Resilience by design & failures forecasting for a connected autonomous vehicle

Jean-Philippe Monteuuis

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2020IPPAT003

Thèse de doctorat

Resilience by design & failures forecasting for a connected autonomous vehicle

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat: Informatique, données, IA

Thèse présentée et soutenue à Palaiseau, le 31 Janvier 2020, par

JEAN-PHILIPPE MONTEUUIS

Composition du Jury :

Hossam AFIFI Professeur, Telecom Sud Paris (SAMOVAR)	Président
Frederic CUPPENS Professeur, IMT Atlantique	Rapporteur
Isabelle CHRISMENT Professeure, Université de Lorraine	Rapporteur
Hassnaa MOUSTAFA Principal Engineer, Intel Corporation	Rapporteur
Jonathan PETIT Principal Engineer, Qualcomm	Examinateur
Oyunchimeg SHAGDAR R&D Team Lead, VEDECOM	Examinateur
Houda LABIOD Professeure, Télécom Paris	Directrice de thèse
Pascal URIEN Professeur (HDR), Télécom Paris	Co-directeur de thèse

Invités :

Alain SERVEL Expert Télématique & ADAS, Groupe PSA	Invité
Eric OLLINGER Adjoint de la sous-direction, Ministère de l'écologie	Invité

To Ugo de la Trévinière
for all the joy he brought into our lives.

Acknowledgments

I would particularly like to thank my supervisor Prof. Dr. Houda Labiod, who gave me the opportunity and freedom to research in this exciting field of technology at Telecom Paris. Her guidance and scientific advice were a great help to write this dissertation. Also, I would like to express my gratitude to Alain Servel who supported my research at Groupe PSA.

The work in the "INFRES" department at Telecom Paris was a great pleasure thanks to the continuous support and encouragement of the academic staff and my colleagues (Mohammed H, Mohammed B., Roberto, Badis, Sara, Safa, Peng, Eduardo...). Besides, I would like to thank my thesis co-supervisor, Dr. Jun Zhang, for providing helpful feedback during the thesis. Also, I would like to thank Dr. Rida Khatoun who supported me since my master's degree until now.

Thank you to Groupe PSA, this work will not have been possible without the support and the fruitful discussions with my colleagues. A special thanks to Dr. Stefano Mafrica, who helped me with the process of patent application and his feedbacks.

I would also like to thank my mentor Dr Jonathan Petit and Dr Aymen Boudguigua for the fruitful discussions and advices during this PhD. Further, I would like to thank the participants of the project "SCOOP" and "ISE" to discuss the topic of V2X communication security in an international context. I have learned a lot thanks to this community.

Finally, I am grateful to my parents, my brother, and friends (Julien, "French-Chinese crew," "Nephews," "Team Montpellier," "UNG"...) for their unconditional support and patience during the course of this work. Last but not least I would like to thank my wife, Mengdi Zhang, to believe in me and the success of my work. As she used to say: "You don't build a house wall by wall but brick by brick. The same goes for a thesis". In brief, this thesis would not have been possible without all their encouragements and understandings.

INSTITUT POLYTECHNIQUE DE PARIS

Abstract

Telecom ParisTech
Department: Network & Computer Sciences (INFRES)

Doctor of Philosophy

Resilience by design & failures forecasting for a connected autonomous vehicle

by Jean-Philippe MONTEUUIS

Autonomous vehicles with an automation level 5 will drive autonomously in any road scenarios such as highways, snowy roads, urban areas, or traffic jams. The integration of V2X communication, as a new source of perception for the vehicle could remove the limitations of local perception by communicating with an occluded pedestrian or by detecting in advance the presence of a vehicle under a heavy mist. However, this V2X communication may be a new source of attacks threatening the vehicle perception. Current countermeasures are not designed for all autonomous vehicles because these countermeasures require the driver assistance or work with a specific set of sensors. Therefore, the thesis aims to propose a generic failure resilient perception architecture for all types of connected and autonomous vehicles supporting different kinds of sensors. In this thesis, we propose a generic perception architecture named GPA with its failure resilient perception algorithm (FRPA). We propose a new threat analysis and risk assessment method named SARA that identifies and assess the risk of attacks targeting connected and automated vehicles with an automation level 5. To identify where and how these attacks occur, we propose an attacker and a security goal model for all automotive perception systems. We implemented two modules of our failures resilient perception algorithm (FRPA): a Machine Learning based Failure Classifier and a V2X-Sensor Correlation Module considering three kinds of source: camera, radar, and V2X.

We highlighted several new attacks in the perception pipeline and raise the need for new security countermeasures such as the physical integrity of road infrastructures and trustworthy perception algorithms. Besides, our countermeasures based on machine learning and sensor correlation showed very accurate results to detect and classifies perception failures (over 90% accuracy score). Finally, the ideas developed in the thesis resulted in 10 filled patents and several publications.

INSTITUT POLYTECHNIQUE DE PARIS

Abstract

Telecom ParisTech
Department: Network & Computer Sciences (INFRES)

Doctor of Philosophy

Resilience by design & failures forecasting for a connected autonomous vehicle

by Jean-Philippe MONTEUUIS

Les véhicules autonomes dotés d'un niveau d'automatisation 5 conduiront de manière autonome dans tous les scénarios routiers tels que les autoroutes, les routes enneigées, les zones urbaines ou les embouteillages. L'intégration de la communication V2X, en tant que nouvelle source de perception du véhicule, pourrait supprimer les limitations de la perception locale en communiquant avec un piéton caché par un obstacle ou en détectant à l'avance la présence d'un véhicule caché par un brouillard épais. Cependant, cette communication V2X peut constituer une nouvelle source d'attaques menaçant la perception du véhicule. Les contre-mesures actuelles ne sont pas conçues pour toutes les architectures de véhicules autonomes, car elles requièrent l'assistance du conducteur ou fonctionnent avec un ensemble spécifique de capteurs. La thèse vise donc à proposer une architecture de perception générique et résiliante aux défaillances pour tous les types de véhicules connectés et autonomes. Dans cette thèse, nous proposons une architecture de perception générique nommée GPA avec son algorithme de perception résiliante aux défaillances (FRPA). Nous proposons une nouvelle méthode d'analyse de menaces et d'évaluation des risques nommée SARA, qui identifie et évalue le risque d'attaques ciblant les véhicules connectés et automatisés de niveau 5. Pour identifier où et comment ces attaques ont lieu, nous proposons un modèle d'attaquant et un modèle d'objectifs de sécurité pour tous les systèmes de perception automobile. Nous avons implémenté deux modules de notre algorithme FRPA: un module classification des défaillances basé sur une méthode de Machine Learning et un module de corrélation V2X-Capteur en considérant trois sources d'information: radar, caméra et V2X.

Nous avons mis en évidence plusieurs nouvelles attaques dans le cycle de perception et soulevé le besoin de nouvelles contre-mesures de sécurité centrées sur l'intégrité physique des infrastructures routières et sur les algorithmes de perception fiables. De plus, nos contre-mesures basées sur l'apprentissage automatique et la corrélation entre capteurs ont permis d'atteindre une très bonne précision pour détecter et classifier les défaillances de perception (score de précision supérieur à 90 %). Enfin, les idées développées dans la thèse ont abouti à 10 brevets déposés et à plusieurs publications.

Contents

List of Figures

List of Tables

List of Abbreviations

ACC	**A**ctive **C**ruise **C**ontrol
ADAS	**A**utomated **D**riving **A**ssistance **S**ystem
AFK	**A**way **F**rom **K**eyboard
ANN	**A**rtificial **N**eural **N**etwork
AOA	**A**ngle **O**f **A**rrival
AV	**A**utomated **V**ehicle
CA	**C**ertification **A**uthority
CV	**C**onnected **V**ehicle
CAV	**C**onnected and **A**utomated **V**ehicle
CAM	**C**ooperative **A**wareness **M**essage
CAN	**C**ontroller **A**rea **N**etwork
C-ITS	**C**ooperative-**I**ntelligent **T**ransport **S**ystem
DAE	**D**eep **A**uto **E**ncoder **M**essage
DENM	**D**ecentralized **E**nvironmental **N**otification **M**essage
DoS	**D**enial **o**f **S**ervice
DR	**D**imensionality **R**eduction
DSRC	**D**edicated **S**hort **R**ange **C**ommunication
ECU	**E**lectronic **C**ontrol **U**nit
ETSI	**E**uropean **T**elecommunications **S**tandards **I**nstitute
FNR	**F**alse **N**egative and **R**ate
FOV	**F**ield **O**f **V**iew
FPR	**F**alse **P**ositive **R**ate
FRPA	**F**ailures **R**esilient **P**erception **A**lgorithm
GG	**G**ood **G**ame
GNN	**G**lobal **N**earest **N**eighbor
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem
GPA	**G**eneric **P**erception **A**rchitecture
GPS	**G**lobal **P**ositioning **S**ystem
HARA	**H**azard **A**nalysis and **R**isk **A**ssessment
HMI	**H**uman **M**achine-**I**nterface
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers
ITS	**I**ntelligent **T**ransport **S**ystem
ITSS	**I**ntelligent **T**ransport **S**ystem **S**tation
KNN	**K**-**N**earest **N**eighbor
KF	**K**alman **F**ilter
LDA	**L**inear **D**iscriminant **A**nalysis
LDM	**L**ocal **D**ynamic **M**ap
LiDaR	**Li**ght **D**etection **a**nd **R**anging
LOS	**L**ine **O**f **S**ight
MCC	**M**atthews **C**orrelation **C**oefficient
MIA	**M**issing **I**n **A**ction
MLP	**M**ulti**l**ayer **P**erceptron
ML	**M**achine **L**earning

MOT	**M**ulti-**O**bject **T**racking
M2MA	**M**easurement-**to**-**M**easurement **A**ssociation
M2TA	**M**easurement-**to**-**T**rack **A**ssociation
NA	**N**ot **A**vailable
NLOS	**N**on **L**ine **O**f **S**ight
OBU	**O**n-**B**oard **U**nit
OEM	**O**riginal **E**quipment **M**anufacturers
OOSM	**O**ut **O**f **S**equence **M**easurements
PC	**P**seudonym **C**ertificate
PDM	**P**erception **D**ata **M**odel
PKI	**P**ublic **K**ey **I**nfrastructure
QDA	**Q**uadratic **D**iscriminant **A**nalysis
RaDaR	**Ra**dio **D**etection **a**nd **R**anging
RetEx	**Ret**urn of **Ex**perience
RSU	**R**oad **S**ide **U**nit
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
SAE	**S**ociety of **A**utomotive **E**ngineers
SVM	**S**upport **V**ector **M**achine
TARA	**T**hreat **A**nalysis and **R**isk **A**ssessment
TPR	**T**rue **P**ositive and **R**ate
TNR	**T**rue **N**egative and **R**ate
T2T	**T**rack-**to**-**T**rack association
VANet	**V**ehicular **A**d-Hoc **Net**work
VRU	**V**ulnerable **R**oad **U**ser
V2I	**V**ehicle-**to**-**I**nfrastructure Communication
V2P	**V**ehicle-**to**-**P**edestrian Communication
V2V	**V**ehicle-**to**-**V**ehicle Communication
V2X	**V**ehicle-**to**-**Everything** Communication

Chapter 1

Introduction

This chapter gives an overview of the thesis context. Then, we highlight the thesis motivations and our contributions.

1.1 Context

The gradual automation of autonomous vehicle aims to reduce accident conditions, optimize energy consumption, and traffic efficiency. To this end, an autonomous vehicle uses a set of sensors to perceive its surrounding environment and drive autonomously (Figure 1.1).

FIGURE 1.1: Autonomous Vehicles

However, autonomous driving may affect the driver' safety and security. Indeed, components of the vehicle may become faulty and, thus, disable autonomous driving. For example, an automotive camera cannot detect the color of the traffic light because the camera lens can be broken. The camera may fail to detect a pedestrian (Figure 1.2). Besides, hackers may decide to take control of autonomous vehicles and cause road accidents remotely. The road environment may also threaten the safety of autonomous driving. For instance, solar lights may blind the vehicle's camera and

disable its ability to detect a pedestrian. Alternatively, intentionally, a pedestrian may attack the autonomous vehicle by flashing its camera.

Moreover, trucks and buildings may occlude the camera vision. For instance, the camera is unable to detect a pedestrian behind a truck. As seen, these intentional and unintentional perception failures should be solved to ensure safe autonomous driving.

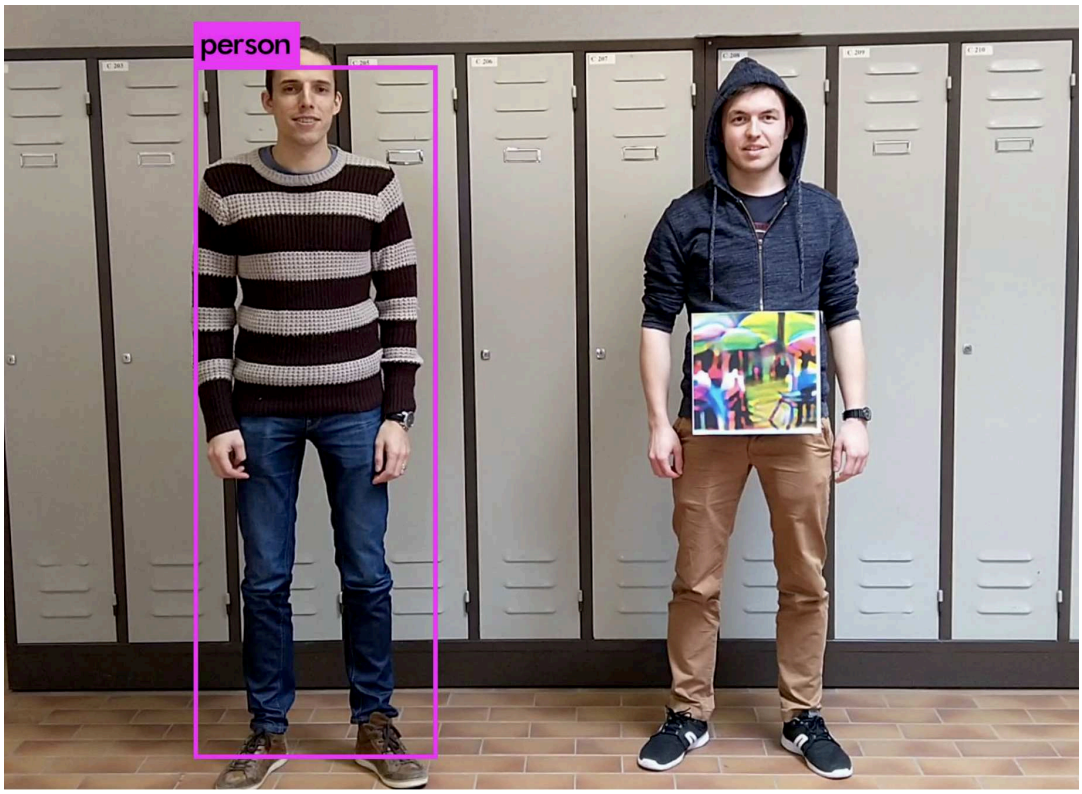


FIGURE 1.2: Failed Object Detection of an automotive camera

The integration of vehicular communication (V2X) in the perception system of connected vehicles partially solves these perception failures. For instance, the traffic light can communicate its light color to the vehicle. Thus, the vehicle knows the light color even if a flash blinds the camera. In fact, the vehicular communication can be viewed as a redundant data source for the perception system. It can be used to prevent an absence of detection or to detect a faulty sensor fulfilling anomaly detection purposes. Thus, V2X communication can be used for anomaly detection. Moreover, as mentioned, sensors may not detect occluded road users. However, an occluded pedestrian can communicate its location to all surrounding road users. We can conclude that V2X communication is an efficient data source able of enhancing the horizon of the perception system and improving failure detection for future connected and automated vehicles.

Unfortunately, V2X communication is also the target of security attacks. Therefore, these attacks jeopardize the benefits of an augmented perception system. It is very critical to take into account the security flaws as well as the security attacks in order to define necessary countermeasures against these attacks.

1.2 Motivations and Objectives

In the absence of security and safety counter-measures, the presence of automotive attacks and faults threaten the proper functioning of any perception system. Therefore, we aim to **design a generic failure resilient perception system for a connected and automated vehicle** (CAV). This perception system must include different types of sensors (radar, camera, lidar, GPS) but also V2X data carried through various V2X communications that can exist in a cooperative intelligent transportation system such as vehicle-to-vehicle (V2V), vehicle-to-vehicle (V2I), vehicle-to-vehicle (I2V), and vehicle-to-pedestrian (V2P).

Firstly, we propose to **define a generic perception architecture** as a key basis for our perception failure resilient system. This generic architecture models any connected and automated vehicle. It includes both physical (sensors, computers, actuators) and logical (processes and data flows) elements. Unlike previous architectures that were designed for a specific CAV architecture. Our architecture for failure perception proposes a generic and modular solution to prevent perception failures.

Secondly, we identify the source of failures in our generic perception architecture. Then, our resilient perception system must counter sources with the highest risk of failures. To do so, we analyze and assess each cause of failures. In the domain of safety, there is a standard method that assesses the risk of failures caused by faults. However, there is no method for connected and automated vehicles that assesses the risk of failures caused by security attacks. Thus, our second objective is **to identify the attacks and assess their risks in the perception system of any connected and automated vehicle**. Indeed, current methods do not consider the fact that the driver may not be able to control the vehicle in case of an attack and the safety repercussion of such attacks on road users.

Besides identifying and assessing attacks, we need to understand where and when these attacks can occur during a driving scenario. Indeed, it is crucial to know the motivation of an attacker while launching its attacks. For instance, the attacker may jam the wireless V2X communication to blind our perception system. Therefore, our third objective is **to define the attacker model**. This attacker model must consider attacks targeting sensors, V2X data, but also the algorithms of the perception systems.

To counter these attackers, we must design a resilient perception system that includes a set of security modules such as the verification of digital signatures and intrusion detection using machine learning methods. For instance, perception data obtained from sensor (radars and cameras) and from V2X communication can allow to identify cross detected objects among these three sources. Thus, a source among these three sources, becoming faulty can be easily detected thanks to crossing perception information of these sources. Therefore, the fourth objective is to **design an efficient and failure resilient perception algorithm**. In our work, we focus on attacks targeting V2X communications. Indeed, few works have studied the consequences of these attacks and their countermeasures in the context of CAVs.

Lastly, this failure resilient perception algorithm must have low computation latency and high detection accuracy. Therefore, the fifth objective is **to evaluate this failure resilient perception algorithm** through a deep analysis of several metrics considering various road scenarios.

1.3 Thesis Contributions and Organization

This thesis brings original contributions, essential in the design of a generic failure resilient perception system for connected and highly automated (level five) vehicles. In our work, we mainly focus on perception failures driven by security attacks that are seldom addressed in the literature. Figure 1.3 highlights the thesis organization and our main contributions.

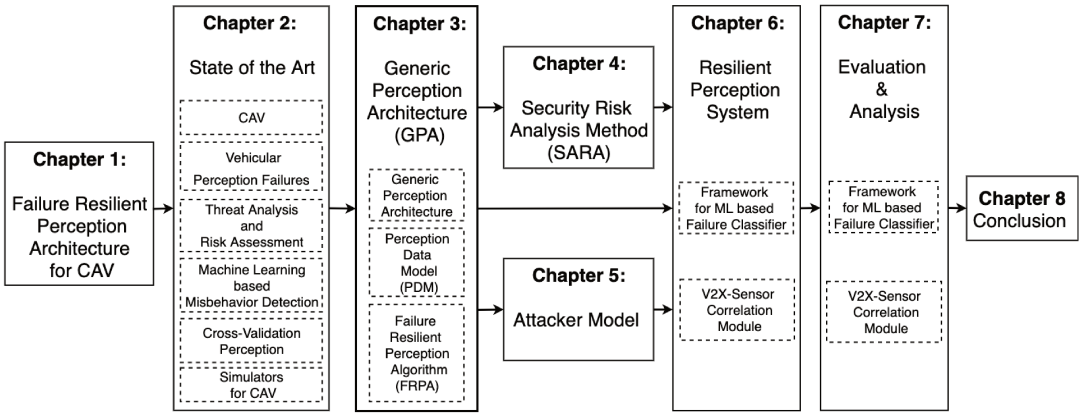


FIGURE 1.3: Overview of the thesis

Chapter 2 presents the state of the art related to topics investigated in our thesis. First, we give an overview of the context and the perception failures related to connected and automated vehicles. Then, we survey methods to assess the security risk of failures in CAV. Also, we present the state of the art of Machine Learning methods to detect attacks related to perception failures. Besides, we present related work on the Sensor-V2X data correlation to detect security attacks leading to perception failures. To evaluate our failures detectors, we survey the current simulators and describes the mandatory modules for the thesis simulations.

Chapter 3 describes our first contribution, which is the definition of our Generic Perception Architecture (**GPA**) based on using a combination of sources (sensors and V2X communication receiver). Our architecture is modular and generic which supports several types of CAV architectures. Alongside our **GPA**, we define of a *Perception Data Model* (**PDM**). Our third contribution is the design of a Failure Resilient Perception Algorithm (**FRPA**) to detect anomalous perception data and, thus, to prevent failures.

In Chapter 4, we propose a new threat analysis and risk assessment method named **SARA** [Mon+18c]. Our method includes a new threat analysis features such as attacker profiles. Besides, SARA includes new risk assessment metrics such as the degree of auto-pilot controllability and the degree of CAV observation. Finally, we apply our method to two use cases to prove its practicability.

In Chapter 5, we propose a new attacker model [Mon+18b]. Accordingly, we extend the security goals model from **SARA** and compare it with existing models to highlight our contribution.

After the analysis of the attacks and the attacker, we focus on two important modules of FRPA algorithm (Chapter 6): a framework for Machine Learning based Failure Classication [Mon+18a] and a V2X-Sensor Correlation Module [Mon+19]. The first contribution detects abnormal data in a V2X message using supervised machine algorithms such as *MLP*, *Adaboost*, and *Random Forest* and our new dimension dataset. The second contribution detects abnormal data in a V2X message using a camera and, then, a radar. In this chapter, we showed how a perception source

could self-detect abnormal data to prevent failures. Moreover, we show how to use another perception source to detect anomalous perception data.

In Chapter 7, we evaluate our framework for Machine Learning based Failure Classifiers and our V2X-Sensor Correlation Module. We test our ML Framework on classification data. During this evaluation, we test 3 ML classifiers and compare them with a threshold model. Besides, we analyze several metrics to assess the performance of a classifier model with an unbalanced dataset distribution (more normal data than abnormal data). Then, we test our sensor correlation module with anomalous position data in a V2X message. During this evaluation, we test different frequencies of V2X message reception and various sensor types (radar and camera).

Chapter 8 concludes the thesis and highlights the short and long term research perspectives of thesis work.

Chapter 2

State-of-the-art

This chapter describes the thesis State of art over six sections. Firstly, Section 2.1 gives an overview of Connected and Automated Vehicles (CAV). Secondly, Section 2.2 defines and presents perception failures related to CAV. Thirdly, Section 2.3 surveys methods to assess the security risk of failures in CAV. Fourthly, Section 2.4 presents the state of the art of Machine Learning methods to detect attacks related to security CAV failures. Fifthly, Section 2.5 presents related work on the Sensor-V2X data correlation to detect security attacks leading to CAV failures. Lastly, Section 2.6 surveys the current CAV simulators and describes the mandatory modules for the thesis simulations.

2.1 Connected and Automated Vehicle (CAV)

Section 2.1 presents the context of the connected vehicle (CV), as well as the context of the automated vehicle (AV).

2.1.1 Connected Vehicle

Global positioning system (GPS) technology has opened the doors for new cooperative intelligent transportation system (C-ITS) applications in which CVs are one of the most promising technological advances. Indeed, CVs introduce wireless communications among vehicles and roadside infrastructures [CSS17] (Figure 2.1).

In the following sections, we review the types of entities involved in C-ITS as well as V2X communication and communications protocols.

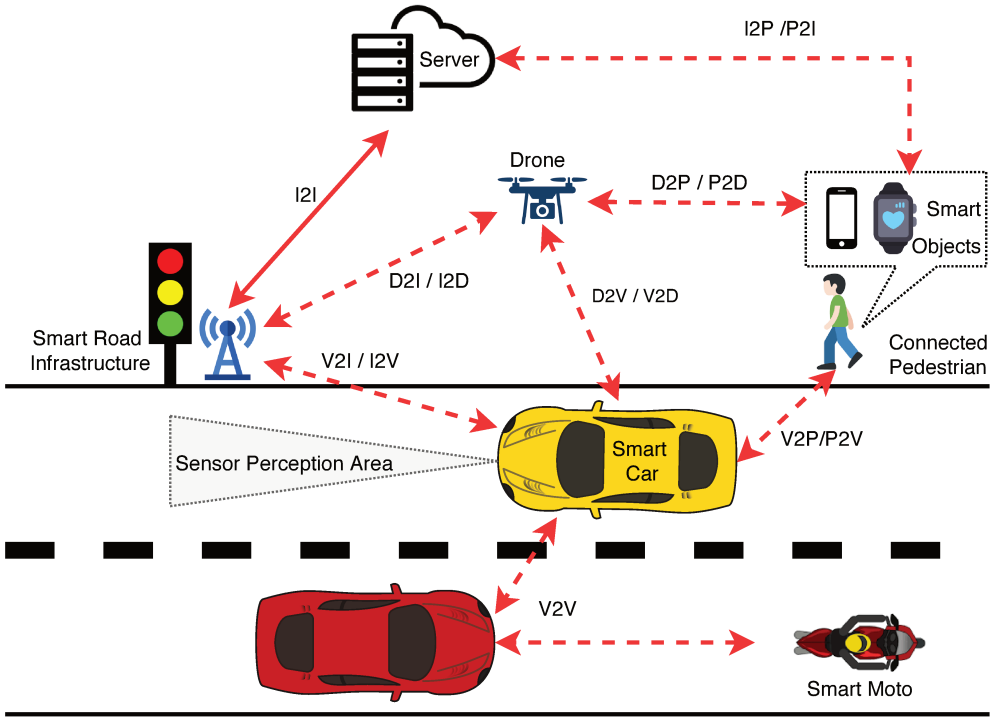


FIGURE 2.1: C-ITS System

2.1.1.1 ITS-Stations Classification

In C-ITS, four types of ITS-Stations can communicate with each other [ETSb].

The first type is named **Connected Vehicle** and provides ITS applications to vehicle drivers and passengers. Each connected vehicle embeds an Onboard Unit (OBU) that acts as V2X gateway that handles the radio communication, as well as the encoding and the decoding of V2X messages.

The second type is named **Central** includes servers from traffic operators, road operators, services providers, or content providers. A centralized ITS-S provides centralized ITS applications. Furthermore, a centralized ITS-S requires further connection with backend systems (e.g., Internet).

The third type is named **Roadside** which regroups connected road infrastructures such as Roadside unit (RSU), smart traffic lights, smart toll. A Roadside ITS-S provides ITS applications independently or cooperatively with central ITS-S or another roadside ITS-S.

The fourth type is named **Personal** and regroups personal and nomadic devices (e.g., drones or cell phones). It provides ITS applications to the device owner.

2.1.1.2 V2X Communication Model

Cooperative Intelligent Transport Systems (C-ITS) use technologies that allow road vehicles to communicate with other vehicles, with traffic signals and roadside infrastructure, as well as with other road users [Pla16]. The V2X Communication Model defines several communication modes, such as Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Infrastructure to Vehicle (I2V), Vehicle to Pedestrians (V2P), Vehicle to Network (V2N), Vehicle to Server (V2S) as well as Vehicle to Cloud (V2C) (Figure 2.1). A vehicle defines any road transporter such as a car, a bus, or a motorbike.

Collectively, these connections refer to V2X (vehicle-to-everything) communication. These communications depend on a set of protocols, application types (e.g., road safety), message types (e.g., Cooperative Awareness Message), wireless technologies (e.g., IEEE 802.11p, 3G, 4G,and 5G), and security mechanisms [17] used in C-ITS use cases.

Besides wired technologies, V2X communication can be supported by numerous wireless access technologies. Some of these communication technologies support medium-range and short-range communications in a distributed manner (e.g., DSRC). In contrast, other technologies support communications in long-range, relying on a centralized infrastructure (e.g., Cellular-V2X) [Vuk+18].

Each country or region follows its own set of standards defining each of these protocols. Our thesis focuses on the standards defined by the European Telecommunications Standards Institute (ETSI). Therefore, the next section describes the protocols stack defined in Europe.

2.1.1.3 Stack of Communication Protocols

Figure 2.2 depicts the protocols stack for V2X Communication. The following sections detail the role of each layer.

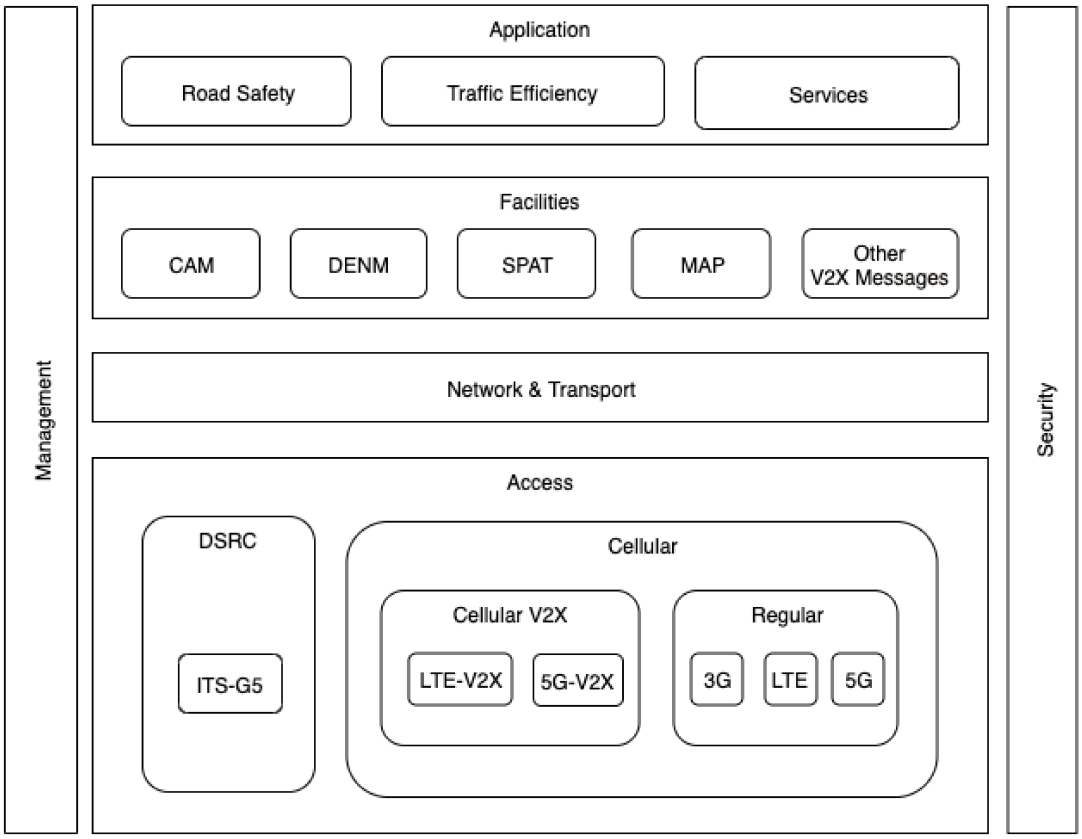


FIGURE 2.2: V2X Protocols Stack

The Access Layer depends on the application requirements (e.g., high bandwidth for real-time application) and the environment context (e.g., do we have cellular coverage?). Indeed, the V2X communication protocol stack must support multiple wireless access technologies. Thus, we briefly introduce some examples. For instance, *DSRC networks* provide vehicles the possibility to communicate with each other within a short-range distance (e.g., 1 km in a rural environment or 400m in an

urban environment). For instance, *Cellular networks* provide broader coverage and higher data rate than DSRC. The full deployment of cellular networks benefits new applications or existing applications. The different cellular technologies regroup two categories. The first category regroups Cellular technologies designed for V2X communication (C-V2X) such as LTE-V2X or 5GV2X. The second category regroups the cellular technology used in cell phones (device). In this configuration, the device needs to discover the neighboring devices before any direct communication. Regular cellular includes 3G, LTE, and 5G technologies.

