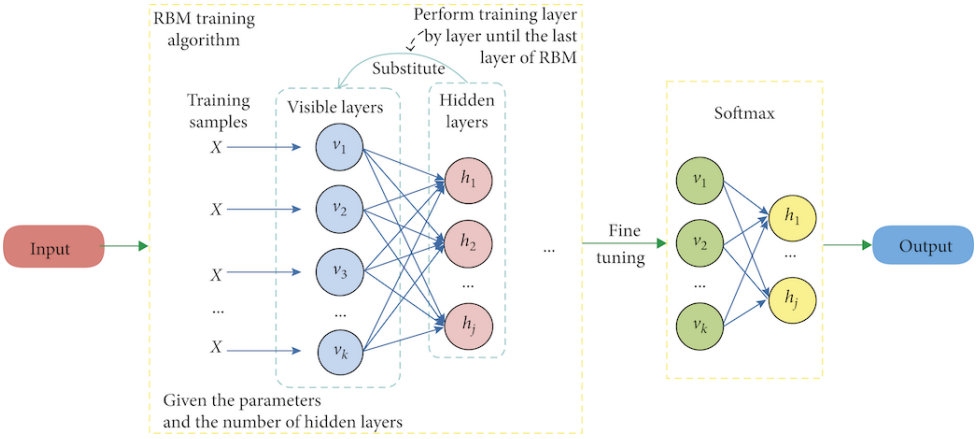


Figure 3.3: Deep Belief Network Diagram [43]

In research by [4] DBN is used in conjunction with the ID3 Decision Tree to provide maximum performance for detecting intrusions in communication between vehicles. Whereas in this study we will only use DBN to detect attacks on ITS and compare it with other ML models. We took the DBN algorithm from [3] and made some modifications so that the DBN algorithm could process the VeReMi dataset according to the preprocessing results.

3.2.2.2 Long Short-Term Memory (LSTM)

LSTM was introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997. The vanishing gradient issue plagues traditional RNNs was the driving force for the creation of the LSTM architecture. The gradient is said to disappear because it becomes smaller until the final layer leaves the weight value unchanged, which results in the gradient never improving or converging. On the other hand, the expanding gradient results from the increasing gradient, which increases the weight values in numerous layers, causing the optimization procedure to diverge [34]. LSTM is a type of Recurrent Neural Network (RNN) where modifications are made to the RNN by adding a memory cell or a cell state that can store information for a long period of time. In addition to the cell state, LSTM uses three processing gates namely input gate, output gate, and forge gate state. Broadly speaking, how the LSTM works is as follows, see figure 3.4 [21]:

(1) Information that is no longer necessary or of low significance for the processed case will be eliminated using the sigmoid function in the forget gate section.

(2) The input gate component handles the information processing. By employing the sigmoid activation function, this procedure will sort and identify specific information that will be updated to the cell state section. The *tanh* activation function, which will be added to the cell state section, is also used in this phase to create a new candidate vector.

(3) Next, change the value of the old cell state to the new cell state. (4)The cell state is set to *tanh* in the output gate component and the algorithm executes

sigmoid to generate an output value in the hidden state. Before moving on to the next phase, the two activation results are multiplied after producing the sigmoid output value and the *tanh* output value.

(4) The LSTM method will then produce a categorization value based on the results of the complete calculation.

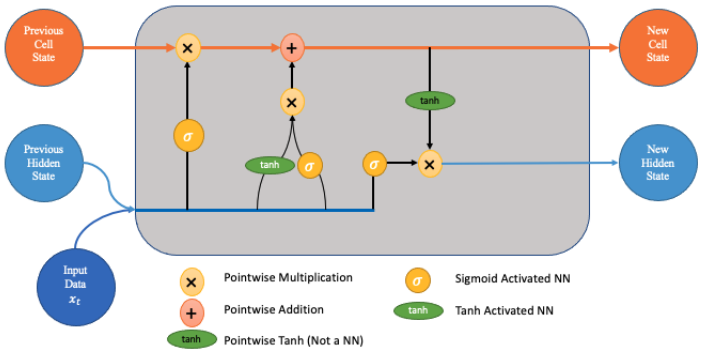


Figure 3.4: Long Short Term Memory Diagram [21]

In the simulation conducted in [45] it has been found that the best detection accuracy performance is obtained by LSTM, although it is also the slowest algorithm.

3.2.2.3 Gate Reccurent Unit (GRU)

GRU, as a newer generation of RNN, is a variant of the LSTM; however, it is claimed to be simpler and capable of producing the same results as the LSTM. GRUs employed the hidden state to transmit information instead of the cell state, and without hiring a forget gate like LSTM, GRU only uses two gates: the reset gate and the update gate. GRU was introduced by [16]. and [15]. In general, the way the GRU algorithm works is almost the same as the LSTM, but what distinguishes it is [72]:

(1) The update gate functions similarly to an LSTM's forget and input gates. It chooses what data to discard and what fresh data to include.

(2) Another gate used to determine how much old data to forget is the reset gate.

The GRU diagram works can refer to figure 3.5.

3.2.2.4 Residual Network (ResNet)

There is a limit to add the number of layers to a neural network. After that threshold is reached, the accuracy of the model starts to saturate and then degrades. This is due to the vanishing/exploding gradients, which causes the gradient to become 0 or too large. Thus when we increase the number of layers, the training and test error rate also increases. The Residual Network (ResNet) will handle this problem. ResNet consists of the residual units or blocks as the main component of the network (see figure 3.6.(A)). Residual network or ResNet in short was introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in their paper [30].
In a residual network, each layer feeds directly to the two or three levels behind it. The residual block is composed of two 33 convolution layers and an identity mapping, commonly referred to as a shortcut link. Following every convolution layer comes a batch normalization layer and a ReLU (Rectified Linear Unit) activation function. Between the identity mapping and the last batch normalization output, an element-wise addition is performed. The residual block allows researchers to construct and train a deeper network without the issue of gradients vanishing or inflating,

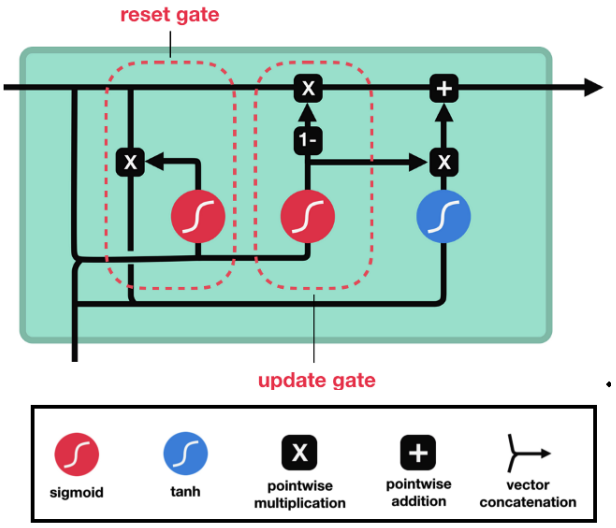


Figure 3.5: Gate Reccurent Unit Diagram [72]

see figure 3.6.(B). The identity mapping or the shortcut connection present in the residual block helps in the following ways: In case the layers in the normal flow do not learn anything, then the identity mapping basically copies the information from the earlier layers. This helps the neural network to perform better even with the deeper architecture. Using the residual network or ResNet can drastically improve the performance of neural networks despite having more layers [90].

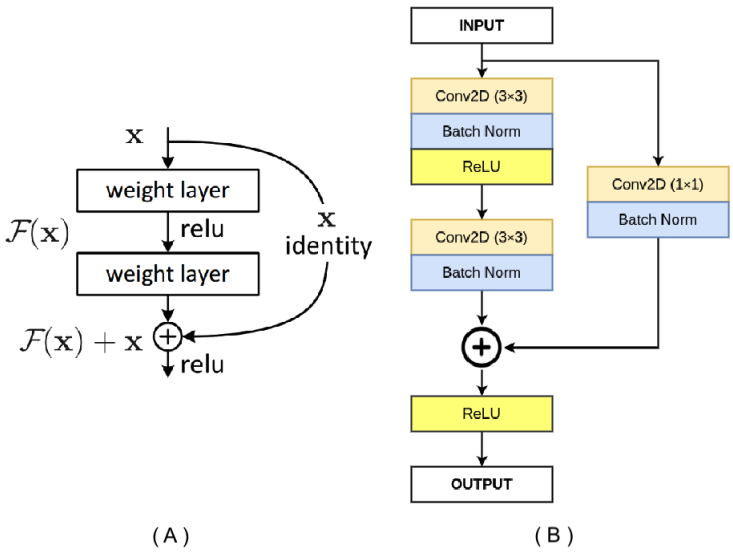


Figure 3.6: (A) Residual Block, (B) ResNet Diagram [90]

3.2.2.5 Mobile Network (MobileNet)

MobileNet was introduced by Andrew et al in their paper [35]. the MobileNet model is designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model. MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular

convolutions with the same depth in the nets. This results in lightweight deep neural networks. A depthwise separable convolution is made from two operations (see figure 3.7) :

(1) Depthwise convolution.

(2) Pointwise convolution.

MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.The main difference between MobileNet architecture and a traditional CNN instead of a single 3x3 convolution layer followed by the batch norm and ReLU, see figure 3.8.(B). Mobile Nets split the convolution into a 3x3 depth-wise convolution and a 1x1 pointwise convolution [74].

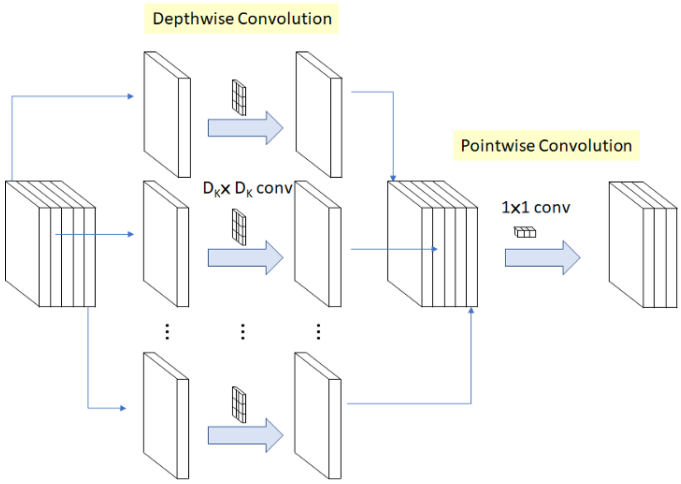


Figure 3.7: Deptwise Separable Convolution [74]

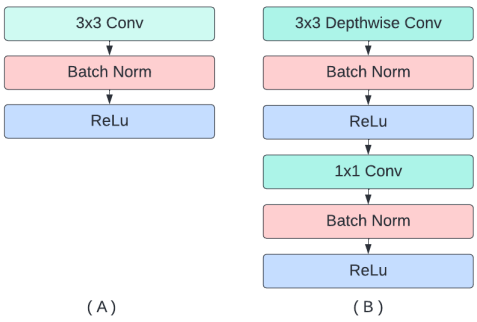


Figure 3.8: (A) Standard Convolutional Layer, (B) MobileNet Diagram [35]

3.2.3 Comparison

A brief comparison of the pros and cons of each ML model that will become the proposed system detection algorithm can be seen in table 3.1

3.2.4 Hyperparameter Optimization

In ML, hyperparameter optimization is challenging in selecting the appropriate set of hyperparameters for a learning algorithm. Hyperparameter optimization is the value

Table 3.1: Comparison of the ML Model

No	ML Model	PROS	CONS
1.	RF	Works well with unbalanced, high-dimensional, and huge data sets.	Data features must be predictive. Hard to interpret.
2.	DBN	Efficient with hidden layers. Has classification robustness.	Takes massive data to improve procedures. Expensive to train.
3.	LSTM	Can model sequence of data (i.e. time series). Typically better for short-term memory issues	Complex computation, gradient vanishing and exploding problems.
4.	GRU	Same with LSTM but less training parameters and memory.	Slow convergence and low learning efficiency for longer sequence dataset
5.	ResNet	Reduce vanishing gradient problem. Low error rate and high accuracy.	Computationally heavy. Need a lot of labelled data.
6.	MobileNet	Low latency. High classification accuracy. Fast training process.	Need a lot of labelled data.

for the parameters used to influence the learning process. In addition, other factors, such as node weights, are also studied. The same ML model will require different constraints, weights, or learning speeds to generalize to diverse data patterns. These values are known as hyperparameters and must be adjusted so that the model can perform ML tasks optimally. Hyperparameter tuning identifies tuples of hyperparameters that produce an optimal model that minimizes a predetermined loss function on the independent data provided [10]. Hyper-parameter tuning refers to the automatic optimization of the hyper-parameters of a ML model. Hyper-parameters are all the parameters of a model which are not updated during the learning and are used to configure either the model (e.g. size of the hashing space, number of decisions trees and their depth, number of layers of a deep neural network, etc.) or the algorithm used to lower the cost function (learning rate for gradient descent algorithm, etc.). This idea can be pushed further to include the optimization algorithm (for neural nets: stochastic gradient descent, Adam, RmsProp, etc.) as an hyper-parameter. The last step is to include the type of model itself (logistic regression, ensembles of trees, neural nets) and also the features which are fed into the algorithm, but here we are venturing in the realm of autoML, which promises to put the human out of the loop of ML model design [12].

3.2.4.1 Tree-structured Parzen Estimator(TPE)

Tree-Structured Parzen Estimator (TPE) algorithm is designed to optimize quantization hyperparameters to find quantization configuration that achieve an expected accuracy target and provide best possible latency improvement. TPE is an iterative process that uses history of evaluated hyperparameters to create probabilistic model, which is used to suggest next set of hyperparameters to evaluate [49]. Basically TPE is an instantiation of Bayesian Optimization. It expect improvement as the acquisition function :

$$a(x, \alpha) = \int max(0, \alpha - f(x))\, dp(f(x)|D) \tag{3.1}$$

Non-parametric Parzen kernel density estimators (KDEs) to model the distribution of good and bad configurations w.r.t. a reference value α:

$$l(x) = p(y < \alpha|x) \text{ and } g(x) = p(y > \alpha|x) \tag{3.2}$$

KDEs in 3.2 can be used to compute 3.1 and optimized via sampling.

3.2.4.2 Hyperparameter ML Model

The hyperparameter values obtained in each ML Model here are only specific to the VeRemi Dataset or the Dataset generated by the F2MD framework. The hyperparameters for each ML Model can be seen in the table 3.2

Table 3.2: Hyperparamert of each ML Model

ML Model	Hyperparameter
RF	estimator : 208 criterion : entropy max depth : 11 max features : 0.399438
DBN	hidden layer structure : [256,256] RBM learning rate : 0.05 learning rate : 0.01 RBM epoch : 10 backpropagation iteration : 400 activation function : relu batch size : 32 dropout : 0.2
LSTM	batch size : 32 classes : 20 hidden layer : [128,128] optimizer : adam
GRU	batch size : 32 classes : 20 hidden layer : [128,128] optimizer : adam
ResNet152V2	batch size : 99 optimizer : nadam
MobileNet	batch size : 122 optimizer : adam

3.3 Detection System Proposed

3.3.1 2-Step History Prediction

3.3.1.1 Architecture

A Two-Step (2-step) prediction system involves two different classifiers. Initially, two distinct training methods with two distinct label datasets are used. The procedure for creating these two datasets is determined by the method depicted in subsection 4.3.1.3. The first dataset consists of two labels vehicle i.e attacker and legitimate. The training task of this dataset produces a classifier we call the "0-1 classifier". The second dataset consists of 14 classes referred to the attack's type afore-mentioned, without including legitimate vehicles. The second dataset will produce a classifier we refer to as the "14 Attack Classifier".The results of this process will be presented in the next subchapter.

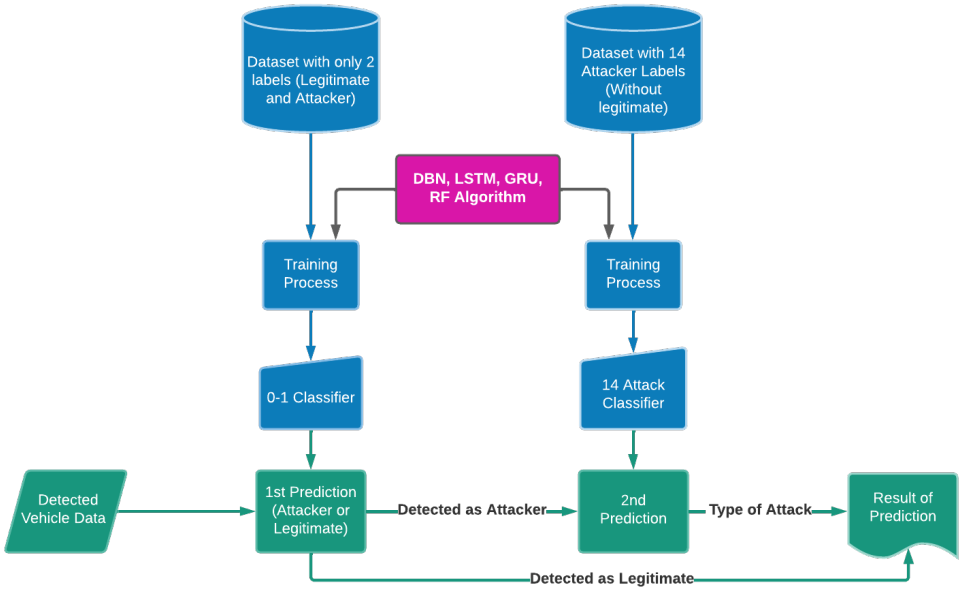


Figure 3.9: 2-Step History Prediction System Scheme [51]

In order to determine if the income messages are from a legitimate vehicle or an attacking vehicle, we have placed two stages classification system (Figure 3.9). At the first stage, a 0-1 classifier is used in order to identify the vehicle behavior (legitimate/attacker).
The output result of this classifier is used to trigger a second stage classification sub-system, in case the vehicle is suspected of being an attacker. Thus, the data is fed into the second predictor, which can figure out the type of attack. This second classifier can recognize a variety of VN Attacks.

3.3.1.2 Dataset

The dataset used in the 2-Step History Prediction training process is the Vehicular Reference Misbehavior (VeReMi) extension version. The purpose of the Veremi dataset is to assess VANET behavior detection methods. A number of simple attack are included in the initial dataset; this release is meant to serve as both a starting point for more complex attacks and a baseline against which others may be compared. This dataset was first compiled by Rens van der Heijden at 2018 at the Institute of Distributed Systems, part of Ulm University, Germany [93]. In 2020 Joseph Kamel et al developed the Veremi Dataset to become a Veremi Extension [48].

VeReMi extension is a simulated dataset, generated using F^{2}MD with a subsection of the Luxembourg SUMO Traffic (LuST) network with a size of 1.61 km^2 and a peak density of 67 Veh/km^2 [45].

VeReMi extension dataset consists of message logs per vehicle, and the details of the message are as follows:

- Message type
- BSM receive time
- BSM sent time
- Sender ID
- Sender Pseudonym
- Message ID
- Position
- Position Error
- Speed
- Speed Error
- Acceleration
- Acceleration Error
- Heading
- Heading Error

It should be noted that the Position info to Heading Error is each divided into a coordinate system x,y,z.
We provide further explanation of this dataset in the appendix B.

3.3.1.3 Preprocessing

The raw dataset will first be converted into the same format as table 3.3, through the steps shown in diagram 3.10. The parsing process of this dataset begins by creating a data list containing the Id of each vehicle and the attack code (see algorithm 1). Then the dataset will be processed by utilizing the attacker's list so that it becomes a dataset that is ready for the training process (see algorithm 2). At the end of this process, we get a labelled dataset.