The Networking and Transport Layers supports the protocols for the dissemination of messages from the source to the destination. This layer is defined by a communication profile that contains at least a transportation protocol and a networking protocol. For instance, ETSI considers the following profiles: BTP over GeoNet, TCP/UDP over IPv6, TCP/UDP over IPv6 over GeoNet [ETS19].

The Facility Layer processes the V2X messages during their reception and their emission. The layer supports several messages types. Each message serves specific driving uses cases. For instance, the message named CAM has a role in warning the surrounding connected vehicle of the message emitter presence. Therefore, the CAM has a function of cooperative awareness used in road safety applications. Indeed, in none line of sight (NLoS) scenarios (e.g., traffic jams), vehicle sensors cannot detect occluded ITS Stations. Therefore, the communication of an occluded vehicle location increases the local perception and may prevent road collision due to an occluded environment. Table 2.1 is a brief description of the information contained in a CAM. Other messages types give details regarding the environmental event (DENM), signal phase, and time (SPaT) and topology specification (MAP). The next section describes the applications developed upon the presented V2X messages.

Data Elements	Description
Standard Elements	
DSRCmsgID	Identifier for message type
SecMark	Timestamp
MsgCount	Message Number for a sequence
TemporaryID	Network ID
Latitude	Position along the latitude axis
Longitude	Position along the longitude axis
Elevation	Elevation relative to the sea level
Speed	Object speed
Heading	Angle between object head and North
Yaw Rate	Heading per second
Lat. Accel	Acceleration along the latitude axis
Long. Accel	Acceleration along the longitude axis
Vet. Accel	Acceleration along the vertical axis
Positional Accuracy	Semi-Major/Minor accuracy at one standard deviation
Brake System Status	Status of the Brake System
Length	Vehicle Length
Width	Vehicle Width
Meta Data	
Sender ID	ID of the emitter
Gentime	Time of message generation

TABLE 2.1: CAM Structure

The Application Layer concerns C-ITS applications. We distinguish three types of applications, which are road safety, traffic efficiency, and services (Table 2.2).

Safety applications prevent road accidents and protect road users' lives. These applications require high communication availability (e.g., radio channel availability) and high bandwidth (e.g., high reception frequency) to reach a highly accurate spatial and temporal visualization of all the surrounding connected objects. *Traffic efficiency applications* optimize the driver itinerary. For instance, an application can choose an optimal path in terms of driving time. Alternatively, an application can adopt an ecology friendly drive by knowing each encountered traffic light duration. *Services* entertain the vehicle resident during their road trip. These applications require a broad bandwidth due to the high amount of downloaded data. Therefore, the used communication stack regroups protocols related to Internet usage (e.g., website reading, media streaming).

With the development of connected and automated vehicles, the mentioned applications may benefit from or support other sources of perception located in the vehicle (e.g., camera, radar, lidar). The current automation race leads to the progressive deployment of Advanced Driver Assistance Systems (ADAS) and their applications regarding road safety and traffic efficiency. In such an aim, ADAS use more and more V2X communications. Thus, in the following section, we highlight the relationship between V2X communication and automotive perception.

Class	**Application**	**Uses case**	**Deployment Phase**		
			1	2	3
Active Road Safety	Cooperative Awareness	Emergency vehicle warning	✓	×	×
		Slow vehicle indication	✓	×	×
		Intersection collision warning	✓	×	×
		Advanced Intersection Collision Warning	×	✓	×
		Motorcycle approaching indication	✓	×	×
		Motorcycle approaching warning	×	✓	×
		Vulnerable Road User Protection	✓	✓	×
		Improved Vulnerable Road User protection	×	×	✓
		Traffic light information	✓	×	×
		In-Vehicle Signage	✓	×	×
	Road Hazard	Emergency electronic brake lights	✓	×	×
		Slow or Stationary Vehicle Warning	✓	×	×
		Traffic condition warning	✓	×	×
		Signal violation warning	✓	×	×
		Road Work warning Short Term	✓	×	×
		Road Work warning Long Term	×	✓	×
		Advanced Precrash Sensing Warning	✓	✓	×
		Adverse Weather Conditions	✓	×	×
Traffic Efficiency	Speed Management	Green Light Optimum Speed Advisory	✓	×	×
		Automated Green Light Optimum Speed Advisory	×	×	✓
		Cooperative ACC	×	✓	×
		Advanced Cooperative ACC	×	×	✓
		Platooning	×	×	✓
	Cooperative Navigation	Traffic information & Itinerary recommendation	✓	×	×
		Enhanced route guide & navigation	✓	×	×
		Limited access warning and detour notification	✓	×	×
Services	Location	Point of Interest notification	✓	×	×
		Automatic access control & parking management	✓	×	×
		ITS local electronic commerce	✓	×	×
		Media downloading	✓	×	×
	Communities	Insurance and financial services	✓	×	×
		Fleet management	✓	×	×
		Loading zone management	✓	×	×
	Data Life Cycle	Provisioning & update	✓	×	×
		Calibration	✓	×	×

TABLE 2.2: Basic Set of C-ITS Applications

2.1.2 Automated Vehicle

Self-driving vehicles incorporate multiple complex systems to sense the surrounding environment, plan a path to a destination, and control steering and speed. Thus, this section gives an overview of an automated vehicle. Firstly, we list the levels and applications in the context of vehicular automation. Secondly, we present the

CAV Architectures. Thirdly, we present the existing sensors used in CAV Perception. Finally, we give an overview of a perception system used in a CAV.

2.1.2.1 Vehicular Automation Levels & Applications

The standard automation level defines the set of ADAS applications, driving conditions (e.g., urban area, snowy, desert, highway), and degree of non-driver assistance supported by the automated driving system of CAV (Figure 2.3). Systems such as the Automated Braking System showed the benefits of vehicular automation and paved the way towards vehicular automation.

The increase of computational power and advances in sensor technology allowed the development of highly automated systems. In 2014, the Society of Automotive Engineers (SAE) released standard defining automation levels known as SAE J 3016 [SAE18]. Table 2.3 gives a brief overview of the SAE automation levels along with a description of an example system for each level. SAE J 3016 also provides a table describing the various responsibilities of the automated vehicle and the driver for each of the automation levels. In recent years, the SAE automation level definitions have become internationally well accepted. Also, Table 2.4 gives examples of an automated system for each automation level with a description of their application and required sensors. As seen in the previous table, the increased automation in automated systems increases the number of needed sensors.

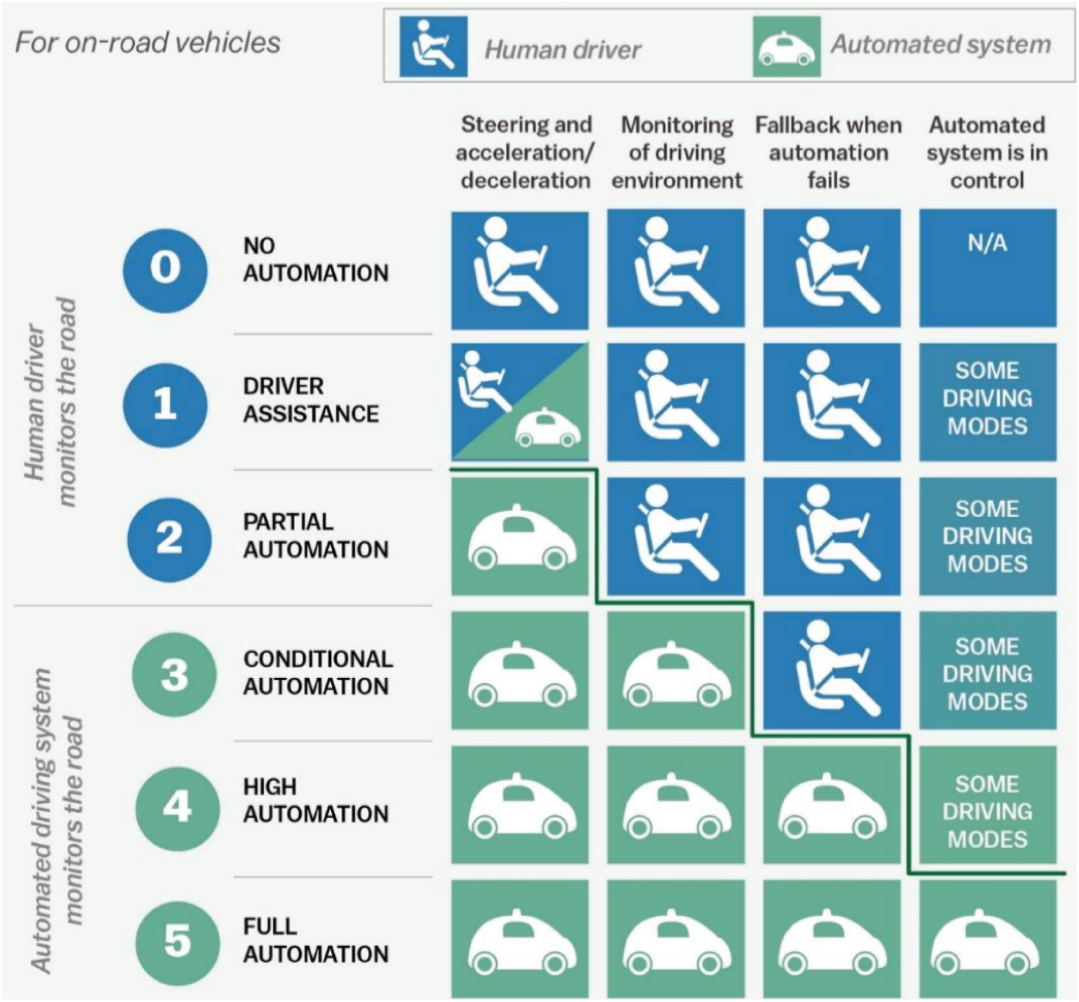


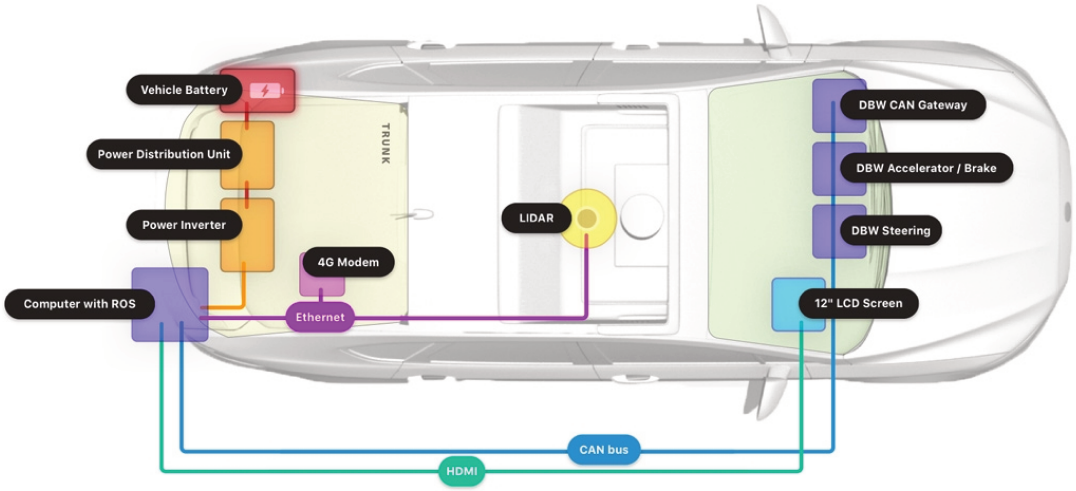
FIGURE 2.3: Standard automation levels [SAE18]

2.1.2.2 Architectures

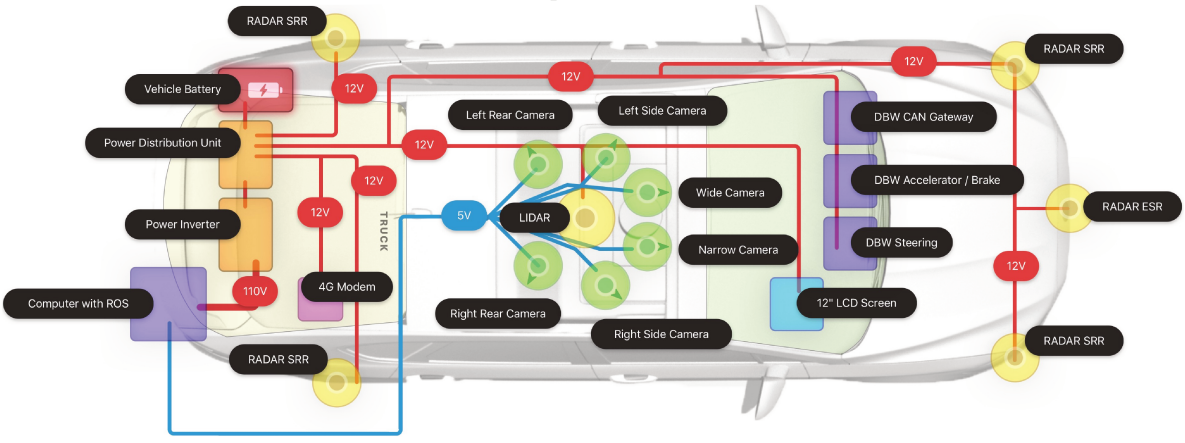
Figure 2.4 depicts physical architecture examples. For instance, Figure 2.4a depicts an architecture with a rotating lidar. Whereas, Figure 2.4b depicts another architecture based on lidars, cameras, and radars. In the first architecture, the rotating lidar has better performance in detection. In contrast, the second needs to integrate with other sensors as it has a less performant lidar.

Sensors sense the surrounding environment (e.g., road infrastructures, cars, motorbikes, pedestrians, animals) through physical signals (e.g., light beams or radio waves). Then, embedded computers process and merge sensor data into a unique representation of the surrounding environment. Finally, computers transmit signals to actuators (brake, accelerator, steering) to react according to the perceived environment.

In our context, the thesis focuses on **exteroceptive sensors** (e.g., camera, radar, lidar) that sense the external environment surrounding the AV. Where each sensor has specific technical features (e.g., acquisition frequency) and environmental performance (e.g., meteorological events, urban environment) surveyed in [Pon17]. Also, automated architectures among car manufacturers are highly heterogeneous [MV14]. Therefore, a set of ADAS applications may not rely on the same sensor.



(A) First example [Mon19]



(B) Second example [Cam17]

FIGURE 2.4: Physical architectures of a self-driving car

Automation Level	Description	Example
0- No Automation	The driver continuously carries out longitudinal (acceleration & braking) and lateral (steering) control of the vehicle.	No driver assistance system is active, which intervenes in the longitudinal and lateral control. Safety systems, such as ABS and DSC, or warning systems, may still be active.
1 - Driver Assistance	The driver continuously carries out either longitudinal or lateral control of the vehicle. The driver must continuously monitor the system and driving environment and at any time be prepared to take over full control of the vehicle.	Adaptive Cruise Control (ACC): the vehicle takes over longitudinal control and keeps a safe distance to the vehicle in front automatically.
2 - Partial Automation	The system takes over longitudinal and lateral control of the vehicle under certain conditions and situations and for a limited amount of time. The driver must still permanently monitor the system and driving environment and be prepared to take over control of the vehicle at any time.	Highway or Traffic Jam Assistant: The vehicle automatically carries out longitudinal and lateral control on the highway or in a traffic jam up to a certain speed. The driver permanently monitors the system and takes over full control when required to do so.
3 - Conditional Automation	The systems take over longitudinal and lateral control of the vehicle under certain conditions and situations and for a limited amount of time. The driver need not permanently monitor the system nor the driving environment when it is activated, but must still serve as a fallback and take over control in case the automation fails or reaches a limit.	Highway Pilot: The vehicle takes over longitudinal and lateral control up to a certain speed. The driver need not permanently monitor the system but must be able to take over control upon request within a specific time frame.
4 - High Automation	The system takes over longitudinal and lateral control of the vehicle under certain conditions and situations and for a limited amount of time. In case the driver fails to take over during a system failure or limit, the vehicle will automatically initiate a maneuver to bring the vehicle to a minimum risk condition (this is the main difference to Level 3 conditional automation).	Remote Valet Parking: Performs all driving tasks at low speeds in a parking environment, autonomously driving the vehicle from a start position to an available parking spot. A driver is not physically necessary, as the vehicle come to a stop in case of a system failure or limit.
5 - Full Automation	The system completely takes over longitudinal and lateral control of the vehicle during all conditions, all driving situations, and at all times. In the event of a system failure or limit, the systems automatically brings the vehicle to a minimum risk condition.	Automated Taxi: The vehicles completely take over longitudinal and lateral control of the vehicle during the complete journey, without any intervention or monitoring necessary by any of the passengers (a driver is most likely not present).

TABLE 2.3: Automation Levels

Level	System	Description	Sensor(s)
0	Lane Departure Warning	Warns the driver if the vehicle is imminently about to unintentionally depart the lane.	Camera
	Blind Spot Detection	Informs the driver, usually visually on the side view mirror, when other traffic is in the vehicle's blind spot, preventing a possible lane change.	Radar
	Forward Collision Warning	Acoustically warns the driver of an imminent impact in case of uninitiated immediate braking intervention.	Radar
	Park Distance Control	Acoustically informs the driver about obstacles around the vehicle during parking maneuvers.	Ultrasonic
	Parking Surround View	Visually gives the driver a reference of the vehicle's surrounding during parking maneuvers.	Camera
	Night Vision	Provides an enhanced visual view at night, sometimes with an integrated pedestrian warning system.	Thermal camera
1	Active Cruise Control	A cruise control system which automatically keeps a safe distance to the vehicle in front.	Radar
	Active Lane Assist	Keeps the vehicle from crossing the lane boundaries by actively applying a force on the steering wheel in the opposite direction when the vehicle is about to cross a lane boundary inadvertently.	Camera
	Lateral Collision Avoidance	Makes an active steering intervention during a lane change situation to avoid a collision with another overseen vehicle during the lane change.	Radar
	Park Assistant	Takes over steering control during a parallel parking maneuver.	Ultrasonic
2	Traffic Jam Assistant	Hands-on steering wheel automated driving in traffic jams up to a specific speed limit.	Several
	Advanced Park Assist	Automated steering, braking and gas control during a complete parallel parking maneuver.	Ultrasonic
	Lane Keeping Assist	Keeps the vehicle actively in the center of the lane while keeping a safe distance to the vehicle in front.	Several
3	Highway Chauffeur	The System performs the longitudinal and lateral driving task in highway	Several
	Traffic Jam Chauffeur	The System performs the longitudinal and lateral driving task in highway traffic jams (e.g., 60 km/h max)	Several
4	Traffic Jam Pilot	The System takes full control the longitudinal and lateral driving task in highway traffic jams.	Several
	Highway Pilot	The System takes full control the longitudinal and lateral driving task in highway.	Several

TABLE 2.4: Current driver assistance systems which inform or warn the driver along with the sensor technology used.

2.1.2.3 Perception Sensors

As shown in Figure 2.5, each sensor provides data to one or several ADAS functions. A Perception based on a set of sensors may depend on many criteria. The first criterion is the type of ADAS function implemented in the AV. For instance, the ADAS application named *Adaptative Cruise Control* relies only on a long-range radar. Indeed, Tables 2.6 and 2.7 depict radars as performant sensors to detect objects. The second criterion is the performances of the sensor. Indeed, Table 2.5 shows that lidars have different features and may perform their tasks with dissimilar performances.

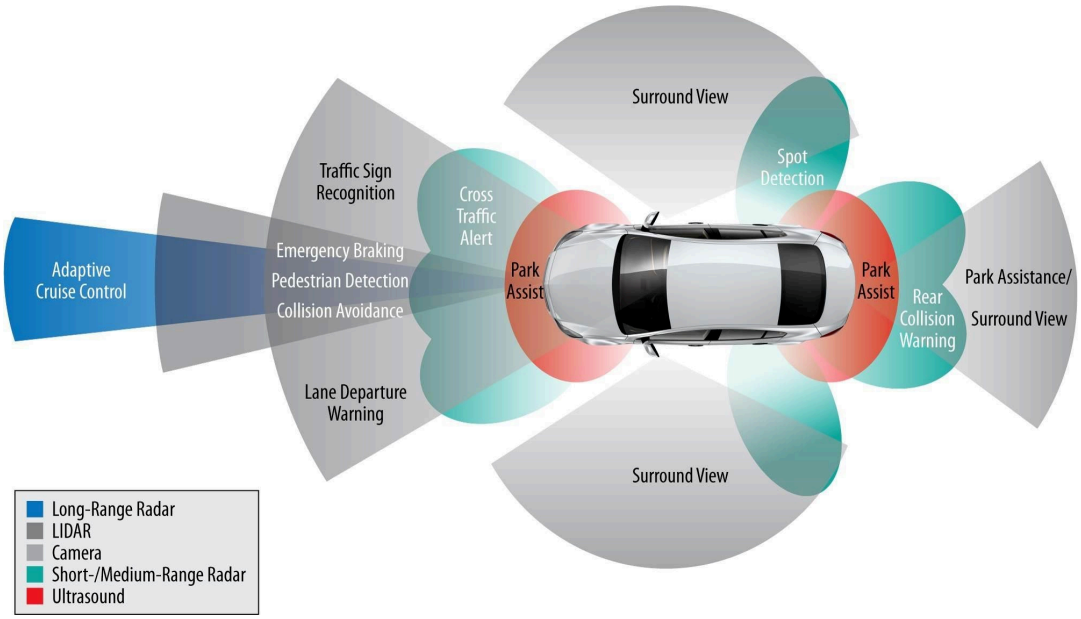


FIGURE 2.5: Examples of ADAS applications [Mur17]

Radar is a technology that uses high-frequency electromagnetic waves to measure the distance and relative speed of target objects. Radars can already be found today in middle-class vehicles for forwarding collision warning applications, lane change assistance or automatic cruise control. Other characteristics of radar sensors are their invisible integration behind electromagnetically-transparent materials, for instance, behind the front bumper. Besides a few exceptions, radars usually have no moving parts and are, therefore, more robust and less prone to mechanical failures than laser scanners. In contrast to vision-based and laser scanners, radars are robust against environmental conditions, such as changes in light or fog and rain. Other characteristics of radar sensors are their invisible integration behind electromagnetically-transparent materials, for instance, behind the front bumper.

Lidars are laser-based ranging systems. Similarly to radars, they are based on the time-of-flight of reflected light pulses and can measure the distance towards an object. The use of laser scanners is mainly in the field of obstacle detection, collision mitigation, and stop-and-go assistance. Nevertheless, the use of laser scanners for ACC is already finding its place in ITS. Future autonomous vehicles may rely on laser scanner information to get information from surrounding obstacles [Göh+11]. Environmental conditions (fog, rain, dust, and dirt) importantly limit the availability of laser scanners. Furthermore, incident sunlight in the morning and afternoon hours can cause significant disturbances on the laser detecting device. The price is still high to incorporate laser scanners into commercial vehicles. Nevertheless, they

are extensively used by many research groups to test novel advanced driver assistant systems or self-driving vehicles. Table 2.5 depicts a set of lidars used in a project in Nevada for Connected Vehicles and New Traffic Application [Xu+18].

Name	Detection Range (m)	# of Beams	Max. Measurement Frequency	FoV	Price
LeddarOne	40	1	140	3°	$115
Leddar IS16	50	16	50	45°	$940
Leddar M16	100	16	50	95°	$740
HDL 32E	100	32	10	360°	NC
VLP 16	100	16	20	360°	NC
HDL 64E	120	64	20	360°	NC
Vu8	215	8	100	100°	$650

TABLE 2.5: LiDARs from various Manufacturers [Xu+18]

Cameras detect and localize objects by processing the images drawn from an imaging device like a camera. Although vision can provide highly valuable information about the environment, image processing techniques are complicated, computationally expensive, and still under research. For automotive vision sensors, processing of road scenes can provide accurate information other sensors fail to obtain. Already today, cameras are being introduced in high-class vehicles for detecting lane marks and offer lane-keeping assistance or lane departure warning systems. Furthermore, automatic traffic sign recognition systems are already able to inform the driver about the current speed limit and other types of hazards along the road. Recently, applications for object detection incorporate camera readings. Especially the detection of pedestrians, which would otherwise fail with radar sensors or laser scanners, can be accomplished with vision-based solutions. Camera sensors are, as human visual perception, sensitive to adverse lighting conditions, for instance, fog and rain or low sun and blinded by the headlights of approaching vehicles.

In our work, we want to fuse V2X data with sensor data. Table 2.6 summarizes and compares the characteristics of V2X with other sensors [Pon17].

Vehicle Type		AV			CV		
Source		Radar	LiDAR	Vision	CAM (V2X) RSSI	CAM (V2X) GNSS	CAM (V2X) DGNSS+INS
Accuracy	position (m)	0.1	0.02	1-5	5-20	2-10	0.5-2
	velocity (m/s)	0.2	0.5	NA	NA	0.01	0.01
	orientation (°)	0.1	0.1	NA	NA	0.25-10	0.25-1
Availability		0			-	-	+
Reliability		++	+	0	0	+	++
Range (m)		250	200	40	400		
Field of View (°)		15	360	20	360		

NA: Not Available

TABLE 2.6: Relative positioning techniques [Pon17]

Task	Operational Condition	AV				CV
		Vision	Radar	LiDAR	Fusion	V2X
Object detection	Normal					
	Bad weather					
	Poor lightning					
	Occluded area					
Object classification	Normal					
	Bad weather					
	Poor lightning					
	Occluded area					
Distance estimation	Normal					
	Bad weather					
	Poor lightning					
	Occluded area					
Object edge precision	Normal					
	Bad weather					
	Poor lightning					
	Occluded area					
Lane tracking	Normal					
	Bad weather					
	Poor lightning					
	Occluded area					
Range of visibility	Normal					
	Bad weather					
	Poor lightning					
	Occluded area					

Performances Definition: □ (Good), ▫ (Fair), ▪ (Poor), ■ (Not Available)

TABLE 2.7: Sensor performances [Sch17]

2.1.2.4 Perception System & Architectures

This section provides a quick overview of perception systems and basic fusion architectures. Recent work has favored the low and feature-level fusion approach [Aeb17]. However, the high-level fusion approach [Aeb17], if implemented correctly, has large potential and many advantages over low/feature-level fusion.

The perception system regroups two parts:

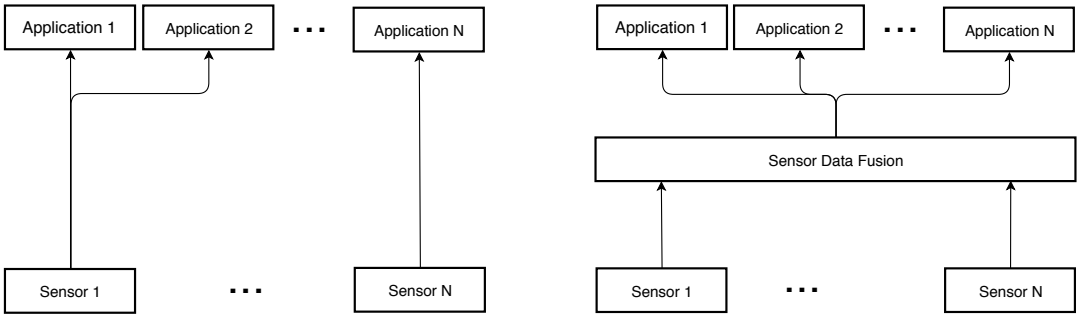
- The **Sense** part concerns non-cooperative sensors and their processes
- The **Understand** part includes a database (e.g., embedded map) and processes for data fusion.

Today, driver assistance systems have fairly basic sensor processing architectures. Indeed, one or more applications use the input from one or multiple sensors. As driver assistance systems become more complex, edging towards autonomous driving technology, this simple sensor processing architecture is insufficient. The combination of each sensor strength culminates in an accurate single perception of the environment surrounding the automated vehicle. The process of data combination, named data fusion, increases the automated system performance and reduces its performance cost [DW05; Kae+04; NF05].

Figure 2.6 shows the difference between these two basic architectural models for processing sensor data.

Direct Sensor architecture is an architecture of a perception system without data fusion (Figure 2.6a). This architecture exists if an automated system relies on a single sensor for a given task. For instance, only a camera can perform the task of traffic light recognition.

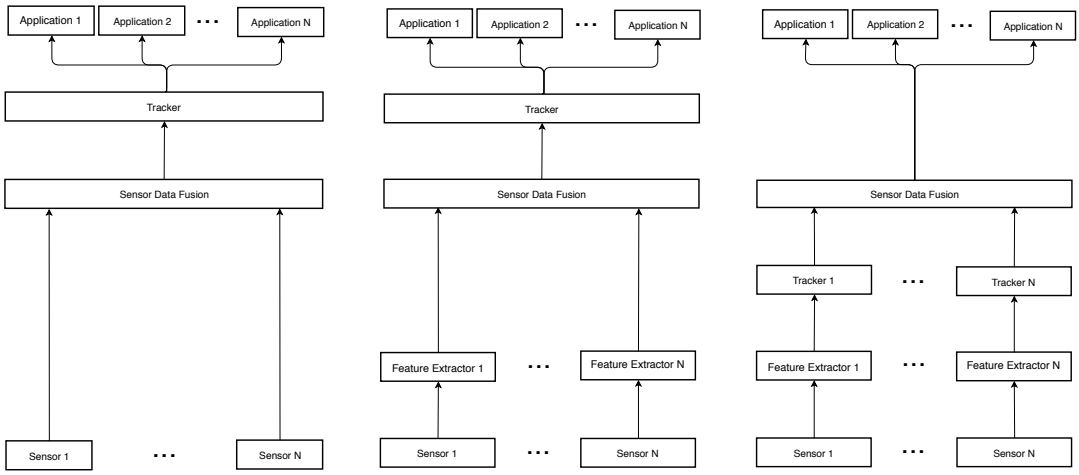
The **Sensor Fusion** architecture is an architecture of a perception system with data fusion (Figure 2.6b).



(A) Direct Sensor (B) Sensor Fusion

FIGURE 2.6: Basic Models of perception Architecture [Aeb17]

In the literature, we can find several generic architecture proposals for automotive perception [BTC13; BT16; Ulb+17; BP99]. In airborne radar applications, the aerospace industry fuses radars data to track aircraft objects. The visualization of a detected object is a point due to the significant distance between the radar station, and the detected aircraft. However, this assumption does not hold in automotive applications, since a detected object, such as an overtaking vehicle, can fill the entire field-of-view of a sensor. Therefore, sensor data fusion algorithms and architectures must be newly investigated for ADAS applications.



(A) low-level (B) feature-level (C) high-level

FIGURE 2.7: Architecture models used in sensor fusion [Aeb17]

Low-level Fusion: Figure 2.7a illustrates a low-level fusion architecture. In a low-level fusion architecture, no pre-processing of raw data takes place at the sensor-level. Each sensor transmits its raw data to the fusion module, which then performs a low-level fusion of the raw data from all of the sensors. Then, the fused raw data serves as inputs to a central tracker before reaching each ADAS application. For example, in the automotive context, a pre-crash application uses a low-level fusion method [Pie+09]. The advantage of low-level fusion is to classify data at a very early stage through the fusion of raw data from different sources. However, low-level fusion requires high data bandwidth and can be complex to implement in practice. The sensor measurements require a high likelihood of being a relevant object. Also, the addition of a new sensor to the architecture requires significant changes to the fusion module, since raw data from different sensor types come in different formats. Thus, the addition of extra processing to align the data format is mandatory.

Feature-level fusion: a feature-level fusion architecture process raw data to extract features before fusion. Then, the tracking algorithm uses the extracted features as inputs. Figure 2.7b depicts the architecture of a feature-level fusion. The main advantage of feature-level fusion is the reduction of required data bandwidth between sensors and the fusion module. Indeed features extraction includes the aggregation of raw sensor data. Also, feature-level fusion retains the same classification and pre-processing capabilities of low-level fusion, allowing for a similar efficient integration of relevant data into the tracking algorithm. Several works use feature level fusion in automotive applications [KBD05; Mah+06].

High-level Fusion: Also named track-to-track fusion, the architecture is the opposite of low-level fusion. Each sensor data is transformed from a raw state (e.g., pixels) to a refined state (e.g., pixels cluster). Figure 2.7c depicts a high-level fusion architecture. The main advantages of high-level fusion are the architecture modularity and the encapsulation of sensor-specific details. All sensor details remain at the sensor-level. Therefore, the fusion module processes abstracted data. For instance, there is a single measurement per sensor object (e.g., centroid center) instead of multiple measurements per sensor object (e.g., measurement within the centroid). Therefore, high-level fusion architectures favor applications with modular design requirements (e.g., camera, radar, lidar). However, object classification becomes more difficult because the sensor-level tracking algorithms have less information when associating raw data measurements to relevant objects. The definition of a high-level fusion is a complex task because each sensor tracker requires a definition specific to the sensor capability and reliability (e.g., acquisition frequency). Wrong tracking settings may impact the overall module performance of data fusion. Despite these disadvantages, several works use high-level fusion architectures in automotive applications [TYI04; Flo+07].

2.2 Vehicular perception failures

While developing a highly autonomous system, the requirements to ensure the driver and the road users' safety increases. Indeed, the autonomous vehicle must drive in challenging environments for sensors. These environments include lousy weather, highly cluttered area, damaged road infrastructures, or broken hardware. In such conditions, the CAV must not fail in achieving an ADAS function. For now, the analysis of the causes leading to a potential failure in the perception system is the focus of two domains: automotive safety and security.

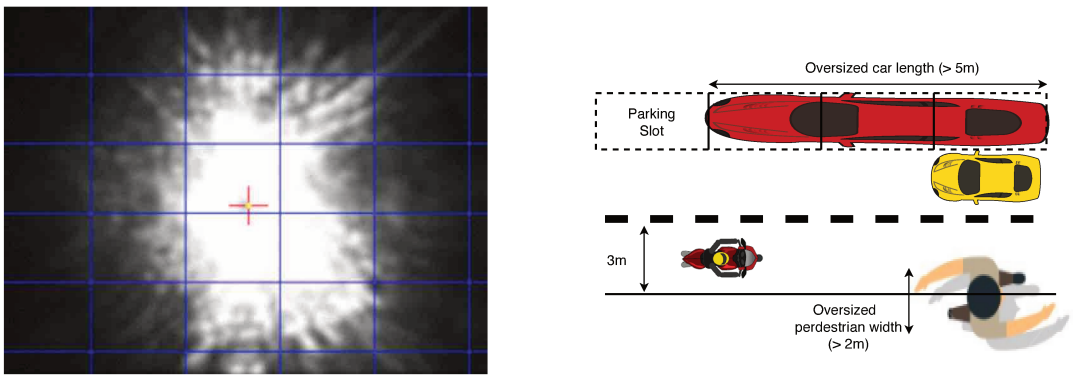
The leading standard defining automotive Safety is ISO26262 [ISO11]. This standard regroups all the words and framework definitions used to perform a safety risk assessment. The purpose of risk assessment is to link each failure case with all potential hazardous events that lead to an accident. In our work, we use the following definitions when referring to automotive safety:

- **Item** is a system or array of systems which implements a safety-related function (e.g., steering, braking, transmission) to which ISO26262 applies.
- **System** consists of elements (sub-systems, components, HW, SW) and relates a sensor, controller, and actuator with each other.
- **Component** is a none system-level element which consists of more than one HW part or more than one SW unit.
- **Hardware Part** is an indivisible hardware component.
- **Software Unit** is an atomic level of the Software architecture tested as a standalone part.
- **Element** is a system or part of a system, including components, hardware, software, hardware parts, and software units effectively, anything in a system that can be distinctly identified and manipulated.
- **Hazard** is a potential source of harm caused by malfunctioning behavior of the item.
- **Fault** is an abnormal condition that can cause an element or an item to fail. There are several types of faults [Jha+18]. First, Data Faults can originate from a faulty sensor due to an alteration of its settings or to world perturbation (weather or degraded road infrastructures). Next, Hardware Faults can originate from a mechanical malfunction of a hardware component leading to bit or stuck-at faults. Timing Faults which can be caused by a high propagation time in a highly obstructed external environment (e.g., urban scenario). Another cause is the processing time caused by a highly dense external environment (e.g., many pedestrians).
- **Error** is a discrepancy between a computed, observed or measured value or condition, and the real, specified or theoretically correct value or condition.
- **Failure** is a termination of the ability of an element to perform a function as required. Failure is systematic or random hardware (e.g., Aging or Oxidation). Systematic Failures relates to processing (e.g., bugged specifications), software (e.g., programming error), or hardware (e.g., insufficient immunity to environmental conditions).
- **Exposure** is state of being in an operational situation that can be hazardous if coincident with the failure mode under analysis
- **Operational situation** addresses the limits within which the item is expected to behave safely. For example, an average passenger road vehicle is not expected to travel the cross country at high speed. Operational situations include visibility, road surface traction, road surface unevenness, road surface bank angle change, road surface pitch change, objects in the path of the vehicle, objects on a trajectory intersecting the path of the vehicle, relative velocity of the vehicle and the object it is approaching, relative to the distance (gap).

In our thesis, we use the following definitions when referring to automotive security:

- **Threat** is a circumstance or event with the potential to cause harm, where harm may be for financial, reputation, privacy, safety, or operational factors.
- **Attack** classification regroups two groups which are basic and sophisticated attacks [RH07]. Basic attacks are a single primary threat performed in a context (e.g., VANET). Sophisticated attacks are combinations of basic attacks.

In our thesis, the cause of a perception failure is intentional (Figure 2.8) or unintentional (Figure 2.9). For instance, a person may flash the front camera of a CAV intentionally, as seen in Figure 2.8a. Similarly, Figure 2.9a shows that an extreme operational situation may also blind the front camera. In both scenarios, the camera may fail to recognize a traffic sign or road marks (Figure 2.9b). In addition to attacks against sensors, V2X communication can be the source of attacks. For example, a malicious CAV may lie on its length to occupy several parking slots, as shown in Figure 2.8b.



(A) Flash blinding [Pet+15c] (B) CAV Communication Deception

FIGURE 2.8: Intentional perception failures

(A) Sun blind (B) Snow occlusion

FIGURE 2.9: Non intentional perception failures

Besides, an attacker may broadcast messages with fake positions. As seen in Figure 2.10, our perception system receives several V2X messages that are legit (identified as number 6, 7, and 8) or malicious (identified as number 9, 10, 11, and 12). As a consequence, the perception system will display these fake messages to the driver.

Besides, the decision module may activate the automated braking system if a malicious message contains the position of a connected vehicle in front of him. In this scenario, the perception system failed to perceive the road environment correctly.

For now, current works in automotive perception do not consider such failures. A failure scenario defines a situation where the perception failure fails to accomplish a task (Table 2.7) due to an attack or a fault. Accordingly, our generic failure resilient perception architecture must solved each failure scenario. Thus, we define several failure scenarios divided in three uses cases: classification, detection, and self-localisation failures. For scenario details, readers may refer to Appendix B. In the scope of thesis, we will focus on **perception failures due to an attack**. Thus, it is mandatory to model an attacker, a threat, and an attack to understand and design security mechanisms for a resilient perception architecture.

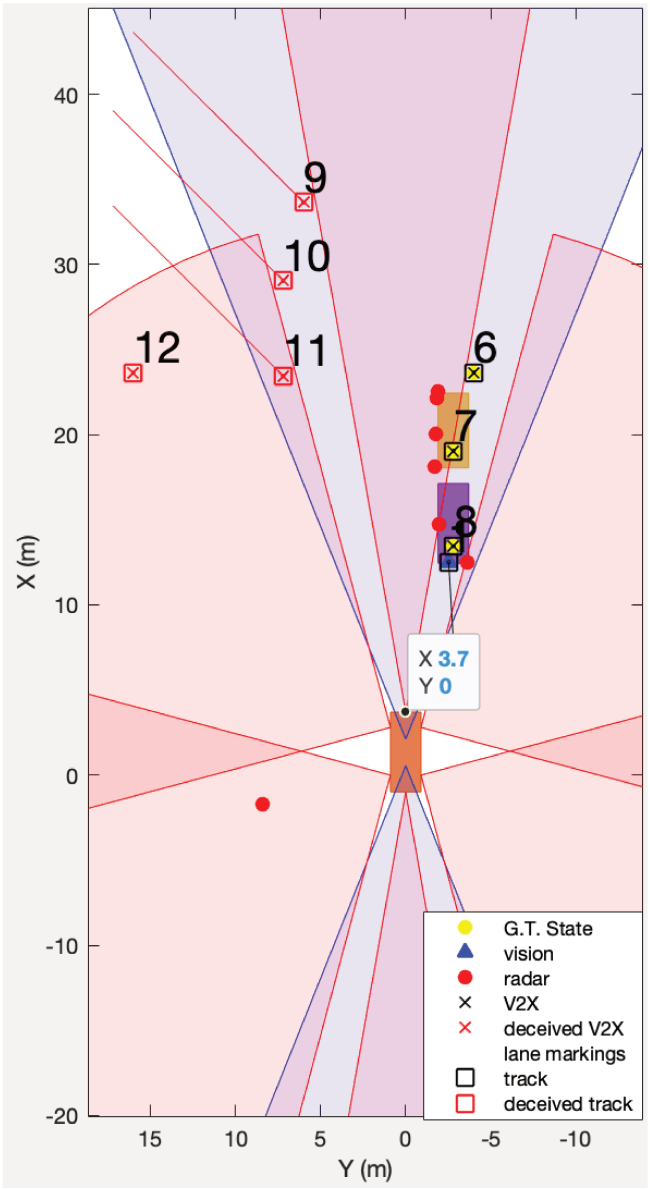


FIGURE 2.10: Perception Failure due to V2X attacks

2.2.1 Attacker Model

An Attacker model classifies the capacities of an attacker. Based on its capacities and on its target, an attacker can launch a specific set of attacks. Therefore, we need an attacker model to identify the types of attacks that can be performed. Then, the identification of these attacks will help to propose adequate security mechanism. Currently, few works has been done on attacker model which are VANET centric [RH07] (Figure 2.11) or in-vehicle centric [Pon+16; PFK14].

In general, an attacker model is composed of the following dimensions:

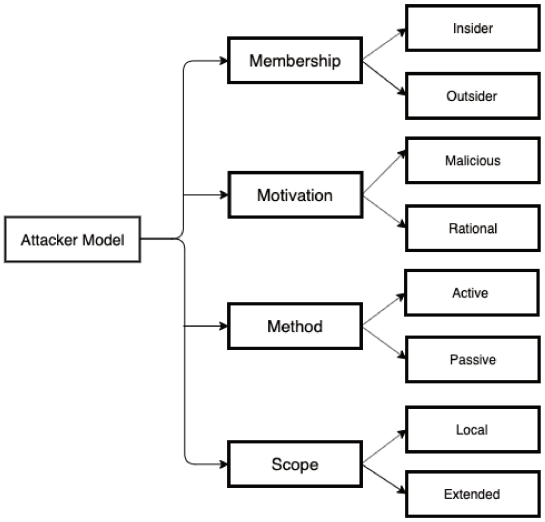


FIGURE 2.11: Attacker Model [RH07]

1. **Membership** (*Insider* or *Outsider*): The insider is an authenticated member of the network that can communicate with other members. The network members consider the outsider as an intruder and hence is limited in the diversity of attacks he can mount (especially by misusing network-specific protocols).

2. **Motivation** (*Malicious* or *Rational*): A malicious attacker seeks no personal benefits from the attacks and aims to harm the members or the functionality of the network. Hence, he may employ any means of disregarding corresponding costs and consequences. On the contrary, a rational attacker seeks personal profit so hence is more predictable in terms of the attack means and the attack target.

3. **Method** (*Active* or *Passive*): An active attacker can generate packets or signals, whereas a passive attacker contents himself with eavesdropping on the wireless channel.

4. **Scope** (*Local* or *Extended*): An attacker can be limited in scope, even if he controls several entities (vehicles or base stations), which makes him local. An extended attacker controls several entities that are scattered across the network, thus extending his scope. This distinction is especially crucial in privacy-violating and wormhole attacks.

However, **our work considers attacks on the perception system which includes attacks on sensors and on V2X communication but also on the perception algorithms (e.g., modifying a speed limit from 90 km/h to 80 km/h). Thus, it is mandatory to investigate the potential attackers targeting perception**.

2.2.2 Threats Model

This section describes the security threats facing vehicular networks. One of the standard approaches to defining a threat is the "Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege" (STRIDE) model [HL02]. STRIDE defines fives attack types that may threaten system elements.

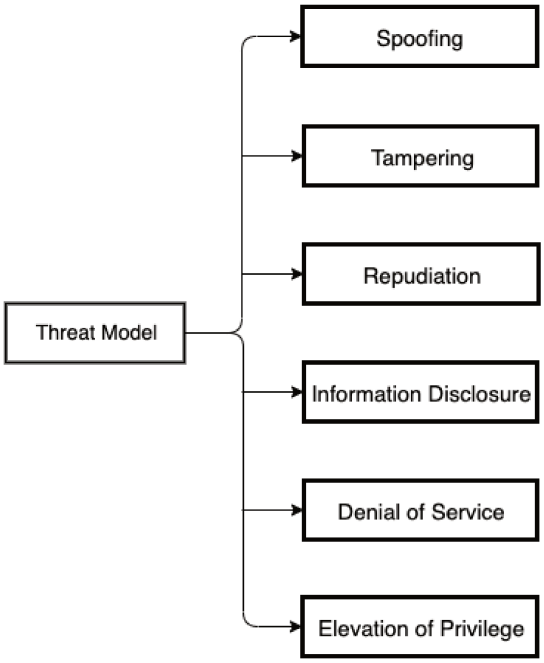


FIGURE 2.12: Threats Model (STRIDE [HL02])

- **Spoofing** is the impersonation of something or someone else. The resulting attack aims to spoof the authentication layer. Authentication layer is used to protect legitimate nodes from rogue insiders and outsiders. A suitable authentication mechanism generally prevents against several attacks, including black holes, spoofing GPS signals, and replay attacks. The objective of this mechanism is that the resources and services should be accessed only by authenticated users. An example of spoofing type of attacks is masquerading attacks. The attack appears as a legitimate node to other connected users. Indeed, the attacker fabricates false messages and sends them to surrounding vehicles to impact the vehicle's decisions. A malicious vehicle node can try to act as an emergency vehicle, and thus deceive other vehicles by broadcasting false safety messages into the V2X network. Thus, the deception results in other surrounding vehicles believing that another vehicle is responsible for this attack. When malicious nodes fail or refuse to forward messages, then they create a black hole attack. The black hole attack can be injected into any ad hoc network and is a standard attack against the authentication mechanism. Blackhole attack means that a legitimate vehicle never receives messages because of the malicious vehicle which pretends to be part of the network. The malicious vehicle is not a legitimate node in the network. As a result, legitimate vehicles become vulnerable to such attacks from vehicles. Insider actors in the network usually perform these attacks. Another spoofing attack is GPS spoofing. In vehicular systems, GPS position information is an essential variable for the vehicle and should be accurate. This information usually comes from Global Navigation Satellite Systems (GNSS). In spoofing attacks, a GPS

satellite simulator is used to generate radio signals or messages that overwrite the signals from the accurate GPS satellite. This way, an attacker can spoof the vehicle to receive and process a different location than the one that they are. The attacks can cause severe consequences for a car.

- **Tampering** is the modification of an element. Attacks attempt to modify or inject malicious code or messages in the execution of the program. The attack has the potential to disrupt the operations of the vehicular network, OBUs, and RSUs because they receive periodic updates. An example of tampering attacks is when they generate and broadcast false safety messages (BSM). The attack is often made to deceive other vehicles and get other vehicles to behave in a specific manner. In addition to broadcast tempering, an attacker can also tamper transaction messages in flight. Tampering attacks belongs to the class of active attacks.

- **Repudiation** is the refutation of a performed. The attack happens when a vehicle refuses to accept the message causing the sender node to resend the message. Usually, this happens when the receiver does not verify the sender authenticity or freshness.

- **Information Disclosure** is the exposition of information to someone or something unauthorized to see it. Attacks attempt to violate the confidentiality of messages. These attacks are often used to track or record certain confidential information and have privacy consequences. A typical example of these is an eavesdropping attack. These attacks only impact one vehicle and attempt to collect user or other information about that vehicle (e.g., payment information or identity information). Another example related to information disclosure is when attackers try to exploit vehicle tracking information. In general, an OBU sends out a safety message to inform other surrounding vehicles for traffic or safety situations. This message contains the OBU certificate and other identifiers. If the attacker can track this piece of information across time, then it can track vehicle location.

- **Denial of Service** is the denial or the degradation of an element. Attacks attempt to bring down or overload the communication medium either by jamming signals at the physical layer, thereby causing channel jamming or by flood-ing the nodes, so the vehicle nodes are prevented from accessing the network. The primary purpose of the attacker in a denial of service attack is to prevent legitimate vehicles from accessing the V2X network and exchanging messages with other vehicles. A DoS intruder may attack either the individual vehicles (OBUs) or roadside units. An implementation of Denial of Service attack is flooding. These attacks flood the network with many false messages generated by malicious vehicle nodes. The attack floods the OBUs and RSUs and unable to communicate with each other over the V2X channel. Also, this attack results in the loss of critical safety messages. Thus, the legitimate vehicle nodes cannot warn other nodes. Spamming attacks are another type of DoS attack, and they occur when an intruder sends a series of messages to consume the network resources. The control of this attack is difficult in V2X as there is no centralized infrastructure. Lastly, jamming attacks disrupt the communication channel at the physical layer by injecting noisy signals to halt message transmission delivery. Thus, the communication channel goes down, and the vehicles are unable to communicate with each other or infrastructure services. Jamming is also used to hide the identity of the attacker.

- **Elevation of Privileges** is the capability gain, without authorization, over an element. An attack happens when messages attempt to obtain higher privileges. For example, fake high priority messages which attempt to flash malicious software would consist of this type of attack. Because this dissertation looks at safety messages, the elevation of privilege attacks has similar properties to the tempering attacks.

For each threat, a counter-measurement must ensure the corresponding security goal.

- **Authenticity** is the process of verifying the uniqueness of an information (e.g., message or a vehicle)
- **Integrity** is the assurance that the information is trustworthy and accurate
- **Non-Repudiation** is the association of indisputable actions with a unique individual.
- **Confidentiality** is ensuring that information is accessible only to authorized entities.
- **Availability** is a guarantee of reliable access to the information by authorized people
- **Authorization** is the selective restriction of access to a place or other resource.

As seen, current models do not include new threats specific to the context of CAV such as data privacy. Thus, a new threat model in the context of CAV perception is needed.

2.2.3 Attack Model

However, in complex systems such as in a cooperative autonomous vehicle, each element of the perception system is vulnerable to one or multiple threats. Table 2.8 present a set of attacks targeting elements related to the automated driving context.

Targeted Element	Attack Goal	Feasibility	Success Probability	Mitigation Technique
Road Infrastructures	Alter Signalisation	high	low-medium	Data Redundancy, harden infrastructure, reporting
	Remove Signalisation	high	low-medium	
Machine Vision	Blind	high	high	Data redundancy
GPS	Spoofing	high	high	Authentication
	Jamming	high	high	Anti-Jammer or IMU
In-vehicle devices	Malware Injection	high	medium	Secured In-Vehicle Architecture, IDS, Anti-virus, Firewall
	Eavesdropping	high	medium	In-vehicle security
RaDAR/ LiDAR	Jamming	high	medium	Data redundancy
	Spoofing	high	medium	Data redundancy
Embedded map	Alteration	low	medium	Server authentication
ITS-Stations	Faulty messages	high	high	Authentication, data redundancy
	Refuse to communicate	high	high	Data redundancy
	Jamming	high	high	Authentication

TABLE 2.8: Subset of Attack Surfaces in a cooperative automated vehicle [PS15]

2.3 Threat Analysis and Risk Assessment (TARA)

Usually, Threat Analysis and Risk Assessment (TARA) methods assess the risk of traditional *IT* infrastructures without considering safety implications. Unfortunately, as mentioned, automotive attacks can have safety impacts. The Society of Automotive Engineers (SAE) defines the security framework (Figure 2.13), in SAEJ3061 standard [SAE16]. This standard proposes tu use one of the following risk assessment methods: EVITA [Hen+09], TVRA 2015 [ETS11], and HEAVENS [Isl+16].

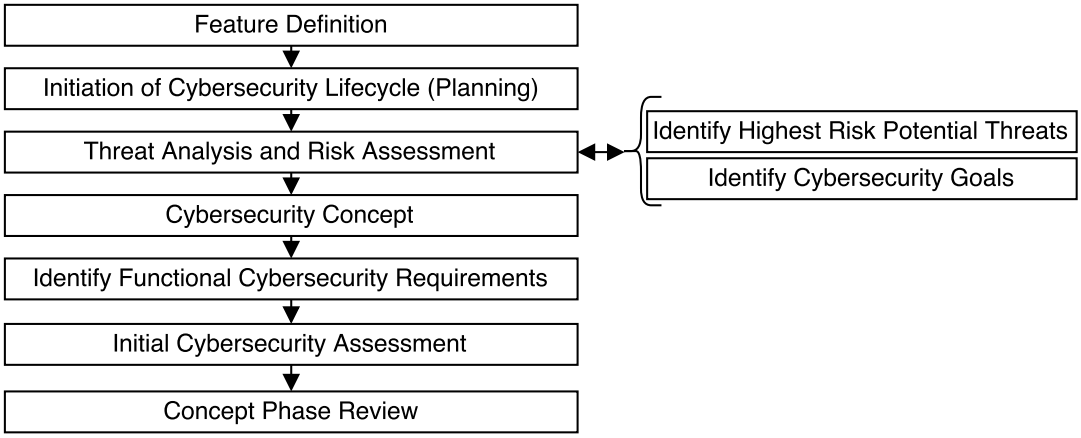


FIGURE 2.13: SAE J3061 Concept Phase [SAE16]

During the literature review, we surveyed automotive security risk assessment methods [Mon+18c]. Most methods rely on the *STRIDE* threat model that do not capture the recent threats related to data trustworthiness [Tan+17] or data linkability [EWS15].

In Table 2.9, we provide a comparison of the related work in terms of: vehicle type, attacker model, threat model, security goal model, attack type, attack model, type of controllability, privacy impact and, safety impact.

Related Work	Vehicle Type	Modeling			Attack Definition		Vehicle Controllability	Impact	
		Attacker	Threat	Attack	Complexity	Scalability		Privacy	Safety
EVITA [Hen+09]	CAV	Missing	STRIDE (Insufficient)	Attack Tree	multi-threats	Severity	Driver Control	EVITA Severity	EVITA Severity
Wolf et al., [WS12]	AV	Missing	STRIDE (Insufficient)	Attack Tree	multi-threats	Severity	Missing	Missing	Severity
Moalla et al., [Moa+12]	CV	Missing	Threats List (Limited)	Missing	mono-threat	TVRA Impact	Missing	Missing	Missing
SAHARA [Mac+15; Mac+16]	AV	Missing	STRIDE (Insufficient)	Missing	mono-threat	R-metric (DREAD)	Driver Control	Missing	ASIL Level
RACE [Bou+15]	CAV	Missing	Threats List (Limited)	Attack Tree	multi-threats	TVRA Impact	Driver Control	EVITA Severity	EVITA Severity
Dominic et al., [Dom+16]	CAV	random List	STRIDE (Insufficient)	Threat Matrix	mono-threat	Missing	Missing	Impact Level	Impact Level
HEAVEN [Isl+16]	AV	Missing	STRIDE (Insufficient)	Data Flow Diagram	multi-threats	Impact Level	Missing	Impact Level	Impact Level
DEWI [Ste+16]	CV	Missing	STRIDE (Insufficient)	Missing	mono-threat	Missing	Missing	Missing	Missing
Schmittner et al., [Sch+16]	CAV	Random List	TID (Insufficient)	Missing	mono-threat	SAE Severity	Driver Control	SAE Severity	SAE Severity
TVRA 2017 [ETSa]	CV	Random List	STRID (Insufficient)	Threat Tree	multi-threats	Missing	Missing	Missing	Missing
SARA [Mon+18c]	CAV-L5	Metric	STRIDELC	Attack Tree with attacker	multi-threats	Revisited SAE Severity	Automated Driving System Control	Revisited SAE Severity	Revisited SAE Severity

TABLE 2.9: Comparison of TARA methods

In 2009, Henniger et al., [Hen+09] proposed EVITA. They defined a risk matrix considering the attack likelihood, the attack severity, and the driver controllability. However, their attack tree definition is unclear. Indeed, the confusion comes from their distinction between attack goals and objectives that are respectively, the roots and the second nodes of their attack tree. Also, they only consider driver control during the risk computation which does not work with driver-less vehicles. To this end, we introduce the **CAV observation and controllability metrics** later to be compliant with SAE standard[SAE16]. Indeed, SAE standard requires the CAV to self-observe potential faults or failures (caused by hazards or threats) to self-control vehicle dynamics and reduce the safety and security risks.

In 2012, Moalla et al., [Moa+12] applied TVRA on a connected vehicle. They did not consider threats from the internal vehicle network. Besides, their considered threats list is not exhaustive, due to the absence of threat modeling despite standards recommendation [SAE16; ETSa]. Wolf and Sheibel [WS12] applied their security risk assessment framework to a generic ECU model. The framework suits for subsystems but not for the whole vehicular system. Also, the method does not assess the privacy impact on security risk.

In 2015, Boudguiga et al., [Bou+15] proposed a method, named RACE, combining TVRA and EVITA. The authors clarified the definition of EVITA attack tree for automotive experts by using automotive functions instead of EVITA attack objectives. Besides, they proposed a unique risk computation method using EVITA controllability that matches TVRA rating of risk. However, they did not demonstrate RACE feasibility nor the impact of scalable attacks on the computation of risk value.

In 2016, Macher et al., [Mac+16] proposed a method named SAHARA. Their method framework just maps attack goals to ISO26262 safety use cases. However, the framework does not allow interactions between security risk and safety metrics. Their method uses the threat model STRIDE [HL02] which does not consider authentic messages with false data attacks. Also, STRIDE fails to consider attack with multiples security goals. Finally, authors used *DREAD* [LH02] to assess the security risk. Unfortunately, the discovery of a new attack affects the computation of DREAD. Indeed, if a blog advertises an attack, the value of the metric *discoverability* increases. Then, it increases the values of metrics *reproducibility*, *exploitability*, and *affected users* because, thanks to the leak, an attacker knows how to reproduce and perform the attack massively. That is, DREAD is not suitable for assessing risk. Islam et al., [Isl+16] combined STRIDE to Data Flow Diagrams (DFD) to categorize vulnerabilities on an automotive speed limiter. However, as they studied only the speed limiter, their approach does not scale to the whole vehicle system. Dominic et al., [Dom+16] proposed a method for autonomous driving systems. As required by standards [ETSa], the authors used attacker profiles to compute the risk value using the metric *Motivation*. However, they only consider surface attacks and not internal attacks such as *ECU confusion attack* [PFK14].

In 2017, the European Telecommunications Standards Institute (ETSI) provides a revised version of TVRA [ETSa]. TVRA relies on industry-proven methods (e.g., Target of Evaluation [Cri17]) and metrics (e.g., attack potential [08; 09]) to assess security risk. Also, TVRA mandates to identify attackers for the computation of risk. However, there is no proposed solution to relate an attacker to its attack. Moreover, TVRA [ETSa] focuses only on telecommunication threats. Therefore, it misses the automation threats domain [PS15] and ISO26262 *safety* for the risk computation.

In Table 2.9, we resume the main related work using multiple criteria. We can conclude that the main used methods do not consider a highly CAV as a system of study. These methods always rely on driver control. However, a CAV must rely only

on self-control vehicle dynamics to reduce risk in case of failure from an automated feature [SAE16]. Also, despite standard requirement, many methods do not link the attacker to its attack. Moreover, many methods has insufficient threat-security goal modeling against *vehicle tracking misbehaving nodes* threats [PS15]. Also, despite having a threat model, methods consider only mono-threat attack instead of multiples threats attacks as mentioned in state of the art [PS15] due to the lack of attack modeling. Finally, some methods do not consider the impact on safety or privacy despite the European Commission recommendations. In this thesis, we propose SARA, an improved security risk analysis method for CAV, which comprises safety experts opinions, a new threat model, attack method/asset map, and attack tree definition including the attacker as a metric. Moreover, we define a new metric which considers driver/CAV controllability for the computation of the risk value.

2.4 Machine Learning based Failure Detection

Currently, machine learning application have gained recent interest. In our work, we investigated the usage of machine learning to detect abnormal perception data. Indeed, the application of machine learning techniques to detect attacks has not been sufficiently evaluated in the context of CAV [Hei+16].

2.4.1 Related Work

In Table 2.10, we review the main ML methods used for detection. We grouped each work per year of publication and authors. Then, we highlight the data location in the V2X Stack (Communication layer), the type of connected object, the ML model, the methodology to train and test their ML model, and the type of threat detected by the classifier.

Raya et al;, [Ray+07] used entropy to represent the "abnormal" and "normal" behaviors of nodes, and k-means clustering to identify outliers which are the assumed attackers. Another assumption is the existence of honest nodes majority. Thus, the suspected nodes eviction relies on distance enlargement and deviation between the attacking node and the majority of honest nodes. The scheme uses the position of each observed station to compute the entropy.

Tian et al., [Tia+10] proposed a centralized intrusion detection system based on RSU for VANET. The network of connected buses, named BUSNet, individually eavesdrops and collects the data packets and routing control messages exchanged in VANET. The BUSNet forward the information to RSU to process and to detect anomalies based on a neural network.

Grover et al., designed a framework for differentiating between legitimate and malicious nodes in VANET [Gro+11]. They used a machine learning approach to classify multiple misbehaviors node in VANET using behavioral features of each node. These features are speed deviation, distance, received signal strength (RSS), the number of packets generated, delivered, dropped, collided. They measured the accuracy of two classifier types. The first one is a binary classifier whereas the second one is a multi-class classifier. Also, the authors extracted the features of packets by performing experiments in NCTUns-5.0 simulator with various simulation scenario and calculated by nearby observer nodes. Also, they used WEKA to classify the misbehavior with several classifiers: Random Forest (RF), J-48, Naive Bayes, Ada Boost1, and IBK. Experimental results show that RF and J-48 classifiers perform better compared to other classifiers. The RF and J-48 classifier gives better classification

due to the boosting and bagging properties. Then, their extension used a majority voting scheme to improve the detection accuracy of their classifier [GLG11]. The voting scheme is plurality vote and decides based on the label which received the most vote among all voting classifier. The proposed system shows a better result than any model used singly.

Dutta et al., [DC13] used a fuzzy time-series clustering for Sybil attack detection. Their proposed scheme leverages the dispersion of vehicles by clustering their locations. Sybil nodes are detected as those closely located and move for an unusually long period.