Algorithm 1 Attacker List Process

```
labelsAttacks = [1,2,3,4,5,..., 19]
for attackType ∈ labelsAttacks do
    for data ∈ veremidataset do
        attackInfo ← data
        BSM ← data
    end for
    vehicleList ← join(attackInfo, BSM)
    for vehicle ∈ vehicleList do
        attackerId ← id number of attacker ∈ vehicle
    end for
    attackerList = openWrite(file)
    attackerList.write ← attackerId
    attackerList.close()
end for
```

Algorithm 2 Parsing Attacker Process

```
function convertData
    Pass in: vehicle, nMessage
    attackerData = openRead(attackerList)
    attackerId,labelAttack ← attackerData.read()
```

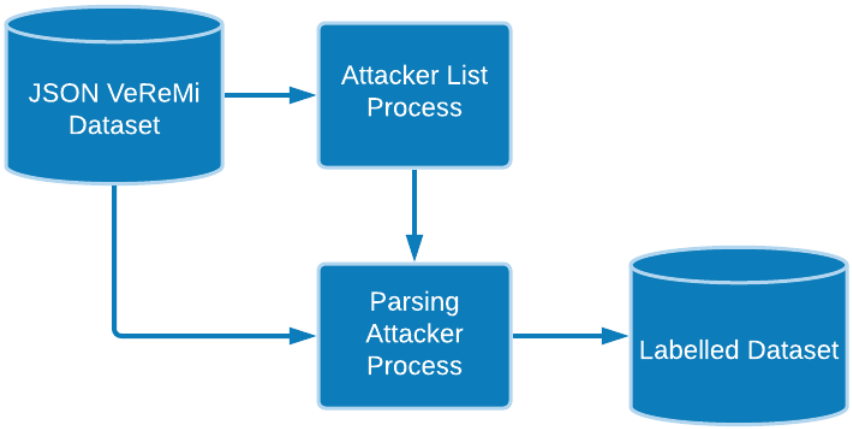


Figure 3.10: Parsing dataset

```
    attackerData.close()
    senderVeh ← data vehicle sender ∈ Vehicle
    for x = 0 to length(senderVehicle) do
        if length(senderVehicle[x]) ≥ nMessage then
            if senderVehicle[x].id ∈ attackerId then
                typeAttack ← labelAttack
            else
                typeAttack ← 0
            end if
            data.attackId ← typeAttack
            data.senderId ← senderVeh[x].id
            data.history.pos ← senderVeh[x].position
            data.history.spd← senderVeh[x].speed
        end if
    end for
    Pass out : data
end function
labelsAttacks = [1,2,3,4,5,..., 19]
nMessage = 5, 10, 15, 20, 25, 30
for attackType ∈ labelsAttacks do
    vehicleList ← vehicle trace data∈ attackType
    for vehicle ∈ vehicleList do
        input ← vehicle,nMessage
        vehicelBSM = ConvertData ← input
    end for
end for
vehicleBSM = openWrite(file)
vehicleBSM.write ← vehicleBSM
vehicleBSM.close()
```

3.3.1.3.1 Training dataset Format Each vehicle in the VN will send Message from the first time, second time, third time and so on $(t_0, t_1, t_2, ..., t_m)$. From each Message only information about its position and speed will be collected. In other words we use the **historical position and speed** of each vehicle as training data for

the ML models. So one input data for the training process is a history of information from one vehicle. We can also define input data as collecting an aggregate number of messages from a vehicle,see figure 3.11.

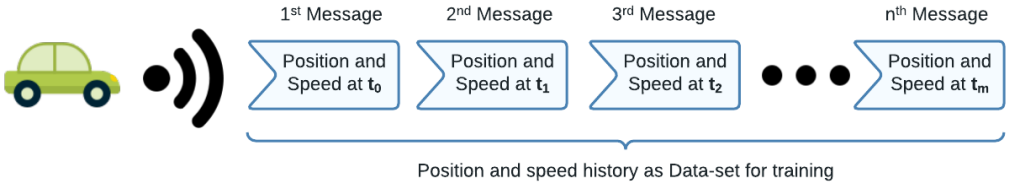


Figure 3.11: Illustration of the Origin of the dataset on 2-Step History Prediction

From the illustration above, it is possible to determine the format of the dataset that corresponds to the original information in BSM. So that the format messages are organized as shown in Table 3.3.

Table 3.3: Format of a Vehicle Data [51]

id	**type**	m_1	m_i	m_n
x	y	$pos/0_1$	$pos/0_i$	$pos/0_n$
		$pos/1_1$	$pos/1_i$	$pos/1_n$
		$spd/0_1$	$spd/0_i$	$spd/0_n$
		$spd/1_1$	$spd/1_i$	$spd/1_n$

- **id: x**: is the vehicle sender number identification
- **type: y**: indicates the type of the vehicle, and refers to 0, it means that this is a legitimate vehicle (non-attacker). Howerever if value equals to 1, it means that this is an attacker vehicle.
- m_1, m_i, m_n: refers respectively to 1st message, ith message, and nth message.
  - **pos**: vehicle position in accord with GPS coordinate x (pos/0) and y (pos/1)
  - **spd**: vehicle speed in accord with speed vector x (spd/0) and speed vector y (spd/1) in meter/second

Of course, determining the number of messages in one vehicle history data must be limited. As an experiment, we created several formats to see how far the accuracy progressed. For dataset training task, we considered different configurations, depending on the group numbers of the aggregated messages used in the input features of the ML model. Thus we consider the following cases:

1) 5 aggregated messages (5_{msg})
2) 10 aggregated messages (10_{msg})
3) 15 aggregated messages (15_{msg})
4) 20 aggregated messages (20_{msg})
5) 25 aggregated messages (25_{msg})
6) 30 aggregated messages (30_{msg})

3.3.1.3.2 Clustering The training process has been carried out using the DBN, LSTM, GRU, and RF models. The training process produces many types of the confusion matrix. Based on observations of all confusion matrices generated from the aforementioned ML models, several types of attacks have been found that have the same characteristics, i.e **sub-cluster** :

1) *Constant Speed* and *Constant Speed Offset* from now on referred to as *Constant Speed+*

2) *Data Replay* and *Data Replay Sybil* from now on referred to as *Data Replay+*

3) *Disruptive*, *DoS Disruptive*, and *DoS Disruptive Sybil* from now on referred to as *Disruptive+*

4) *DoS Random* and *DoS Random Sybil* from now on referred to as *DoS Random+*

As a result, if some attacks are gathered together into a small group, it will be easier to identify them. Regrouping these types of attack groups resulted in **14 types of attacks** compared with the list of misbehavior in chapter 3. The new misbehavior list that will be implemented in the next training process is the following [51]:

1. Constant Position
2. Constant Position Offset
3. *Constant Speed+*
4. *Data Replay+*
5. Delayed Messages
6. *Disruptive+*
7. DoS
8. *DoS Random+*
9. Eventual Stop
10. Grid Sybil
11. Random Position
12. Random Position Offset
13. Random Speed
14. Random Speed Offset

As an example of a comparison of the accuracy of the training results of the aforementioned models for 30msg, it can be seen in the table 3.4, and it is clearer from Figure 3.12, that there is an increase in accuracy for all models with clusterization

Table 3.4: Comparison of Accuracy With and Without Clustering for 30_{msg}

	Accuracy	
	Without Clustering	**Clustering**
DBN	0.496553	0.601792
LSTM	0.683134	0.859605
GRU	0.701057	0.8642
RF	0.644301	0.823989

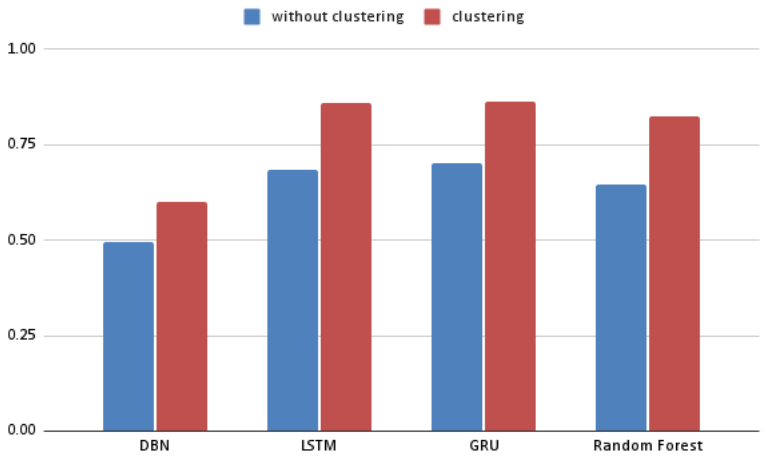


Figure 3.12: Chart Comparison of Accuracy With and Without Clustering for 30_{msg}

3.3.2 2-Step 2-D BSM Prediction

3.3.2.1 Architecture

The 2-Dimension(2-D) BSM technique is different from the 2-Step History Prediction technique. This technique utilizes the capabilities of ResNet152V2 and MobileNet Deep Learning in performing classifications based on a 2-D matrix. The process of choosing these Machine Learning model is described in the appendix B. In general, the input for ResNet152V2 and MobileNet is an image to be classified. To put it another way, we treat a BSM as an image. Meanwhile, a BSM is only a one-Dimension collection of information mentioned in subchapter A. Therefore, it is necessary to pre-process the data so that ResNet152V2 and MobileNet can accommodate the BSM data.

In the first process in this technique, the ML model conducts training using a 2-D BSM dataset (as explained in the following subchapter) with the assumption that the ML model has used the optimum hyperparameter. The training results are in the form of a classifier model, which has a target to classify at the same time 19 types of BSM from attack vehicles plus 1 type of BSM from legitimate vehicles. This classifier will be directly used to classify the detected BSM. The detection results decide whether the BSM comes from a legitimate vehicle or an attacker's vehicle. If the BSM comes from the attacker's vehicle, then the classifier is also expected to determine the attack type directly, see figure 3.13. This is different from the 2-Step History Prediction, which determines the kind of attack from the second detection result. One of the essential things distinguishing the 2-Step 2-D BSM Prediction technique from 2-Step History Prediction is: 2-Step 2-D BSM Prediction detects message by message; of course, this has advantages and disadvantages.

3.3.2.2 Dataset

2-Step 2-D BSM Prediction using the dataset generated by F2MD on the UPHF map (UPHF stand for Université Polytechnique Hauts-de-France in Valenciennes France); we want to be able to directly use the results of the ML model training to be applied to real-time applications. There is a slight difference between the dataset generated by F2MD and the VeReMi dataset, although the two datasets come from the same source. The UPHF dataset has GPS information in the BSM, which the vehicle sends and receives in the VN simulation. Meanwhile, in the Veremi dataset, the gps information is separated in data type 2. In addition to the UPHF dataset there is no message ID as in the VeReMi dataset; this is because the available F2MD

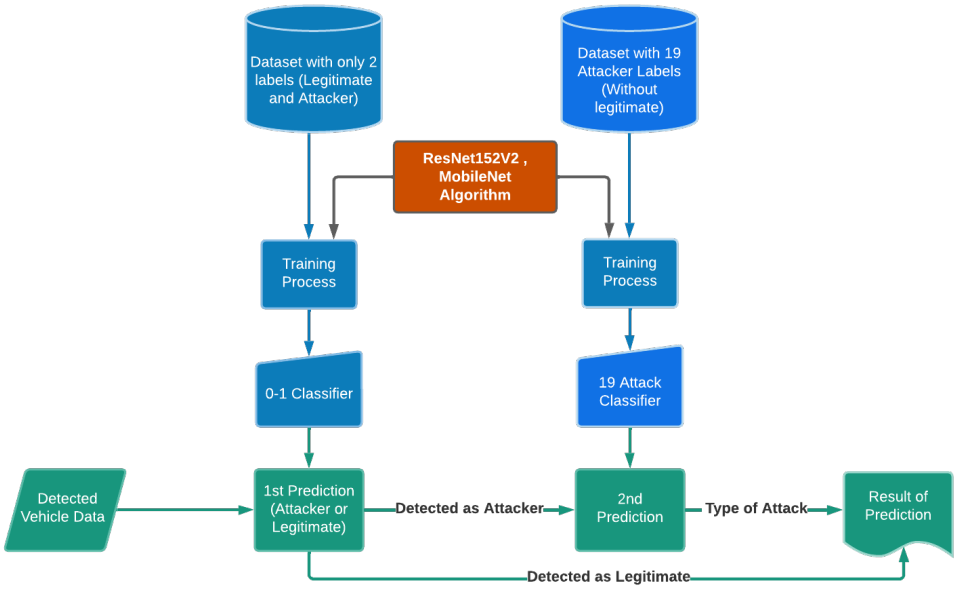


Figure 3.13: 2-Step 2-D BSM Prediction System Scheme

application is the latest version (Ver.2) while the VeReMi dataset (which has been published) comes from the first version of F2MD. 2-Step 2-D BSM Prediction utilizes all information in BSM, so every detail of data becomes important, while 2-Step History Prediction only uses position and speed information.

UPHF Map dataset consists of message logs per vehicle, and the details of the message are as follows:

- Attack type
- BSM create time
- BSM arrival time
- Sender ID
- Sender Pseudonym
- GPS
- Position
- Position Confidence
- Speed
- Speed Confidence
- Acceleration
- Acceleration Confidence
- Heading
- Heading Confidence

It should be noted that the GPS info to Heading confidence is each divided into a coordinate system x,y,z.

3.3.2.3 Preprocessing

ResNet152V2 and MobileNet require input in the form of a 32×32 matrix, then the information in the BSM will be converted into a matrix according to the shifting technique. If we have a dataset with 32 features, it's the same as a 1×32 matrix; this matrix will be the first row of the converted 32×32 matrix. Then the second, third, and so on are obtained by shifting the components into a 2-D format, as shown in Figure 3.14.

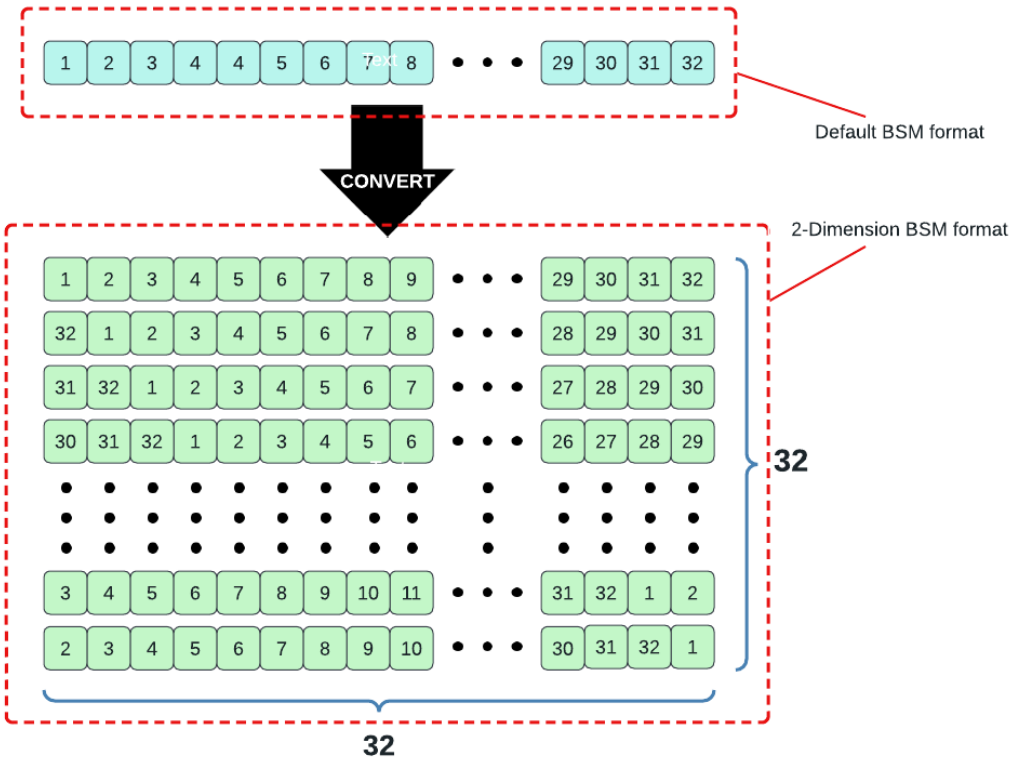


Figure 3.14: 2-Dimension BSM Shifting Mechanism

All BSM data formatted according to 2-D format will be input for the ResNet and MobileNet model training process. Of course, the training process will require significant resources because the data set size is quite large.

The preprocessing algorithm 3 in 2-Step 2-D BSM Prediction aims to change the BSM standard into a 2-D format. This algorithm first works by converting the BSM one-dimensional data into a 1×32 data array. We will rearrange this array data in a 32×32 matrix whose row components are shifts from 1×32 data arrays. The final result is matrix $X_{32\times32}$ as the input data for the training process and Y as a label for each input $X_{32\times32}$.

3.4 Performance Analysis

3.4.1 2-Step History Prediction

3.4.1.1 Evaluation

The training process has been carried out and gives a classifier for each model. Then an accuracy test is imposed to evaluate the performance of each classifier. First, we look at the single prediction simulation outcomes in table 3.5. Single prediction means that the classifier immediately detects the type of attack that appears in the network and classifies it into an attack-type according to the subsection 3.3.1.3. In general, there is an accuracy increment when the number of messages is increased for each training and validation process. This also happens in 2-Step History Prediction, see table 3.6. In a single prediction, the highest accuracy was obtained using GRU architecture, with an accuracy of 86,42% using 30 aggregated messages. In a 2-Step History Prediction configuration, the best accuracy was obtained using LSTM architecture, with an accuracy of 95,88% using also 30 aggregated messages. In this

Algorithm 3 Converting BSM to 2-D BSM

```
data ← array(data-set[drop:sender,senderPseduo])
Y_output ← data(attack_type)X_input← array.zero[0,32]
for i = 0 to data.row do
    x ← data[i,30]
    S = data.col - 1
    X_input[i,S] ← x
end for
X_temp ← array.zero[1,32,32]
for j = 0 to X_input.row do
    for t = 0 to X_temp[j].row do
        if t==0 then
            X_temp[j,t] ← X_input[j]
        else
            X_temp[j,t] ← array.roll(X_temp[j,t-1])
        end if
    end for
end for
X ← X_temp
Y ← Y_output
```

latter case, a result obtained with the GRU model gives an accuracy of only 0.16% less than the LSTM model. We can note that input data that contains more information or features tend to be easier and better detected by the classifier, as seen from the increase in accuracy from 5 messages to 30 messages.

For each model, we compare the results obtained with the single prediction system and with the 2-Step History Predictions system, where we note that all simulations results show a significant increase in terms of accuracy. The performance disparity between single prediction and 2-Step History Prediction is rather large, with optimal gain accuracy for LSTM and GRU hitting **95%** (see Figures 3.16 and Figure 3.17). These two models exhibit similar accuracy, which is understandable given that the GRU architecture is nearly identical to the LSTM model, except of the kind of gate and memory. Figure 3.18 shows that the accuracy of RF has increased significantly while approaching that of LSTM and GRU. DBN architecture has the highest accuracy increment, which is equal to 23.24% for 10 messages, 77.01% from the 2-Step History Prediction minus 53.77% from the single-step prediction (see figure 3.15). However, overall DBN has the lowest accuracy compared to the three other models (table 3.5 and table 3.6).

Table 3.5: Accuracy of Single Prediction

	5_{msg}	10_{msg}	15_{msg}	20_{msg}	25_{msg}	30_{msg}
DBN	45.87%	53.77%	59.63%	56.73%	61.10%	60.17%
LSTM	78.53%	79.65%	83.32%	82.62%	85.72%	85.96%
GRU	80.34%	80.06%	83.06%	85.86%	86.28%	**86.42%**
RF	75.91%	77.83%	79.35%	81.62%	82.52%	82.39%

One thing that is quite interesting to observe is whether the addition of the number of messages has a major effect on the accuracy of attack detection. According to table 3.5 and table 3.6, we calculate the **average accuracy($\bar{x}$)** from 5 messages to

Table 3.6: Accuracy of 2-Step History Prediction

	5_{msg}	10_{msg}	15_{msg}	20_{msg}	25_{msg}	30_{msg}
DBN	64.32%	77.01%	77.27%	73.63%	79.94%	79.26%
LSTM	89.72%	91.05%	92.62%	94.07%	94.56%	**95.88%**
GRU	90.07%	89.97%	92.31%	94.19%	95.13%	95.72%
RF	88.65%	89.03%	91.28%	92.40%	93.89%	93.41%

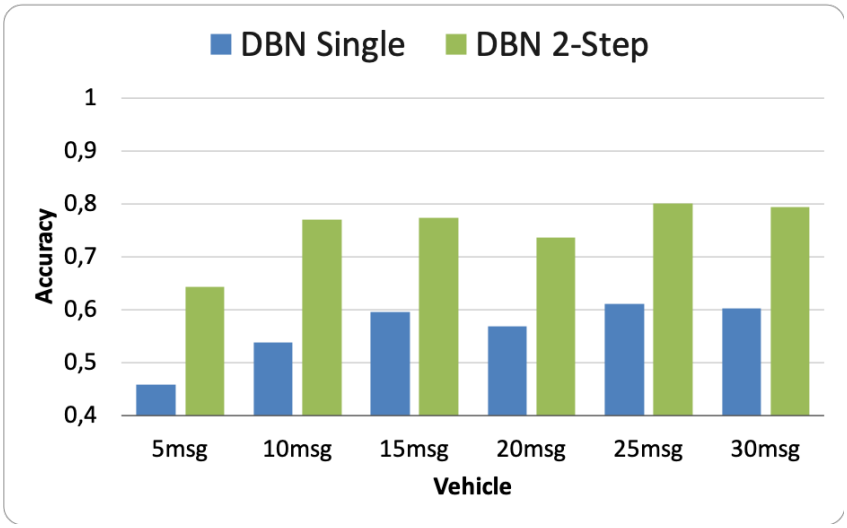


Figure 3.15: Comparison of Single Prediction and 2-Step History Prediction in DBN Model

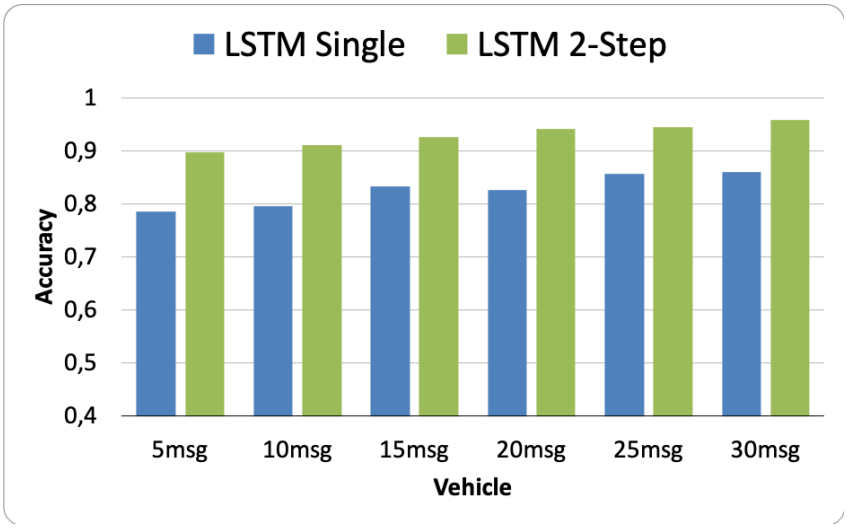


Figure 3.16: Comparison of Single Prediction and 2-Step History Prediction in LSTM Model

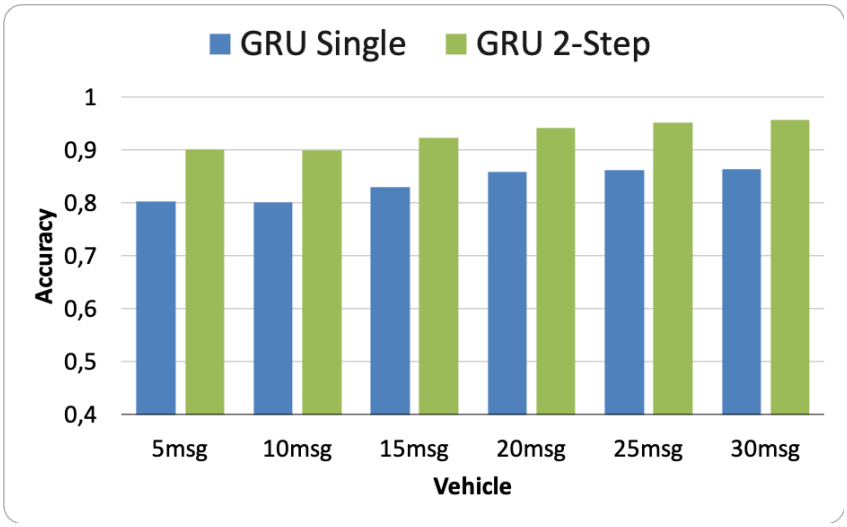


Figure 3.17: Comparison of Single Prediction and 2-Step History Prediction in GRU Model

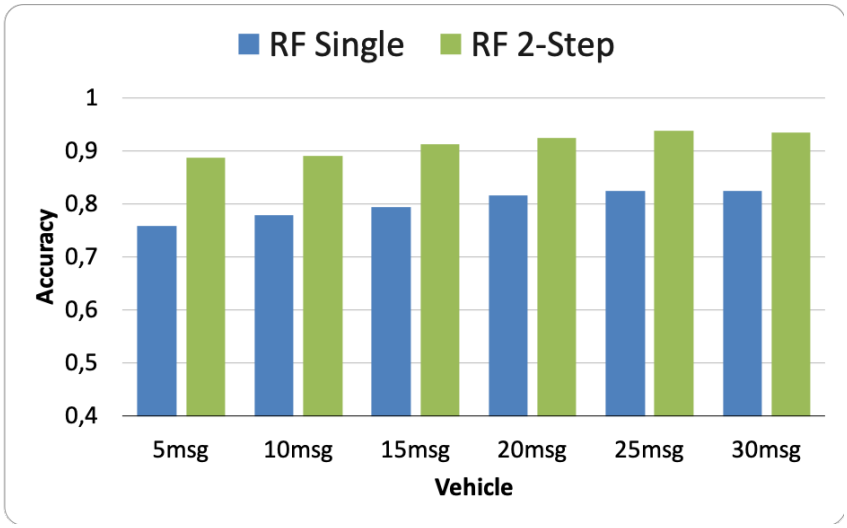


Figure 3.18: Comparison of Single Prediction and 2-Step History Prediction in RF Model

30 messages, then calculate their **standard deviation** (σ) and the overall results are presented in the table 3.7. The standard deviation value for all models is found to be significantly lower than the average value, indicating that there is no major data deviation between the apparent number of messages. However, the quantity effect of messages has no major influence on detection accuracy, this is true for every model examined. Even though the accuracy of the DBN model is noticeably smaller than the other three models.

Table 3.7: The Accuracy Significance of Single and 2-Step Prediciton

	Single Prediction		**2-Step History Prediction**	
	$\bar{x}$	σ	$\bar{x}$	σ
DBN	0.56	0.057	0.752	0.058
LSTM	0.82	0.03	0.930	0.023
GRU	0.83	0.029	0.929	0.025
RF	0.80	0.027	0.914	0.022

3.4.1.2 Timing Comparison

The performance evaluation of misbehavior/attack detection in terms of accuracy is quite important. Furthermore, the detection speed process factor is also an important indicator since our proposed system is attended to be used at a crucial time in a vehicular environment. This timing process comparison simulation is performed on an Intel Xeon 3.70 GHz processor (16 Cores) workstation,64 GByte of DRAM. In a 2-step detection system, the detection speed is more affected. This is due to the algorithm mechanism of each model being different and also due to the amount of data that can be captured by the detector. In the table 3.8, 3.9, 3.10, and 3.11, a comparison of the detection process speed between single prediction and 2-Step History Prediction is presented. It is shown that LSTM and GRU models require higher detection times. Meanwhile, RF and DBN architectures require less time. The gap of time between the single prediction system and the 2-Step History Prediction system for each model is as follow : for DBN the average gaps time is equal to 0.08 ms. This gap is less for RF model, which is equal to 0.02 ms, while for LSTM and GRU architectures the gap time is more significant, it has an average of 0.63 ms and 0.67 ms respectively.

In general, the 2-Step History Prediction technique will require more time process than a single prediction in attacks identification. This is because one input attack data must pass through two classifiers in a 2-Step History Prediction before it can be detected. However, the accuracy of the 2-Step History Prediction system is highly important and more promising in terms of security compared to the single prediction system.

Table 3.8: Comparison of LSTM Timing Process Predictions

	Vehicle Timing on Average (ms)	
	Single Prediction	**2-Step History Prediction**
5_{msg}	0.281357	0.729124
10_{msg}	0.367216	0.832994
15_{msg}	0.467365	1.105935
20_{msg}	0.552322	1.278447
25_{msg}	0.654835	1.369581
30_{msg}	0.739149	1.537104

Table 3.9: Comparison of GRU Timing Process Predictions

	Vehicle Timing on Average (ms)	
	Single Prediction	**2-Step History Prediction**
5_{msg}	0.255114	0.594953
10_{msg}	0.325892	0.779323
15_{msg}	0.404575	0.959176
20_{msg}	0.494021	1.159909
25_{msg}	0.572481	1.336444
30_{msg}	0.658712	1.917452

Table 3.10: Comparison of DBN Timing Process Predictions

	Vehicle Timing on Average (ms)	
	Single Prediction	**2-Step History Prediction**
5_{msg}	0.025007	0.100025
10_{msg}	0.02337	0.105609
15_{msg}	0.025306	0.11346
20_{msg}	0.02614	0.114469
25_{msg}	0.027162	0.120092
30_{msg}	0.028292	0.117776

Table 3.11: Comparison of RF Timing Process Predictions

	Vehicle Timing on Average (ms)	
	Single Prediction	**2-Step History Prediction**
5_{msg}	0.035613	0.059768
10_{msg}	0.036318	0.059768
15_{msg}	0.038094	0.058037
20_{msg}	0.038072	0.065021
25_{msg}	0.039105	0.06608
30_{msg}	0.045618	0.061529

3.4.2 2-Step 2-D BSM Prediction

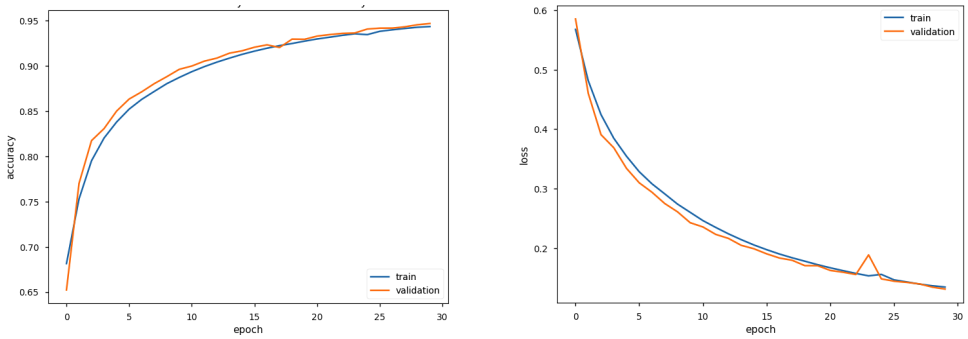
3.4.2.1 Evaluation

The training process is quite time-consuming, especially for ResNet152V2 compared to MobileNet. However, the resource requirements of these two ML models are more or less the same, which requires the support of GPU and CUDA parallel computing platforms. Otherwise, the training process that relies on the CPU alone will take a long time.

Before doing the training process, we did Hyperparameter Optimization (HPO) first for these two algorithms. The HPO process can be seen at B.4. Then we retrained using HPO for both ML Models by increasing the number of epochs. Similar with 2-Step History, we prepare classifier for 1st Prediction and 2nd Prediction. 1st prediction process is expected to produce a classifier that can detect legitimate and attacker messages. The training results can be seen in table 3.12. ResNet152v2 only uses 30 epochs, because the loss and accuracy graphs at 30 epochs have converged and show good results (figure 3.19). Likewise with MobileNet, at this stage the loss and accuracy graphs are quite convergent at epoch 200, see graphic in figure 3.20.

Table 3.12: Accuracy of ML Model 2-Step 2-D BSM Prediction For 1st Prediction

ML Model	Epoch	Accuracy
ResNet152V2	30	94,7%
MobileNet	200	96,78%



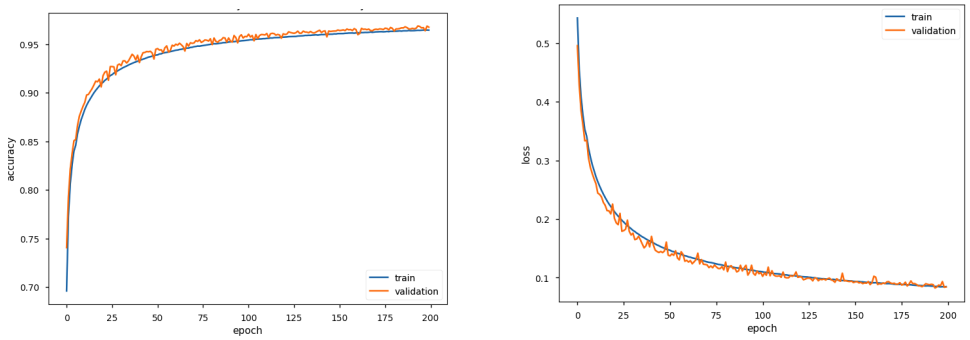
(a) Accuracy (b) Loss

Figure 3.19: ResNet152V2 Train vs Validation For 1st Prediction

For the needs of 2nd prediction, ResNet152V2 conducted training for 80 epochs and MobileNet for 350 epochs, then resulted in an accuracy of 97.78% for ResNet152V2 and 96.23% for MobileNet, see table 3.13. Epoch is limited to a particular value after the accuracy and loss graph show convergence between the training data and validation data.

Table 3.13: Accuracy of ML Model 2-Step 2-D BSM Prediction For 2nd Prediction

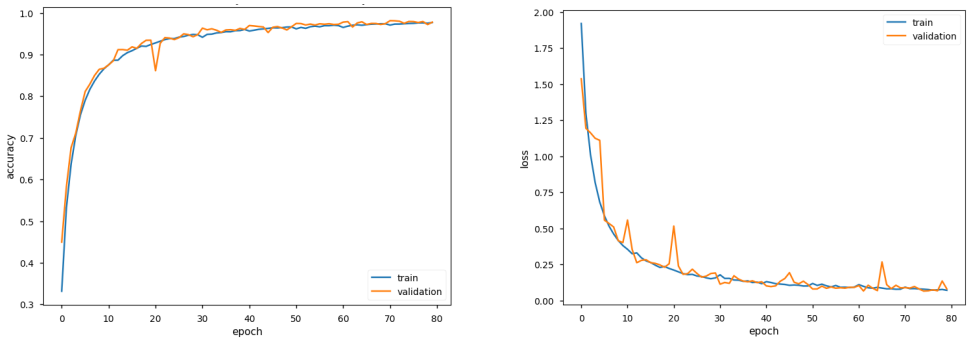
ML Model	Epoch	Accuracy
ResNet152V2	80	97.78%
MobileNet	350	96.23%



(a) Accuracy (b) Loss

Figure 3.20: MobileNet Train vs Validation For 1st Prediction

If we look at the graph 3.21 Accuracy on ResNet152V2, we can see that the validation graph follows the train graph from the beginning of the epoch. The accuracy increases significantly starting from epoch 15 and converging at epoch 30, and so on; the accuracy does not increase too much. This means that the accuracy at epoch 80 is already the optimum condition for ResNet152V2. This is also illustrated on the Loss chart.



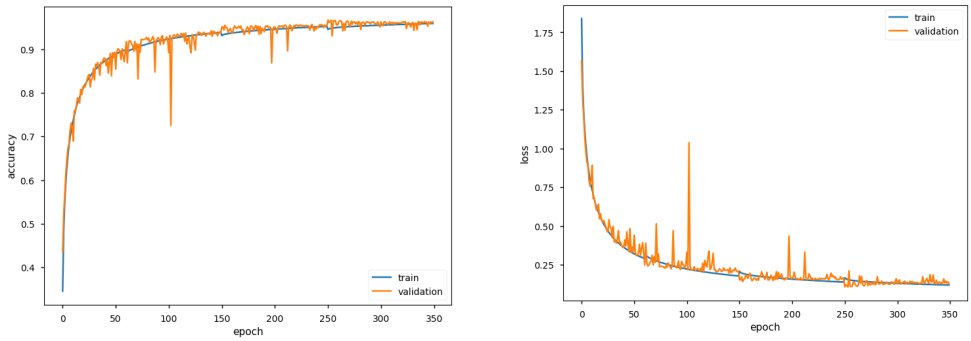
(a) Accuracy (b) Loss

Figure 3.21: ResNet152V2 Train vs Validation For 2nd Prediction

For MobileNet, we can look at graph 3.22; for accuracy and loss, both graphs show that the validation can follow the train graph well, even though there are many spikes. It indicates that MobileNet requires a large enough epoch to achieve optimum conditions. Unlike ResNet152V2, MobileNet showed a significant increase in accuracy in the 100th epoch despite a reasonably high spike. The graphs start to converge at epoch 150 onwards. So the accuracy of epoch 350 is already optimum in this case.

3.4.2.2 Timing comparison

Like in 2-Step History Prediction, in the 2-Step 2-D BSM Prediction technique, it is also necessary to compare the timing process. The hardware base used in this timing comparison is the same as that used in the 2-Step History Prediction. The technique is also not much different; 1000 messages are taken and then detected by each model so that we will obtain the total process time. The whole detection time will be divided by the number of messages, so we will get the average time for each ML model to detect one message. From table 3.14, it can be seen that MobileNet is faster than ResNet152V2. We also remember that the training speed on MobileNet is



(a) Accuracy (b) Loss

Figure 3.22: MobileNet Train vs Validation For 2nd Prediction

also faster than ResNet152V2. This shows that MobileNet has a promising potential when applied to real-time applications in term of timming process.

Table 3.14: Comparison of Timing Process 2-Step 2-D BSM Predictions

	BSM Timing on Average (ms)	
	Single Prediction	**2-Step 2-D BSM Prediction**
MobileNet	1.4806	2.5312
ResNet152V2	4.0723	14.387

3.5 Conclusion

The primary purpose of an IDS in VN is to distinguish between normal behavior and abnormal behavior of a vehicle and sound an alarm if an attack is detected. Misbehavior detection methods generally revolve around three techniques: *signature, specification, and anomaly* detection system. Each group has its own strengths and weaknesses. In the cyber security world, a huge amount of information data is obtained from network sensors, logs, agent endpoints, and others. The data obtained is extensive in volume, speed, and variation, so it is included in Big Data. Problems arise when attempting to analyze Big Data, and traditional methods of intrusion detection are inadequate. On the other hand, ML, as part of AI, is considered capable enough to overcome this problem.

The 2-Step History Prediction technique is able to improve the accuracy performance of each model ML even better because this technique focuses on how to classify the type of attack after separating the attacker's vehicle from the legitimate vehicle first. The best accuracy for training results and predictions using clustering and 2-Step History Prediction is GRU and LSTM, while the lowest accuracy is DBN. The slowest 2-Step History Prediction speed is LSTM, while the best speed for this technique is RF. The increase in the number of messages per vehicle does not have much effect on the detection speed of the model. Even though the Accuracy of DBN is slightly affected by the increase in the number of messages per vehicle compared to the other 3 models. The LSTM detection speed is slightly affected by the increase in the number of messages per vehicle compared to the other 3 models. Doing some clustering on the VeReMi dataset which consists of 19 types of attacks into 14 types of attacks is sufficient to increase the predictive accuracy performance of each ML model.

The main value of 2-Step History Prediction are: The system can detect several types of misbehavior using one kind of ML classifier; the system inputs are only position and speed information, so it is more flexible for various traffic scenarios; the system does not require high resources in the training process.

2-Step 2-D BSM Prediction is a more straightforward detection system than 2-Step History Prediction because it process message directly without need of history messages. However, in its preparation, the 2-Step 2-D BSM Prediction technique requires a more extensive resource when compared to 2-Step History Prediction. This system requires GPU hardware capable enough to carry out the training process. This technique treats the BSM data stream as if it were an image. From the results of the training that has been carried out, ResNet152V2 and MobileNet both show good results in terms of accuracy.

The Main value of 2-D BSM Prediction are: The system can detect several types of misbehavior using one kind of ML classifier; The system can detect misbehavior based on one BSM, so it can produce relatively fast decisions.

Chapter 4

Real Time Implementation

Contents

4.1 Introduction

The term "real-time simulation" refers to a computer model of a physical system that is capable of being executed at the same pace as "wall clock" time in the real world. To put it another way, the computer model and the real-world physical system have the same pace of operation. Studying the Vehicular Network in a real-time application is very important. Unlike computer networks that tend to connect between stagnant nodes and are slightly affected by external factors, vehicular networks are very dynamic and will be heavily influenced by external factors. Each node in the VN constantly changes quickly and moves, so the vehicle information data will continually change. Application of the system offline and online (real-time) can have different results. A system that gives good results significantly will not necessarily produce the same results in real-time implementation.