Sedjelmaci et al. [SS14; SS15] proposed an intrusion detection framework, named AECFV, which monitors node mobility and frequent changes in a network topology. At its core, there is a clustering algorithm, where cluster-heads are selected based on the trust level of each vehicle and a boundary distance. Trust levels are evaluated based on majority voting and a reputation protocol and are broadcast periodically within the network. The proposed framework uses two detection systems and a single decision system. The first system runs locally at each cluster member and monitors the neighboring vehicles and the cluster-head. The second system runs globally at the cluster-head level and evaluates the trustworthiness of its cluster members. The global decision system runs at the roadside unit (RSU) level, computes, and classifies each vehicle based on the level of trust. Together, these systems constitute a network IDS as they take a decision based on monitoring of behaviors of nodes within their radio range. The two IDSs use rules and support vector machines to classify vehicle behavior. The network simulator is NS-3. Their scheme outperforms T-CLAIDS [KC14].

Li et al., [LJF15] proposed a context-aware security framework for VANETs based on SVM algorithm. The objective of the proposed framework is to automatically differentiate between malicious nodes from abnormal nodes due to contextual reasons such as movement speed, temperature, and transmission range. The proposed framework has three functional modules, start with behavior data collection, then context sensing and processing, finally the misbehavior detection. In the experiment, they generated a dataset thanks to the simulator named GloMoSim [ZBG98]. The results demonstrate that the proposed framework achieves excellent accuracy, recall values, and an acceptable value of communication overhead.

Alheeti et al., [AGM15a] an intrusion detection mechanism for the VANETs based on Artificial Neural Networks (ANNs) to detect a specific type of Denial of Service (DoS) attack known as black hole attacks. The classifier uses spatial, temporal, and networking features as inputs for the training and testing phase. Their simulation framework is NS2, SUMO, and MOVE. The proposed mechanism shows high error rate despite having a high accuracy score and a low false-positive alarm rate.Their first extension includes a method, named Proportional Overlapping Scores (POS), which reduces the number of features extracted from the trace file. In the context of black holes detection, the POS method ranked networking features above state features. Also, the extension uses the fuzzy set method, which improves the separation between label types. Besides, the classier is a feed forward neural network (FFNN). Overall, the classifier performances improved. However, the mechanism requires more memory and computation resources than the previous work. In a second extension [AGM15c; AGM16], the intrusion detection mechanism focuses on the detection of a new type of DoS attacks known as the grey hole and rushing attacks. This extension used two classifiers which are FFNN and SVM. Overall, FFNN has the best detection rate for grey holes. However, SVM has the best detection rate for rushing attacks. In a third extension [AM16], the authors includes a new feature based on an

hashed ICMetric number. Mathematical functions generate the "ICMetric" number based on sensors readings (magnetometer [AM16], gyroscope [Alh+17], and infrared sensors [AM17]). Additionally, the hash function outputs a hash from the ICMetric number. The classifier is K-NN. Overall, the scheme shows a higher accuracy rate of detection with low false alarms rate than their previous proposal [AGM15a]. In their last extension, the authors evaluated and compared the performance of the Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) to detect Dos attacks [AGM17]. Unlike previous, the framework uses only V2X data without features selection. Overall, the LDA classifier outperforms the QDA in both detection accuracy and computation time. In comparison with previous work, their scheme outperforms previous proposals in terms of error rates and false alarm rate [AGM15b].

Wahab et al., [Wah+16] proposed a mechanism based on SVM to detect misbehaving node.

Berlin et al. [Ber+16] proposed the idea of a security information and event management system (SIEM) for connected vehicles based on machine learning. However, the author did not provide any specification and implementation.

Kim et al., [Kim+17] proposed a collaborative security attack detection mechanism in a software-defined vehicular cloud architecture. Each vehicle analyses the received information and transmits the result periodically to the controller for training the support vector machine. After training, each vehicle classifies nodes. However, this method is energy inefficient.

Ghaleb et al., [Gha+17]proposed a model based on Artificial Network (ANN) using feedforward and backpropagation. The mechanism uses historical data from to classify normal or malicious data. The classifier uses the NGSIM dataset.

Gu et al., [Gu+17b; Gu+17a] used driving patterns of vehicles and machine learning model (nearest neighbors and support vector machines) to detect Sybil attacks.

Sharanya et al., [SK17] proposed the use of the Support Vector Machine (SVM) algorithm with Modified Fading Memory (MFM) to classify legitimate and malicious nodes. The purpose of the MFM is to reduce the computational overhead for the machine learning algorithm by only considering as eligible nodes those in the range of the VANET communication only for a limited time.

So et al., [SSP18] proposed a machine learning-based mechanism to detect V2X message with malicious content. The scheme training and testing use the VeRemi dataset. The classifier used are KNN and SVM. In an extension, the authors used an additional feature which is RSSI [SPS19]. Gyawali et al., [GQ19] used a similar approach by using the same dataset. However, they tested their classifier through the simulator VEINS. In addition to previous work, the author compared different models.

Subba et al. have combined several promising ideas for VANET IDSs into a single multi-layered framework, which they have shown to be effective against a variety of different attacks [SBK18]. In all cases, detection compares audit features against thresholds. These include packet delivery rates (PDR) and Received Signal Strength Information (RSSI) for selective forwarding (gray hole) and blackhole attacks; duplicate packet rate and packet forwarding rate for denial of service; RSSI and PDR for wormhole attack; and the z-score of RSSI for Sybil attack. Evaluation based on NS-3 simulation has shown that this framework can achieve greater accuracy and lower overhead in terms of IDS-specific network traffic generated than [DR13; SS15; KC14]. The reduction of IDS traffic overhead is the result of adopting a game-theoretic approach in modeling the interaction between the IDS and the malicious vehicle as

a two-player non-cooperative game and using the Nash Equilibrium to inform the choice of the monitoring strategy.

Sharma et al., [SPL18] used the Pearson Correlation to detect location forging attacks. The proposed solution works in real-time and requires at least four to seven seconds of history to be fully efficient. Experiments used the real datasets from Wyoming Connected Vehicle Pilot Deployment. [USDb].

Eziama et al., [Ezi+18] proposed a Bayesian deep learning approach to detect VANET anomalies. However, the author did not evaluate their contribution.

Zeng et al., [Zen+18] proposed a machine learning-based intrusion detection methods to detect intruders in VANET automatically. They used ANNs and SVMs for implementing their approach.

Kamel et al., [Kam+19] proposed confidence based plausibility checks to detect anomalies inside the V2X message. The authors tested and trained their classifier using the VEINS simulator. The used classifier is a non-optimal MLP.

Wang et al., [Wan+19] used unsupervised machine learning to detect VANET anomalies. The classifier is a deep autoencoder. The goal is to identify the abnormal position in the V2X message based on the vehicle location and the RSSI.

Kaja et al., [Kaj19] used the WCVP dataset. Thus, their classifier analyses only BSM fields. Also, the authors created their attacks by modifying the dataset.

Kosmanos et al., [Kos+18] used supervised learning models (KNN and RF). Features include the variation of relative speed (VRS). Radar measures the relative speed between the jammer and the receiver.

Current works do not consider several type of ITS-Station. Therefore, the anomalies related to a vehicle mobility may be be different from the one related to a pedestrian mobility. Besides, perception data does not include only mobility data but also classification data. Thus, we propose a method which fill these gaps [Mon+18a] (Chapter 6).

Year	Authors	Reference	Comm. Layer					ITS Station Type			Train.		ML algorithm	Eval.		Threat	
			Appl. (A)	State (S)	Class. (C)	Net. (N)	PHY. (P)	4 Wheels	2 Wheels	Human	Simul	DataSet		Simul.	Experi.	DoS	Tamper.
2007	Raya et al.,	[Ray+07]	×	S	×	×	×	✓	×	×	✓	×	K-means	✓	×	×	✓
2010	Tian et al.,	[Tia+10]	×	×	×	N	×	✓	×	×	✓	×	ANN	✓	×	✓	×
2011	Grover et al.,	[Gro+11] [GLG11]	×	S	×	N	P	✓	×	×	✓	×	RF, NB, IBK, J-48, A-Boost	✓	×	✓	✓
2013	Dutta et al.,	[DC13]	×	S	×	×	×	✓	×	×	✓	×	Fuzzy STS	✓	×	✓	✓
2014	Sedjelmaci et al.,	[SS14]	×	×	×	N	P	✓	×	×	✓	×	SVM	✓	×	✓	×
2015	Maglaras et al.,	[Mag15]	×	×	×	N	×	✓	×	×	✓	×	SVM, K-means	✓	×	✓	×
	Li et al.,	[LJF15]	×	S	×	N	×	✓	×	×	✓	×	SVM	✓	×	✓	✓
	Alheeti et al.,	[AGM15a]	×	S	×	N	×	✓	×	×	✓	×	ANN	✓	×	✓	×
		[AGM15b]	×	×	×	N	×						FFNN				
		[AGM15c]											FFNN, SVM				
	Sedjelmaci et al.,	[SS15]	×	×	×	N	P	✓	×	×	✓	×	SVM	✓	×	✓	✓
2016	Alheeti et al.,	[AM16]	×	S	×	N	×	✓	×	×	✓	×	KNN	✓	×	✓	×
	Wahab et al.,	[Wah+16]	×	×	×	N	×	✓	×	×	✓	×	SVM	✓	×	✓	×
	Berlin et al.,	[Ber+16]	?	?	?	?	?	✓	×	×	×	×	?	×	×	?	?
	Fan et al.,	[Fan+16]	×	×	×	N	×	✓	×	×	✓	×	SVM	×	×	✓	×
2017	Kim et al.,	[Kim+17]	×	×	×	N	×	✓	×	×	×	6	SVM	✓	×	✓	×
	Ghaleb et al.,	[Gha+17]	×	S	×	×	×	✓	×	×	×	3	ANN	✓	×	×	✓
	Gu et al.,	[Gu+17b]	×	S	×	×	×	✓	×	×	✓	×	SVM	✓	×	×	✓
		[Gu+17a]											KNN				
	Alheeti et al.,	[AGM17]	×	S	×	N	×	✓	×	×	✓	×	LDA, QDA	✓	×	✓	×
	Sharanya et al.,	[SK17]	×	S	×	N	P	✓	×	×	×	×	SVM-MFM	✓	×	×	✓
2018	Monteuuis et al.,	[Mon+18a]	×	×	C	×	×	✓	✓	✓	×	5	MLP, RF, A-Boost	✓	×	×	✓
	So et al.,	[SSP18]	×	S	×	×	×	✓	×	×	×	2	KNN, SVM	✓	×	×	✓
	Amirat et al.,	[Ami+18]	×	×	×	N	×	✓	×	×	✓	×	Fuzzy C-means	✓	×	✓	×
	Subba et al.,	[SBK18]	×	×	×	N	P	✓	×	×	✓	×	ANN	✓	×	✓	×
	Sharma et al.,	[SPL18]	×	S	×	×	×	✓	×	×	×	1	PCA	✓	×	×	✓
	Singh et al.,	[Sin+18]	A	×	×	×	×	✓	×	×	✓	×	MLP, LSTM	✓	×	✓	×
	Eziama et al.,	[Ezi+18]	×	S	?	?	?	✓	×	×	×	×	BNN	×	×	×	✓
	Zeng et al.,	[Zen+18]	×	×	×	N	×	✓	×	×	✓	×	SVM, ANN	✓	×	✓	×
2019	Kamel et al.,	[Kam+19]	×	S	×	×	×	✓	×	×	✓	×	MLP	✓	×	×	✓
	So et al.,	[SPS19]	×	S	×	×	P	✓	×	×	×	2	KNN, SVM	✓	×	×	✓
	Gyawali et al.,	[GQ19]	×	S	×	×	P	✓	×	×	×	2	LR, KNN, DT, Bag, RF	✓	×	×	✓
	Singh et al.,	[Sin+19]	×	S	×	×	×	✓	×	×	×	2	LR,SVM	✓	×	×	✓
	Wang et al.,	[Wan+19]	×	S	×	×	P	✓	×	×	✓	×	DAE, SVM	✓	×	×	✓
	Kaja	[Kaj19]	×	S	×	×	×	✓	×	×	×	1	CC, DBKM, FC, KM, FF	✓	×	×	✓
	Zeng	[Zen+19]	×	×	×	N	×	✓	×	×	✓	×	2*CNN+LSTM	✓	×	✓	×
	Kosmanos	[Kos+18]	A	×	×	×	P	✓	×	×	✓	×	KNN, RF	✓	×	✓	×

TABLE 2.10: Machine learning based Attacks Detection for a CAV

2.4.2 Datasets

In the context of CAV, we did not succeed to find a dataset which covers both failures due to faults and security attacks. However, we found the following datasets that can be used to dealt with V2X failures classification:

1. NGSIM [Tra18] has recorded vehicle movements on various roadways in the United States.
2. Safety Pilot Model Deployment (SPMD) [USDa] dataset has more than 5.6 TB of recorded Basic Safety Messages (BSM)
3. The Wyoming Connected Vehicle Pilot (WCVP) [USDb] dataset contains V2X data collected during an experiment that runs on US I-80.
4. VeReMi [HLK18] is a simulated dataset that contains several types of misbehavior attacks.

Then, in Table 2.11, we classified each dataset based on the several characteristics: covered safety message fields, the presence of failure causes, physical signal data, meta data, and the CAV context.

We observed the following:

- datasets are mostly unrealistic (simulated data)
- datasets are mostly incomplete (missing safety message fields or absence of attacks).
- datasets focus on vehicles and do not include others types of C-ITS stations (pedestrian, road infrastructures stations, and motorbike)

Therefore, in chapter 6, we propose a new dataset [Mon+18a] considering C-ITS stations dimensions and type (for more details, see Chapter 6).

Data Elements	Description	Dataset				
		1 [Tra18]	2 [USDa]	3 [USDb]	4 [HLK18]	6 [Mon+18a]
Standard Elements						
DSRCmsgID	Identifier for message type	x	✓	✓	✓	x
SecMark	Timestamp	✓	✓	✓	✓	x
MsgCount	Message Number for a sequence	x	✓	✓	✓	x
TemporaryID	Network ID	x	✓	x	x	x
Latitude	Position along the latitude axis	x	✓	✓	✓	x
Longitude	Position along the longitude axis	x	✓	✓	✓	x
Elevation	Elevation relative to the sea level	x	✓	✓	✓	x
Speed	Object speed	✓	✓	✓	✓	x
Heading	Angle between object head and North	x	✓	✓	x	x
Yaw Rate	Heading per second	x	✓	x	x	x
Lat. Accel	Acceleration along the latitude axis	x	✓	x	x	x
Long. Accel	Acceleration along the longitude axis	x	✓	x	x	x
Vet. Accel	Acceleration along the vertical axis	x	✓	x	x	x
Positional Accuracy	Accuracy at one standard deviation	x	✓	x	x	x
Brake System Status	Status of the Brake System	x	✓	✓	x	x
Vehicle Length	Vehicle Length	✓	x	x	x	✓
Vehicle Width	Vehicle Width	✓	x	x	x	✓
Physical Signal Data						
RSSI	Received Signal Strength Indication	x	x	x	✓	x
Meta Data						
Sender ID	Emitter Identifier	x	✓	✓	✓	x
Gentime	Time of message creation	x	✓	✓	✓	x
Receiver Data						
Receiver ID	Receiver Identifier	x	✓	x	✓	x
Receiver Position	Position along the X-Y-Z axis	x	x	x	✓	x
Cause of Perception Failures						
Attacks	Presence of attacks	x	x	x	✓	✓
CAV Type						
#Type	Diversity of CAV types	x	x	x	x	✓

Our contribution

TABLE 2.11: Datasets for CAV Perception Failures

2.4.3 Classifier evaluation

The following sections cover the evaluation tool and metrics used to evaluate the performance of machine learning algorithms used to detect CAV failures.

2.4.3.1 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classifier on a set of test data for which the actual labels are known. In a binary classification problem (Table 2.12), it is crucial to define which labels are a positive case and a negative case. For instance, Table 2.12 defines "normal" data as a positive case and "attack" is related to a negative case.

Actual \ Predicted	Normal	Abnormal
Normal	TP	FN
Abnormal	FP	TN

TABLE 2.12: Confusion Matrix for Binary Classification

- True Positive (TP): The classifier predicts the data label as *Normal*, which is the actual (ground truth) label of the data.
- False Positive (FP): The classifier predicts the data label as *Normal*. However, the actual label is *Abnormal*.
- True Negative (TN): The classifier predicts the data label as *Abnormal*, which is the actual (ground truth) label of the data.
- False Negative (FN): The classifier predicts the data label as *Abnormal*. However, the actual label is *Normal*.

A confusion matrix is a visualization tool providing an overview of the classifier performance for a given data set. However, the confusion matrix does not measure classifier performances. Therefore, several metrics measure different parameters related to classifier performances.

2.4.3.2 Metrics

Generally, the metrics computed from the confusion matrix are recall, precision, and accuracy.

$$Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP}, \quad Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.1}$$

The *recall* defines the number of message containing *Normal* data that are classified as *Normal* divided by the total number of *Normal* or *Abnormal* data that actually are *Normal*.

The *precision* defines the number of *Normal* data classified as *Normal* divided by the total number of messages which contain data classified as *Normal* that actually are *Normal* or *Abnormal*.

The *accuracy* defines sum of messages correctly classified over the entire dataset.

Besides, other research fields studied used other performance metrics [Hag+18]. Therefore, we will compare and analyse the differences between the regular metrics and these other.

2.5 Cross-validated perception

Section 2.5 presents the related work of cross-validation perception to detect perception attacks.

An attacker may create a V2X ghost vehicle which copy the mobility of a real vehicle. In the absence of exhaustive attack dataset, a classifier may label this attack as normal. An approach proposed to alleviate this drawback is to use vehicle sensor to confirm the existence of V2X object. is the use of vehicle sensors to detect the physical existence of a V2X object. Table 2.13 surveys the related work and positions

our work. As seen, only few works use data cross-validation between perception sources to detect anomalies.

Yan et al., used radar measurements to detect Sybil Attacks [YOW08]. Indeed, the lack of physical object presence hints that the V2X object does not exist. Their scheme depends on radars. In this scheme, the radars acts as the system eye. if the radar can see the node then the V2X object exists, otherwise it does not exist. Thus, the proposed scheme verifies the physical existence of a connected vehicle in radar LoS. However, their work ignores the highly dynamic and harsh road environment. For instance, radar echoes increases the false alarm rate. Also, the current scheme classify occluded object as an attack. Thus, the proposed does not handle NLoS scenarios.

Obst et al., compared the similarity between a V2X and a camera detection to detect a ghost Attack [OHR14]. However, the paper does not consider multiple objects scenarios which require to associate measurements to their respective object. Thus, the paper assumes both measurements are related to the same object.

Zacharias et al., measured the local traffic density using its local sensors to detect Sybil Attacks [ZF18]. However, the lack of simulation results questions their solution feasibility.

Our work [Mon+19] uses V2X and a camera detection to detect a ghost attack. Our scheme handles multiple objects scenarios thanks to a multiple objects tracker. Besides, our scheme includes evidence from a surrounding RSU to tackle objects which are in camera NLoS. Finally, our scheme uses subjective logic to measure the certainty in our security scheme decisions.

Work	**Observer Type**		**Data Alignment**		**Multiple Objects Tracker**	**Performance**	
	Target	Sensor	Time	Spatial		Scalability	Real-Time
Yan et al., [YOW08]	✓	Radar	✘	✘	✘	✘	✘
Obst et al., [OHR14]	CAM	Camera	Buffer	polar	✘	1 vehicle	✓
Zacharias et al., [ZF18]	CAM	Camera	✘	✘	✘	✘	✘
Monteuuis et al., [Mon+19]	CAM	Camera	✘	cartesian	✓	N vehicles	✘

Our contribution

TABLE 2.13: Sensor-V2X Correlation for perception anomalies

2.6 Simulators for autonomous and cooperative perception

Lastly, Section 2.6 overviews the existing CAV simulators. First, we give the main modules found in a simulator. Then, we survey the existing simulators to verify if they support each module. Lastly, we explain the reasons behind choosing Matlab as our thesis simulator.

2.6.1 Main Simulation modules

The simulation platform must include several modules to evaluate the failure-resiliency of a cooperative and autonomous vehicular perception system.

2.6.1.1 V2X Communication module

A communication module simulates features related to V2X communication. The first goal is to model protocols used in vehicular communication. The second feature is to model the behavior of a wireless communication given a driving context (e.g., packet drop, path loss, and signal fading).

2.6.1.2 Traffic Mobility module

A traffic module simulates the mobility of each ITS-Station (e.g., pedestrian, CAV, bike) for a given driving context. For instance, The mobility model depends on the type of the ITS Station (e.g., speed based on the station dimension) and also on the driving context (e.g., highway)

2.6.1.3 ADAS module

The ADAS module simulates each component defining an ADAS such as sensors, controllers, and actuators with their specifications. For instance, the module can include sensor models such as a radar or a lidar. For the generalization sake, the ADAS module must provide models which mimic the behavior of each component. The ADAS module must provide algorithms library to associate and process the data provided and needed by each component. For instance, these algorithms include state predictors, classifiers, and data fusion algorithms.

2.6.1.4 Security module

The security module must provide a library of security mechanism defined in the research literature or standards at various level of the V2X protocol stacks. For instance, the security module can provide a model for each standardized cryptography operation such as key generation, digital signature computation, and digital signature verification. The security library aims to provide an implementation of existing security mechanism or an imitation of the expected behavior (e.g., the computation time of a digital signature) depending on a list of parameters (e.g., message size).

2.6.2 Existing Simulators

As shown in Table 2.14, we briefly review the main simulators in the following sections.

		VANETsim	VEINS	iTETRIS	Prescan	SiVIC-RTMaps	CARLA	Matlab
Modules	V2X Communication	~	✓	✓	✓	✓	✗	✗
	Traffic Mobility	✓	✓	✓	✗	✓	✓	✓
	ADAS	✗	✗	✗	✓	✓	✓	✓
	Security	✓	✓	✗	✗	✗	✗	✗
	Update	✗	✓	✓	✓	✓	✓	✓

TABLE 2.14: Comparison of automotive simulators

2.6.2.1 VANETSim

VANETSim [Tom+14] (VANET Simulator) is a VANET simulator developed in java. The communication module focuses on the application layer, such as beacon message (e.g., CAM) and event message (e.g., DENM). Therefore, the lower protocol layers are excluded from the simulation.

As a result, the simulator does not take into account the risk of packet drop due to signal collision or attenuation. The module related to traffic mobility, VANETSim uses a microscopic model [Kra98] of traffic flow where each vehicle makes its own decisions based on the simulated traffic context and personal observation. Unfortunately, VanetSim has no ADAS module. However, the simulator includes a security module with privacy schemes. Lastly, VanetSim is no longer maintained.

2.6.2.2 VEINS

Veins [SGD11] (Vehicles in network simulation) is an open-source simulation framework. The V2X communication module integrates several standardized V2X protocol stacks and the network simulator OMNeT++ [Var10]. The Traffic mobility module uses SUMO as its traffic simulator. Unfortunately, VEINS has not an ADAS module which prevents to simulate sensor measurements [Van18]. However, VEINS provides a security module with an exhaustive library of security mechanisms related to misbehavior detection [Van18] or privacy preservation [Van18].

2.6.2.3 iTETRIS

iTETRIS [Ron+13] (An Integrated Wireless and Traffic Platform for Real-Time Road Traffic Management Solutions) is an open-source simulation framework. The V2X communication module uses NS-3 [RH10] as its network simulator and is compliant with the protocols stack defined by the ETSI standard [Jem+17]. The Traffic mobility module uses SUMO as its traffic simulator. Unfortunately, iTETRIS does not provide an ADAS module and a security module.

2.6.2.4 PreScan

Prescan [19c] is a proprietary simulator for Advanced Driver Assistance Systems for driving systems. The V2X communication module integrates statistical V2X communication models within its network simulator and several standardized protocols stacks. Also, Prescan uses the traffic simulator named Vissim [Fel94], which provides microscopic traffic simulation. The ADAS module proposes several sensors models. Also, the ADAS module proposes an interface which enables users to design and verify algorithms for data processing, sensor fusion, decision making, and control implemented in Matlab/Simulink [19a]. Unfortunately, Prescan has not a Security module.

2.6.2.5 SiVIC-RTMaps

The SiVIC-RTMaps is a simulation framework [Gru+06] for ADAS Evaluation. The framework regroups two proprietary simulation platforms named Pro-SiVIC and RTMAPS (Real-Time Mines Automotive Prototyping System). Pro-SiVIC is a virtual prototyping platform that enables 3D simulations of physically realistic environments and sensors. Whereas RTMAPS is a software platform capable of recording, replaying, managing, and processing multiple data flows in real-time. For now, the framework does not include a V2X module. However, The framework includes a traffic module which allows the definition of the driving object (e.g., vehicle or pedestrian) trajectory. Also, the framework has its own ADAS module, which includes sensors models and algorithms to process sensor acquisitions. Finally, the framework does not include any security module despite some work in the context of V2X Privacy [Lef+13].

2.6.2.6 CARLA

CARLA [Dos+17] (Car Learning to Act) is an open-source simulator for autonomous driving research. The simulator does not include a V2X communication module. Also, CARLA integrates its traffic simulator for both vehicle and pedestrians. Besides, CARLA integrates an ADAS module which includes several sensors models and tools to conceive perception, planning, and control systems. However, CARLA does include any security module.

2.6.2.7 Matlab

Matlab [19a] is a multi-paradigm numerical computing environment and proprietary programming language. Recently, Matlab provides libraries to simulate, conceive, and evaluate system related to an ADAS context. Matlab does not include a V2X module. However, some work proposes an open-source V2X module based on Matlab [Wan19]. The traffic module uses basic traffic modeling based on the object trajectory. Indeed, Matlab provides driving scenarios or tools to set the object trajectory (e.g., position and speed). Thus, in our work, we choose to use Matlab as our simulator.

2.6.3 Thesis Simulator

As shown in Table 2.14, the number of simulators to evaluate cooperative and autonomous system is low. As far as we know, there is none open-source simulator that permits to evaluate simultaneously cooperative and autonomous aspects. Therefore, the development of such a platform which must provide security and safety mechanism is one of the main challenge encountered during the thesis. However, the thesis goal was not to develop such a platform but the definition of mechanisms or methods towards a generic cooperative perception architecture which is failure-resilient. Regarding its high number of users and its ties with other simulators, we decide to use Matlab for our research perspectives in the context of ADAS.

2.7 Synthesis

In this chapter, we presented state of the art related to the thesis context.

Section 2.1 presents the CAV context. Firstly, we remind the CV and its relationship with the C-ITS domain. Secondly, we presented the AV context and detailed the perception system of the CAV. Secondly, Section 2.2 defines and presents failures in CAV Perception. Thirdly, Section 2.3 presents the related work to assess the risk of security perception failures. Fourthly, Section 2.4 presents the concepts and related work of machine learning classifiers in the context of CAV Perception. Fifthly, Section 2.5 presents the related work of cross-validated perception for V2X attacks in the context of CAV. Lastly, Section 2.6 overviews the CAV simulators.

From this chapter, we made several observations:

- observed CAV perception architectures have specificities but also similarities (e.g., type of sensors, perception algorithms). Therefore, current failure resilient modules may be designed for a specific CAV architecture. Consequently, our contribution, described in Chapter 3, is the definition of generic architecture (**GPA**) and its failure resilient perception algorithm (**FRPA**). Thus, this failure resilient algorithm fits to any CAV architectures.

- A second observation is the lack of methodology which assesses the risk of failures in the context of CAV. Indeed, current methods focus on perception faults without considering the presence of attacks or focus on V2X and sensors attacks without considering the surrounding environment and the driverless context. Therefore, we propose in chapter 4, a security risk assessment method for CAV architecture. This contribution aims to integrate the safety expert assignment on safety metrics for safety-related attack goals. Besides, we propose in chapter 5, an attacker model adapted to CAV which surveys the cause of failures.

- A third observation is the lack of security modules to detect attacks originating from incoming V2X communication or local sensors in CAV perception. Thus, we propose two security mechanisms to detect malicious perception data. In Chapter 6, we propose a machine learning to detect sensor failures and prevent a perception failures. Secondly, we propose a mechanism that uses local sensor to detect V2X anomalies.

Chapter 3

A generic perception architecture

In this chapter, our contribution is three folds. First, we propose a generic perception architecture for CAV named GPA. Secondly, we propose its associated perception data model, named PDM. Finally, we define a Failures Resilient Pseudo Algorithm, named FRPA.

3.1 Generic Perception Architecture (GPA)

In this section, we present our generic perception architecture. We follow a systemic approach to consider various perception architectures as well as different kinds of CAVs. Generally, a CAV interacts with road users and infrastructures, as depicted in Figure 3.1. The latter are divided into C-ITS Stations and the environment externally surrounding the CAV. These interactions are possible thanks to CAV components. For instance, the CAV interacts with the external environments, thanks to sensors. Nevertheless, also, a CAV can use its OBU to communicate with C-ITS station. Then, the CAV will react according to each interaction. At First, he detects (e.g., the presence of a road user) and perceives (e.g., the user shape) the surrounding environment. After analyzing the road scene (e.g., collision distance), the CAV decides an action or inaction that may affect its behavior externally (brake lights) and internally (oil consumption). Currently, the connections and the data between each CAV components are specific to each car manufacturer. Therefore, if we want to propose a generic failure resilient perception system, then we need to define a generic perception architecture. Thus, our contribution is threefold:

- a physical architecture model, which aims to define with a systemic approach any physical CAV architecture.
- a logical architecture model highlighting the main functional modules of a perception architecture.
- a model of perception lifecycle abstracting the perception flows and processes.

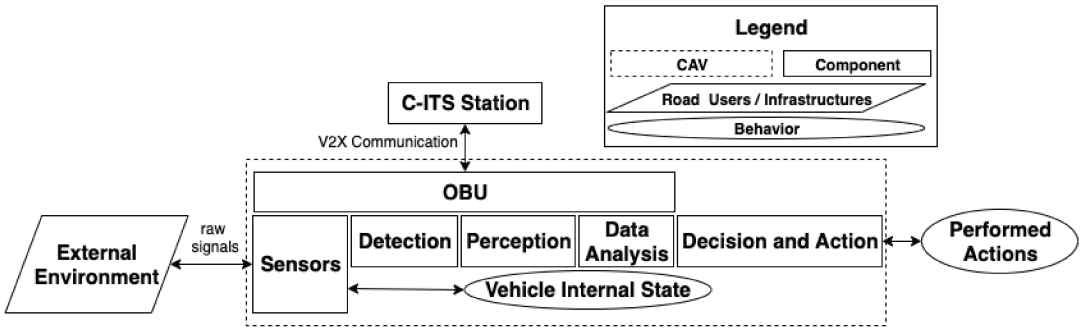


FIGURE 3.1: An architecture of CAV

3.1.1 Physical Architecture Model [Mon+18c]

Our physical vehicle architecture (Figures 3.2) is derived from the state of the art disclosed architectures. Our considered architecture is composed of *Electronic Controller Unit* (ECU), sensors, and actuators connected through several field buses (CAN, FlexRay, Ethernet...).

Each ECU achieves an automotive function (i.e., powertrain, infotainment, body, chassis, safety, communication, ADAS...) by collecting and processing data from various sources. For instance, sensors (e.g., camera, lidar, and radar) sense vehicle internals and its environment to detect mechanical problems, road lines, and traffic signs. ECUs process sensors information using data fusion and tracking techniques to extract advanced data features (e.g., obstacle class, speed value, localization). Then, the ADAS controller processes the perceived data into a real-world data model. The latter relies on V2X data collected by the On-Board Unit and on the driver inputs via the Infotainment Controller. Once we establish the environment model, the ADAS controller improves vehicle driving by ensuring functions such as *Automatic Emergency Braking*, *Automatic Parking*, and *Lane Keeping Assist System*.