VEINS is one of the most commonly used real-time simulators in the C-ITS field [88]. This application is open-source and quite reliable in running vehicle network simulations. See figure. VEINS generally utilizes the OMNet++ application to create vehicle nodes in the simulation and pair the node movements with vehicle movements in the road traffic simulator (SUMO). This mechanism will form a comprehensive V2X simulation. Network and mobility simulations can run in parallel with the help of two-way coupling achieved by a standard connection protocol: Traffic Control Interface (TraCI). TraCI allows OMNeT++ and SUMO to exchange messages while the simulation runs as part of a TCP connection [95]. VEINS also provides the ability to generate custom data sets for different road networks. But by default, it does not include misbehavior detection algorithms. Figure 4.1 demonstrates the several modules that come together to form the VEINS architectural design.

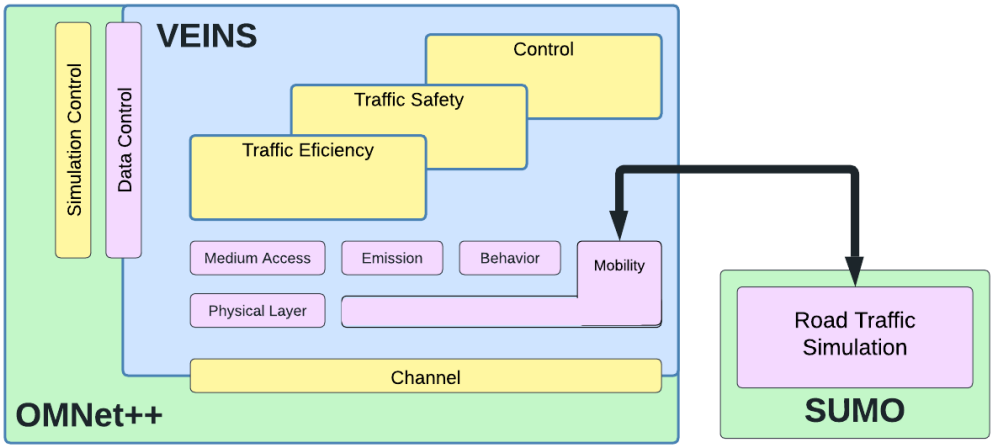


Figure 4.1: Building Design for the VEINS Platform

In this subsection, we will explain the results of the implementation of the proposed detection system, including an explanation of the application framework that we use and evaluation metrics related to the results.

4.2 Application Framework

4.2.1 F²MD

In the real-time implementation, we use the F²MD application, which we have partially explained in subchapter 2.6.1.5. Basically this application is an additional framework for VEINS. F²MD offers a comprehensive solution for modeling and assessment of a MisBehavior Detection (MBD) system in real time. It expands VEINS with a vast array of modules for MBD, assessment, and other C-ITS modules in general. Modularity is one of F²MD's most prominent traits. This framework uses the Luxembourg SUMO Traffic (LuST) network as a real traffic scenario and also OMNET++ for simulations involving parameter beacons.

The simulation of the vehicle network in real traffic will be displayed extensively by this application by presenting OMNET++, which shows data communication, and SUMO which displays vehicle traffic and terminals to run services and view messages that appear, see figure 4.2.

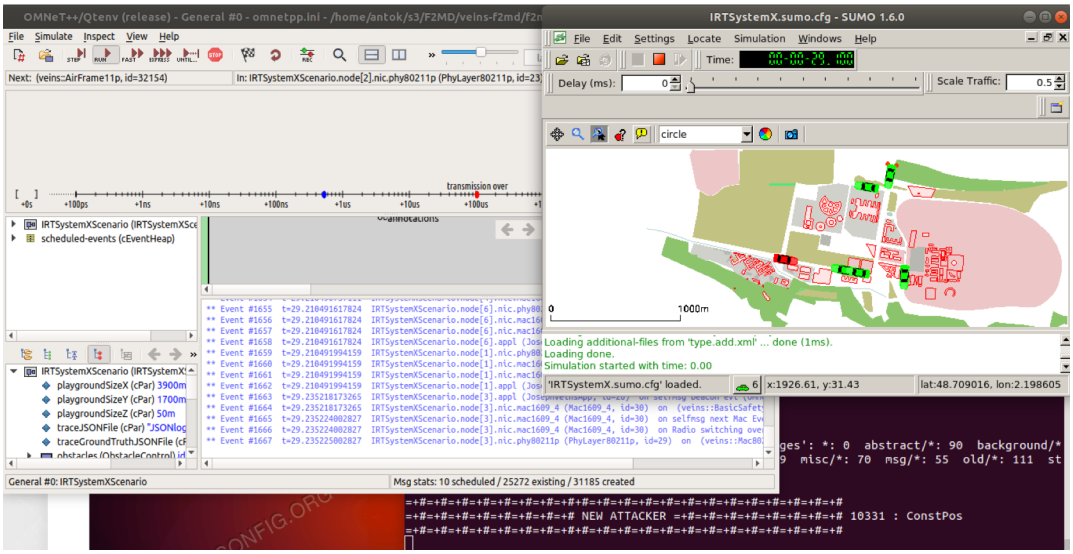


Figure 4.2: Simulation Display on F²MD

F²MD support network technologies:

- ITS-G5 (IEEE 802.11p)
- C-V2X (3GPP PC5 Mode 4)

F²MD features:

- Checks of Received Beacons for Their Most Fundamental Sense of Plausibility.
- Investigation of the Plausibility at the Node Level.
- The Use of Real-Time Machine Learning in the Investigation of Plausibility (HTTP to the Python Server: machine-learning-server)
- Output of Real-Time Detection Status in Real Time.
- Assistance with a Number of Different Reporting Mechanisms.
- Assistance in the Collation and Investigation of Global Reports.
- Basic Psudonym Change Policies.

- Implementation of Misbehavior Attacks on a Local and Global Scale.
- Attacks Can Be Performed in Real Time.

4.2.2 Architecture

The F^{2}MD architecture consists of 5 main module level [45], see figure 4.3 :

1. *Input Dataset*
   The dataset input of this application comes from the BSM which is sent and received by the vehicles in the vehicular network in the simulation according to the selected scenario.

2. *Local Detection*
   Local detection consists of two types of algorithms:

   - Fixed Algorithm
     - Threshold App
     - Aggregation App
     - Behavioral App
   - ML Algorithm
     - SVM
     - MLP
     - LSTM

   The detection system receives input as a plausibility check, for example, range plausibility, speed plausibility, and position plausibility. Likewise, the ML algorithm gets input on feature datasets derived from this mechanism.

3. *Local Visual Output*
   The appearance of every vehicle, both legitimate and attacking vehicles, will appear in real-time on the SUMO application, as well as in the form of plot graphs.

4. *Data Output*
   The output data is a report of the overall detection results. This report will be forwarded to the Misbehavior Authority.

5. *Global Detection*
   Reports received from the Data Output section will be stored in the Data Collection and Format sections. Furthermore, the Analysis and Decision section will analyze the report collection to determine the proper reaction. Then the results of the analysis decision will be issued by the Reaction section, whether it is no reaction, alarm, or revocation of the vehicle's certificate suspected of being the attacker.

The F^{2}MD application comes with a support module, see figure 4.4. Support modules are helpful to help the main module run smoothly. The storage mechanism is used by Global Detection in storing reports. We can also choose the pseudonym mode we want. In this simulation, we choose the Car2Car method. Meanwhile, what is very important from this module is to provide misbehavior modules according to the type of attack that has been discussed in subchapter **??**.

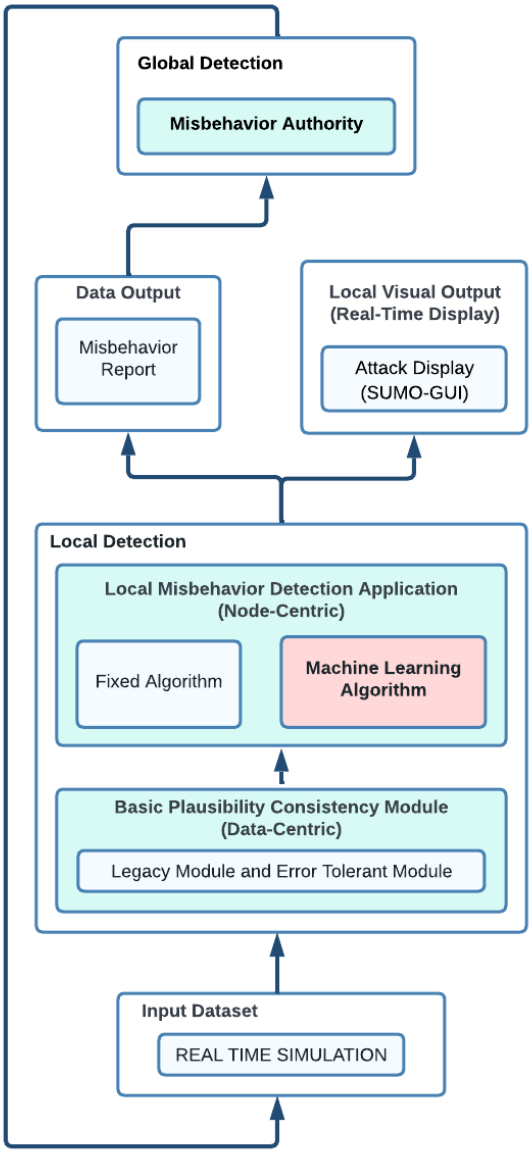


Figure 4.3: F^{2}MD Architecture Diagram

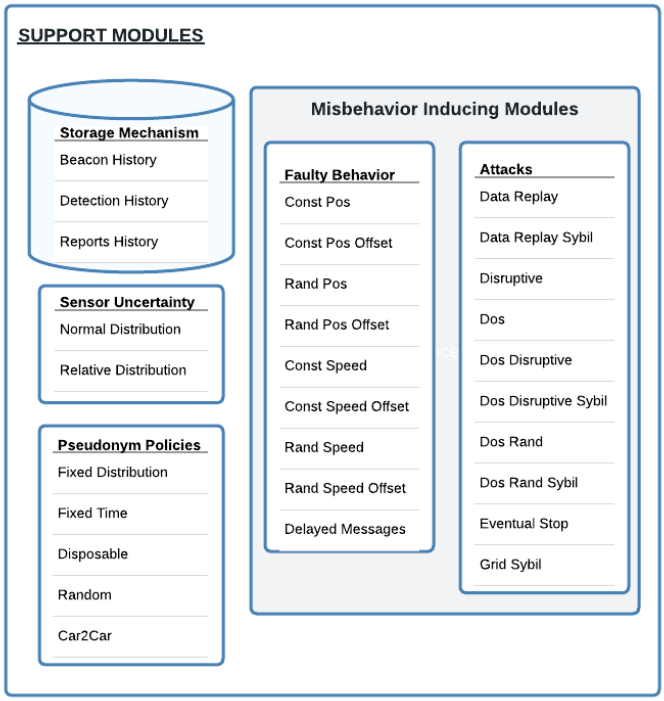


Figure 4.4: F^{2}MD Support Module

4.2.3 System Proposed Implementation

The main modification we made to the F^{2}MD application to accommodate our proposed system is in the *Local Detection* section. So it's within our implementation limitation that we don't handle *Global Detection*. In this section, we do not use a fixed algorithm and a plausibility check mechanism but replace it with a direct BSM converter mechanism. In the ML section, we also use the classifier that we have prepared in section 3.3, See figure 4.5. From the diagram, we can see that we have replaced the default fixed algorithm, ML algorithm, Legacy Module, and Error Tolerant Module modules with two main modules:

1. *Machine Learning System Proposed*
   ML will use the DBN, LSTM, GRU, and RF classifiers for 2-Step History Prediction. Then for 2-Step 2-D BSM Prediction, ResNet152V2 and MobileNet classifiers will be used.

2. *BSM Converter System Proposed*
   In 2-Step History Prediction, BSM will be converted into position and speed history, while in 2-Step 2-D BSM Prediction, BSM will be converted into 2-Dimension BSM.

We also made some modifications to Map Scenario for implementation in this application. We use the UPHF map scenario, which we generated from OpenStreetMap and SUMO. See figure 4.6. We hope that further research can be more sustainable by using our campus map.

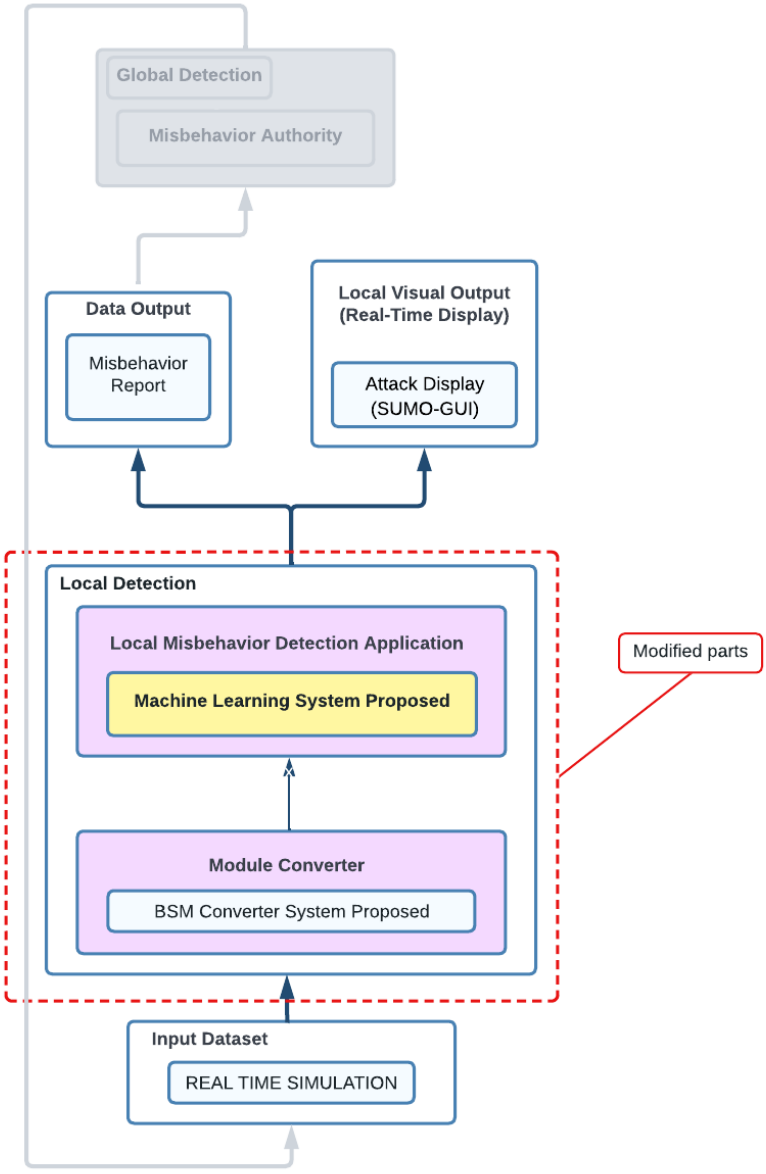


Figure 4.5: F^{2}MD Architecture Diagram Modification

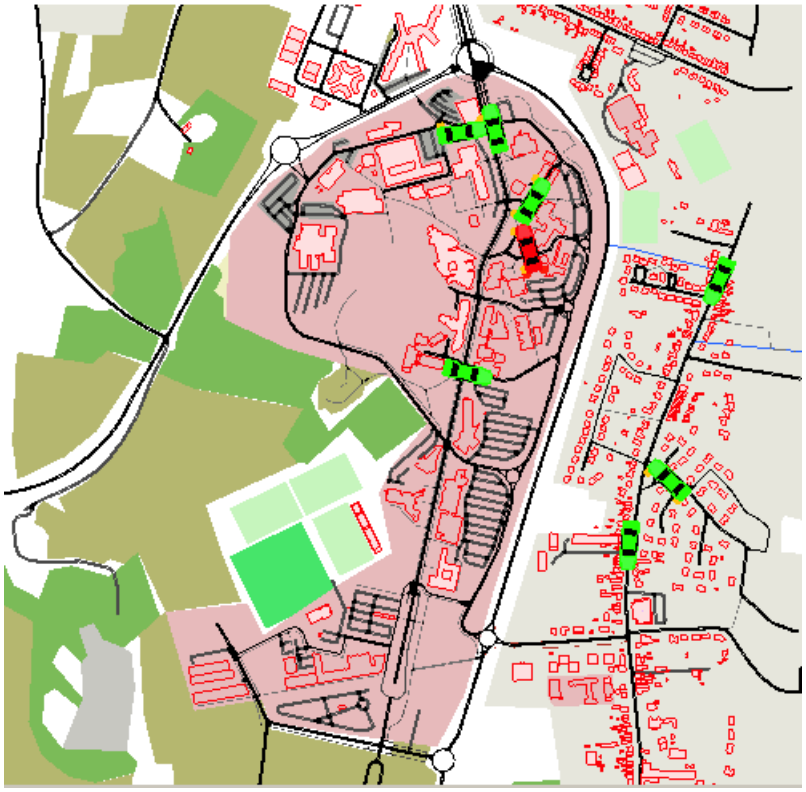


Figure 4.6: UPHF Map on SUMO GUI

4.2.4 Platform

Users of the F^{2}MD application are advised to use the Linux operating system as the base OS. If we want to use Microsoft Windows, it is recommended to use the virtual machine version of this application. We can read installation instructions at the following link: https://github.com/josephkamel/F2MD.git
The basis of our system in the implementation of F^{2}MD is as follows:

- OS : Ubuntu 18.04.6 LTS
- CPU : Intel i5-4300CPU @ 1.90 GHZ (dual core)
- RAM : 8052 MB
- HD : 120 GB SSD

4.3 Evaluation Metrics

The detection simulation will produce several possible results according to the sensitivity level of each ML model classifier. The possible results obtained are as follows:

True Positive (TP) : The system detects a vehicle data that is **actually the attacker** and is predicted to be **the attacker**.

True Negative (TN) : The system detects a vehicle data that is **actually the legitimate** and is predicted to be **the legitimate**.

False Positive (FP) : The system detects a vehicle data that is **actually the legitimate** and is predicted to be **the attacker**.

False Negative (FN) : The system detects a vehicle data that is **actually the attacker** and is predicted to be **the legitimate**.

We will arrange the possible results in a confusion matrix. This matrix will produce parameters that are useful for measuring the performance of a detection system. We can see this in matrix 4.1.

Table 4.1: Confusion Matrix of Detection Result

		Predict Value	
		Attacker	Legitimate
Actual Value	Attacker	**TP**	**FN**
	Legitimate	**FP**	**TN**

To measure performance based on the confusion matrix, we will use the following ratios parameter:

Recall

Recall is a comparison between True Positive and the number of data that are positive or we can call it the *sensitivity to detect attacker*. This could mean that *Recall* answers the question, "What percentage of vehicles are predicted to be the attackers compared to the total number of vehicles that are actually the attackers?".

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

Precision

Precision is the ratio between True Positive and the amount of data that is predicted to be positive (Its means "Attacker Detected" = Positive). In other words, precision can be interpreted as *the ability to distinguish between attackers and legitimate ones*. We can say that *Precision* answers the question, "What percentage of the vehicles were actually attackers out of the total predicted as attackers?"

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

F_1-Score

F_1-*Score* is the harmonic average between *Precision* and *Recall*. The best value for F_1-*Score* is 1.0 and the worst value is 0. In representation, if the F_1-*Score* has a good score, it indicates that our classification model has good *Precision* and *Recall*. F_1-*Score* becomes a good performance indicator of an ML model if the dataset is not balanced.

$$F_1 - Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{4.3}$$

Accuracy

Accuracy is the ratio of all data that is correctly detected, whether it is detected as an attacker's vehicle or correctly as a legitimate vehicle. In other words, *Accuracy* answers the question, "What percentage of vehicles are correctly predicted as attackers and legitimate from all vehicles?".

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.4}$$

4.4 2-Step History Prediction

4.4.1 Implementation Setup

The real-time simulation will be run and developed using an established real-time simulation application. At this stage, we modify the F^{2}MD [44] application so that, it can run all machine learning model classifiers that are the tasks of the current research works.

The Parameters of real-time simulation are as follows:

- Software environment : OMNET++ v.5.4, SUMO 1.10.0*
- Protocol communication : ITS-G5 (IEEE 802.11p)
- Duration : 86400 second (24 hours)
- Type Attacker : all of attacker type (mix)
- Scenario : UPHF Map
- Attacker density : 10% and 30%
- Format data input : Vehicle history position and speed (5 messages and 30 messages)
- ML model : DBN, LSTM, GRU and RF

The use of 5_{msg} and 30_{msg} configurations is intended to see changes from the lowest accuracy to the highest accuracy according to the results of table 3.6. In real-time simulations, attacker density is defined as the number of attack vehicles divided by the total number of vehicles. The default on F^{2}MD framework is 5%, so for this study, we use 10% and 30% attacker density intending to increase the chances of detecting attack vehicles and also concerning evaluate the impact of raising the number of attackers on each ML model's performance. To adapt our anomalies detection system to the new map scenario, all ML model classifiers are retrained using data generated from the UPHF map scenario.

Implementation in a real-time simulation environment has to consider real-time data and training data-sets are in different formats, even though they have the same substance. To be able to retrieve real-time data by the prediction model, an intermediary algorithm is needed (see algorithm 4). This algorithm works by filtering the received BSM and then retrieving the main information such as vehicle ID, vehicle type (legitimate or attacker), vehicle position, and speed. Especially for position and speed data, the format will be changed according to the prediction model that will be used.

Algorithm 4 2-Step History Prediction Implementation Algorithm for Real-time Simulation

```
1: function convertBSM
2:     Pass in : position, speed
3:     pos.array ← position(x,y)
4:     speed.array ← speed(x,y)
5:     Pass out : pos.array, speed.array
6: end function
7: bsmLoad ← loads(bsmDataStrem)
8: clf1 ← load(1st classifier)
9: clf2 ← load(2nd classifier)
10: idVehicle ← bsmLoad[VehRealID]
11: createTime ← bsmLoad[MsgCreateTime]
12: posBSM ← bsmLoad[Position]
13: speedBSM ← bsmLoad[Speed]
14: pos.array = convertBSM ← posBSM
15: speed.array = convertBSM ← speedBSM
16: tempVehicle ← idVehicle
17: target ← tempVehicle[a]
18: if idVehicle = target then
19:     if CrTime != tempCrTime then
20:         tempCrTime ← createTime
21:         X.Pos ← pos.array
22:         X.Speed ← speed.array
23:         X.Vehicle ← join(X.Pos, X.Speed)
24:         YPrediction1 ← clf1.predict(X.Vehicle)
25:         if Yprediction == 1 then
26:             YPrediction2 ← clf2.predict(X.Vehicle)
27:             Attacker ← X.Vehicle
28:         else
29:             Legitimate ← X.Vehicle
30:         end if
31:     end if
32: end if
```

4.4.2 Evaluation

In this simulation, we present two types of tables for each simulation under different attacker densities:

1. Result of *Detection of The Attackers* (Table 4.2 and 4.4). For 2-Step History Prediction we can say that this is a *1st prediction*. These tables show the ability of each ML model in **distinguishing the attacker's vehicle from the legitimate vehicle**.

2. Result of *Identification of the Attackers* (Table 4.3 and 4.5). For 2-Step History Prediction we can say that this is a *2nd prediction*. These tables show the ability of each ML model in **distinguishing the types of attacks**, including distinguishing them from legitimate vehicles. In other words, the ability to classify attack types is measured in this table.

4.4.2.1 Case 1 : 10% Density Attacker

Reviewing when the attacker density is 10% (table 4.2) and the number of messages is increased from 5 messages to 30 messages, each ML model has increased sensitivity in detecting the attacker's vehicle. It can be seen from the *Recall* value which has increased for all models, even LSTM, GRU, and RF can detect all attacker vehicles that appear. Even though, the number of attacker vehicles is much less than the legitimate vehicles.

Table 4.2: Result of *Attacker Detection* of 2-Step History Prediction (Real-Time 10% Density Attacker)

		Recall	**Precision**	**F_1-Score**	**Accuracy**
DBN	5_{msg}	0.3333	0.0811	0.1304	0.6226
	30_{msg}	0.75	1	0.8571	0.9773
LSTM	5_{msg}	0.6667	0.1304	0.2182	0.5943
	30_{msg}	1	0.1818	0.3077	0.5909
GRU	5_{msg}	0.625	0.1042	0.1786	0.578
	30_{msg}	1	0.6667	0.8	0.9545
RF	5_{msg}	0.7778	0.3684	0.5	0.8679
	30_{msg}	1	0.5	0.6667	0.9091

On the other hand, the LSTM *Precision* value does not experience a significant increase. In the case of the 30 msg, the F_1-*Score* value and its *Accuracy* are the lowest compared to other models, this indicates that in real-time implementation, LSTM does not have a good performance in differentiating between attacker vehicle and legitimate vehicle when there is only a small number of attacker vehicle emerging.

In table 4.3 it will be more concerned with *Recall* value because the classification performance is influenced by changes in the *True Positive* and *False Negative* values. If there is an incorrect classification by the 2nd prediction, it will reduce the TP value and increase the FN value, causing the *Recall* value to decrease.

However, we can see that in the use of 5_{msg}, the ability of each model to classify types of attacks, decreases and looks very low in value. So the use of 5_{msg} to classify low-density attackers is less effective. As for the use of 30_{msg}, the decrease only occurred in DBN.

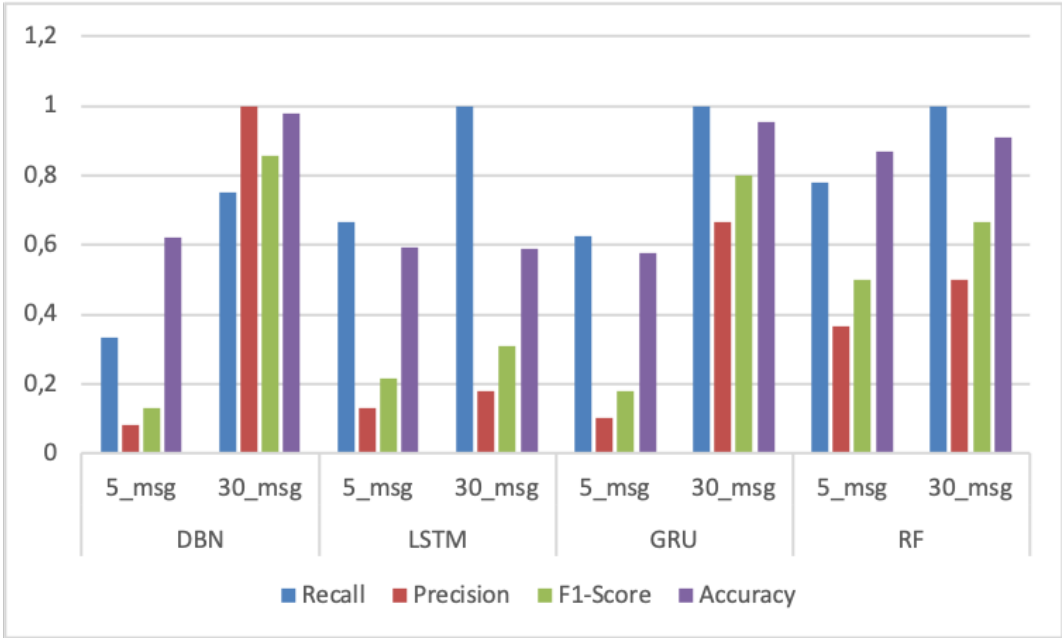


Figure 4.7: Graphic of *Attacker Detection* of 2-Step History Prediction (Real-Time 10% Density Attacker)

Table 4.3: Result of *Attacker Identification* of 2-Step History Prediction (Real-Time 10% Density Attacker)

		Recall	**Precision**	F_1**-Score**	**Accuracy**
DBN	5_{msg}	0.2222	0.0556	0.0889	0.6132
	30_{msg}	0.5	1	0.6667	0.9545
LSTM	5_{msg}	0.3333	0.0698	0.1154	0.566
	30_{msg}	1	0.1818	0.3077	0.5909
GRU	5_{msg}	0.5	0.0851	0.1455	0.5688
	30_{msg}	0.75	0.6	0.6667	0.9318
RF	5_{msg}	0.6667	0.3333	0.4444	0.8585
	30_{msg}	1	0.5	0.6667	0.9091

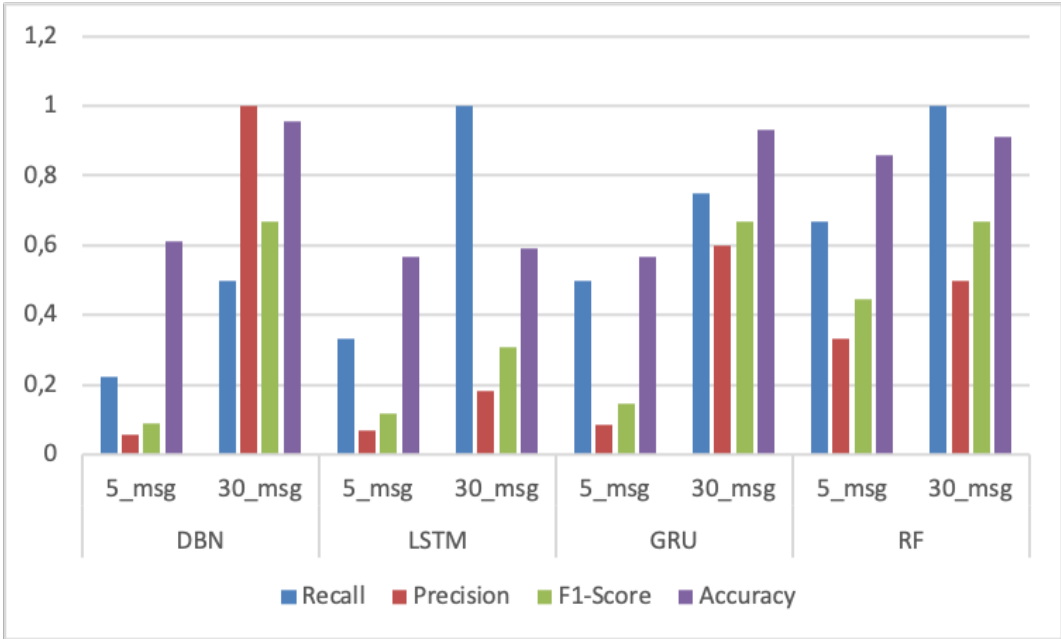


Figure 4.8: Graphic of *Attacker Identification* of 2-Step History Prediction (Real-Time 10% Density Attacker)

4.4.2.2 Case 2 : 30% Density Attacker

Considering the addition of the density of attackers to 30% (comparing table 4.4 and table 4.2), we see that at 5_{msg}, DBN, LSTM, and GRU experienced an increase in attack detection sensitivity, indicated by an increase in the *Recall* value. Meanwhile, the sensitivity of RF has decreased slightly. The ability to distinguish between attack vehicles and legitimate vehicles is also improved for all models. This is indicated by the increase in the *Precision* value compared to the 10% density condition. This is caused by an increase in the value of TP when there is an expansion in the number of density attackers.

Table 4.4: Result of *Attacker Detection* of 2-Step History Prediction (Real-Time 30% Density Attacker)

		Recall	**Precision**	F_1**-Score**	**Accuracy**
DBN	5_{msg}	0.75	0.5455	0.6316	0.6111
	30_{msg}	0.2857	1	0.4444	0.6552
LSTM	5_{msg}	0.8333	0.5738	0.6796	0.6333
	30_{msg}	0.8214	0.7419	0.7797	0.7759
GRU	5_{msg}	0.963	0.5652	0.7123	0.7123
	30_{msg}	0.6842	0.8667	0.7647	0.8261
RF	5_{msg}	0.7241	0.6562	0.6885	0.7432
	30_{msg}	0.7222	0.9286	0.8125	0.8696

Still, in comparison between table 4.4 and table 4.2, accuracy at 5_{msg} also improves for every model except RF as the detection sensitivity decreases. At 30_{msg}, all models experienced a decrease in attack detection sensitivity. At a density of 30%, the number of attacks increases 3 times, and 30_{msg} requires a longer detection time than 5_{msg} resulting in more attacks that can be falsely detected. However, the ability

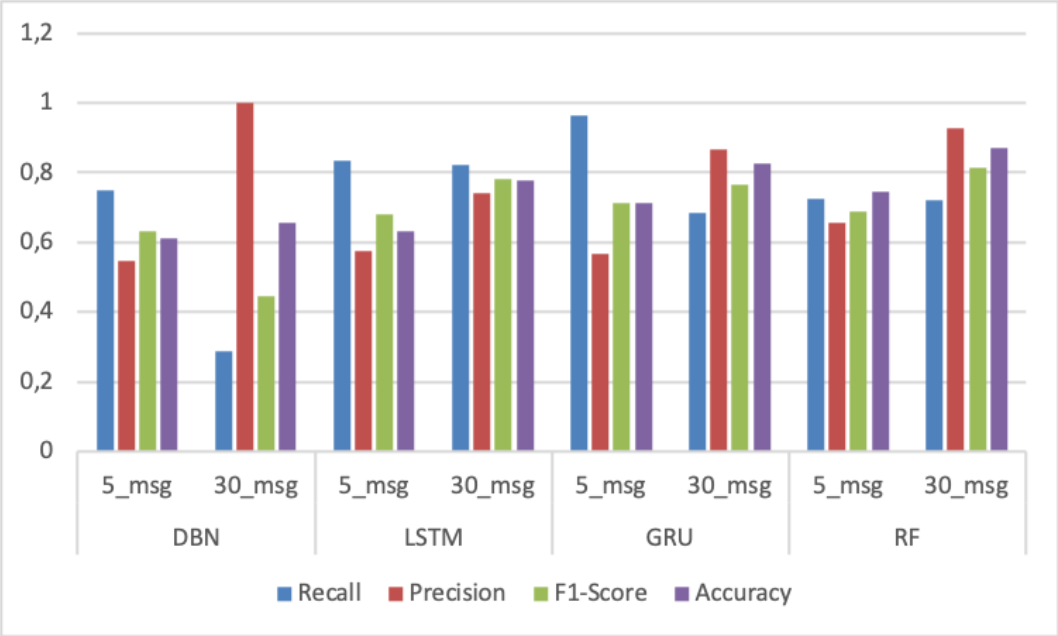


Figure 4.9: Graphic of *Attacker Detection* of 2-Step History Prediction (Real-Time 30% Density Attacker)

to distinguish between attack vehicles and legitimate vehicles increased significantly, especially for LSTM. it can be seen from the LSTM. *Precision* and F1 Score at a density of 30% which increased significantly both at 5_{msg} and at 30_{msg}, compared to a density of 10%. This makes the LSTM's performance closer to GRU and RF in terms of distinguishing between attacking and legitimate vehicles (see table 4.4).

To see the performance of the classification of attack types in 30% density attacker, we can review the *Recall* value in the table 4.5. It can be seen that the classification ability of all models has decreased, meaning that each model has difficulty in distinguishing the type of attack, especially at 5_{msg}. However, at the use of 30_{msg}, only LSTM decreased slightly, while for other models the *Recall* value decreased significantly. This makes the LSTM *Recall* value at 30_{msg} is the highest, this means that LSTM has a better ability in terms of classifying types of attacks than DBN, GRU, and RF. But for total accuracy, GRU and RF are somewhat better than LSTM.

Table 4.5: Result of *Attacker Identification* of 2-Step History Prediction (Real-Time 30% Density Attacker)

		Recall	**Precision**	**F_1-Score**	**Accuracy**
DBN	5_{msg}	0.275	0.3056	0.2895	0.4
	30_{msg}	0.1429	1	0.25	0.5862
LSTM	5_{msg}	0.3095	0.3333	0.321	0.3889
	30_{msg}	0.7857	0.7333	0.7586	0.7586
GRU	5_{msg}	0.4074	0.3548	0.3793	0.5068
	30_{msg}	0.5789	0.8462	0.6875	0.7826
RF	5_{msg}	0.4828	0.56	0.5185	0.6486
	30_{msg}	0.5	0.9	0.6429	0.7826

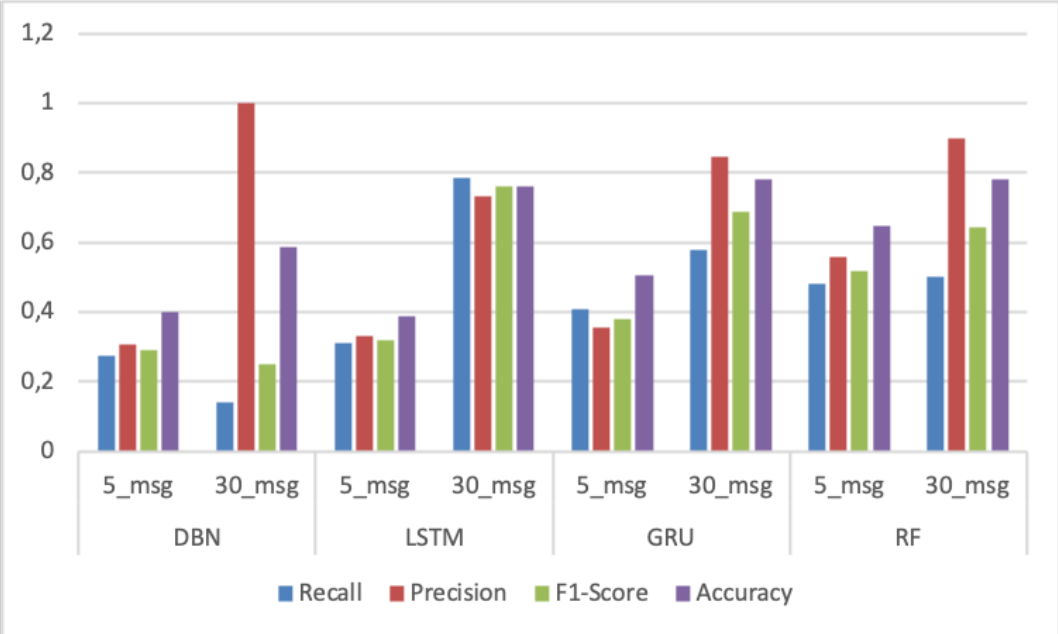


Figure 4.10: Graphic of *Attacker Identification* of 2-Step History Prediction (Real-Time 30% Density Attacker)

4.5 2-Step 2-D BSM Prediction

4.5.1 Implementation Setup

In 2-Step 2-D BSM Prediction, the application platform for real-time implementation still uses the same application as 2-Step History Prediction. There is only a slight difference in parameters.

The Parameters of real-time simulation are as follows:

- Software environment : OMNET++ v.5.4, SUMO 1.10.0*
- Protocol communication : ITS-G5 (IEEE 802.11p)
- Duration : 86400 second (24 hours)
- Type Attacker : all of attacker type (mix)
- Scenario : UPHF Map
- Attacker density : 10% and 30%
- Format data input : 2-Dimension BSM
- ML model : ResNet152V2 and MobileNet

For the same reason as 2-Step History Prediction, in 2-Step 2-D BSM Prediction, the UPHF-Map scenario is also used, see figure 4.6. Like 2-Step History Prediction, 2-Step 2-D BSM Prediction also requires an algorithm 5 to modify the F^2MD application. This algorithm will converting BSM streaming into a 2-D format using shifting technique. BSM consist of Message Create Time, Message Arrival Time, Data Position, Position Confident, Speed, Speed Confident, Acceleration, Acceleration Confident, Heading, and Heading Confident. This 2-D format will be the input for the ResNet152V2 and MobileNet ML classifiers.

Algorithm 5 2-Step 2-D BSM Prediction Implementation Algorithm for Real-time Simulation

```
bsmLoad ← loads(bsmDataStrem)
clf_net ← load(net_classifier)
bsm_in ← bsmLoad[MsgCrTime,MsgArvTime, Pos, Pos_Conf, Spd, Spd_Conf, Acl, Acl_Conf, Head, Head_Conf]
bsm_array ← bsm_in.array
X_input ← array.zero[bsm_array.row,32]
for i = 0 to bsm_array.row do
    x ← bsm_array[i,30].array
    sp ← bsm_array.col
    X_input[i,sp] ← x
end for
X_temp ← array.zero[X_input.row,,32,32]
for j = 0 to X_input.row do
    for t = 0 to X_temp[j].row do
        if t==0 then
            X_temp[j,t] ← X_input[j]
        else
            X_temp[j,t] ← array.shifting(X_temp[j,t-1])
        end if
    end for
end for
X.BSM ← X_temp
YPrediction ← clf_net.predict(X.BSM)
if YPrediction ≠ 0 then
    YPrediction2 ← clf2_net.predict(X.BSM)
    Attacker ← X.BSM
else
    Legitimate ← X.BSM
end if
```

4.5.2 Evaluation

In this simulation, we present two types of tables for each simulation under different attacker densities:

1. Result of *Detection of The Attackers* (Table 4.6 and 4.8). These tables show the ability of each ML model in distinguishing the attacker's vehicle from the legitimate vehicle.