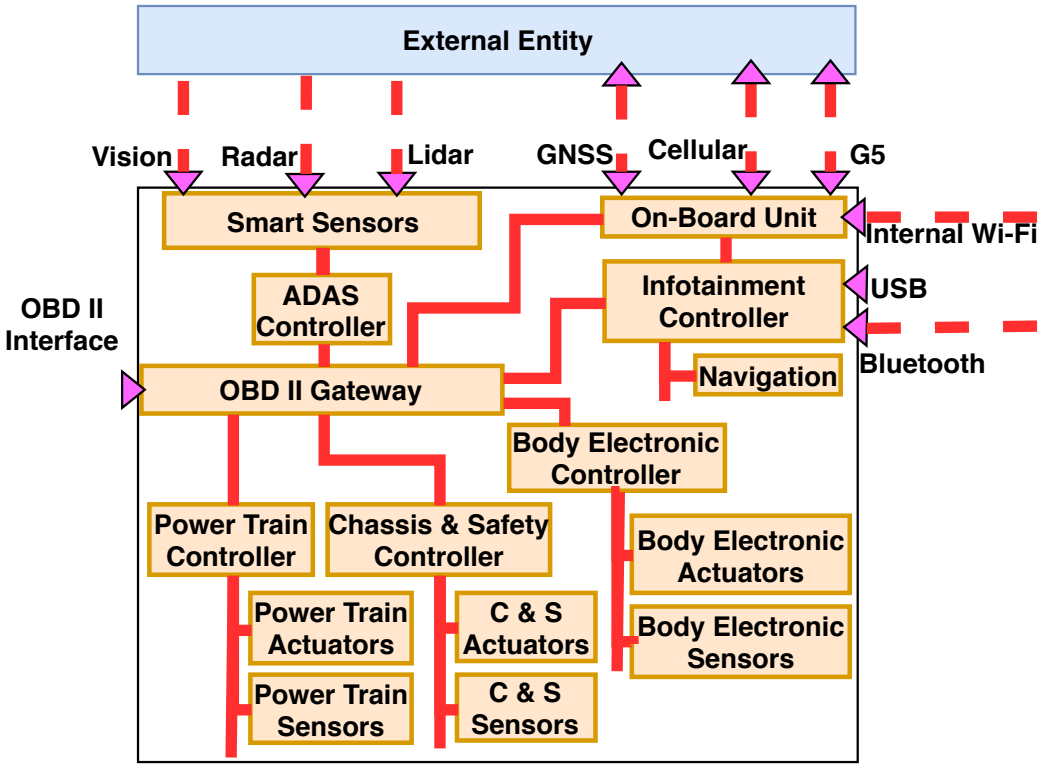


FIGURE 3.2: A Physical Architecture Model for CAV

3.1.1.1 Vehicle Assets categories

The components above are critical for vehicle control. They are vehicle assets and, therefore, targets of malicious road users. We divide assets into three categories.

The first category is named **Equipements**, which groups controller, sensors, and actuators (with their installed software and stored data).

The second category is named **External Entities** groups entities interacting with our architecture (e.g., other vehicles, road infrastructures, pedestrians...).

The third category, named **Data Flow** are flows between **External Entities** and **Equipements**. This category regroups in-vehicle communication (e.g., CAN bus, automotive Ethernet), V2X Communications, and sensors acquisition.

3.1.1.2 Interfaces

Besides, each equipment has one or multiple Interfaces such as OBD II interface, USB, Bluetooth, Cellular, G5, GNSS, internal Wi-Fi, and ranged sensors receiver/transmitter (e.g., lidar, radar, camera). Interfaces lead to a various number of surface attacks that can occur on the vehicle.

3.1.2 Logical Architecture Model

Our logical vehicle architecture (Figure 3.3) is based on the state of the art disclosed architectures. This section models the perception data and flows in our GPA. Figure 3.3 depicts the logical architecture of a CAV. This architecture has several features. The first feature is the architecture modularity. Indeed, it is independent of any algorithms and technologies and used in CAV Perception. For instance, we can choose the number and the type of sensors. Also, the interconnection between modules are interchangeable. For instance, we can choose to fuse or not V2X data with sensor data. The second feature is the integration of V2X communication. In this architecture, we consider the following specificities of V2X communication. Firstly, V2X share mutual data with sensors. Thus, V2X Data can serve as the input of the exterioperception modules. The last feature is the integration of security modules for CAV Perception. For instance, our architecture considers the cryptography module, which authenticates and verifies the integrity of V2X data.

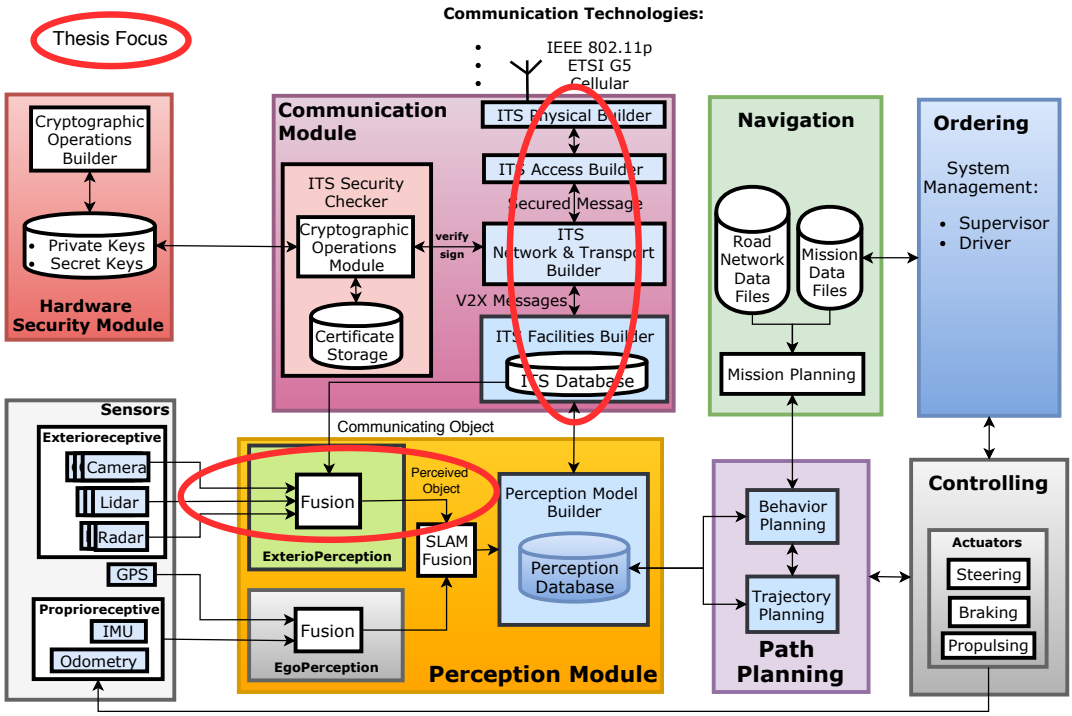


FIGURE 3.3: A logical Architecture Model for CAV

3.1.3 Perception Lifecycle Model [Mon+18b]

Figure 3.4 outlines the perception lifecycle which has two main components. The first one is *Objects* which regroups:

- the perceiver of the perception system named *ego-vehicle*,
- the perceived entities named *Road Object*.

The second component is *Data Stages* which are the stages followed by the data through the perception lifecycle defined as follows:

- *Data Acquisition* is the transition of the physical signal (e.g., light intensity, radio wave, pulsed laser light, sound waves) between a detected road object and the ego-vehicle and its acquisition processes. It includes communication signals for V2X and measurement signals for ranging sensors. The acquisition processes include message encoding/decoding, security modules (e.g., cryptographic verification) [17], object detection (e.g., Doppler Shift [SLG17]), and object classification (e.g., dots and pixels clustering).
- *Data Processing* regroups the data fusion modules applied to the acquired data such as association and/or tracking [BP99]. Their localization and their implementation within the Perception Lifecycle model vary among OEMs [Yua+17; Aeb17; Raw+17].
- *Data Storage* contains the data stored temporarily (e.g., tracks) or permanently (e.g., algorithms). Indeed, these data are a keystone in ensuring the monitoring (e.g., tracks) or the operation of the perception system (e.g., association algorithm).
- *Phenotype Data* is the observable traits of a *Road Object*, such as its morphology (e.g., dimensions), physiological properties (e.g., color), behavior (object state over time), and behavior actions (e.g., human-made tags).

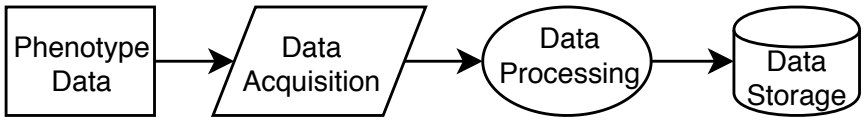


FIGURE 3.4: Perception Lifecycle Model

Thank to our model, we can define a perception lifecycle, as depicted in Figure 3.5, that works independently of any communication protocols, sensors, or data fusion algorithms. Such abstraction exhibits the primary assets of a perception system to derive our attacker model.

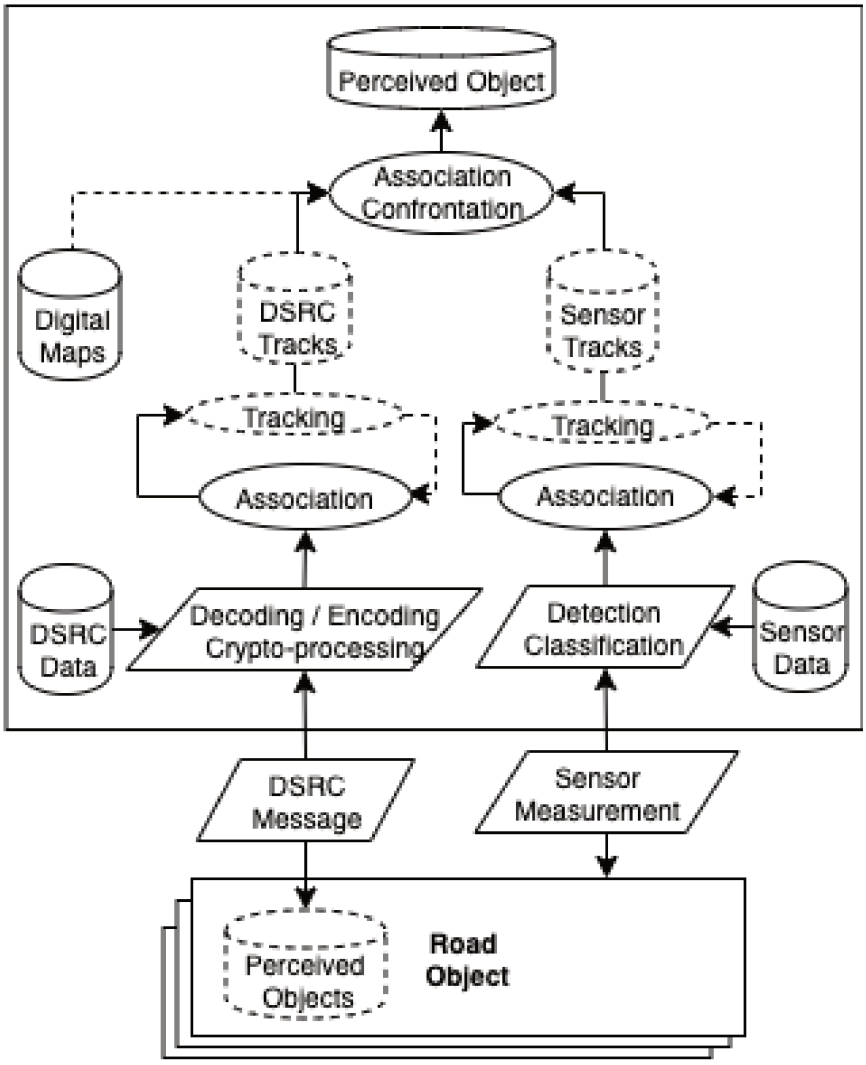


FIGURE 3.5: Example of a Perception LifeCycle

3.2 Perception Data Model (PDM)

In this section, we present our perception data model named PDM. The DMCAVP contribution is twofold:

The first part is the Source Data Model which aims to model the mutual and non-mutual data provided by each perception source.

The second part is the Data Model which defines the different data layers during the perception process.

3.2.1 Perception Object Data Model

We define a data model for objects perceived by a CAV. Such a model aims to identify data provided by a unique source of perception and data provided by multiple sources. This goal will allow us, for instance, to define anomaly detectors based on multiple sources. Our model has five features five features: **Identification**, **State**, **Classification**, **Time**, and **Security**.

The **Identification** feature includes data related to the identity and the authenticity of an object. For instance, the station identifier of a C-ITS Station confirms the object identity and authenticity.

The **State** feature includes data related to the object mobility such as its position.

The object type and its dimension are part of the **Classification** feature.

The **State** feature regroups all temporal data.

Finally, the **Security** feature includes all data related to cyber security.

Category	Sub-Category	Source	
		Sensors	V2X
Identification	Source Identifier	×	✓
	Object Identifier	✓	✓
	Signal Strength	✓	✓
	Existence	✓	✓
State	Position	✓	✓
	Heading	✓	✓
	Speed	✓	✓
	Acceleration	✓	✓
Classification	Dimension	✓	✓
	Type	✓	✓
Time	Object Creation	✓	✓
	Message reception	×	✓
	Validity Duration	✓	✓
Security	Digital Certificate	×	✓
	Digital Signature	×	✓
	Global Trust	×	✓

TABLE 3.1: Data Model for CAV Perception

3.2.2 Multi-Data Layers Perception Model

This section presents our Multi-Data Layers Perception Model (Figure 3.6) which has six layers:

3.2.2.1 Layer 0: Phenotypic Object

A phenotypic object is defined by all information visible by the human eye. For instance, such an object can be defined by its car matriculation (identification data) or color (dimension data).

3.2.2.2 Layer 1: Raw Object

Raw sensor measurements define a raw object. Depending on the sensor type, raw objects can be defined by pixels (camera) or physical signals suc as light beams or radio waves. For instance, the received signal strength indication of a radio wave can provide the position of a road object.

Raw object resolution: The resolution of a raw object depends on the detection performance of the sensor. For instance, some sensors may measure only the position of an object. However, some sensors can compute other information such as the speed or its existence (multiple object tracking). Thus, the amount of data detected by a sensor defines the resolution of a raw object.

Raw objects Density: A sensor resolution defines the number of times a sensor can detect an object in a single measurement. For instance, a high-resolution sensor may assign multiple detections to a single object. In this case, a raw object has a high density of detections. In a dense traffic scenario (multiple pedestrians), multiple raw objects with high detection density may affect the detection performance of the sensor. On the contrary, in the absence of crowds, raw objects with high detection density may help to determine object dimensions and type, as seen in the next section.

3.2.2.3 Layer 2: Sensed Object

A sensed object is defined by all information processed and extracted from a raw object. For instance, a cluster of pixels can define an object type (car) and dimensions.

Sensed object density: In general, the number of sensed objects is lower than the number of raw objects. Indeed, the clustering of several laser beams into a cluster centroid reduce the computation latency of the sensor in crowded environments.

Sensed object resolution: A sensed object has higher detection resolution than a raw object. For instance, clustering several raw objects provides more information such as the object shape or dimensions. Besides, the tracking of a sensed object can provides new information such as an identifier (track id) for a specific duration.

3.2.2.4 Layer 3: Communicated Object

Communicated objects are defined by data contained in a safety message. For instance, data may be the internal state of CAV (action intention, driver state, vehicle health). In the thesis, a communicated object contains all the information in Table 3.1.

Communicated object density: The number of sensed objects is lower or equal to the number of phenotypic objects. Indeed, all phenotypic objects do not communicate at the same time or at all (not connected vehicles and roadsigns).

Communicated object resolution: The resolution of a sensed object depends on the safety message content. For instance, security algorithms allow to verify the authenticity and the integrity of a secured safety message.

3.2.2.5 Layer 4: Fusion Object

Fusion objects regrouped objects detected and processed from sensors and V2X communication. This layer aims to align each detection (spatial transformation) structurally.

Fusion object density: The number of fusion object equals to the sum of mutual and non-mutual objects detected from sensors and V2X communication.

Fusion object resolution: The resolution of a fusion object is the same as the one from its originating layer (Layer 1 or Layer 2) and according to its originating sensors noise (Radar, Lidar, emitter GNSS), or Camera).

3.2.2.6 Layer 5: Perceived Object

In this layer, detections from sources referring to a single phenotypic object are fused into a single detection.

Perceived object density: The number of perceived objects equals to the sum of fused and non-fused objects between sources.

Perceived object resolution: The data contained in the perceived object are identical to the data contained in the fusion object. Regarding data accuracy, the fusion

process depends on the type of fusion architecture. In Low and Feature fusion, the perception system uses the fused object, which is the most similar to the predicted mobility data of a perceived object. Thus, the perceived object is at least as accurate as of the most similar fused object. In high-level fusion, the perception system exploits the strength of each sensor to output a perceived object. Thus, a perceived object will rely on the fused object of a radar for the data related to velocity. However, the perceived object will use classification data from V2X communication due to its high accuracy.

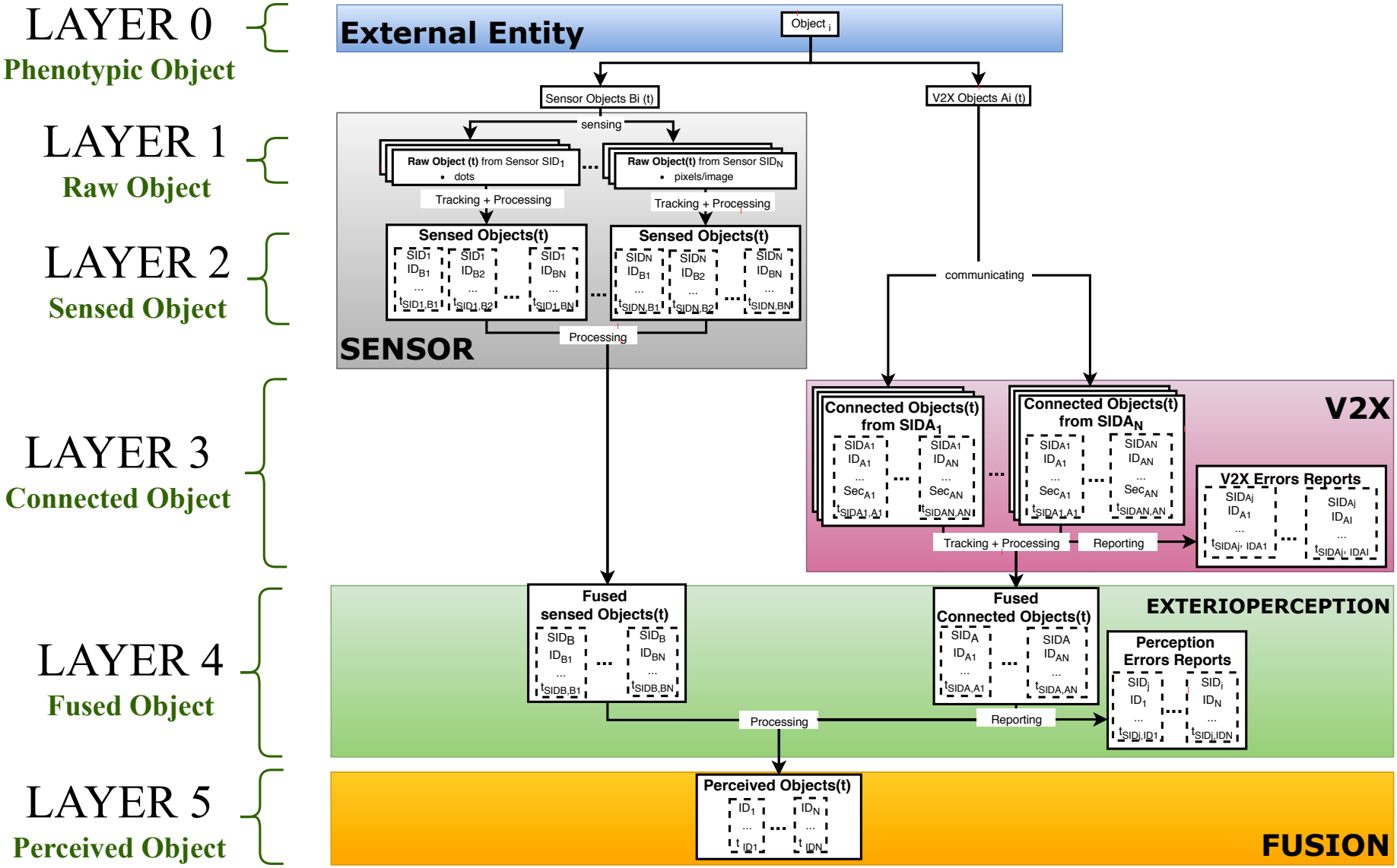


FIGURE 3.6: Multi-Data Layers Perception Model

3.3 Failures Resilient Perception Algorithm (FRPA)

This section presents our Failures Resilient Perception Algorithm (**FRPA**) depicted in Figures 3.7). Our FRPA design is compliant with our generic perception architecture by including V2X messages, sensor acquisition, and perception algorithms (tracking and fusion).

Besides, based on our Perception Data Model, our FRPA has six modules that detect perception anomalies. Each module is compliant with security standards (e.g., verification of digital signature). Accordingly, modules are divided into two groups named *Self-Resilient* and *Multi-Sources Resilient*. Each modules group is described in the following sections.

3.3.1 Self-Resilient modules

Self-Resilient modules do not rely on other perception sources to detect perception anomalies, which are faults or attacks.

3.3.1.1 Cryptography module

According to the ETSI Stack, cryptography modules are the first performed self-resilient modules. Besides, cryptography modules are standardized and mandatory security modules implemented in CAVs. As identified in our DMP, cryptography modules are specific to V2X communication. Examples of cryptography modules are digital signature verification or message encryption, as seen in our work [Mon+17].

3.3.1.2 Facility module

Facility modules verify the plausibility of the data contained in a V2X message or measured from a sensor at the time of object detection. Facility modules use methods such as static threshold or ML (Section 2.4).

3.3.1.3 Physical module

Physical modules measure features from the received physical signal, such as the Angle of Arrival or Time of Arrival. These modules aim to compare the measured features issued from V2X signal and computed features contained in a safety message [SLG17; Han+17].

3.3.1.4 Temporal module

Temporal modules verify the behavior of a source (object or CAV sensor) over time to detect anomalies. Such methods exist for perception sensors to remove errors (e.g., radar echoes). However, such methods are not fit for V2X attacks and to CAV perception yet. Thus, our FRPA fills the gaps.

3.3.2 Multi-Sources Resilient modules

Multi-Sources modules exploit measurements from other sources in the perception system to detect anomalies or converges toward an accurate perceived object.

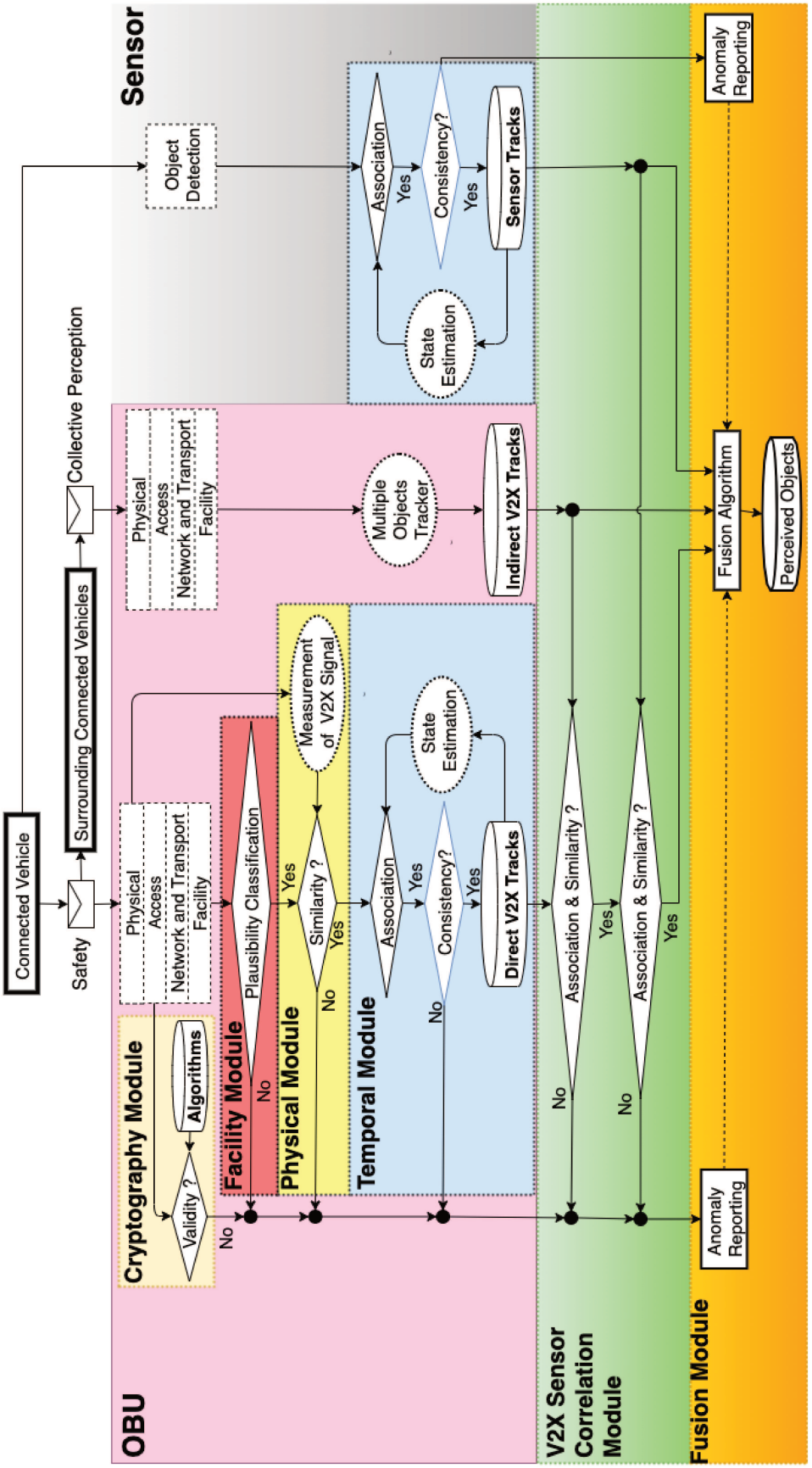


FIGURE 3.7: Failures Resilient Perception Algorithm (FRPA)

3.3.2.1 Cooperative-Sensor module

According to our DMP, cooperative-Sensor modules detect anomalies among mutual data contained in an object detected by several perception sources (Section2.5). However, Cooperative-Sensor modules require a trusted source. Therefore, FRPA ensures the existence of a trusted source with self-resilient modules per source.

3.3.2.2 Opinion Fusion module

Each presented module has a confidence value regarding its output (e.g., malicious data). An opinion Fusion module merges the confidence value of each module into a single confidence value.

3.4 Conclusion

In this chapter, we presented three contributions. The first contribution is the definition of *Generic Perception Architecture* (**GPA**) composed of a *Physical Architecture Model* and a *Logical Architecture Model* which are modular and generic to define any physical and logical architecture of a CAV vehicle. The associated Perception Lifecycle Model, which models the main flows and the processes of the Perception Lifecycle is also described

The second contribution is the definition of a *Perception Data Model* (**PDM**). The first part is the Data Model per Source, which models the data from each perception source into several categories. The second part is the Multi-Data Layers Perception which defines the processes and data structures for an object perception.

The last contribution is the design of a *Failure Resilient Perception Algorithm* (FRPA). The latter includes security modules to prevent perception failures in the CAV.

Our **GPA** offers a systemic and generic definition of the target of evaluation for automotive threat analysis and risk assessment (Chapter 4). Besides, the extensive analysis of our **GPA** leads us to the definition of a new attacker model for CAV (Chapter 5).

Also, our PDM helped us to identify data that are specific and common to all perception sources. The result of these findings lead to the definition of two failure resilient modules (Chapters 6 and 7).

In the next chapter (Chapter 4), we will present a new method for threat analysis and risk assessment adapted for the CAV.

Chapter 4

Security automotive risk analysis method (SARA)

As mentioned in Chapter 2, current threat analysis and risk assessment methods do not consider CAV architectures.

Therefore, this chapter aims to fill this gap by proposing a new threat analysis and risk assessment method named SARA. Our contribution is three folds. Firstly, Section 4.1 presents the SARA framework. Secondly, Section 4.2 describes our new threat analysis method. Section 4.3 presents SARA's risk assessment method and its application to two use cases. Section 4.4 describes SARA's countermeasure method. Finally, Section 4.5 concludes the chapter.

4.1 SARA method [Mon+18c]

SARA method is organized into four blocks (Figure 4.1):

- **Feature definition** describes the defense perimeter[1] of the assessed system. The system definition follows two architectures. The *physical* architecture represents interfaces, controllers, sensors, actuators, and communication links. The *logical* architecture represents the data flows issued by aforementioned physical entities. Indeed, a CAV relies on data flows to observe its surrounding environment and control the vehicle dynamics [MSN17]. By knowing the threaten data flows, the expert forecasts the severity of attacks on assets, the capabilities of self-observation, and self-controllability of the CAV.

- **Threat specification** describes SARA *threat to security goal* map, *attack method to asset* map, and SARA *attacker list* definition. The SARA *threat to security goal* map associates our threat model (**STRIDELC**) to our security goal model (**AINCAAUT**)[2]. Then, SARA *attack method to asset* maps a set of assets categories and threats/security goals to an attack method. The latter is a single threat or a set of threats performed by an attacker on an asset. SARA *attackers list* maps an attacker profile and its *attacker capability* score. The latter is the sum of the standardized metrics values (*expertise*, *knowledge*, and *equipment*) required as a minimum to perform an attack[3]. The *attacker capability* serves to compute the *attack likelihood* in the next building block.

- **Risk assessment** returns the risk value of an attack. SARA attack tree defines the attack goal as the tree root and selects its related threats from those identified in the previous threat specification step. Then, we define the *attacker* as the

[1] Defense perimeter is defined in Target of Evaluation from Common Criteria [Cri17]
[2] STRIDELC and AINCAAUT are detailed later in section 4.2.1.
[3] Attacker capability metrics are detailed later in section 4.2.3.1

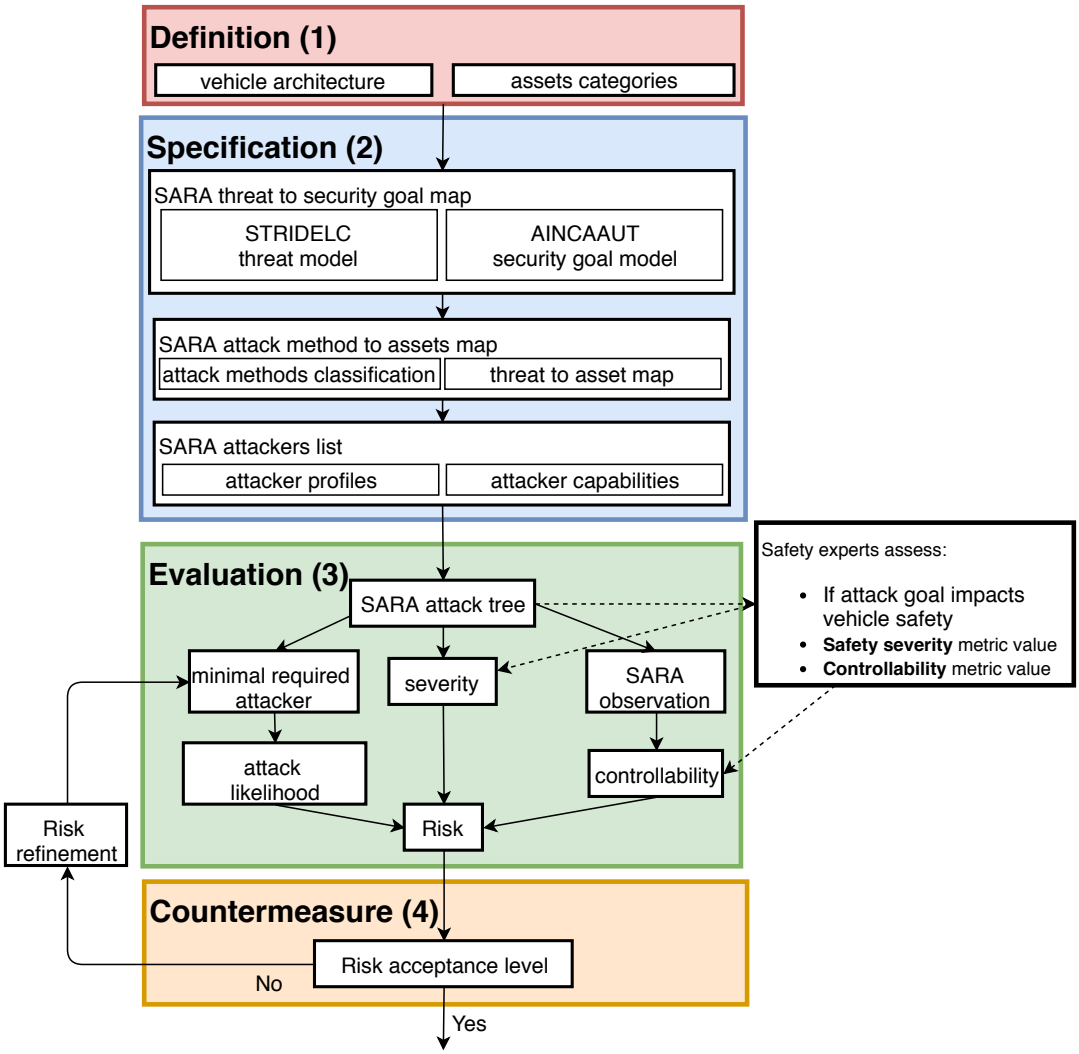


FIGURE 4.1: SARA framework

minimally required profile to perform a threat using SARA *attacker list*. Therefore, we compute the attack likelihood of a threat. Then, security and safety experts define attacks goal severity, observation and control values. Finally, experts compute the risk value of an attack goal from the following metrics: severity, observation, controllability and the highest attack likelihood. Section 4.3 details the risk computation using SARA attack tree.