2. Result of *Identification of the Attackers* (Table 4.7 and 4.9). These tables show the ability of each ML model in distinguishing the types of attacks, including distinguishing them from legitimate vehicles. In other words, the ability to classify attack types is measured in this table.

4.5.2.1 Case 1 : 10% Density Attacker

The implementation of the ML model for Attacker Detection at a density of 10% show that both ResNet152V2 and MobileNet has similar value. See table 4.6. The highest accuracy is 75.33% by MobileNet. At this stage, the number of legitimate messages is much larger than the attacker's, affecting the low Precision value for both ML models. However, let's look at the much higher Accuracy. We can conclude that the system can detect legitimate messages correctly, far above the error in predicting legitimate attackers. Then the recall value, which is still relatively high, is almost the same as the accuracy value, indicating that the sensitivity level of the system is still quite good in recognizing an attacker's message.

Table 4.6: Result of *Attacker Detection* of 2-Step 2-D BSM Prediction (Real-Time 10% Density Attacker)

	Recall	**Precission**	**F1_Score**	**Accuracy**
ResNet152V2	0,7340	0,2445	0,3668	0,7490
MobileNet	0,7093	0,2391	0,3577	0,7533

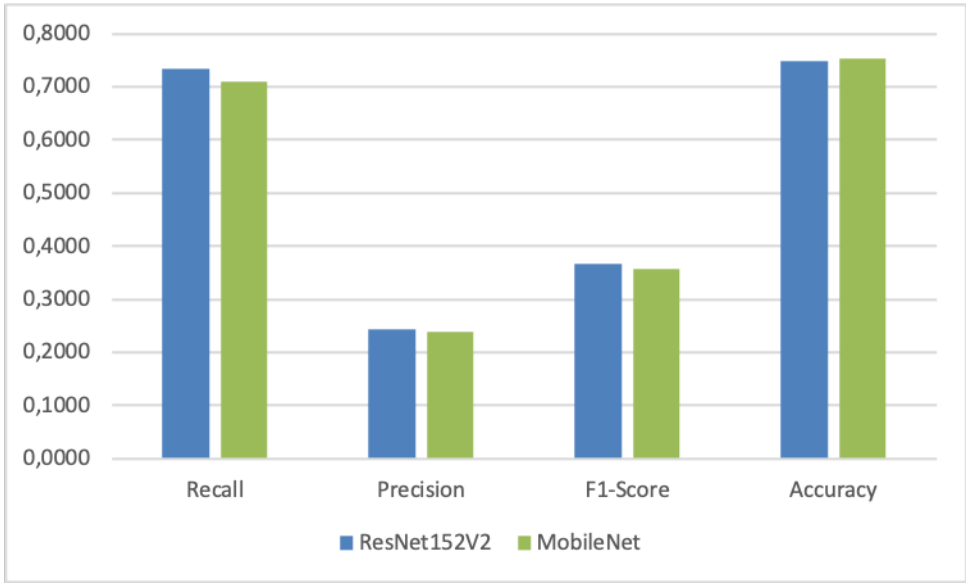


Figure 4.11: Graphic of *Attacker Detection* of 2-Step 2-D BSM Prediction (Real-Time 10% Density Attacker)

However, this is different when viewed from the model's accuracy in determining the attack type. This can be seen in table 4.7. ResNet152V2 recalls were drastically reduced to 35.07%. This means that only 35.07% of the attacker's BSM successfully identified the type of attack correctly. This significant decrease in recall is due to the fact that, the 2-Step 2D BSM prediction task is heavy enough to classify 19 types of attacks simultaneously correctly. However, the level of accuracy did not decrease significantly, meaning that more messages that could be detected correctly by the system were higher than those that the system incorrectly predicted.

Table 4.7: Result of *Attacker Identification* of 2-Step 2-D BSM Prediction (Real-Time 10% Density Attacker)

	Recall	**Precission**	**F1_Score**	**Accuracy**
ResNet152V2	0,3534	0,1282	0,1882	0,7133
MobileNet	0,3115	0,1140	0,1669	0,7175

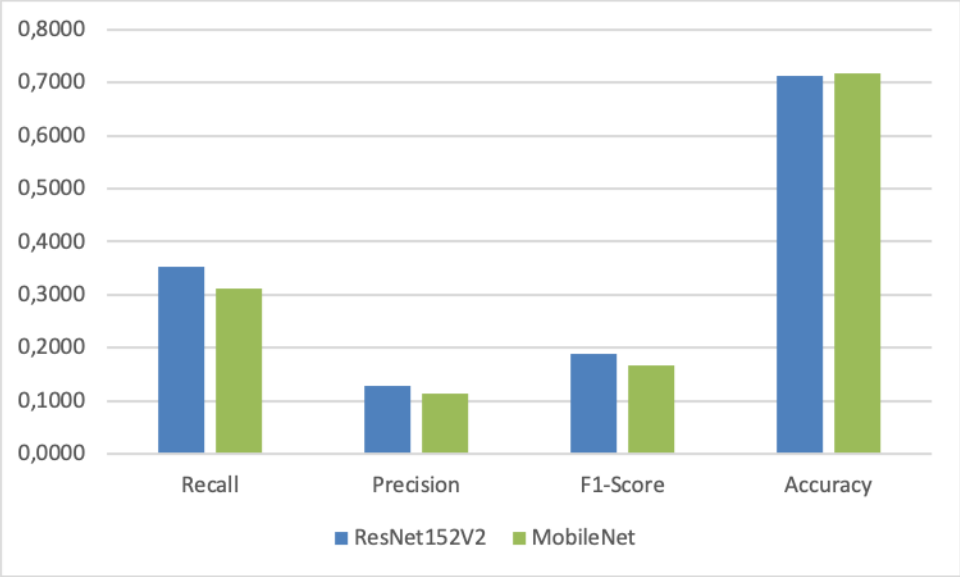


Figure 4.12: Graphic of *Attacker Identification* of 2-Step 2-D BSM Prediction (Real-Time 10% Density Attacker)

4.5.2.2 Case 1 : 30% Density Attacker

The assumption that the ML model has decreased accuracy is quite large because the number of attacking BSM is much less than the legitimate BSM, can be seen in the 4.8 table. When the attacker's density is increased to 30% the precision value increases considerably. As in ResNet152V2 which increased into 75% compared to when the density was 10%. This shows that the model's ability to distinguish the attacker's BSM from the legitimate BSM is increasing. And when viewed from the Recall value, there are no significant decrease. This means that the sensitivity of the model in detecting is quite the same. In the case of 30% density attacker, both ML models, both ResNet152V2 and MobileNet, get better performance than before, namely in terms of the ability to distinguish between legitimate BSM and attacker BSM, which has improved quite well.

In detecting the type of attack in table 4.9 the recall value has decreased, as is the case with 2-Step history prediction. This is due to the fact that various kinds of attacks are detected with the wrong type, which will increase the FN value, which

Table 4.8: Result of *Attacker Detection* of 2-Step 2-D BSM Prediction (Real-Time 30% Density Attacker)

	Recall	**Precission**	**F1_Score**	**Accuracy**
ResNet152V2	0,6873	0,7511	0,7178	0,8103
MobileNet	0,7403	0,7913	0,7649	0,8298

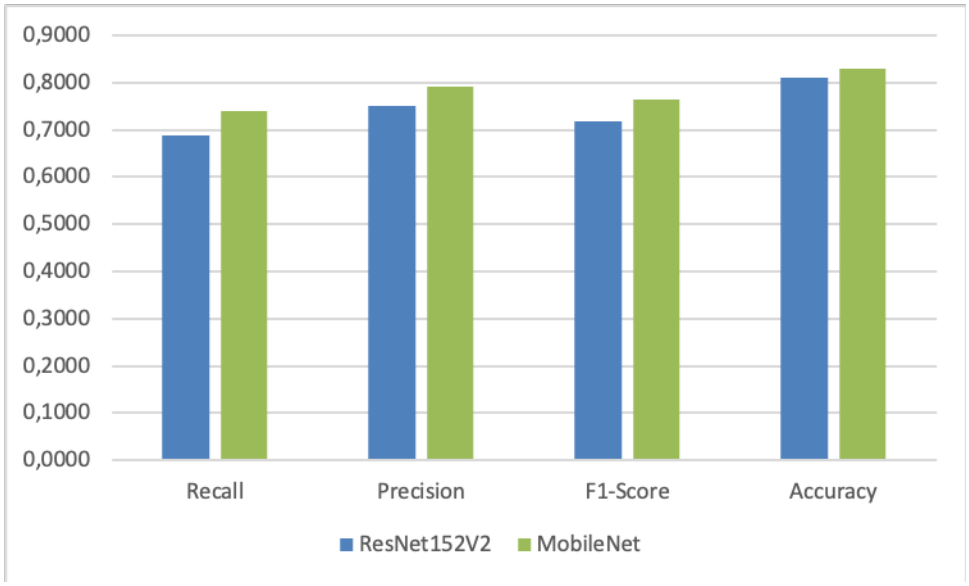


Figure 4.13: Graphic of *Attacker Detection* of 2-Step 2-D BSM Prediction (Real-Time 30% Density Attacker)

automatically causes the recall value to decrease. However, if we look at the accuracy, there is not much decrease, meaning that the BSM detected correctly is still the majority of the prediction results.

Table 4.9: Result of *Attacker Identification* of 2-Step 2-D BSM Prediction (Real-Time 30% Density Attacker)

	Recall	**Precission**	**F1_Score**	**Accuracy**
ResNet152V2	0,2215	0,4805	0,3032	0,6542
MobileNet	0,1962	0,4823	0,2790	0,6382

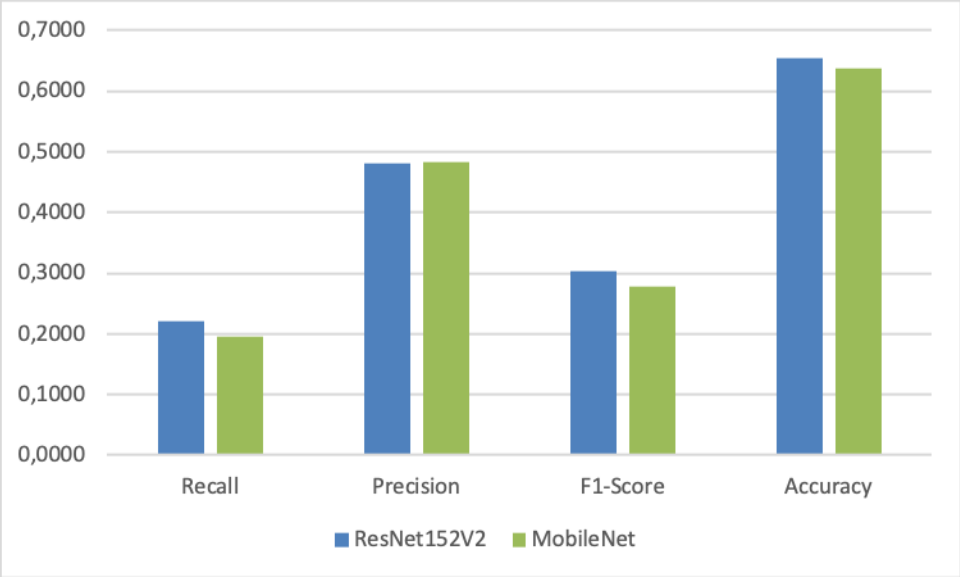


Figure 4.14: Graphic of *Attacker Identification* of 2-Step 2-D BSM Prediction (Real-Time 30% Density Attacker)

4.6 Conclusion

There is a decrease in *Accuracy* in the real-time simulation compared to the simulation in subsection 3.4. This is due to the data created in the real-time simulation is always changing and never the same from time to time, particularly the vehicle's position and speed data, which are the study's major parameters. Especially with an unbalanced data distribution, where the amount of legitimate data is far more than the attacker's data, they are quite influential in changing the *Accuracy* of each ML model. Considering that there are quite a lot of different types of attacks that must be distinguished.

In *real-time systems* implementation, all ML models have decreased in terms of accuracy, considering the complexity is relatively high. However, by using only the vehicle's position and speed information, 2-Step History Prediction can distinguish between the attacker's vehicle and the legitimate vehicle with an accuracy rate of 87% and 83%, respectively, for Random Forest and GRU. LSTM also has an advantage in the level of sensitivity to distinguish types of attacks compared to other ML models. Although in terms of total accuracy, it is still below GRU and Random Forest. While the lowest is DBN. It's also worth mentioning that the DBN and RF models have the fastest detection speeds, while the LSTM and GRU models have the slowest.

Another conclusion based on the implementation in real-time systems, an increment of messages and density of attackers can affect the detection performance of the machine learning model. The higher the number of messages employed in the detection, the higher the level of accuracy, but on the other hand, the detecting speed will slow down. While the number of attackers increases, the type of attack gets more difficult to classify.

ResNet152 and MobileNet actually obtained quite good results at 30% density attackers compared to conditions at 10% density attackers. And also ResNet152V2 and MobileNet have their own advantages in real time detection. These two algorithms do not need to collect the history messages of a vehicle, but can directly process the detected BSM

Chapter 5

General Conclusion

Contents

5.1 Conclusion

In this work, we have presented our research work which results in the following conclusions:

- We are interested in big data from the track record of vehicle communication messages on the vehicular network when an attack or misbehavior occurs. Misbehavior is not only one but up to 19 types of misbehavior that appear in one scenario. Based on the existing database, we have proposed a 2-step history prediction that can predict the attacker's vehicles and distinguish them from legitimate vehicles. Prediction does not depend on a particular threshold value but is based on the history of the position and speed of the vehicle. Our method obtains the best results with 95% accuracy using the LSTM and GRU algorithms based on the VeReMi extension database.

- We also have proposed the 2-step 2-D BSM prediction method, which aims to predict misbehavior based on vehicle BSM. This method has been proven to perform classification for 19 types of attacks well for misbehavior datasets on VN. The method that utilizes the ResNet152V2 algorithm gets an accuracy of 97%, and the one that uses the MobileNet algorithm gets an accuracy of 96%.

- We are interested in the application of misbehavior prediction methods to real-time systems. 2-Step History Prediction which has been confirmed to be implemented in real-time simulations to detect attacking vehicles and distinguish them from legitimate ones. At this stage, it is also demonstrated that the more position and speed history that is used to input the detection system, the higher the accuracy will be. However, this has the consequence of a longer detection time. The best algorithm in this method is GRU which shows good performance. If the attacker's density increases, the RF algorithm can also be a good alternative. The advantage of 2-Step History Prediction in real-time implementation is that the file system classifier is relatively small and does not overload the system, so it is suitable to be applied to limited resources.

- We have also proposed a 2-Step 2-D BSM Prediction method implemented in real-time. This method gives good results and can predict messages from attacking vehicles and distinguish them from messages from legitimate vehicles. The ResNet152V2 and MobileNet algorithms produce better performance when the density of the attacker's vehicle increases. The advantage of 2-Step 2-D BSM Prediction in real-time implementation is the speed in making decisions on predictions which is relatively fast because it is only based on a message. However, the classifier file system is quite large, especially ResNet152V2, so it requires more significant resources than 2-Step History Prediction.

5.2 Perspective

- Develop 2-Step History Prediction using historical vehicle acceleration and heading data, then compare it to existing systems.
- Develop 2-Step Prediction by combining different Machine Learning in one system model for real-time simulation, then compared to existing systems. For example 1st prediction using GRU and second prediction using RF and etc.
- Improve timing detection and memory management in the real time simulation, to get optimal detection capacity and performance.
- Validate all system predictions on the larger map with more vehicles, such as big city map in France.

Acronyms

3GPP 3rd Generation Partnership Project.

AI Artificial Intelligent.

ANN Artificial Neural Networks.

AV Autonomous Vehicle.

AVh Autonomous Vehicle.

BSM Basic Safety Messages.

CAV Connected Autonomous Vehicles.

C-V2X Cellular V2X.

CACC Cooperative Adaptive Cruise Control.

C-ITS Cooperative ITS.

CV Connected Vehicle.

D2D Device to Device.

DATP Driver Assistive Truck Platooning.

DBN Deep Belief Network.

DNN Deep Neural Network.

DoS Denial of Service.

DOT Department of Transportation.

DSRC Dedicated Short-Range Communication.

ETSI European Telecommunications Standards Institute.

EV Electric Vehicle.

FHSS Frequency Hopping Spread Spectrum.

FHWA Federal Highway Administration.

GAN Generative Adversarial Network.

GNN Graph Neural Network.

GPS Global Positioning System.

GRU Gate Reccurent Unit.

HSDPA High Speed Downlink Packet Access.

IDS Intrusion Detection System.

IoT Internet of Things.

IRU International Road Transport Union.

ITS Intelligent Transportation System.

IVP Infrastructure-to-Pedestrian.

KNN K-Nearest Neighbors.

LSTM Long Short-Term Memory.

LTA Land Transport Authority.

LTE Long Term Evolution.

LuST Luxembourg SUMO Traffic.

MA Misbehavior Authority.

MAC Media Access Control.

MACs Message Authentication Codes.

MBD MisBehavior Detection.

MDS MisBehavior Detection System.

MFAs Message Falsification Attacks.

MITM Man In The Middle Attack.

ML Machine Learning.

MLP Mutli-Layer Perceptron.

NDRL New Deep Reinforcement Learning.

OBU On-Board Unit.

PKI Public Key Infrastructurs.

PSAP Public-Safety Answering Point.

PVRS Position Verification using Relative Speed (PVRS.

R2L Remote to Local.

RBM Restricted Boltzmann Machines.

RF Random Forest.

RFID Radio-Frequency Identification.

RSU Road Side Unit.

SDN Software-Defined Networking.

SpaT Multipath Signal Phase and Timing.

SV Smart Vehicle.

SVM Support Vector Machine.

TraCI Traffic Control Interface.

TSB Topollogically-Scoped Broadcast.

U2R User to Root.

UMTRI Michigan Transportation Research Institute.

UMTS Universal Mobile Telecommunications System.

V2D Vehicle-to-Device.

V2G Vehicle-to-Grid.

V2I Vehicle-To-Infrastructure.

V2P Vehicle-to-Pedestrian.

V2R Vehicle-to-Roadside.

V2V Vehicle-To-Vehicle.

V2X Vehicle-To-Everything.

VeReMi Vehicular Reference Misbehavior.

VN Vehicular Network.

VRU Vulnerable Road Users.

WPAN Wireless Personal Area Network.