- **Countermeasures** minimize the computed risk from an attack tree. The applied countermeasures refine the risk level or end the risk assessment process. Indeed, risk analysis is an iterative process that ends once countermeasures have been applied to critical threats until the risk value converges to an acceptable level.

4.2 SARA Threat Specification

Risk assessment requires security experts to define the evaluated vehicle architecture and its features. The detail level of the system description reflects the architecture

maturity and affects risk assessment results. In this section, we consider the vehicle architecture described in Chapter 3.

We identify assets and their related threats using our systematic threats specification. Our method follows three steps: SARA threat/security goal mapping, SARA attack method/asset mapping and SARA attackers list definition.

4.2.1 *Threat to security goal* mapping

To identify considered threats, we define a new threat model named **STRIDELC** (Table 4.1). The latter extends STRIDE by adding two categories that are **Linkability** and **Confusion**. The latter refers to the processing of authentic data structure with incorrect content that does not reflect the ground truth state. For instance, a traffic light emits authentic V2X messages with incorrect traffic light states. Therefore, The incorrect state confuses the ADS which must rely on another reliable data source.

Confusion differentiates from threats such as *Spoofing*, *Tampering*, and *Elevation of Privilege*. Indeed, a source sending authentic messages with incorrect content neither usurps another source identity nor alters a data structure [PS15]. **Confusion** related security goal is **Trustworthy** which includes countermeasures assessing the trustworthiness of the content and/or its source [Bis+12].

Linkability refers to the ability to link pseudonymous or anonymous data to identify the data owner. **Linkability** differentiates from *Confidentiality* as follows. For instance, malicious observers collect vehicle signed cooperative awareness messages (CAMs) on a predefined road path. By tracking vehicle localization contained in the messages, the attacker extract private data such as preferred driving path [Pet+15a], housing localization, children localization, health status (e.g., hospital, gym, fast-food), and its customers localization. After data processing, the data allow to map to confidential information such as vehicle owner identity using its house localization (despite anonymous/pseudonymous message [Wie+10]). The related security goal *Unlinkability* includes countermeasures providing dynamic confidentiality such as pseudonym certificate change scheme [PS15] or obscuring proxies [Why+13].

For the remaining threat/security goal associations, we refer to Microsoft SDL *STRIDE to security goals* map [Her+06]. Also, we prioritize *Confidentiality*, *Integrity*, *Availability*, and *Trustworthy* for their strong impact on ensuring system safety and as a standard procedure (CIA model). That is, Table 4.1 depicts the considered threat model and security goal model in SARA.

STRIDELC Threats Categories	Explanation	**AINCAAUT** Security Goals Categories
Spoofing	impersonate someone or something else	**A**uthenticity
Tampering	to modify data or functions	**I**ntegrity (*)
Repudiation	cannot traced back the author actions	**N**on-Repudiation
Information Disclosure	to access to confidential data	**C**onfidentiality (Privacy)(*)
Denial of Service	interrupt a system legitimate operation	**A**vailability (*)
Elevation of Privilege	perform unauthorized actions	**A**uthorization
Linkability [PFK14]	deduce the owner identity from owner public unidentified data	**U**nlinkability (Privacy) [Why+13]
Confusion [PFK14]	a data source confuses the system by sending incorrect data within authentic data structure	**T**rustworthy (*) [PS15]

(*) identifies prioritized goal, (■) (Contribution Thesis [Mon+17])

TABLE 4.1: SARA Threat-Security goals

4.2.2 *Attack method to asset* mapping

Once the considered threats and security goal defined, we map them to system assets.

4.2.2.1 Mapping threats to asset categories

To this end, we revisit *Microsoft STRIDE-per-Element* map [Her+06]. First, we associate defined assets categories with elements. As mentioned in section 3.1.1.1, *Equipment* stores data and processes it using the software. Therefore, we map this asset category to *Data Process* and *Data Store* elements. The remaining associations between asset categories and elements are straightforward. Then, we match our asset categories to our **STRIDELC** threat model. As defined in Table 4.1, only data emitters such as *Equipment* and *External Entity* produce authentic messages with incorrect content. Therefore, we map *Confusion* threat to *Equipment* and *External Entity* asset categories. *Linkability* threat targets data related to our system of definition. This includes both *Data Store* and *Data Flow* which are, if not confidential, public or semi-public information (e.g., logs) potentially used to gather confidential information as mentioned. Finally, Table 4.2 maps STRIDELC threats to our defined assets categories.

Element [Her+06]	Assets (Section 3.1.1.1)	STRIDELC threats (Table 4.1)							
		S	T	R	I	D	E	L	C
External Entity	External Entity	✓	✘	✓	✘	✘	✘	✘	✓
Data Process	Equipment	✓	✓	✓	✓	✓	✓	✓	✓
Data Store									
Data Flow	Data Flow	✘	✓	✘	✓	✓	✘	✓	✘

TABLE 4.2: STRIDELC-per-asset categories map

4.2.2.2 Defining attack methods classes

Once the threats-assets map defined, we define the *attack methods classes*. An attack method groups one or multiple threat categories. For instance, an attack *greedy jamming vehicular communication channel* includes an *Elevation of Privilege* threat and a *Denial of Service* threat. To define attack methods, we use *CIA* model and TVRA *Threat Tree* [ETSa]. As a result, we define four attacks method classes as follows:

- **A***lter* attacks aim to modify data which relate to *Tampering* threats/*Integrity* security goals
- **L***isten* attacks aim to monitor data which relate to *Information Disclosure* threats/*Confidentiality* security goals
- **D***isable* attacks aim to deny access to data which relate to *Denial of Service* threats/*Availability* security goals
- **F***orge* attacks aim to create incorrect data which relate to *Confusing* threats/*Trustworthy* security goals

4.2.2.3 Mapping attack method classes to asset categories

Once the attack method classes defined, we map the major threat of each attack method (e.g., Tampering) to our *STRIDELC-per-asset categories map* (Table 4.2). As a result, we obtain the *SARA attack method per asset map* (Table 4.3). This map allows a systematic tool to map multiple threats/security objectives to assets which can be useful to build attack tree, Petri-nets or graphs. Also, non-security experts can use *SARA attack method per asset map* to avoid security goal omission/misidentification. For instance, in[Ste+16], the author identifies a spoofing threat from a malicious diagnostic tester as an *Authorization* security goal whereas it is an *Authenticity* security goal.

Attack Method Classes	Alter	Listen	Disable	Forge
Prioritized Security Goals (Table 4.1)	**Integrity** Non-Repudiation Authorization	**Confidentiality** Non-Repudiation Authorization Unlinkability	**Availability** Non-Repudiation Authorization	Authenticity Non-Repudiation Authorization **Trustworthy**
Asset Categories (Section 3.1.1.1)	Data Flow / Equipment	Data Flow / Equipment	Data Flow / Equipment	Ext. Entity / Equipment

TABLE 4.3: Mapping SARA attacks method classes to assets categories

4.2.3 Attackers list

This section defines SARA attackers list and provides an analysis of its advatanges.

4.2.3.1 Attackers profiles definition

We define an attacker as the combination of an attacker profile and an attacker capability. To define the attacker profile, we refer to previous methods [Dom+16; PFK14] and the attacker model defined in [PS15].

Table 4.4 depicts a list of attackers (A) and their corresponding capabilities (Ca_A). Ca_A depends on multiple standardized factors [08; 09]:

SARA	ISO metrics [08; 09]			SARA
Attacker Profiles (A)	Expertise (Ex)	Knowledge (K)	Equipment (Eq)	Attacker Capabilities Ca_A
Thief, Mr.Nobody	Layman (0)	Public (0)	Standard (0)	0
Researchers	Experts (8)	Public (0)	Specialized (4)	12
Evil Mechanic	Expert (6)	Restricted (3)	Specialized (4)	13
Organized Crime	Proficient (2)	Sensitive (10)	Specialized (4)	16
Hacktivist	Experts (8)	Sensitive (4)	Mult-bespoke (9)	21
Foreign Government	Experts (8)	Critical (11)	Mult-bespoke (9)	28

TABLE 4.4: ISO metrics to Attackers Capabilities

- **Knowledge factor** (K) refers to the attacker knowledge regarding the chosen system. K can be public, restricted, sensitive and critical.
- **Expertise factor** (Ex) refers to an attacker expertise. It specifies four attacker categories. A layman is a person without specific security knowledge. A proficient is a person with basic security knowledge. An expert has a strong security culture learned from his past hacks and attended conferences. Multiple experts are a group of experts united around a common attack goal. Multiple experts launch simultaneous attacks to achieve their attack goal.
- **Equipment factor** (Eq) refers to the equipment needed by an attacker to perform an attack. There are 4 types of equipment. A standard equipment is an equipment already available for the attacker. A specialized equipment needs to be ordered from a specialized shop. A bespoke equipment is not easy to purchase and is expensive to create. Some attacks require multiple bespoke pieces of equipment which are hardly available and very expensive.

Security expert sums factor values (C_j) to compute the capability Ca_A of an attacker A as follows:

$$Ca_A = \sum_{j \in \{K,Ex,Eq\}} C_j \tag{4.1}$$

4.2.3.2 Attacker profile analysis

This approach differs from previous work in two ways. First, an attacker profile defines the attacker capability whereas previous work defined attacker capability based on threats. Second, we optimize previous approaches by reducing the total decision time and the total number of choice combinations. The latter is the number of possible combinations proposed to the expert to evaluate the attacker capability. Our method proposes 7 possible combinations (7 attacker profiles) to evaluate the attacker capability whereas the standard offers 48 combinations. Therefore, our method reduces the decision time. Indeed, if we assume the same time of evaluation per metric(t), an expert needs only a single t to evaluate the attacker capability instead of three t.

Metric $j \in \{K, Ex, Eq, A\}$		K	Ex	Eq	A
Available choices per metric	n_j	4	4	4	7
Setting time per metric	t_j	t	t	t	t
Total of choices combination	$\prod_j n_j$	64			7
Total setting time	$\sum_j t_j$	$3 \times t$			t

TABLE 4.5: Decision-Gain regarding Choices and Time

That is, SARA attacker profiles optimize the computation of attacker capability.

4.3 Risk Assessment

This section presents SARA risk assessment. We first define SARA attack tree and its metrics used for risk computation. Then, we apply our risk assessment method to two use cases.

4.3.1 SARA attack tree

An attack tree defines threats used by attackers to reach an attacking goal (Figure 4.2). Attackers reach their goal through attacked automotive functions. *Attacked functions* simplify targeted components identification within the CAV and clarify the attack description for automotive experts. Then, SARA *attack method* (Section4.2.2) maps an attack method to the impacted assets using SARA *attack method to asset* map (Table 4.3). Finally, we associate a minimally required *attacker* (Table 4.4) to the *attack on an asset* which maps one or multiple threats to the impacted asset(Table 4.3). Experts compute the attack goal risk score based on the highest attack likelihood score among all attacked assets.

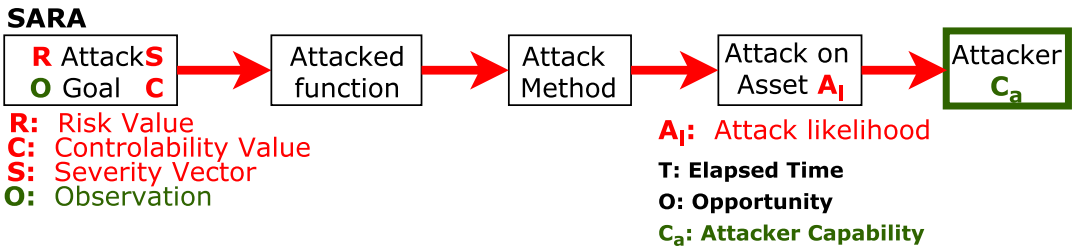


FIGURE 4.2: SARA attack tree

4.3.2 Attacker profile and attack likelihood

SARA attacker profiles help experts to identify the minimally required attackers regarding an attack goal. As mentioned, an attack is composed of one or multiples threats which are performed by attackers. Therefore, as mentioned, the success of an attack depends on the attacker capability but also on the elapsed time (T) and on the required opportunity (WO) to perform the attack [Hen+09]. The time factor is the time needed to identify and successfully realize an attack considering the attacker capability. The opportunity tells if an attack requires a special window of opportunity to be executed or it can be easily executed.

TABLE 4.6: Standardized Mapping Attack Likelihood [ETSa; Int16; Hen+09]

AP_A	Description	AI
$[0, 9]$	Basic	5
$[10, 13]$	Enhanced basic	4
$[14, 19]$	Moderate	3
$[20, 24]$	High	2
> 24	Beyond high	1

Experts compute attack potential (AP_A) using the values of *attacker capability* Ca_A, normalized *elapsed time* T and *opportunity* metrics WO as follows:

$$AP_A = Ca_A + T + WO \tag{4.2}$$

That is, attacks requiring the lowest minimal attack potential are more likely to occur (Table 4.6). As a result of improving attacker capability, we reduce the total metrics for attack potential from initially 5 to 3.

4.3.3 Attack goal severity

Standardized severity factors are safety (S_s), privacy (S_p), financial (S_f) and operational (S_o) [Int16]. SARA severity relies on the previous factors values and expert motivation for severity vector computation (Table 4.8):

$$S = (S_s, S_p, S_f, S_o) \tag{4.3}$$

We choose a maximization approach by assuming that all severity factors have equal importance. That is, SARA severity value is the highest severity vector coefficient. For instance, we consider as attack goal *an unauthorized braking from one CAV at low speed* with specific severity vector (e.g., $\tilde{S} = (1, 1, 0, 2)$). The maximized severity value is $S = S_o = 2$. Therefore, we reduce risk assessment time by avoiding the full risk vector computation. However, our approach still supports vector approach if threat risk must be evaluated for each severity factor separately. SARA severity considers attack goal scalability. For instance, if the aforementioned attack goal occurs in a traffic jam. An unauthorized brake has a strong impact on multiple CAV safety and their operational state. The severity of this situation ($S = 3$) is higher than in the single-CAV case ($S = 2$). That is, SARA Severity is flexible (e.g., maximization or vector approach), supports severity absence (e.g., $S = 0$) and is scalable (e.g., single or multiple attacks).

O	C	Meaning
1	0	ADS observation is available but no accident avoidance is required
	1	ADS observation is available and accident avoidance is required using ADS response
0	2	ADS observation is unavailable/uncertain, driver response is required
	3	ADS observation is unavailable/uncertain, driver response is impossible/unavailable

TABLE 4.7: Observation and controllability classification

S	Safety	Privacy	Financial	Operational
0	No injuries	undisclosed or unlinkable data	No loss	Intact vehicle performance
1	single light to moderate injury	one identified vehicle	$> \$100$	one small impact on a vehicle
2	single severe injury or multiples moderates injuries	one vehicle tracking or identification of multiple vehicles	$> \$1000$	one big impact or many small impacts
3	single life threatening injury or multiple severe injuries	multiple vehicles tracking	$> \$10000$	big impact on many vehicles

TABLE 4.8: SARA Severity

4.3.4 Attack goal observation and controllability

System control requires system internal and external observation to anticipate system failures caused by hazards or threats. The fully autonomous vehicle cannot rely on human perception. We tackle this issue with a new metric called *Observation* (O). The latter defines system tolerant default and its ability to detect errors and faults. Therefore, it controls system security risks. *Observation* has two values: perceptible ($O = 1$) and imperceptible ($O = 0$). Figure 4.3 illustrates the use of *Observation* in practice.

A mechanic attacker targets a sensor connected to a vehicle by altering a sensor calibration. The faulty sensor creates system error and probably a system failure. At initialization, expert considers threat observation as null. However, with appropriate countermeasures, the threat becomes perceptible which leads to risk reduction. The metric *Observation* advantages are considering vehicle safety without human control and forecasting vehicle architecture countermeasures to failures (Table 4.7). Architecture countermeasures control fault propagation and rely for example, on data redundancy, watchdog or IDS. Note that data redundancy increases vehicle cost.

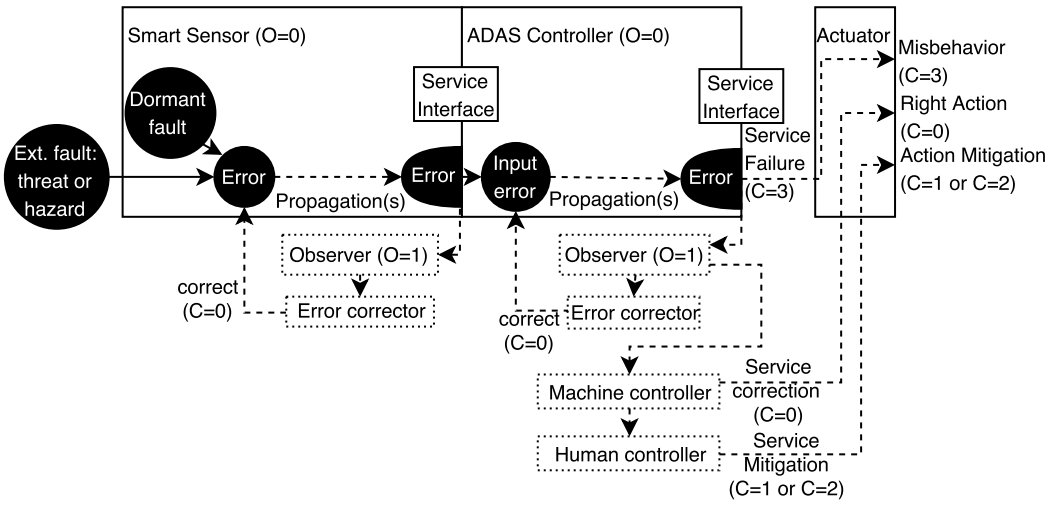


FIGURE 4.3: Concept of Observation and Controllability

Controllability (C) quantifies the autonomous system or driver influence on security risk [ISO11]. C ranges from 0 to 3: 3 refers to the absence of driver/ADS controllability over the vehicle, whereas 0 is the opposite (Table 4.7).

4.3.5 Risk computation

SARA computes the risk score using the SARA matrix function (f) defined in Table 4.9.

$$R = f(C, S, AI) \tag{4.4}$$

The risk score ranges from insignificant (R0) to unacceptable (R7+). Expert uses risk score to evaluate a threat and decide if countermeasures are needed. Besides considering machine controllability, our approach advantage is to rely on the same matrix for safety and none safety-related use cases. Also, it is similar to ASIL computation method which reduces the gap between security and safety.

C [4]	S [5]	AI [6]				
		1	2	3	4	5
0	1	R0	R0	R1	R2	R3
	2	R0	R1	R2	R3	R4
	3	R2	R3	R4	R5	R6
1	1	R0	R1	R2	R3	R4
	2	R1	R2	R3	R4	R5
	3	R3	R4	R5	R6	R7+
2	1	R1	R2	R3	R4	R5
	2	R2	R3	R4	R5	R6
	3	R4	R5	R6	R7	R7+
3	1	R2	R3	R4	R5	R6
	2	R3	R4	R5	R6	R7
	3	R5	R7	R7	R7+	R7+

TABLE 4.9: SARA Risk Matrix
(C: Controllability, S: Severity, AI: Attack Likelihood)

4.3.6 SARA risk assessment application

In this section, we assess the security risk of two use cases: the *Vehicle Tracking* [PFK14] and the *Comfortable Emergency Brake Failure* [Li+16].

4.3.7 Vehicle Tracking use case

A vehicle broadcasts periodically signed *cooperative awareness messages* (CAMs) in a defined area. The latter contains public anonymous/pseudonymous data related to the vehicle localization. However, despite anonymous data, an observer eavesdrops and tracks vehicles messages. Then, knowing vehicles positions history in the neighborhood, an observer maps the pseudonym certificate of the car owner to its house address and so, its identity. A global observer can track information on the supply chain of a company such as the supply chain path or the position and or

[4] refer to Table 4.7
[5] refer to Table 4.8
[6] refer to Table 4.6, if S=0, it means an absence of risk.

clients localization/identity. Indeed, The observer reconstructs the vehicle path history using the position data in its CAM. Also, the observer checks where the vehicle stopped, then maps the localization to potential customers address. Robbers could check the presence of police vehicles in the area visually or through the rights of their pseudonym certificate then tracking them [Pet+15a]. That is, the privacy impact is high. Figure 4.4 depicts an attack tree of the scenario based on SARA framework (Figure 4.1).

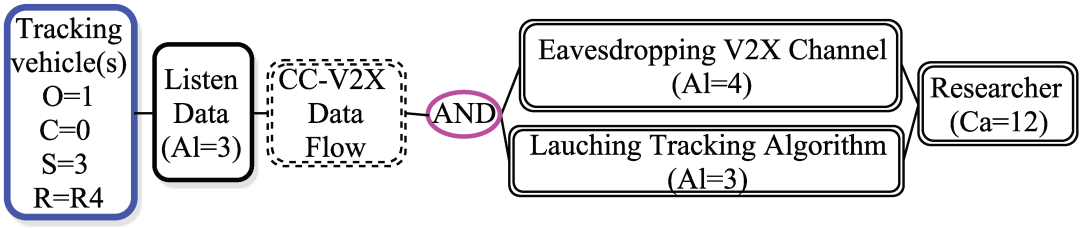


FIGURE 4.4: Attack tree for Vehicle Tracking use case

First, we initialize the attack goal settings (Figure 4.2) as follow.

- The severity factor concerns only privacy as the attack goal does not affect safety, financial or operational aspects of the system. Also, the attack allows the tracking of multiple vehicles. That is, the Severity factor value is $S_p = 3$ (Table 4.8).
- The attack does not impact the ADS observation operational state ($O = 1$, (Table 4.7).
- Therefore, the attack does not require any control over the vehicle dynamic ($C = 0$, Table 4.7).

Second, using the attack tree goal and SARA *attack method/asset map* (Table 4.3), we define the potential attack methods, threatened asset category, possible threats, and the minimum required attacker.

- the attack method class is *Listen* because the attack goal requires to monitor vehicular communication.
- the threaten asset category is *Data Flow* because the attack focus on the data elements contained in CAMs.
- the identified security goals are *Confidentiality*, *Unlinkability*, *Authorization*, and *Non-Repudiation*.
- the minimum required attacker is *Researchers* [Pet+15a].

Based on the identified security goals and the *SARA threat to security map* (Table 4.1), we identify the following threats. To achieve its attack goal, the attacker eavesdrops the communication channel to "read" the message content. *Confidentiality* is the issue due to disclosed message content. The second threat concerns *Unlinkability* because the attacker can extract private information from public data using tracking algorithms [Wie+10; Bis+12; Pet+15a] as aforementioned. In this case, threats against *Authorization* and *Non-Repudiation* are not part of the scope. Indeed, vehicles broadcast their messages that are accessible to everyone in the transmission range.

Third, we compute the attack likelihood value of *Eavesdropping on the V2X channel* and *Tracking CAMs* threats by assessing the following metrics (Equation 2).

- *attacker capability* value (Ca),
- *Elapsed Time* value (T), and
- *Window of Opportunity* value (WO).

As defined in [Pet+15a], the attacker profile is a *Researcher* which *attacker capability* value ($(Ca_{researcher})$) is 12 (Table 4.4). To perform *Eavesdropping on the V2X channel*, we assume an attacker requires less than a day. Therefore, referring to *Elapsed Time* map [Hen+09; ETSa], the *Elapsed Time* value is 4 ($T = 4$). *Tracking CAMs* requires 2 weeks [Pet+15a]. Therefore, the *Elapsed Time* value is 4 ($T = 4$). The attacker does not need a *Window of Opportunity* to reach his attack goal. Therefore, the *Window of Opportunity* value for both threats is null ($WO = 0$) based on [Hen+09; ETSa]). Then, using Equation 2, we compute *Eavesdropping on the V2X channel* and *Tracking CAMs* attack potential values (respectively, $AP = 0$ and $AP = 0$). Finally, using Table 4.6, we map their attack likelihood values (respectively, $AI = 4$ and $AI = 3$).

Fourth, we compute the attack likelihood value of the attack method class *Listen* using:

- each threat attack likelihood value, and
- if multiple threats, the logical operator definition.

Tracking CAMs is possible only if the attacker is *Eavesdropping on the V2X channel*. Therefore, the logical operator is *AND*. Using the threats attack likelihood value and the logical operator definition, we compute *Listen* attack likelihood value ($AI = 3$).

Fifth, we compute the security risk value on the attack goal using:

- the attack likelihood value of each attack method, and
- if multiple attack methods, the logical operator definition.
- Controllability and Severity values defined during the attack goal setting.

Listen is the only attack method for this attack goal. Using Equation 4 and the risk matrix (Table 4.9, we compute the risk value of the attack goal *Tracking vehicles* ($R = R4$).

Although not high, SARA risk value is pertinent regarding current work. As ongoing researches on the autonomous vehicle enhance tracking algorithms, the price decrease and the accessibility increase of tracking device will increase the spectrum of attackers and therefore the attack likelihood over time. Countering such attack remains difficult. Indeed, the removal of the identifier from V2X messages may increase the identification process and favors spoofing attacks. On the other hand, the removal of data elements from V2X messages threatens cooperative awareness applications that rely on both classification and location data from the lidar and the CAM to detect accurately pedestrian [MSN17]. Even though countermeasures such as *pseudonym change strategies* exist, their efficiency still need to be evaluated [Bis+12; Jae+12].

This analysis confirms two facts. First, the assessed risk score reflects the current situation regarding this attack. No satisfying solution has been proposed yet despite mutual efforts from standardization and industrials. Second, the need to revise or extend current threat models such as STRIDE. As we see, privacy is not just a matter of confidentiality or anonymity but of unlinkability of public data which may variate following the needs of cooperative awareness applications.

4.3.8 Comfortable Emergency Brake Failure use case

To assess the impact of a faulty traffic light on a driver-less vehicle, we assess the risk of an attacked automated and connected feature called *Comfortable Emergency Brake* feature (CEB) [Raw+17].

This feature uses the content of the Signal Phase and Timing (SPaT) and of the MAP messages emitted from a connected traffic light. Then, once the traffic light is in the camera line of sight, the automated driving system (ADS) collects, processes the output of the camera and the V2X messages. The ADS compares the state of the traffic light inside the SPaT message with the camera output for color matching. The camera output assess the correctness of the SPaT before braking.

Due to the lack of security mechanisms [Raw+17], we assess the security risk of this feature. A potential attack goal is *CEB fails to trigger braking at a red light*. We define the following settings:

- The severity factor concerns mainly safety. Therefore, the security experts need to discuss the safety impacts with the safety experts to assess the following metrics. We assume in this case the chosen severity metric value is 3($S_p = 3$, Table 4.8).
- The attack impacts the system observation ($O = 0$).
- There is no driver response with a driver-less vehicle ($C = 3$).

We assume the ADAS controller is the automotive function that processes the data and decides to brake. Therefore, the attacked automotive function is ADAS.

To reach their attack goal, attackers must send a color different than red to the vehicle driving system. Based on SARA *attack method classification*, three attack classes fulfill such conditions: *Alter*, *Disable*, *Forge*. Figure 4.6 depicts the potentially obtained attack tree.

For the sake of clarity, we will discuss on the most interesting threats:

First, we discuss about attacks disabling the optical flow are efficient. They require no effort from the attacker and hardly detectable by the system. Indeed, a delivery van standing in front a traffic light is hardly seen as an attack. Also, physical attacks on the road infrastructure are hardly detectable. Indeed, Figure 4.5a depicts a physically damaged traffic light. Despite targeting the infrastructure, this attack has an impact on the vehicle waiting for the red light to turn green. Moreover, this attack is easily scalable for an attacker which means the system cannot rely on other surrounding traffic lights. Even in a big city such as Paris, it took one month to report and repair the damaged infrastructure.

We assume most vehicular communications to be cryptographically secured as requested by standards [17]. Therefore, we do not focus on attacks altering the content of the SPAT message using MITM.

Next, we discuss attacks forging data. However, as mentioned, road infrastructures are easily accessible and can misbehave. Indeed, an altered traffic light [Li+16] can emit a SPAT with an incorrect red state instead of a ground truth green state. If an attacker physically damage the traffic light (Figure 4.5a) or blind the camera [PS15], the driving system can only rely on an incorrect SPaT. If the SPAT emits incorrect green state, the vehicle can cross the intersection without seeing unconnected objects coming from its left or its rights leading to a potential collision. This threats goal is to confuse the system [PFK14] and require appropriate countermeasure to allow the automated driving system to take safe decision.

(A) Faulty Traffic Light (B) Faulty Road Sign

FIGURE 4.5: Examples of Faulty Road Infrastructures

The present analysis shows that the threats impacting the vehicle are not always vehicle-oriented but infrastructure-oriented. Indeed, road infrastructures are not all connected and render cryptographic approaches useless (Figure 4.5b). Although some work attacked the camera using faulty road Sign [Evt+17] or faulty traffic light [Li+16], none of them assessed the security risk of a CAV.