Bibliography

[1] Emad E. Abdallah, Wafa' Eleisah, and Ahmed Fawzi Otoom. Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey. *Procedia Computer Science*, 201:205–212, 2022. 44

[2] Indonesian Central Statistics Agency. Traffic accident, killed person, seriously injured, slight injured and expected of material losses value, indonesia 1992-2018. https://www.bps.go.id/dynamictable/2016/02/09/1134/jumlah-kecelakaan-koban-mati-luka-berat-luka-ringan-dan-kerugian-materi-yang-diderita-tahun-1992-2017.html. 14

[3] albertbup. A python implementation of deep belief networks built upon numpy and tensorflow with scikit-learn compatibility, 2017. Available at https://github.com/albertbup/deep-belief-network [Online]. 47

[4] Moayad Aloqaily, Safa Otoum, Ismaeel Al Ridhawi, and Yaser Jararweh. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks*, 90:101842, July 2019. 35, 47

[5] José J. Anaya, Edgar Talavera, Felipe Jiménez, José G. Zato, Nuria Gómez, and José E. Naranjo. Geonetworking based v2v mesh communications over wsn. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 2421–2426, 2013. 27

[6] Jason Andress. *The Basics of Information Security, CHAPTER 10 Network Security*, page 157–158. Syngress, 2nd edition, 2015. 44

[7] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, Alessandro Guido, and Mirco Marchetti. On the effectiveness of machine and deep learning for cyber security. In *2018 10th International Conference on Cyber Conflict (CyCon)*, pages 371–390, Tallinn, May 2018. IEEE. 14

[8] Fabio Arena and Giovanni Pau. An Overview of Vehicular Communications. *Future Internet*, 11(2):27, January 2019. 27

[9] Singapore Land Transport Authority. Joint release by the land transport authority, jtc a*star - a savi step towards autonomous transport. https://www.lta.gov.sg/content/ltagov/en/newsroom/2014/8/2/joint-release-by-the-land-transport-authority-jtc-astar---a-savi-step-towards-autonomous-transport.html, 2014. 25

[10] James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. 51

[11] Debby Bezzina and James Sayer. Safety pilot model deployment. https://www.nhtsa.gov/sites/nhtsa.gov/files/812171-safetypilotmodeldeploydeltestcondrtmrep.pdf, 2015. 23

[12] Alois Bissuel. Hyper-parameter optimization algorithms: A short review. https://medium.com/criteo-engineering/hyper-parameter-optimization-algorithms-2fe447525903, Apr 2019. 51

[13] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. 46

[14] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in vanet. pages 19–28, 01 2007. 32, 34

[15] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv:1406.1078 [cs, stat]*, September 2014. arXiv: 1406.1078. 48

[16] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*, December 2014. arXiv: 1412.3555. 48

[17] Joyce Dargay, Dermot Gately, and Martin Sommer. Vehicle Ownership and Income Growth, Worldwide: 1960-2030. *The Energy Journal*, 28(4), October 2007. 14

[18] Davis David. Random forest classifier tutorial: How to use tree-based algorithms for machine learning. https://www.freecodecamp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/, Aug 2020. 9, 46

[19] Jose Maria de Fuentes, Jorge Blasco, Ana Isabel González-Tablas, and Lorena González-Manzano. Applying information hiding in vanets to covertly report misbehaving vehicles. *International Journal of Distributed Sensor Networks*, 10(2):120626, 2014. 32

[20] José María De Fuentes, Ana González-Tablas Ferreres, and Arturo Ribagorda. Overview of security issues in vehicular ad-hoc networks. *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts*, 01 2011. 33

[21] Rian Dolphin. Lstm networks: A detailed explanation. https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9, Dec 2021. 9, 47, 48

[22] Enguo Dong and Lei Zhang. Vehicle stability control system of emergency brake on split-mu road. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 1, pages 252–255, 2017. 27

[23] Zeinab El-Rewini, Karthikeyan Sadatsharan, Niroop Sugunaraj, Daisy Flora Selvaraj, Siby Jose Plathottam, and Prakash Ranganathan. Cybersecurity Attacks in Vehicular Sensors. *IEEE Sensors Journal*, pages 1–1, 2020. 14

[24] Masayuki Endo and Kenji Tanaka. Evaluation of storage capacity of electric vehicles for vehicle to grid considering driver's perspective. In *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / ICPS Europe)*, pages 1–5, 2018. 27

[25] Richard Gilles Engoulou, Martine Bellaïche, Samuel Pierre, and Alejandro Quintero. Vanet security surveys. *Computer Communications*, 44:1–13, 2014. 32, 33

[26] The Federal Highway Administration (FHWA). Expanding the freight capacity of america's highways platooning and connectivity to increase efficiency. https://www.fhwa.dot.gov/publications/research/ear/17045/index.cfm, 2017. 24

[27] Sohan Gyawali and Yi Qian. Misbehavior Detection using Machine Learning in Vehicular Communication Networks. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, Shanghai, China, May 2019. IEEE. 35

[28] Sohan Gyawali, Yi Qian, and Rose Qingyang Hu. Machine Learning and Reputation based Misbehavior Detection in Vehicular Communication Networks. *IEEE Transactions on Vehicular Technology*, pages 1–1, 2020. 45, 46

[29] Dalton A. Hahn, Arslan Munir, and Vahid Behzadan. Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. *IEEE Intelligent Transportation Systems Magazine*, pages 1–1, 2019. 33

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 48

[31] Carlos Hidalgo, Myriam Vaca, Mateusz P. Nowak, Piotr Frölich, Martin Reed, Mays Al-Naday, Asterios Mpatziakas, Aikaterini Protogerou, Anastasios Drosou, and Dimitrios Tzovaras. Detection, control and mitigation system for secure vehicular communication. *Vehicular Communications*, page 100425, October 2021. 36

[32] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, July 2006. 46

[33] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1, 1995. 45

[34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. 47

[35] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 9, 49, 50

[36] The Innovation and Networks Executive Agency (INEA). Nordicway. https://ec.europa.eu/inea/sites/default/files/fiche_2014-eu-ta-0060-s_final.pdf, 2015. 21

[37] The Innovation and Networks Executive Agency (INEA). Solred c-its monitoring network (solc-its). https://ec.europa.eu/inea/sites/default/files/fiche_2015-es-tm-0079-s_final.pdf, 2016. 21

[38] International Road Transport Union (IRU). I_heero - ecall Project. https://www.iru.org/what-we-do/being-trusted-voice-mobility-and-logistics/iru-projects/i-heero, 2015. 20

[39] Muhammad Awais Javed, Elyes Ben Hamida, Ala Al-Fuqaha, and Bharat Bhargava. Adaptive Security for Intelligent Transport System Applications. *IEEE Intelligent Transportation Systems Magazine*, 10(2):110–120, 2018. 14

[40] Felipe Jiménez. *INTELLIGENT VEHICLES Enabling Technologies and Future Developments*, volume I. Butterworth-Heinemann, Spain, i edition, September 2017. 29, 30, 31

[41] Diala Jomaa, Siril Yella, and Mark Dougherty. A comparative study between vehicle activated signs and speed indicator devices. *Transportation Research Procedia*, 22:115–123, 2017. "19th EURO Working Group on Transportation Meeting, EWGT2016, 5-7 September 2016, Istanbul, Turkey". 27

[42] Ronald Jurgen. *V2V/V2I Communications for Improved Road Safety and Efficiency*, pages i–viii. 2012. 28

[43] Debasish Kalita. An overview of deep belief network (dbn) in deep learning. https://www.analyticsvidhya.com/blog/2022/03/an-overview-of-deep-belief-network-dbn-in-deep-learning/, Mar 2022. 9, 46, 47

[44] J. Kamel. F^{2}md github repository, 2019. Available at https://github.com/josephkamel/F2MD.git [Online]. 81

[45] Joseph Kamel, Mohammad Raashid Ansari, Jonathan Petit, Arnaud Kaiser, Ines Ben Jemaa, and Pascal Urien. Simulation Framework for Misbehavior Detection in Vehicular Networks. *IEEE Transactions on Vehicular Technology*, 69(6):6631–6643, June 2020. 36, 48, 54, 74

[46] Joseph Kamel, Farah Haidar, Ines Ben Jemaa, Arnaud Kaiser, Brigitte Lonc, and Pascal Urien. A Misbehavior Authority System for Sybil Attack Detection in C-ITS. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1117–1123, New York City, NY, USA, October 2019. IEEE. 32

[47] Joseph Kamel, Farah Haidar, Ines Ben Jemaa, Arnaud Kaiser, Brigitte Lonc, and Pascal Urien. A Misbehavior Authority System for Sybil Attack Detection in C-ITS. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1117–1123. IEEE, October 2019. event-place: New York City, NY, USA. 39

[48] Joseph Kamel, Michael Wolf, Rens W. van der Hei, Arnaud Kaiser, Pascal Urien, and Frank Kargl. VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6, Dublin, Ireland, June 2020. IEEE. 11, 39, 53, 111, 114

[49] Brent Komer, James Bergstra, and Chris Eliasmith. Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn. pages 32–37, Austin, Texas, 2014. 51

[50] Dimitrios Kosmanos, Apostolos Pappas, Leandros Maglaras, Sotiris Moschoyiannis, Francisco J. Aparicio-Navarro, Antonios Argyriou, and Helge Janicke. A novel Intrusion Detection System against spoofing attacks in connected Electric Vehicles. *Array*, 5:100013, March 2020. 35

[51] Nur Cahyono Kushardianto, Yassin El Hillali, and Charles Tatkeu. 2-step prediction for detecting attacker in vehicle to vehicle communication. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pages 1–5, 2021. 9, 11, 53, 56, 57, 114

[52] Seigo Kuzumaki. Sip-adus automated driving for universal service. https://en.sip-adus.go.jp/, 2019. 9, 26

[53] Rémi Lang, Jung Hoon Lee, and Atsuko Okuda. *Intelligent Transportation Systems for Sustainable Development in Asia and the Pacific*. United Nations ESCAP, 2015. 14

[54] Huang Lu, Jie Li, and Mohsen Guizani. A novel id-based authentication framework with adaptive privacy preservation for vanets. In *2012 Computing, Communications and Applications Conference*, pages 345–350, 2012. 32, 34

[55] Rongxing Lu, Xiaodong Lin, Tom H. Luan, Xiaohui Liang, and Xuemin Shen. Pseudonym changing at social spots: An effective strategy for location privacy in vanets. *IEEE Transactions on Vehicular Technology*, 61(1):86–96, 2012. 33, 34

[56] I. Malygin, V. Komashinsky, and V. V. Tsyganov. International experience and multimodal intelligent transportation system of Russia. In *2017 Tenth International Conference Management of Large-Scale System Development (MLSD)*, pages 1–5, Moscow, October 2017. IEEE. 19

[57] Kimi Manchanda and Amarpreet Singh. Article: Covert communication in vanets using internet protocol header bit. *International Journal of Computer Applications*, 123(17):10–14, August 2015. Published by Foundation of Computer Science (FCS), NY, USA. 32

[58] Shimpei Matsumoto, Nobuyuki Ohhigashi, and Takashi Hasuike. Developing a transportation support system for vulnerable road users in local community. In *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, pages 797–800, 2016. 27

[59] Mohamed Nidhal Mejri, Jalel Ben-Othman, and Mohamed Hamdi. Survey on vanet security challenges and possible cryptographic solutions. *Vehicular Communications*, 1(2):53–66, 2014. 32, 33, 34

[60] France Minister of Transport. Scoop project : Connected road and vehicle. http://www.scoop.developpement-durable.gouv.fr/en/, May 2020. 21

[61] Anthony Obiri-Yeboah. Connected vehicles dsrc vs. c-v2x (in perspective to japan). https://www.linkedin.com/pulse/connected-vehicles-dsrc-vs-c-v2x-perspective-japan-obiri-yeboah/, Jan 2020. 29, 30

[62] Office of the Assistant Secretary for Research and Technology (OST-R). Cv pilot deployment program. https://www.its.dot.gov/pilots/technical_assistance_events.htm, 2018. 24

[63] Connecting Europe Facility of the European Union. C-roads - the platform of harmonised c-its deployment in europe. https://www.c-roads.eu/platform.html, 2016. 21

[64] France Ministry of Transport. Indid - infrastructure digitale de demain. https://www.c-roads.eu/pilots/core-members/france/Partner/project/show/indid.html, 2019. 9, 22, 23

[65] Indiana Dept. of Transportation. The maasto tpims project - indiana. https://www.in.gov/indot/files/TheMAASTO_TPIMSProject_FactSheet_2016_08_11.pdf, 2019. 9, 24, 25

[66] Department of Transportation of the United States of America. Vehicle-to-infrastructure (v2i) resources. https://www.its.dot.gov/v2i/index.htm, 2017. 28

[67] Thiago Rodrigues Oliveira, Cristiano M. Silva, Daniel F. Macedo, and José Marcos S. Nogueira. Snvc: Social networks for vehicular certification. *Computer Networks*, 111:129–140, 2016. Cyber-physical systems for Mobile Opportunistic Networking in Proximity (MNP). 27

[68] Seoul Transport Operation and Information Service. Seoul topis. https://topis.seoul.go.kr/eng/english.jsp, 2005. 24

[69] World Health Organization. Global status report on road safety 2018. https://www.who.int/publications/i/item/9789241565684, 2018. 14, 27

[70] Yuanyuan Pan and Jianqing Li. Cooperative pseudonym change scheme based on the number of neighbors in vanets. *Journal of Network and Computer Applications*, 36:1599–1609, 02 2013. 34

[71] Giovanni Pau, Tiziana Campisi, Antonino Canale, Alessandro Severino, Mario Collotta, and Giovanni Tesoriere. Smart pedestrian crossing management at traffic light junctions through a fuzzy-based approach. *Future Internet*, 10(2), 2018. 27

[72] Michael Phi. Illustrated guide to lstm's and gru's: A step by step explanation. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21, Jun 2020. 9, 48, 49

[73] Polaris. Electronic toll collection market size global report, 2022 - 2030. https://www.polarismarketresearch.com/industry-analysis/electronic-toll-collection-system-market, 2022. 31

[74] Abhijeet Pujara. Image classification with mobilenet. https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470, Jul 2020. 9, 50

[75] Kazi Atiqur Rahman and Kemal E. Tepe. Towards a cross-layer based mac for smooth v2v and v2i communications for safety applications in dsrc/wave based systems. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 969–973, 2014. 28

[76] Iftikhar Rasheed, Fei Hu, and Lin Zhang. Deep reinforcement learning approach for autonomous vehicle systems for maintaining security and safety using LSTM-GAN. *Vehicular Communications*, 26:100266, December 2020. 35

[77] Vinayakumar Ravi, Soman Kp, Mamoun Alazab, Sriram Srinivasan, and Simran Ketha. A comprehensive tutorial and survey of applications of deep learning for cyber security. 01 2020. 45

[78] The Transport Research, Innovation Monitoring, and Information System. C-its for trucks. https://trimis.ec.europa.eu/project/c-its-trucks. 22

[79] Soheyb Ribouh. *Identification de l'environnement basée sur l'estimation de canal et génération de clés de sécurité pour les communications véhiculaires.* PhD thesis, Université Polytechnique Hauts-de-France; Institut national des sciences . . . , 2020. 30

[80] Soheyb Ribouh, Yassin Elhillali, and Atika Rivenq. Multiple sequential constraint removal algorithm for channel estimation in vehicular environment. In *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–7. IEEE, 2020. 14

[81] Shin Sakaki, Kazunori Ooshima, Tomohito Imamura, and Yuji Ikeda. An interim report on joint research in developing technology for the realization of next-generation C-ITS. *ITS World Congress Singapore*, 26:10, October 2019. 26

[82] Fatih Sakiz and Sevil Sen. A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov. *Ad Hoc Networks*, 61:33–50, 2017. 34

[83] Faisal Saleem and Scott Nodes. Arizona emergency vii (e-vii). http://www.u.arizona.edu/~heqing/jjs_files/az-overview-Feb08.pdf. 23

[84] Roshan Sedar, Charalampos Kalalas, Francisco Vázquez-Gallego, Luis Alonso, and Jesus Alonso-Zarate. A comprehensive survey of v2x cybersecurity mechanisms and future research paths. *IEEE Open Journal of the Communications Society*, 2023. 32

[85] Jaydip Sen and Sidra Mehtab. Machine learning applications in misuse and anomaly detection. In *Security and Privacy From a Legal, Ethical, and Technical Perspective*. IntechOpen, sep 2020. 44

[86] Miguel Sepulcre, Javier Gozalvez, Onur Altintas, and Haris Kremo. Context-aware heterogeneous v2i communications. In *2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 295–300, 2015. 28

[87] Syed Sarmad Shah, Asad Malik, Anis Ur Rahman, Sohail Iqbal, and Samee Khan. Time barrier-based emergency message dissemination in vehicular adhoc networks. *IEEE Access*, PP:1–1, 01 2019. 9, 28

[88] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. *IEEE Transactions on Mobile Computing*, 10(1):3–15, 2011. 72

[89] Amin Tahmasbi-Sarvestani, Hossein Nourkhiz Mahjoub, Yaser P. Fallah, Ehsan Moradi-Pari, and Oubada Abuchaar. Implementation and evaluation of a cooperative vehicle-to-pedestrian safety application. *IEEE Intelligent Transportation Systems Magazine*, 9(4):62–75, 2017. 27

[90] Nikhil Tomar. What is residual network or resnet? https://medium.com/analytics-vidhya/what-is-residual-network-or-resnet-idiot-developer-6a1daa7c3b09, Oct 2021. 9, 49

[91] Sergio M. Tornell, Subhadeep Patra, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni. A novel on-board unit to accelerate the penetration of its services. In *2016 13th IEEE Annual Consumer Communications amp; Networking Conference (CCNC)*, page 467–472. IEEE Press, 2016. 27

[92] European Union. Intercor project. https://intercor-project.eu/, 2017. 22

[93] Rens W. van der Heijden, Thomas Lukaseder, and Frank Kargl. VeReMi: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs. *arXiv:1804.06701 [cs]*, April 2018. arXiv: 1804.06701. 36, 39, 53

[94] Baoyi Wang, Shan Sun, and Shaomin Zhang. Research on feature selection method of intrusion detection based on deep belief network. In *Proceedings of the 2015 3rd International Conference on Machinery, Materials and Information Technology Applications*, pages 556–561. Atlantis Press, 2015/11. 35

[95] Julia Silva Weber, Miguel Neves, and Tiago Ferreto. VANET simulators: an updated review. *Journal of the Brazilian Computer Society*, 27(1):8, May 2021. 72

[96] Celimuge Wu, Tsutomu Yoshinaga, Yusheng Ji, and Yan Zhang. Computational intelligence inspired data delivery for vehicle-to-roadside communications. *IEEE Transactions on Vehicular Technology*, 67(12):12038–12048, 2018. 27

[97] Takahito Yoshizawa, Dave Singelée, Jan Tobias Muehlberg, Stephane Delbruel, Amir Taherkordi, Danny Hughes, and Bart Preneel. A survey of security and privacy issues in v2x communication systems. *ACM Computing Surveys*, 55(9):1–36, 2023. 32

[98] Kexiong Curtis Zeng, Yuanchao Shu, Shinan Liu, Yanzhi Dou, and Yaling Yang. A practical gps location spoofing attack in road navigation scenario. In *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, HotMobile '17, page 85–90, New York, NY, USA, 2017. Association for Computing Machinery. 32

[99] Guangzhen Zhao, Cuixiao Zhang, and Lijuan Zheng. Intrusion Detection Using Deep Belief Network and Probabilistic Neural Network. In *22017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 639–642, Guangzhou, China, July 2017. IEEE. 35

[100] Man Zhou, Lansheng Han, Hongwei Lu, Cai Fu, and Yekui Qian. Attack detection based on invariant state set for SDN-enabled vehicle platoon control system. *Vehicular Communications*, page 100417, September 2021. 36

Appendix A

Vehicular Reference Misbehavior (VeReMi)

The VeReMi extension dataset is organized by attack type and time span. There are 19 types of attacks previously mentioned in 3. There are two-time spans: rush hour between 07.00 and 09.00 and low traffic between 14.00 and 16.00. Based on this condition, there are a total of 38 data groups. One data set comprises message logs of each simulated vehicle in a one-time span.
We can download VeReMi extension freely at the link: https://github.com/josephkamel/VeReMi-Dataset.git.

The VeReMi extension dataset parameters can be seen in the table A.1

Table A.1: Information Regarding VeReMi Datasets Per Described Scenario [48]

		Dataset ID		
		Attack 0709	**Attack 1415**	**MixAll 0024**
Scenario	Time span	07h-09h	14h-16h	00h-24h
	Density (Vehicle/km2)	37.03 Veh/km2	16.36 Veh/km2	23.29 Veh/km2
Attacker	Vehicles (numbers)	1,220	505	7,399
	Messages (numbers)	924,251	249,612	7,505,418
Genuine	Vehicles (numbers)	2,846	1,179	17,264
	Messages (numbers)	2,221,825	569,723	11,951,021
Average Size	Plain (File Size)	1.92 GBs	0.59 GBs	0.91 GBs
	Gzipped (File Size)	0.40 GBs	0.12 GBs	0.91 GBs
Total Size	Plain (File Size)	40.51 GBs	11.92 GBs	10.90 GBs
	Gzipped (File Size)	8.41 GBs	2.42 GBs	2.25 GBs

A.1 File Structure

Veremi Extension divide its Dataset into each simulation for each type of attack in two time span

1. rush hour : 7h-9h,
2. low traffic time : 14h-16h.

for example , there is data simulation with folder name *ConstPos_0709*, so this folder consist of BSM data from simulation with attacker scenario : Constant Positioon at rush hour. see figure A.1

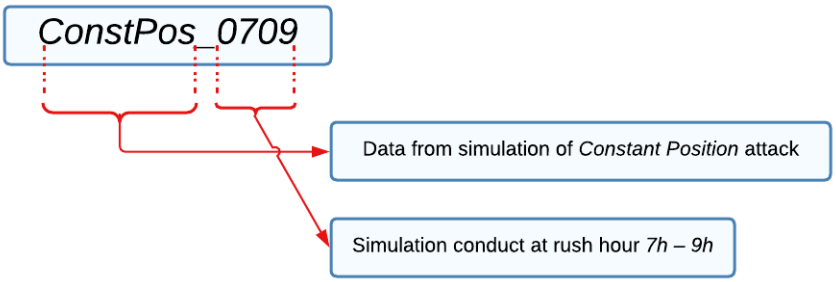


Figure A.1: Dataset Folder Naming Rules

Each folder contains of :

- a ground truth file for every message
  - example : traceGroundTruthJSON-7.json
  - A ground truth file is updated whenever a message is sent by any vehicle.
- a set of message logs for every vehicle that received messages.
  - example : ***traceJSON-39-37-A0-25211-7.json***
  - this file name of a message log identifies the receiver by vehicle number and OMNeT++ module number and another identity :
    - ***traceJSON*** is the file name that identify that this json file is log message of a vehicle.
    - ***39*** refers to id number of vehicle or 39th vehicle who owns this json log message.
    - ***37*** refers to OMNET++ module ID
    - ***A0*** refers to the fact that this vehicle is not an attacker. Attacker will be denote as ***A1, A2, A3, ... A19*** (19 types of attack)
    - ***25211*** refers to the time stamp when this vehicle appeared at the simulation.
    - ***7*** refers to time span 07.00 - 09.00 (simulation at 14.00 - 16.00 will be denote as ***14***)

Files structure at Veremi Extension dataset can refers to the figure A.2.

A.2 Log Messages Composition

VeReMi dataset consists of message logs per vehicle, and the details of the message are as follows:

- GPS data of the local vehicle (labeled as **type=2**).
- Basic Safety Messages from other vehicles through Dedicated Short Range Communication (labeled as **type=3**).
- Messages labeled as **type=4** are basically the same as messages type=3 but are collected in the ground truth file.

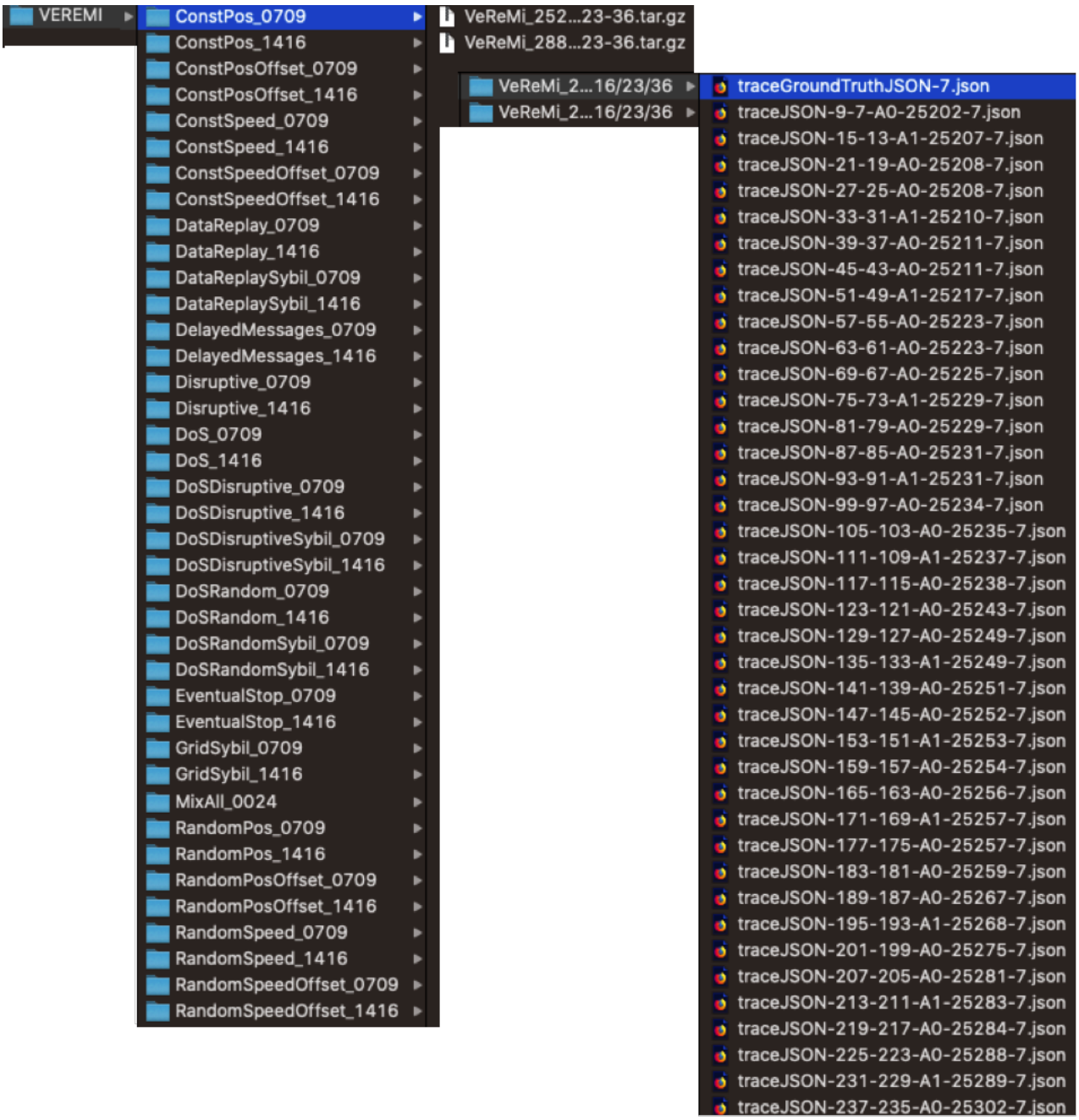


Figure A.2: Veremi Extension Files Structure

- Each message contains four primary data fields [51]:
  1. Position
  2. Speed/Velocity
  3. Acceleration
  4. Heading

We can review the VeReMi extension dataset field format in the table

Table A.2: Format Field Dataset VeReMi Extension [48]

Field	**Format**	**Description**
type	$\mathbb{R}_{[0,20]}$	message type
rcvTime	$\mathbb{R}_{[0,+\infty]}$	message receive time
sendTime	$\mathbb{R}_{[0,+\infty]}$	message send time
sender	$\mathbb{Z}_{[0,+\infty]}$	sender ID
senderPseudo	$\mathbb{Z}_{[0,+\infty]}$	sender Pseudonym
messageID	$\mathbb{Z}_{[0,+\infty]}$	message ID
pos	$[\mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}]$	position
pos_noise	$[\mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}]$	position error
spd	$[\mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}]$	speed
spd_noise	$[\mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}]$	speed error
acl	$[\mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}]$	acceleration
acl_noise	$[\mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}]$	acceleration error
hed	$[\mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}, \mathbb{R}_{[-\infty,+\infty]}]$	heading
hed_noise	$[\mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}, \mathbb{R}_{[0,+\infty]}]$	heading error

Appendix B

2-Step 2-D BSM ML Model

Stage to determine machine learning model for 2-Dimension BSM:

B.1 Preliminary

Select the latest ML model for classification. Determined 20 ML models that will be used in the initial training process :
DenseNet121, DenseNet169, DenseNet201, MobileNet, ResNet101V2, ResNet152V2, EfficientNetB2, ResNet50, EfficientNetV2B0, ResNet101, EfficientNetB0, EfficientNetB1, EfficientNetB3, EfficientNetB4, NASNetMobile, EfficientNetB6, NASNetLarge, ResNet152, EfficientNetB7, EfficientNetB5.

B.2 Stage 1 Training

Carry out the training process using all selected ML models and present the results. The datasets used are sample datasets in the form of BSM, which have been converted to 2-dimensional form according to chapter 4.
Because the training process will take a long time, the datasets used are only 5% of the total UPHF map datasets at this stage.
From table B.1 and graph B.1, the ML model with an accuracy above 40% will be selected to be used in the training process stage 2.

B.3 Stage 2 Training

This training uses the entire dataset from the UPHF Map and involves 8 ML Models. The results of ML stage 2 are presented in the form of tables B.2 and graphs B.2 of Train and Validation Loss. We selected the best 2 ML models based on accuracy and from the best loss graph at this stage.

We can see that the highest accuracy is obtained by ResNet101V2 and ResNet152V2, see figure B.2. However, to decide which ML Model is the best, we must review each model's Train vs. Validation Loss chart. We can see that the ResNet152V2 graph is better than the ResNet101V2 graph, which has too many spikes, figure B.3 (a) and (b) . So the first choice fell to ResNet152V2 as the ML Model to be optimized. For comparison, we will choose another second ML model. So we see that the MobileNet graph is better than the graph of the other models, apart from ResNet152V2, see figure B.3 (c). The worst charts are owned by EfficientNetB2 and ResNet50, which will not be selected.

Table B.1: Result of Stage 1 Training

No	ML Model	Accuracy
1	**DenseNet169**	**51.50%**
2	**DenseNet121**	**48.04%**
3	**MobileNet**	**46.31%**
4	**ResNet101V2**	**46.17%**
5	**ResNet152V2**	**44.26%**
6	**DenseNet201**	**41.81%**
7	**EfficientNetB2**	**41.29%**
8	**ResNet50**	**41.10%**
9	EfficientNetV2B0	36.72%
10	ResNet101	34.74%
11	EfficientNetB0	29.85%
12	EfficientNetB1	29.42%
13	EfficientNetB3	25.77%
14	EfficientNetB4	23.66%
15	NASNetMobile	22.96%
16	EfficientNetB6	19.95%
17	NASNetLarge	19.69%
18	ResNet152	17.30%
19	EfficientNetB7	16.10%
20	EfficientNetB5	15.93%

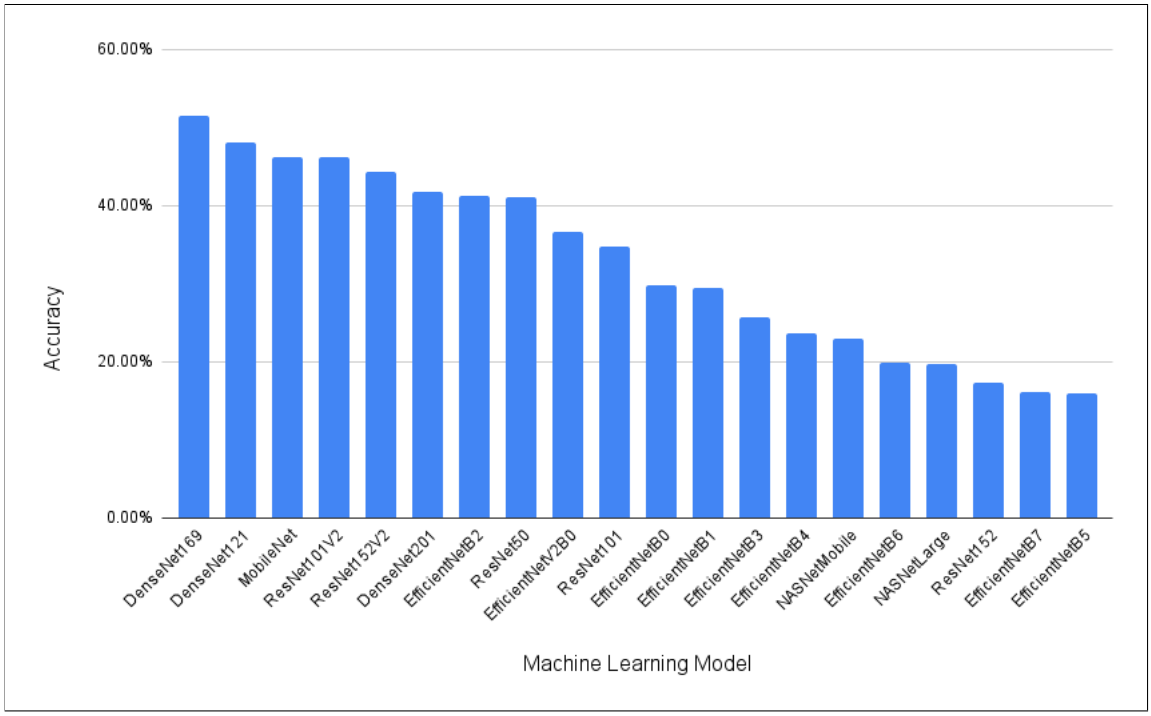


Figure B.1: Graphic Accuracy Stage 1

Table B.2: Result of Stage 2 Training

No	ML Model	Accuracy
1	ResNet101V2	91.92%
2	ResNet152V2	89.79%
3	ResNet50	89.29%
4	DenseNet169	87.85%
5	DenseNet121	86.27%
6	MobileNet	81.03%
7	DenseNet201	75.18%
8	EfficientNetB2	42.84%

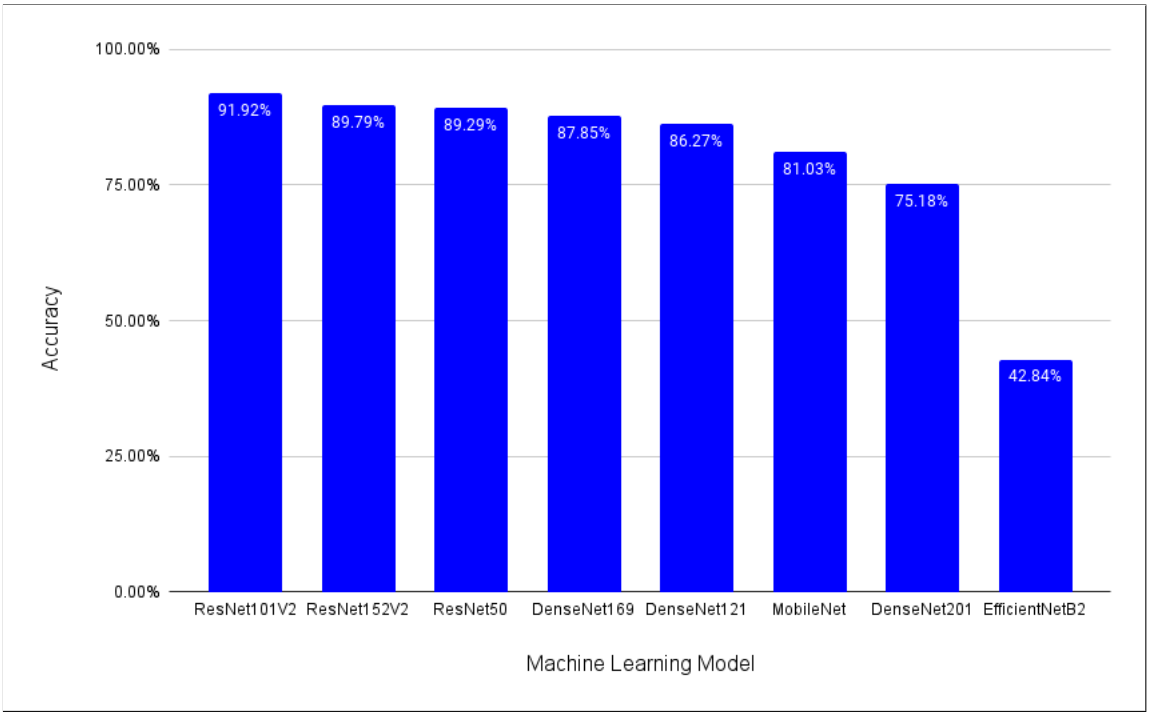
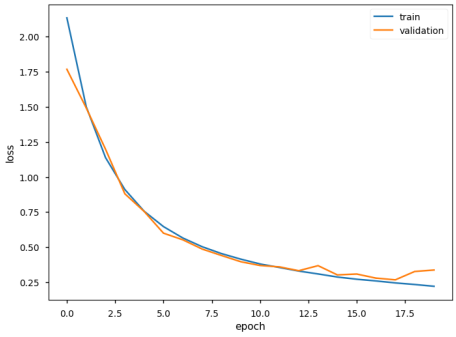
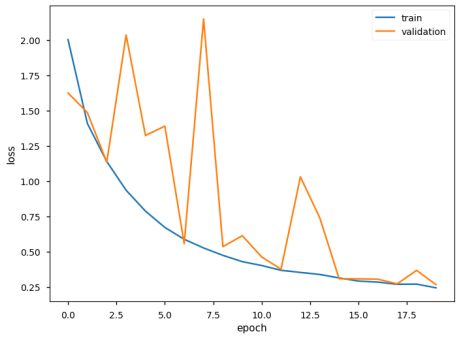


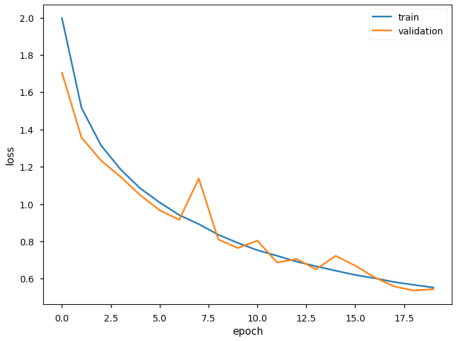
Figure B.2: Graphic Accuracy Stage 2



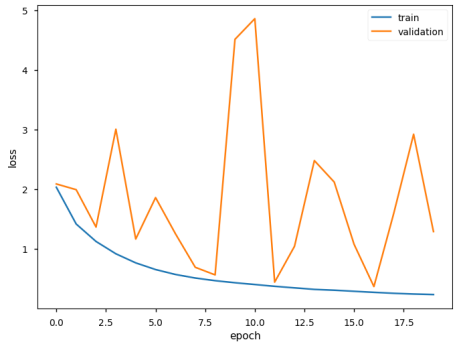
(a) ResNet152V2



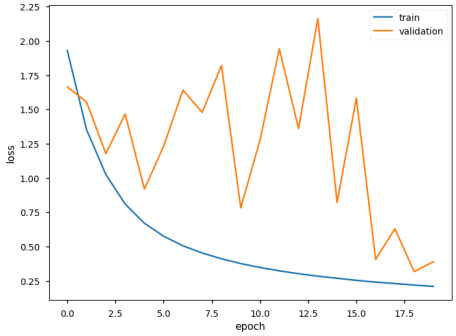
(b) ResNet101V2



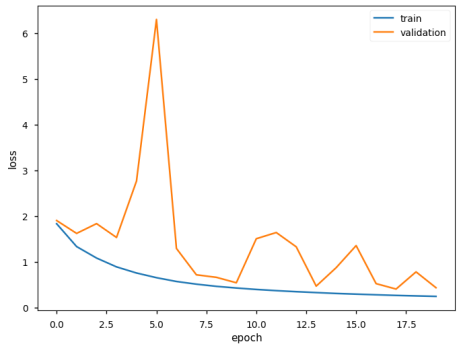
(c) MobileNet



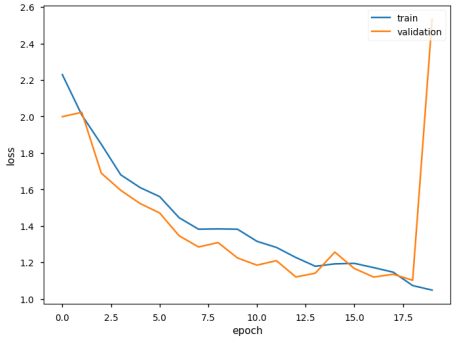
(d) DensNet201



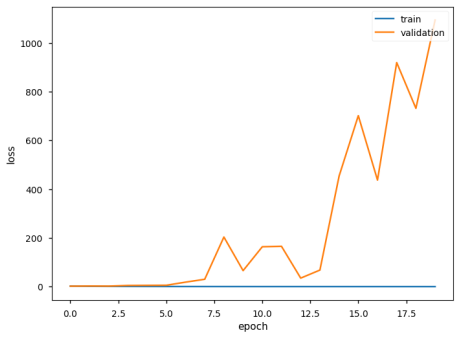
(e) DensNet169



(f) DensNet121



(g) EfficientNetB2



(h) ResnNet50

Figure B.3: Train Loss VS Validation Loss Graphics

B.4 Hyperparameter Optimization (HPO)

After second stage, we retrain the ML model after conducting Hyperparameter Optimization(HPO) using TPE, which is explained in the subchapter 3.2.4 The epoch for both ML is limited to 18 epochs for ResNet152V2 and 60 epochs for MobileNet. The number of epochs is not the same for both models because ResNet152V2 requires a longer training time than MobileNet. We can see in table B.3 that by using hyperparameter optimization, the accuracy of both ML increases significantly. ResNet152V2 increased by 2.9%, while MobileNet increased by 8.55%, with the same number of epochs.

Table B.3: Accuracy Before and After Hyperparameter Optimization

ML Model	**Epoch**	**Accuracy**	
		before HPO	**after HPO**
ResNet152V2	18	89.79%	92.69%
MobileNet	60	81.03%	89.58%