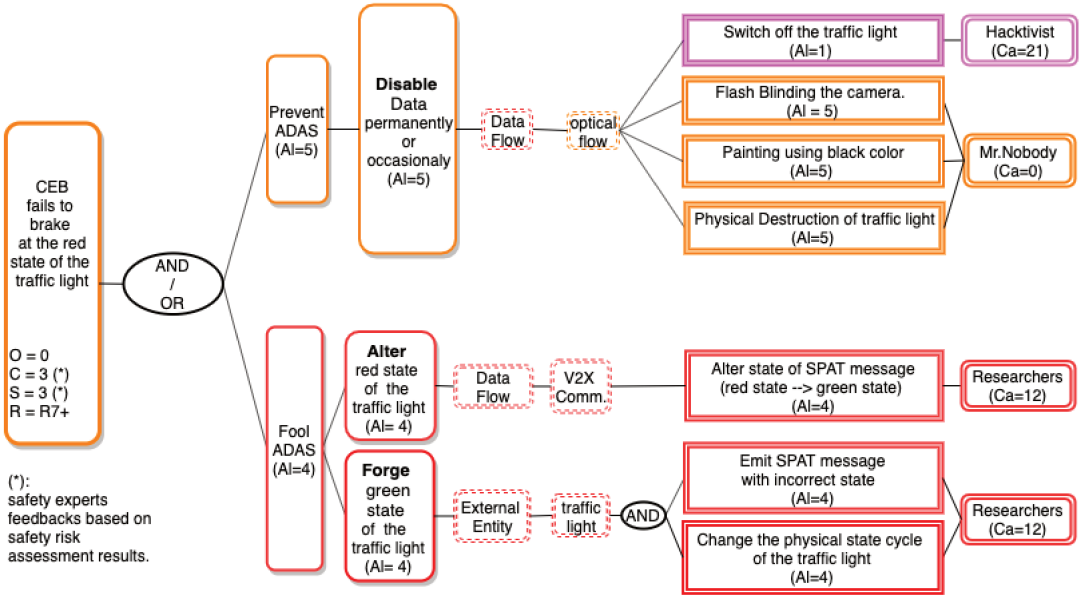


FIGURE 4.6: Attack Tree for Comfortable Emergency Brake Failure

4.4 Countermeasures

SARA final step is to apply countermeasures to reduce highest attack risk values. Then, we re-iterate SARA risk assessment application process until reaching an acceptable risk value. We initially reduce the reiteration process by setting an acceptable risk value for each attack goal. The setting of R and S values define the maximal accepted attack likelihood ($\mathsf{AI}_{\mathsf{wanted}}$) for all attacks on asset related to that attack goal. Finally, we apply countermeasures on attacks on asset until all their attack likelihood values (AI) verify:

$$\mathsf{AI} \leqslant \mathsf{AI}_{\mathsf{wanted}} \tag{4.5}$$

For instance, in the case of the *Comfortable Emergency Brake Failure* (Figure 4.6), we set a risk value of *R5* as a satisfying requirement without changing *Severity* and *Controllability* values. Then, we compute Al_{wanted} using *SARA Risk Matrix* (Table 4.9). The wanted attack likelihood value is 1. Therefore, we know that we need to assess all the threats with an attack likelihood value greater than 1. Doing so, we do not re-iterate SARA risk assessment application process and we know which threats require to be countered first.

4.5 Conclusion

In this chapter, we propose a new method for threat analysis and risk assessment named SARA. First, we present the method which introduces safety experts feedbacks in various security processes. Next, we highlight the need for methods for proper threat analysis coverage against human omissions to consider recent concerns regarding the trustworthiness and privacy of the driver-less vehicle. Also, we propose some improvements to existing standards. Finally, SARA proposes a new metric for attack observation for CAV controllability. Indeed, automated driving system-dedicated vehicles can be designed without having human interaction and, therefore, must be able to detect an attack to control and reduce risk value. To this end, we presented the potential risk of two uses cases named "a malicious observer" and "faulty road infrastructures on the vehicle." Finally, we refined the risk value after applying counter-measurements for each attack identified in each use case.

Currently, SARA attacker model focus on the V2X and in-vehicle domain. Therefore, we need to define an attacker model which includes attacker related to the domain of perception (e.g., attack on perception algorithms). Thus, in the next chapter, we propose a new attacker model with its security goal model for CAV.

Chapter 5

Attacker and Security Goal Models for Perception

As mentioned in Chapter 2, current attacker models focus on VANET or inside-vehicle. Thus, there is a gap for CAV, to combine both domains with Perception.

In this chapter, we propose a new **attacker model (AM)** and a new **Security Goal Model (SGM)**. In Section 5.1, we present our attacker model derived from the identified assets. In Section 5.2, we provide the corresponding security goals model. Finally, Section 5.3 concludes the chapter.

5.1 A New Attacker Model [Mon+18b]

First, we define a generic attacker model. Next, using the assets identified in the **PLM** (Section 3.1.3), we describe specific attacker models for the perception system. To do so, we assume that cryptographic mechanisms yield against cryptanalytic attacks (e.g., message forgery or side channel attacks).

5.1.1 Generic Attacker Model Definition

Firstly introduced in VANETs [RH07] then extended for automotive sensors [PS15], a general attacker model defines attacker actions and potential targets. However, previous works assume that the attacker always reaches its goal directly which is false. Indeed, a malicious node can badmouth to neighboring nodes to provoke its victim exclusion from the network [Tan+17]. Also, the alteration of road sign impacts the vehicle perception indirectly [Evt+17; Sit+18]. Thus, we propose a new generic attacker model with a five-dimensional set as follows:

- **Membership** stands for an *Insider* or an *Outsider* attacker. An *insider* attacker is an authenticated member of one or multiple CAV networks (e.g., CAN, LIN, V2X). Therefore, he can mount a diverse set of attacks using his given credentials. Whereas, an *outsider* is an unauthenticated member who can mount a limited set of attacks due to her restricted network access.

- **Motivation** stands for *Malicious* or *Rational*. A *malicious* attacker seeks no personal benefits from the attacks and aims to harm an asset. Whereas, a *rational* attacker seeks profit and thus is predictable regarding her attack means and target(s). Such attribute may help to define the financial severities of an attack in security risk analysis process [Int16]. For instance, a *rational* attacker will aim the perception algorithms contained in a CAV to sell them to hackers on the black market.

- **Scope** stands for *Local* or *Extended*. A *local* attacker controls few entities (e.g., car or traffic light [Ghe+14]) within a limited scope (e.g., road intersection). However, an *extended* attacker controls several entities scattered across an extended scope (e.g., university campus [Pet+15a]).

- **Method** stands for *Active* or *Passive*. While an *active* attacker must act to attack, a *passive* attacker simply listens or observes its target (e.g., network eavesdropping). For instance, in the context of standardized efforts towards the cooperation between safety and security risk analysis [Int16], a meteorological hazard could be a *passive* attacker.

- **Goal** stands for *Direct* or *Indirect*. A *direct* attacker reaches its primary target directly, whereas an *indirect* attacker reaches its primary target through secondary targets.

This attacker model has a different purpose than the one defined in SARA. In SARA, the attacker is defined by its equipment, vulnerabilities knowledge, and level of expertise. Differently, our attacker is defined by the context before the attack. For instance, an insider attacker implies that the attacker has some credentials prior launching the attack (e.g. private key with a digital certificate). Therefore, the context defines the attacks that an attacker can launch. Indeed, an attacker cannot emits malicious messages with fake position without credentials.

As depicted in Table 5.1, the attack goal helps to define attackers in the perception domain. However, *Goal* does not situate wherein the perception domain the attacker may perform an attack. Therefore, we need to specify *Goal* explicitly. To do so, we derive each sub-attacker model from the *Data Stages* (Table 5.2). We define these sub-attacker models and their attacker profiles in the following sections.

Attacks \ Attacker Model	Membership	Motivation	Scope	Method	Goal
Alter road signs to fool sensors	Outsider	Malicious	Local	Active	Indirect
Alter road signs for "fun"					Direct
Camera blinding towards unperceived stop sign	Outsider	Malicious	Local	Active	Indirect
Camera Blinding for "fun"					Direct
Communication Badmouthing	Insider	Malicious	Both	Active	Indirect
Faulty Safety Message			Local		Direct

TABLE 5.1: Examples of similar attacks with different goals

Data Stage	Sensor Disrupter	Evil Mechanic	Malicious Communicator	Fusion Persuader
Phenotype	✓	✘	✘	✓
Acquisition	✓	✘	✓	
Storage	✘	✓	✘	✓
Processing	✘	✓	✓	✓

TABLE 5.2: Sub-Attacker Models in the Perception Lifecycle

5.1.2 Sensor Disrupter

Sensor Disrupter is an attacker that aims at vehicle sensors. Indeed, CAV perception relies on the acquisitions of exteroceptive sensors (e.g., camera, lidar, or radar) to perceive the surrounding environment. Thus, sensors are assets which a *Sensor Disrupter* can disturb through various attack means.

5.1.2.1 Sensor Illusionist

Sensor Illusionists target sensors *directly* during the *acquisition* stage. During this stage, ranging sensors (e.g., lidar) provide a closely real-time, trusted, and more or less accurate depiction of the surrounding by measuring the reflected physical signal [MSN17]. However, at signal impact, an *Illusionist* can capture, delay, and replay it forcing the sensor to produce erroneous measurements. For instance, from the position of sensors target (the detected object), Petit et al. [Pet+15c] captured, delayed, and replayed the lidar signal. Also, they relayed the signal and replayed it from a different position leading the way towards signal forgery attacks.
Thus, *Illusionist* attack means include signal delay, relay, replay, and forgery.

5.1.2.2 Sensor Blinder

Similarly, *Sensor Blinders* target exteroceptive sensors *directly* during the *acquisition* stage. During this stage, *Blinders* can alter the physical signal trajectory transiting between ego-CAV sensors and its surrogating environment. For instance, a camera cannot detect a facing traffic light state due to the parked vehicle blocking the view. Or, *Blinders* can maximize or minimize signal intensity to emit signals outside the sensing domain of the sensor [Pet+15c; YXL16]. For instance, using fog light against an automotive camera is a realistic and accessible attack to perform.

5.1.2.3 Evil Sensor Calibrator

Evil Sensor Calibrators target exteroceptive sensors *directly* during the *storage* stage.*Evil Calibrators* aims to modify sensor settings to provoke incorrect/missing measurements. Indeed, range sensors measure the distance between the *Road Object* and itself. Then, the measurement system of the sensor computes the absolute position by moving from the local referential base of the sensor to a global referential base. However, *Evil Calibrators* can modify the local referential base by changing the physical position, orientation, or internal settings of the sensor. Such actions lead to an incorrect perception of the *Road Object*. For instance, taking the case of *Lenticular Printing* attack which is an optical process used to create road signs that look different when viewed from different angles [Sit+18]. Sitawarin et al. demonstrated that if the localization of the camera used for road signs recognition is at a different height from the human controller, then the camera classifier performances are diminished while appearing to be correctly positioned to the human operator. Thus, an *Evil Sensor Calibrator* can drastically modify the sensor orientation to provoke an absence of measurements. Rarely mentioned, *Evil Sensor Calibrator* attacks remain easy to perform physically and may extend to other in-vehicle hardware (e.g., *Evil Mechanic* attacks).

5.1.2.4 Ground Truth Falsifier

Ground Truth Falsifiers target exteroceptive sensors *indirectly* through *Road objects* at *phenotype* stage. *Falsifiers* physically alter *Road objects* (e.g., road signs) to provoke incorrect sensors measurement. For instance, *Falsifiers* can forge counterfeit road marks [Els17]. Therefore, an automotive camera can detect fake road marks as real ones which may influence vehicle trajectory. Also, the alteration of road signs known as ***D**eceiving **A**utonomous ca**R**s with **T**oxic **S**igns* (DARTS) leads to camera misclassification from the camera which may affect vehicle dynamic [Evt+17; Sit+18]. Thus, mentioned attacks are *indirect Illusionists* attacks.

Finally, the massive alteration of a *Road Object* can provoke an acquisition absence. Indeed, *Falsifiers* can destroy, remove, or severely deface road infrastructures. Therefore, mentioned attacks are *indirect Blinder* attacks.

5.1.3 Evil Mechanic

As depicted in Figure 3.4, the perception lifecycle takes place mostly within the *ego-CAV*. Each ECU performs an automotive function (e.g., powertrain, infotainment, body, chassis, safety) by collecting and *processing data* from various sources such as sensors and ECUs. Therefore, attacking *processing data* is valuable for an attacker willing to force the CAV into a wrong assessment or to extract valuable data (e.g., data fusion algorithms). Related attack sets are *In-vehicle Manipulator* and *In-vehicle Miner*.

5.1.3.1 In-vehicle Manipulator

This attacker aims to add, modify, or remove automotive components or data contained in it. Indeed, an attacker with elevated physical access (e.g., mechanic) could easily replace a smart camera by one with a dysfunctional detection algorithm. Although the camera is recording, its detection capabilities are abnormal which may catch off-guard the driver. Besides safety, the removal or injection (e.g., odometer manipulation [PFK14]) of vehicle history permits data repudiation. Therefore, a vehicle owner can repudiate facts in case of fraud insurance, resale, or crime investigation because the falsified vehicle history confirms her statement. Moreover, the intentional manipulation of tamper-resistant automotive equipment [PFK14; WG11] may activate defense mechanisms that erase all the data contained in such hardware which, thus, benefits to the attacker. Finally, a mechanic can flash equipment with a modified firmware to increase her attack range [MV16]. Therefore, a malware installation in this equipment allows the injection of CAN message with incorrect content without requiring the mechanic to remain plugged into the vehicle.

5.1.3.2 In-vehicle Miner

This attacker eavesdrops in-vehicle data for personal deeds. For instance, a *Miner* can sell the vehicle history to third parties (*rational attacker*). Indeed, robbers can use the sole localization history to identify the driver routine and rob her house. Moreover, eavesdropping *Storage* and *Processing* steps help to analyze the behavior of perception algorithms. Once reviewed, this information is valuable to *Sensor Disrupter*, *Malicious Communicator*, or *Fusion Persuader*.

5.1.4 Malicious Communicator

As introduced, V2X communications aim to improve vehicular automation reliability, safety, and traffic efficiency. Like in all social group, some participants behave against the interest of the community. Such behaviors threaten communication. We define such attacker as *Malicious Communicator* which regroups *Fully Adversarial Networking*, *Voyeur* and *Communication Deceiver*.

5.1.4.1 Fully Adversarial Networking

This attacker inserts arbitrary messages and performs selective Denial Of Service attack [PFK14].

5.1.4.2 Voyeur

This attacker surveys anonymous public data exchanged in cooperative ITS to obtain confidential data (e.g., car owner identity). For instance, in VANET, localization and trajectory of the vehicle are willingly broadcast. Indeed, cooperative awareness through communication requires a frequent update of surrounding vehicles localization. Therefore, it is mandatory to be able to track vehicles locally. However, *Voyeurs* can use tracking to track broadcasting vehicles in a neighborhood or a campus [Pet+15a]. By tracking vehicle localization contained in the messages, the attacker extracts private data such as preferred driving path, house localization, children localization, or health status (e.g., hospital, gym, fast-food). After being processed, the anonymous data allow extracting confidential information such as vehicle owner identity using its house localization [Wie+10].

5.1.4.3 Communication Deceiver

This attacker emits authentic messages with erroneous content. For instance, a malicious traffic light can send incorrect Signal Phase and Timing (SPaT) messages with a color state which differs from the *phenotypic* state. At best, it creates two different outputs which confuse the automated driving system. At worst, if the real state color is unavailable (e.g., NLoS), the system relies on a single incorrect output from the SPaT. Another example of erroneous message content is the definition of a node dimension for the standardized *Cooperative Awareness Message* [ETS14]. Indeed, the absence of correlation between the class of a V2X node (e.g., pedestrian) and the node dimensions could allow *Communication Deceivers* to emit a message defining an object with an implausible size. Therefore, a pedestrian node may have a length that is between 10 centimeters and 102 meters. Despite some standards recommendations, the choice of plausibility mechanisms regarding V2X Data are left open. Thus, if these erroneous content remain unchecked that may lead to some mis-associations between a V2X message and a sensor measurements.

5.1.4.4 OTA Poisoner

This attacker sends any malicious updates Over-The-Air (OTA). Indeed, CAVs will update OTA their software, firmware, *Data Storage* to fix vulnerabilities, inaccurate information, bugs [Bri14]. A malicious update can alter the integrity of the *Data Storage* by modifying the processing algorithms (e.g., cryptographic algorithms) or the perception data (e.g., cartography data).

5.1.5 Fusion Persuader

Persuaders disrupt the *processing* and *storage* stages to disable or to deceive the perception system. *Persuaders* can perform the followings attacks:

5.1.5.1 Misbehaving Ground Truth

is a road object behaving against the CAV mission (e.g., pedestrian crossing the road at red or traffic light blocked on a red state). These attacks have safety, functional, financial, and privacy impacts on the system. Indeed, a pedestrian faking a collision can block the CAV, extort money from the car company/driver, or provoke an emergency braking threatening passenger safety [Tat17]. Such a behavior questions the need to register and report such actions using the camera recording as juridical proof. Indeed, the recording and storage of identifiable traits of an individual may imply some privacy issues.

5.1.5.2 Sybil Gating

The *Gating* process is a filtering/screening mechanism to determine which objects observations (e.g., V2X messages or sensor measurements) are valid candidates to update existing objects tracks. *Gating* aims primarily to reduce unnecessary computation during data association and tracks maintenance processes [BP99]. Therefore, an attacker could create valid virtual candidates to increase the computation load of *Data Processing*. Although lidar spoofing is possible [PS15], its feasibility in dense or/and highly dynamic scenario may be unrealistic. Indeed while the targeted vehicle is moving fast or is highly surrounded by *Road Objects*, aiming its lidar to achieve a detection is challenging. However, the creation through V2X communication of ghost vehicles [GGS04] fitting the gate conditions is achievable. Therefore, the attacker could create *Sybil Attacks* to disable the filtering benefits of the *Gating*. We define such attack as *Sybil Gating* (Figure 5.1).

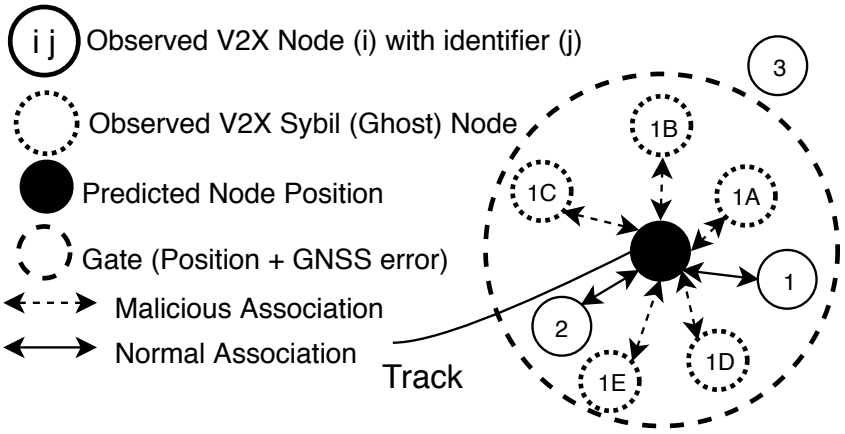


FIGURE 5.1: Sybil Gating

5.1.5.3 Tracking Poisoner

Tracking algorithms aim to predict the state of an object at the next step according to the measurement of object state at the current step. Thus, the system must update each track of its tracking database to ensure the next prediction [BP99]. However, it remains unclear how to perform the track management in pseudonymous V2X communication [Jae+12]. According to the European Certification Policy [gro17], a vehicle can contain simultaneously valid pseudonymous certificates. As mentioned, a vehicle can create a ghost vehicle per pseudonymous certificates. Without proper trustworthiness mechanisms, the ego-CAV will have its tracking database poisoned by tracks of ghost vehicles (Figure 5.2).

Thus, we called such attack *Tracking Poisoning*. Also, such attacks require to adapt existing tracks update mechanisms. Indeed without proper tracks update, these attacks could impact the association process which aims to find the most plausible acquisition-to-track association.

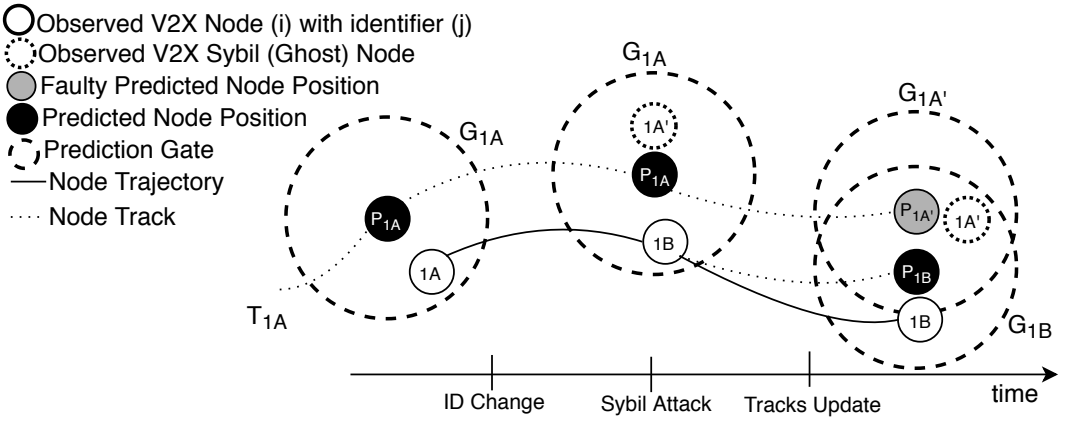


FIGURE 5.2: Tracking Poisoner

5.1.5.4 Fusion Manipulator

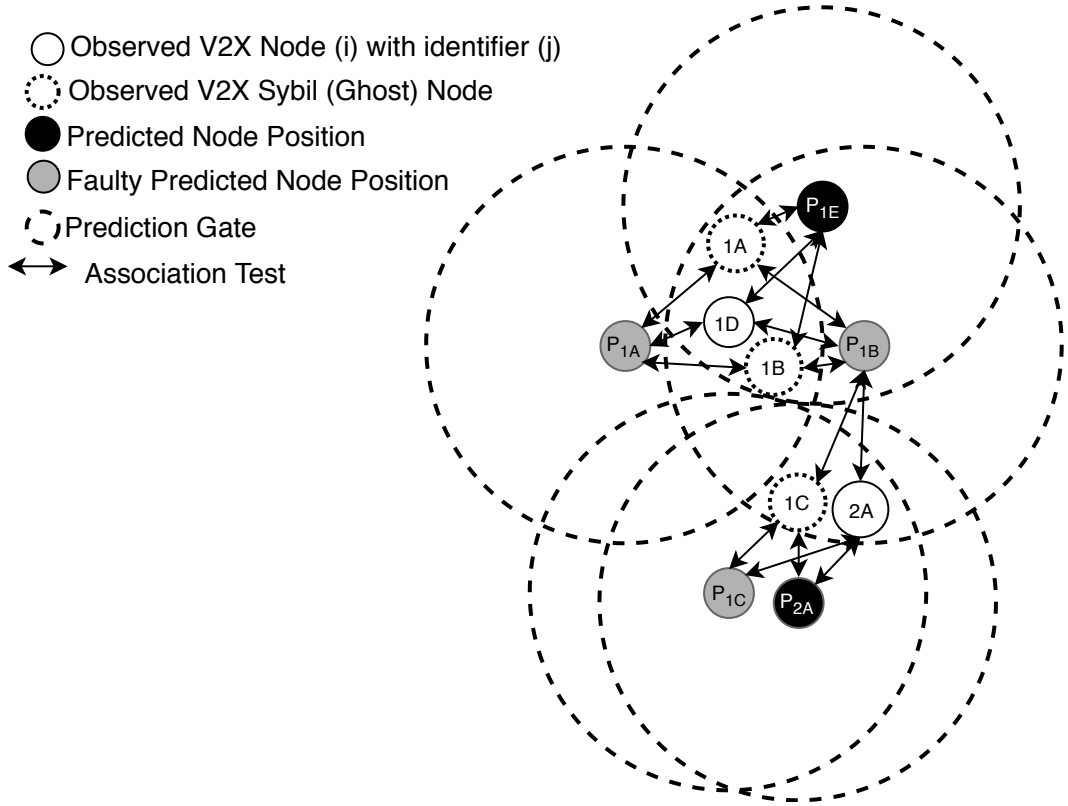


FIGURE 5.3: Association Manipulator Attack

Association algorithms aim to search the most likely acquisition from each source observation set (e.g., sensor measurements or V2X messages) that share the same

detected object as subject. Therefore, a *Fusion Manipulator* has multiple ways to manipulate the association process.

First, an attacker can increase the computation time by increasing the number of potential measurements (*Sybil Gating* or/and *Tracking Poisoner*). For instance, Merdrignac et al., [MSN17] proposed a perception system associating lidar measurements and V2X messages. Despite working with few connected pedestrians, their system does not scale in dense scenarios. Indeed, the perception system associates each lidar observations within the *Gating* area, which is the Global Navigation Satellite System (GNSS) error defined as a circle of 5 meters radius around the GNSS position of the pedestrian emitting V2X messages. Thus, we can assume that the increasing number of potential associations between lidar measurements-V2X Tracks in an urban scenario leads to an increase in the association time.

Second, an attacker can provoke conflicting *acquisition* between two acquisitions. For instance, let us consider a Green Light Optimal Speed Advice (GLOSA) system that uses camera acquisition to verify the content correctness of a SPaT message [Raw+17]. A *Misbehaving Ground Truth* attacker alters the physical signal state of a traffic light [Ghe+14]. Hence, the perception system would disapprove all SPaT message thinking that the camera acquisition represents the correct state.

5.2 Security Goals Model [Mon+18b]

To secure a CAV perception, both identifying and defining proper security goals against identified attackers are mandatory. This section defines such security goals. First, we identify security countermeasures based on the attackers defined in Section 5.1. Then, we derive the security goals model from the identified countermeasures. Finally, we evaluate the model against standardized models.

5.2.1 Security Goals

This section identifies security Goals for each attacker defined in Section 5.1.

5.2.1.1 Security Goals for Sensor Disrupter

To do so, we define the security goals for a *Sensor Disrupter* which regroups *Sensor Illusion*, *Sensor Blindness*, and *Evil Sensor Calibrator*.

Sensor Illusion requires mechanisms which assess the trustworthiness of sensor measurements. Approaches checking the measurement *consistency* assume that a ghost (e.g., Radar) or spoofed measurements are not or hardly repeatable. Therefore, the use of metrics such as *Object Existence* [Aeb17] computed on the object past detections allows to down-weight newly appearing and inconsistent objects during the fusion process.

Sensor Blindness attacks target the *availability* of sensor measurements. Therefore, it is crucial to ensure the *redundancy* of *Data Acquisition*. Current solutions include hardware redundancy (redundant sensor) or data redundancy (different sensor type). For instance, SPaT messages can provide an accurate state of a traffic light while the camera is under *Sensor Blindness*.

Evil Sensor Calibrator targets the sensor *integrity*. Therefore, the security goal to ensure is physical *integrity*. Indeed, a sensor should not be easily manipulated or moved. For instance, tamperproof hardware can store valuable data (e.g., detection algorithms). Also, the access to the sensor settings must be restricted. Thus, *Access Control* is mandatory to identify and to authenticate authorized personnel. Hence,

binding authorized actions to a person profile limits its actions on the sensor according to its function (e.g., developer, mechanic).

Ground Truth Falsifier requires to harden the physical structure of *Road Objects* to ensure their *Phenotype integrity* and *availability*. For instance, the use of anti-graffiti coatings is a solution to avoid the alteration of road signs.

Overall, *Accountability* is a significant security goal against a *Sensor Disrupter*. Indeed, the sensor inability to perform its task must be recorded to understand the causes of misperception. For instance, if the radar detects an object forward but the camera does not, an analysis of the images recorded by the camera can explain that the mis-detection was due to the dense fog which blinded the camera. However, the global acceptance of this mechanism is unsure due to privacy concerns. For instance, the recording of a person face to identify and punish the author of road marks forgery is a possible option [Els17]. But, it requires a strict Privacy Policy regarding the data recorded by the camera of a road infrastructure or a CAV.

5.2.1.2 Security Goals for a Malicious Communicator

Against recent attackers such as a *Voyeur*, standardized countermeasures are ineffective. Thus, the need for new security goals is necessary. Although mentioned [PFK14; Pet+15c], the need for *Linkability* and *Anonymity* as distinct security goals remain unsettled. Despite *Privacy* recommendations from the European C-ITS Platform group [gro17], efficient countermeasures such as pseudonym change in V2X communication are still unsolved [Pet+15b]. Despite common belief, it is the association algorithm and not tracking that decides whether two observations (e.g., track, sensor measurement or V2X message) belong to the same observed object [BP99]. In the sensor domain, range sensor measurements may be incorrect and anonymous. Therefore, it is essential to define the temporal window between two messages which disallows an association algorithm to match two observations based on just their dynamic state.

Communication Deceiver requires the use of *trustworthiness* countermeasures which regroups:

- *Consistency* mechanisms that check how often the emitting node state deviates from the predicted normal behavior (e.g., Kalman Filter [Jae+12]). Unlike *Voyeur Linkability*, V2X messages linkability is necessary for automotive perception. Indeed to track the V2X node state, the association algorithm must associate each V2X messages to its corresponding tracks.

- *Plausibility* mechanisms which rely on plausibility rules (e.g., maximal emitting distance) [Jae+12], multi-source checking [Raw+17], or single source various means checking [SLG17]. The latter compares the object state contained in the V2X message to the measured object state from the communication radio wave (e.g., Doppler Shift).

- *Reputation* mechanisms which rely on the computation of trust score relative to a V2X node behavior [Tan+17]. The *Scope* of node trust can be global or local. *Local trust* implies that the trust value of a node is computed in the vehicle using trust mechanisms (e.g., *Consistency* and *Plausibility* mechanisms). *Local trust* defines a subjective opinion of the perception system towards a V2X node and therefore should not be extended in a cooperative system to avoid badmouthing attacks [Tan+17]. Whereas, global trust values are computed by a global authority (Public Key Infrastructure) and acknowledged by all authenticated VANET members. A specific authority of the Public Key Infrastructure

(e.g., Misbehavior Authority) collects misbehavior reports and decides on revocation of node [Bre+18].

Overall, *Malicious Communicators* also require the following security goals:

- *Accountability* is mandatory to report and revoke malicious nodes.
- *Adaptability* is a major security goal for communication. Indeed, most of the related work assume that cryptographic algorithms will ensure security goals such as *Confidentiality* and *Integrity*. However, few questioned the algorithms obsolescence due to advances in quantum computing. Therefore, without a backup plan, communication system relying on Public Key Infrastructures based on these algorithms are vulnerable [WZ16]. The need to define a system able to adapt by supporting other algorithms in case of such attacks becomes mandatory. For instance, the SCMS PKI uses such system thanks to the integration of specific authorities named Elector CAs [Bre+18].

5.2.1.3 Security Goals for an Evil Mechanic

Evil Mechanics are difficult to counter because non-expert can hardly detect the malicious actions of an expert. However, some security requirements can be implemented to prevent such attacks.

Security goals against *In-vehicle Manipulator* attacks include *Integrity*, *Availability*, *Access Control*, and *Non-Repudiation*. To perform *In-vehicle Manipulator* attacks, an *Evil Mechanic* will first try to access the hardware or the data. Therefore, *Access Control* mechanisms are important to ensure that only authorized personnel can access the data. For instance, such mechanisms include multiple *authentication* factors to ensure that the personnel or installed programs are authorized to access such data. *Authorization* mechanisms restrict actions from an *Evil Mechanic* or malware. Instructions to modify *Data Storage* should be signed using asymmetric cryptography to avoid communication alteration, hardware replacement or the spoofing of administrator session. Also, *Integrity* mechanisms mandatory to avoid the removal of any hardware components and ensure the overall *availability* of the perception system. Finally, *Accountability* mechanisms (e.g., events logs) is mandatory to monitor actions performed a hardware and its data. For instance, during a hardware replacement, the hardware logs indicate if it is new or already used.

Security goal against *In-vehicle Miner* attacks focus on *Confidentiality*. As mentioned, *Data Storage* contains valuable information such as private information or fusion algorithm. Therefore, they should be encrypted.

5.2.1.4 Security Goals for a Fusion Persuader

We define security goals for a *Fusion Persuader* that require the following *Trustworthy* mechanisms:

- *Consistency mechanisms* that detect a potential deviation between the estimated state and the observed state of a data source.
- *Plausibility mechanisms* that confront multiple data sources and detect disagreements among sources. For instance, the disagreement regarding a traffic light state between a camera recording and a SPaT message will raise an anomaly report [Raw+17].

- *Reputation mechanisms* that compute the opinion value of the perception system regarding a perceived *Road Object* by using sensor confidence and V2X node trust metrics. The former assigns a weight to the sensor observations based on sensor past performances such as the number of successful detection of a *Road Object*. The latter is the trust computed based on the detection number of malicious messages emitted by a V2X node.

Also, *Accountability* mechanisms require to record every conflict between data sources that occurs during the fusion process. The aftermath goal is to provide meaningful reports to the Misbehavior Authority [17]. For instance, law enforcement authorities or insurance companies can request these reports to verify the events occurred in an accident. But also, it could help OEMs to detect, understand, and improve vehicles automation. Experts can extract events logs and misbehavior reports to reconstruct the road scene and correct potential weaknesses in the cooperative perception.

Also, *Freshness* mechanisms are mandatory to update the tracks database. For instance, the temporal freshness of tracks is a criterion to remove ghost tracks caused by *Sybil Gating* attacks or outdated *Road Object* tracks that are out of the perception range.

Finally, *Adaptability* mechanisms require to patch the fusion algorithms against potential undiscovered weaknesses during the vehicle lifetime. Moreover, in the case of a detected faulty/malicious source of acquisition, the system can only rely on its communication mode or on its local sensors mode to achieve perception [Abu+16].

5.2.2 Our Security Goals Model (SGM) [Mon+18b]

Figure 5.4 presents our security goals model derived from the attackers security goals identified previously. Our model includes new security goals defined follow:

Privacy is the degree to prevent unauthorized parties to obtain sensitive information. Note that *Privacy* includes *Confidentiality* because sensitive information does not only imply private data but also confidential data (e.g., source code) [Fir04].

- *Anonymity* is the degree of identity disclosure of data users. Thus, *Pseudonymity* is one degree of anonymity that uses pseudonyms (e.g., pseudonym certificate) to identify users.

- *Linkability* is the degree of linking anonymous or pseudonymous data to their owner risking a potential disclosure of its private identity (e.g., home localization).

Trustworthy is the degree of trust assessed by the system regarding perceived *Road Objects* and perception data (Section 3.1.3). Trustworthy mechanisms rely on reputation, consistency, plausibility security goals.

- *Reputation* is the perception system opinion of a V2X system entity. This opinion is subjective. Its validity domain ranges from local to global.

- *Consistency* is the degree of temporal plausibility of a *Road Objects* behavior or products of behavior assessed by the perception system along the perception lifecycle (Section 3.1.3).

- *Plausibility* is the degree to which the system verifies that the perceived data are consistent with the ground truth (Section 3.1.3). As mentioned, other acquisition sources, *Road Objects* model, maximum-minimum thresholds, or Highway Code can be system ground truth assuming they are trustworthy.

Phenotype Integrity is the degree of protection of the *Phenotype* of a *Road Object* from malicious alterations.

Accountability is the degree of mapping security-related events to system entities.

- *Non-repudiation* is the degree of actions recognition of the entity that performs it.
- *Reporting* is the degree of recording Non-repudiated actions.
- *Security Auditing* is the degree of prevention, analysis, and evaluation of occurring, occurred, and potential security-events within a system.

Adaptability is the degree of attack recovery and defense of a system against future similar attacks.

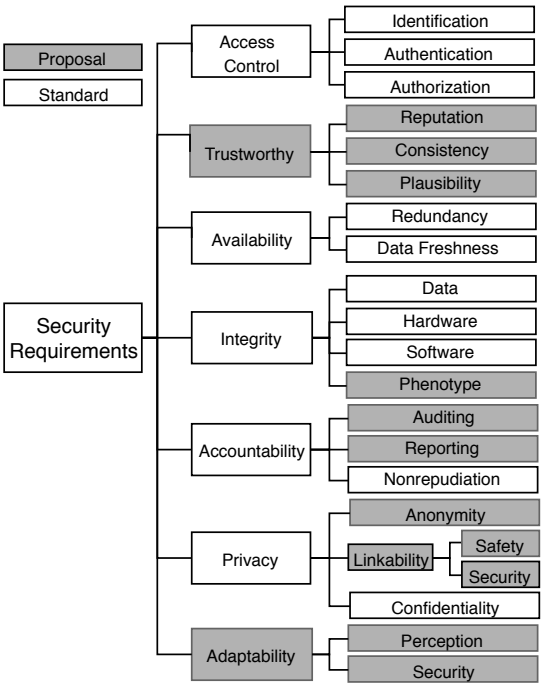


FIGURE 5.4: SGM

5.2.3 Comparison

This section analyses our security goal model through the comparison Table 5.3.

To build this table, we define the *Target of Evaluation* which is the perception domain (Section 3.1.3). Then, we set the involved entities which are *Objects* which regroup *Ego-CAV* and *Road Object*. Then, we link each *Object* to its *Data Stages* (Figure 3.4). This approach avoids speculating on the chosen architecture for data fusion. Indeed, acquisitions tracking is either decentralized (acquisitions source) or centralized (fusion ECU) [BP99]. Accordingly, we relate each *Data Stages* to a sub-attacker (Table 5.2). Finally, we match to each sub-attackers its security goals (Section 5.2.1).

Second, we compare our proposal to standardized security goal models such as *STRIDE* and *CIA*. Where *STRIDE* stands for *Spoofing*, *Tampering*, *Repudiation*, *Information Disclosure*, *Denial of Service*, and *Elevation of Privilege*. *CIA* stands for *Confidentiality*, *Integrity*, and *Availability*. Therefore, both do not consider *Trustworthiness* and *Adaptability* as security goals. An explanation is that both models were designed

for traditional IT environment and not for the CAV domain. Also, both do not distinguish *Authentication* and *Identification* which is not adapted to the CAV domain. Indeed, in the case of Sybil attack on V2X nodes, the system allows a single identity to authenticate itself using multiple authenticators (e.g., Pseudonyms). Finally, *STRIDE* refers to *accountability* only through *non-repudiation*. However, CAV domain may rely on trust between entities and therefore will need security reports from CAVs to report malicious *Data Objects*.

That is, we showed the need for a new security goal model in the domain of CAV. Our model can answer this need. Indeed, our SGM includes new security goals which fits the recent attack occurring in the CAV domain. Besides, our SGM maps road objects, perception data, attacker model and security goals to ease the security analysis of a CAV.

Object	Data Stage	Attacker Model	Security Goals	STRIDE	CIA	**SGM**
Ego-CAV	*Acquisition*	*Voyeur* *Fully Adversarial* *Sensor Blindness* *Sensor Illusion*	Access Control	≈	✘	✓
			Trustworthy	✘	✘	✓
			Availability	✓	✓	✓
			Integrity	✓	✓	✓
			Accountability	≈	✘	✓
			Privacy	≈	≈	✓
			Adaptability	✘	✘	✓
	Processing	*Fusion Manipulator* *Communication Deceiver* *Sybil Gating*	Availability	✓	✓	✓
			Trustworthy	✘	✘	✓
			Accountability	≈	✘	✓
			Adaptability	✘	✘	✓
	Storage	*In-vehicle Miner* *In-vehicle Manipulator* *Tracking Poisoner* *Evil Sensor Calibrator* *OTA Poisoner*	Access Control	≈	✘	✓
			Availability	✓	✓	✓
			Integrity	✓	✓	✓
			Accountability	≈	✘	✓
			Privacy	≈	≈	✓
Road Object	*Phenotype*	*Ground Truth Falsifier* *Misbehaving Ground Truth*	Availability	✓	✓	✓
			Integrity	✓	✓	✓

≈: the full security goal is not covered as depicted in Figure 5.4

TABLE 5.3: Comparison of SGM with STRIDE and CIA

5.3 Conclusion

In this chapter, at first, we describe a data lifecycle within generic perception system model from which we identified its primary assets. Therefore, we derived an attacker model based on such assets and state of the art attacks. Following, we determined related countermeasures then accordingly we defined a security goals model. Finally, we compared our SGM against some standardized models and highlighted missing security goals.

As a result, this chapter showed the need for costless and straightforward countermeasures against attacks performed on the surrounding environment. Also, despite the use of pseudonym certificate, we explained the need to investigate privacy mechanisms furthermore against Voyeur attacker. Overall, we demonstrated that sensor and V2X data are untrustable and may lead to new attacks within data fusion processes which were not designed for an uncooperative environment. Therefore, we explained the need to revisit such processes which led to the identification of three trustworthy sub-goals. Also, we showed the current lack of adaptability countermeasures of a perception system which remains an unsettled issue. Indeed, few works analyzed the obsolescence of perception algorithms such as the break of cryptographic algorithms. Finally, by focusing on the automotive perception, we demonstrated that current tools for threat analysis are insufficient. To conclude, we believe that standardizing automotive perception will help security experts to deepen existing automotive security analysis.

In the next chapter, we will present two modules of our FRPA to counter a *Malicious Communicator*. For the first module, we propose a Framework for a Machine Learning based Failure Classifier to detect anomalies in the V2X message. Then, we will describe the V2X-Sensor Correlation module that uses local sensor to detect V2X anomalies.

Chapter 6

Classification & Correlation Modules

In this chapter, we present two modules of our FRPA to counter the attackers described in chapter 5. In Section 6.1, we describe our framework to design a ML based Failure Classifier. Then, Section 6.2 describes our V2X-Sensor correlation module. Finally, Section 6.3 concludes this chapter.

6.1 Framework for ML based Failure Classifier

Figure 6.1 depicts our framework which describes the processes and flows involved in the realization of a ML based Failure Classifier. The framework has two phases named *offline* and *online*. Each phase contains several stages with processes, data, checks, and databases. The stages of the offline phase are:

- Acquisition that aims to collect the data to train and test our classifier.
- Preprocessing aims to format the acquired data.
- Training aims to train the classifier model and optimize its hyper-parameters value using the formatted data.
- Testing aims to evaluate the trained model.

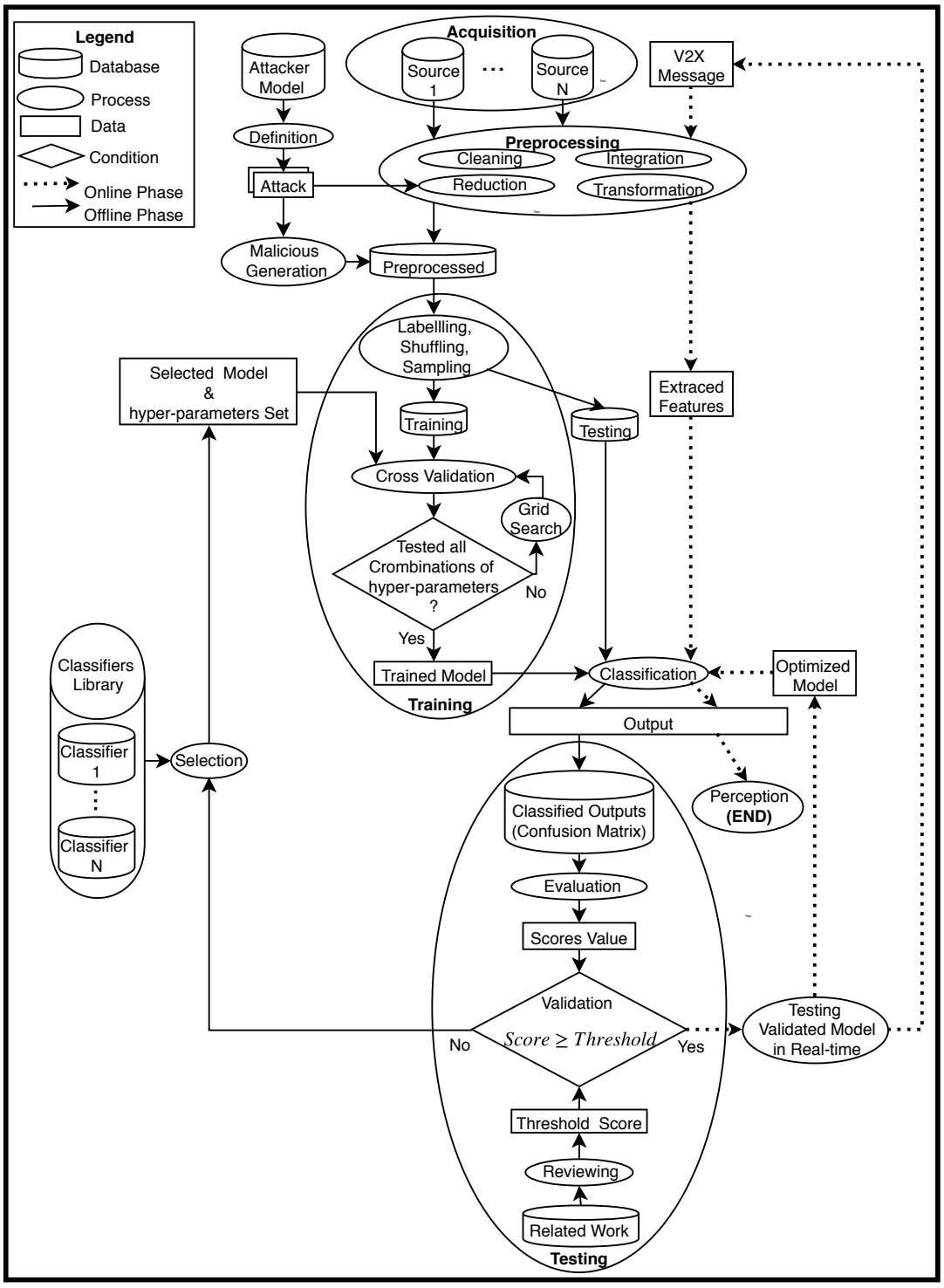


FIGURE 6.1: Framework for ML based Failure Classifier

6.1.1 Acquisition

During this step, data are collected to train and test our classifier. we found three types of datasets for three ITS stations:

- pedestrians,
- motos, and
- CAV.

Table 6.1 summarizes each *source database* where *raw* is an unprocessed dataset.

Source	**ITS-Station Type** [ETS14]	**#Entries** (Row)		**#Features** (Column)	
		Raw	Cleaned	Raw	Reducted
[18a]	Car	70847	55261	37	2
[Teo18; Fur18]	Moto	236	201	2	
[18b]	Pedestrian	501	501	15	

TABLE 6.1: Collected Datasets

6.1.2 Preprocessing

Each *source database* is cleaned and formated into a *preprocessed database*. Indeed, unformatted and cleaned *source databases* may contain duplicated, noisy, or incomplete instances. Also, each *source database* has a different total number of features (Table 6.1). Therefore, we must select the required features for our classifier goal. To do so, we follow these *preprocessing* steps: *cleaning*, *integration*, *transformation*, and *reduction*.

6.1.2.1 Cleaning

An algorithm processes missing, noisy, and duplicated data. The causes of this type of data could be technical problems during data gathering, human mistakes, or attacker manipulation during data entry. For example, a Random Forest classifier does not support null values. To tackle this issue, there are three techniques that are *removal*, *manual filling*, and *computed values filling*.

In our work, we use the *removal* technique due to its straightforwardness while allowing us to maintain a high number of instances for each *source database*. While the second method is time-consuming due to the number of missing values (Table 6.1), the third method can generate a bias due to the number of redundant entries that may influence the mean or median values used to fill the missing values.

Figure 6.2 highlights the presence of outliers in the *raw datasets* that are instances far from the instances cluster. In our work, an outlier is a data point that is distant from a group of data points. By cross-validation, we remove outliers that do not fit the dimension range of another car *data source* [18c]. For instance, in Figure 6.2, we circled in black outliers such as a car with a width smaller than 1 meter.

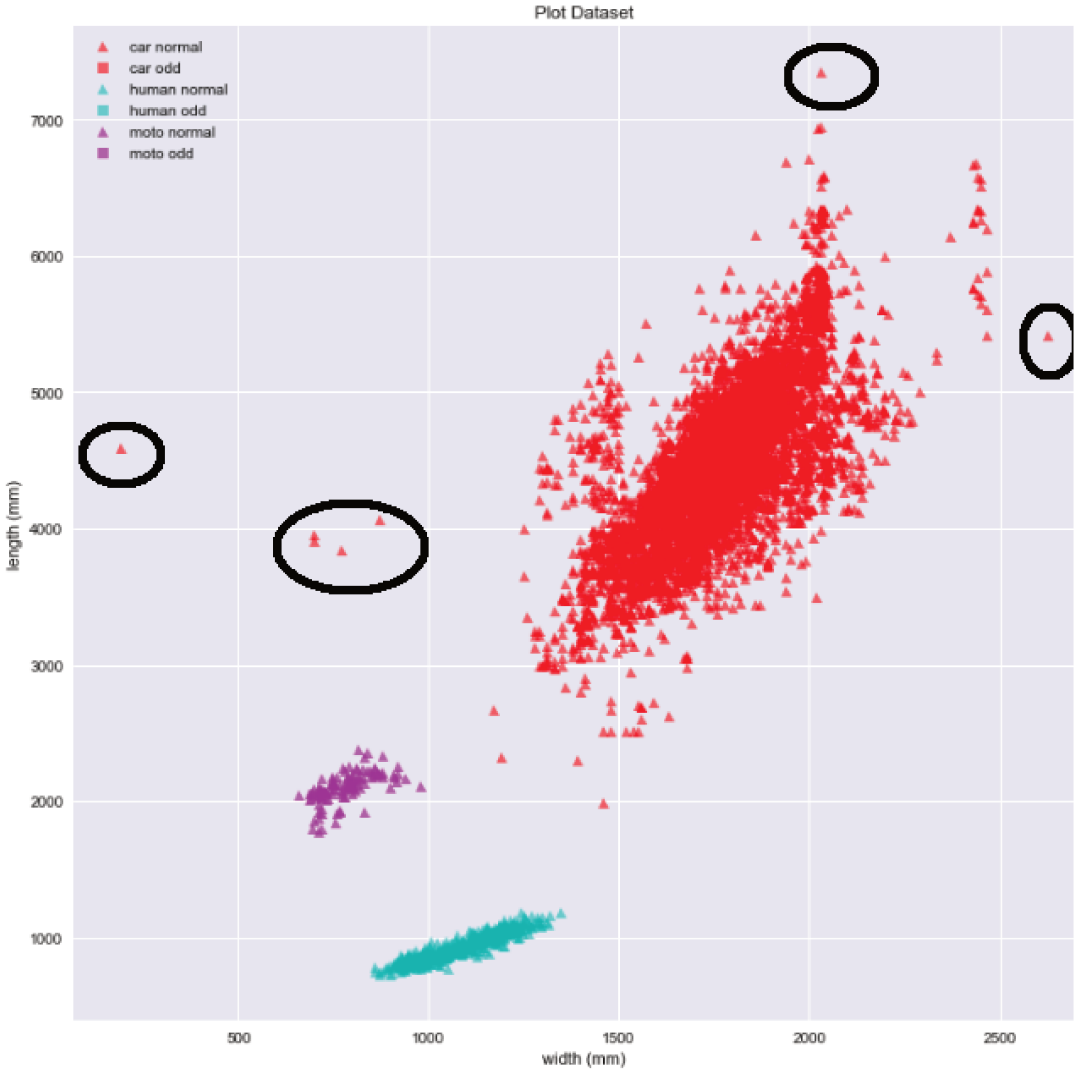


FIGURE 6.2: Dimension Datasets with outliers

6.1.2.2 Integration

An algorithm transforms data from each source into consistent data. Some *source database* had Italian or English words or used millimeters or centimeters as their dimension metric. Thus, during *integration*, we chose a common language (English) and common dimension metric (millimeter). Also, we aggregate each *source database* format (e.g., csv) into a common database named *preprocessed database*. We use panda *dataframe* as *preprocessed database* format.

Note that, for the pedestrian dataset, we assume as a *length* feature the *chest girth* column. Indeed, while the value of the vehicle length is independent of its dynamic state (stopped or moving), a pedestrian length depends on its dynamic state. For instance, while walking, the pedestrian length is the distance between the back foot heel and the front foot toes. At rest, the pedestrian length is the foot length. Therefore, we chose a walking pedestrian length due to its greater length which increase the safety distance when the pedestrian is at rest. Also, the walking length noises the ground-truth value of a pedestrian length at rest and, therefore, increases data privacy. Thus, our assumption holds despite sacrificing V2X dimension accuracy to detect and correct errors from sensor measurements for automotive perception fusion.

6.1.2.3 Transformation

We transform raw features into a specific format needed by the model as follows. For instance, we use *Normalization* to scales raw numerical values into a specified range (i.e., between 0 and 1). This technique ensures that features with large domains will not dominate features with smaller domains.

6.1.2.4 Reduction

The attacker targets the width, length, and type of an ITS-Station. Thus, we select the needed features according to our attacker model.

Then, we remove the redundant instances among the selected features. For instance, different car models may have the same dimension. As a result, there are duplicated instances in the future *preprocessed database* that must be removed. Indeed, duplicated instances may provoke bias on the performance of learning techniques by affecting the inferred statistical distribution of data features.

Table 6.2 gives the number of non-misbehaving instances within our *preprocessed database*. As seen in Table 6.1, the car *source database* has around 70000 instances and at most 37 features before the *preprocessing*. After the *preprocessing*, *preprocessed database* which is the aggregation of all *data sources* has 5500 instances and 3 features.

Database Type	**#Instances (Row)**		**#Features (Column)**
	Non-Misbehaving	*Misbehaving*	
Pre-Processed	5646	1000	3
Training	4490	832	
Testing	1156	168	

TABLE 6.2: Testing Dataset

6.1.3 Generation of our dataset

According to the attacker model, we generate our dataset of implausible dimension for each ITS-Station type (Figure 6.3). The dimension values are generated within the graph minimum and maximum of each axis. To match to the ground truth, the number of misbehaving instances must be much lower than the number of non-misbehaving instances. We arbitrarily generated a fifth of the size of the non-misbehaving dataset.

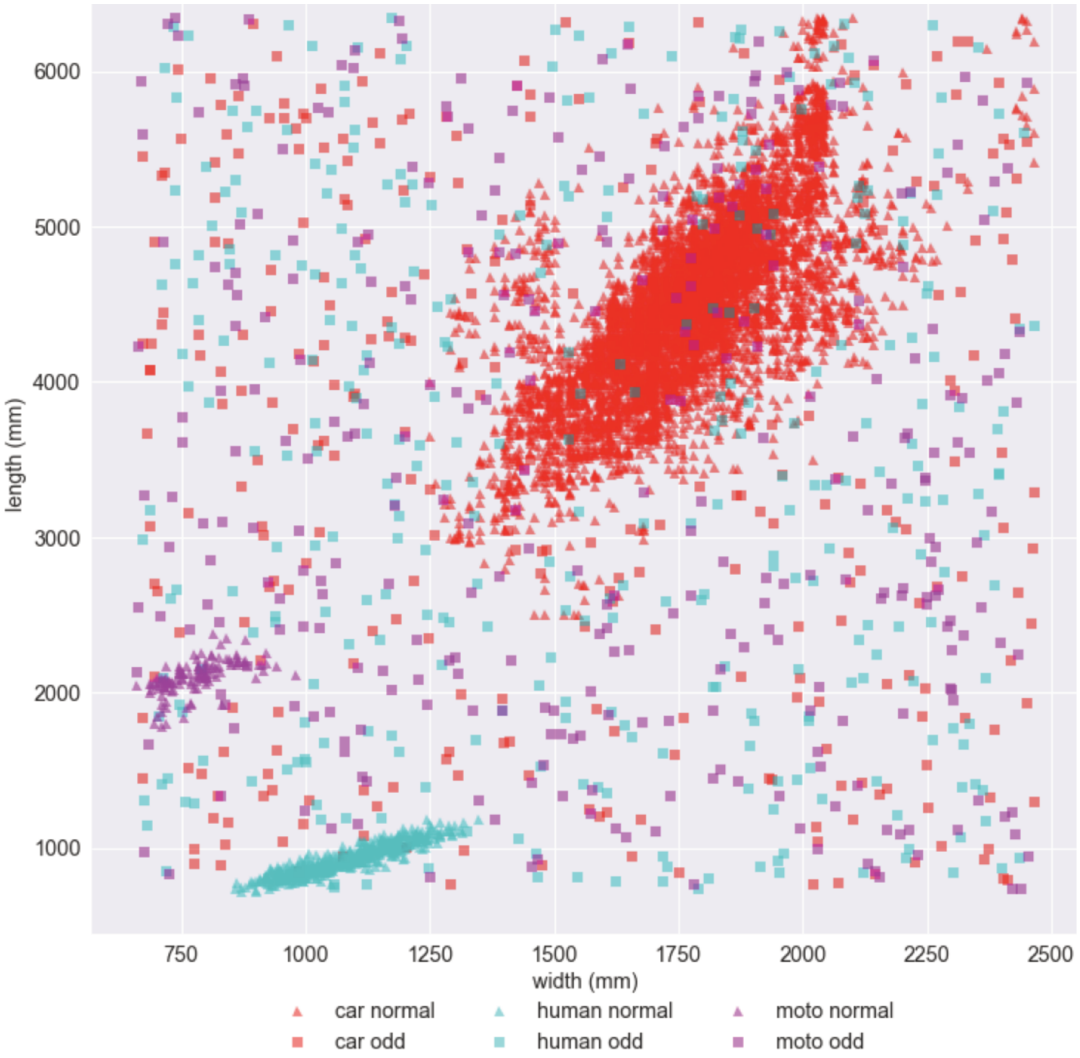


FIGURE 6.3: Generated Misbehaving dataset

6.1.4 Training

In this stage, we optimize the classification accuracy of our classifier.

6.1.4.1 Labeling, Shuffling & Sampling

The *labeling* process aims to label an instance according to its actual class (e.g., misbehaving or not misbehaving). Our classifier must learn to distinguish a misbehaving instance from a non-misbehaving one. Therefore, we label each instance (non-misbehaving or misbehaving) within the *preprocessed database*. During *cross-validation*

process, the classifier model learns to associate a given instance to its actual label. After training, we compare the similarity between the predicted instance label by the classifier against the actual label of the instance.

The *Shuffling* process sets random instances order given a *preprocessed database*. For instance, this process avoids having a testing database that contains only a dataset of misbehaving instances.

The *Sampling* process divides the *preprocessed database* into multiple databases. A common approach is to divide the database into a *training database* and a *testing database*. Where the first database is for training the classifier model while the second database is for testing the model. In our work, we follow the *Pareto Principle* for the *sampling* process.

6.1.4.2 Selection

The *Selection* process defines the used machine learning algorithm and its statistical model. This model has higher-level properties named *hyper-parameters* which influences the model complexity, learning speed, and its application results. In our work, we use four classifiers and their corresponding models defined in Table 6.3.

6.1.4.3 Cross-Validation & Grid Search

The *Cross-Validation* process aims to train the classifier model given a *training database* and a set of hyper-parameters combination. The second goal of *Cross-Validation* is to avoid over-fitting. Indeed, during its training, the statistical model must minimize its performance error while maximizing its correctness during the *testing* stage. Therefore, by training on the same dataset, the model minimizes its performance error on this specific dataset but not on other datasets. To avoid this behavior, the *Cross-Validation* splits the *training database* into multiple datasets named folds that will serve for the model validation and training. The splitting strategy depends on the chosen type of *Cross-Validation* (exhaustive or non-exhaustive). For each model trained on a given combination of hyper-parameters, the cross-validation computes a performance score based on a defined metrics (e.g., accuracy).

The *Grid search* searches all the combinations of hyper-parameters and hyper-parameters values tested in the *Cross-Validation* process (Table 6.3). At the end of the search, the ML model with the highest performance score becomes the trained model to be used in the *testing stage*.

Classifier	Tested hyper-parameter	Tested Range Values	Optimized Value
Naive	`length_interval`	`[min_l;max_l]`	No Optimization
	`width_interval`	`[min_w;max_w]`	
MLP	`learning_rate`	`{constant, invscaling, adaptive}`	`invscaling`
	`alpha`	$\{10^{-x} \mid x \in [\![4,7]\!]\}$	0.0001
	`activation`	`{identity, logistic, tanh, relu}`	`logistic`
	`solver`	`{lbfgs, sgd, adam}`	`lbfgs`
	`hidden_layer_sizes`	$\{(p_{ij} \in \mathcal{R}^{1\times k} \mid k \in [\![0,5]\!],\ p \in [\![1,50]\!]\}$	`[12, 12, 12]`
Boost	`base_estimator`	`{Default}`	`DecisionTreeClassifier`
	`algorithm`	`{Default}`	`'SAMME.R'`
	`random_state`	`{Default}`	`None`
	`n_estimators`	$[\![1,99]\!]$	2
	`learning_rate`	$\{k/10 \mid k \in [\![1,9]\!]\}$	0.6000000000000001
R. Forest	`Default Hyper-parameter Set`	`{Default Values}` Set	No Optimization

TABLE 6.3: Classifier Hyper-parameters

6.1.5 Testing

The *testing stage* evaluates the performance of a trained model on the *testing dataset*. To do so, this stage has three steps that are *Classification*, *Evaluation*, and *Validation*.

6.1.5.1 Classification

This process tests the performance of the trained model on the *testing database*.

Given a trained model and a *testing dataset* where each instance label is unknown, the classifier model predicts the label of each encountered instance.

Then, we store the comparison between the predicted label of the instance and its actual label in a *confusion matrix database* as defined in Table 6.5. Table 6.4 defines the terms related to this matrix.

In Table 6.5, the positive cases (TP, TN) regroup instances predicted as non-misbehaving, whereas the negative cases (FP, FN) regroup instances predicted as misbehaving. The definition of the positive and the negative case is important during the analysis of classifier performance. Indeed, some metrics such as *F1-score* relies only on positive cases. Therefore, if the classifier performs better in detecting misbehaving instances than non-misbehaving ones, then the assignment of the cases influences the metric score.

For the *online phase*, the real-time system outputs the predicted label and does not create a confusion matrix because the system does not know the actual label of the incoming instance.

Terminology	**Notation**	**Definition**
True Positive	TP	instance predicted as non-misbehaving and is labeled as non-misbehaving
False Positive	FP	instance predicted as non-misbehaving but is labeled as misbehaving
True Negative	TN	instance predicted as misbehaving and is labeled as misbehaving
False Negative	FN	instance predicted as misbehaving but is labeled as non-misbehaving

TABLE 6.4: Terms related to the confusion matrix

Actual \ Predicted	Non-Misbehaving	Misbehaving
Non-Misbehaving	TP	FN
Misbehaving	FP	TN

TABLE 6.5: Matrix for Misbehavior Classification

6.1.5.2 Evaluation

This step evaluates the classifier performance on the *testing database*. Table 6.6 summarizes the list of metrics derived from the confusion matrix.

Metric	Notation	Equation
Recall (Sensitivity)	TPR	$\frac{TP}{TP+FN}$
False Positive Ratio (Fallout)	FPR	$\frac{FP}{FP+TN}$
Specificity	TNR	$\frac{TN}{TN+FP}$
False Negative Ratio	FNR	$\frac{FN}{TP+FN}$
Positive Predictive Value (Precision)	PPV	$\frac{TP}{TP+FP}$
Negative Predictive Value	NPV	$\frac{TN}{TN+FN}$
Accuracy	ACC	$\frac{TP+TN}{TP+FP+TN+FN}$
F1-Score	F1	$2\times\frac{PPV\times TPR}{PPV+TPR}$
Cohen Kappa	κ	$\frac{ACC-\frac{(TP+FP)\times(TP+TN)+(TN+FP)\times(TN+FN)}{(TN+TP+FP+FN)^2}}{1-\frac{(TP+FP)\times(TP+TN)+(TN+FP)\times(TN+FN)}{(TN+TP+FP+FN)^2}}$
Informedness (Youden's J)	J	$TPR+TNR-1$
Matthews Correlation Coefficient	MCC	$\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

TABLE 6.6: Metrics of a confusion matrix

In our work, we analyse the following metrics:

- The *Accuracy* metric measures the proportion of predictions correctly classified among all predictions.
- The *F1-Score* metric is the harmonic mean of precision and sensitivity.
- The *Cohen Kappa* metric [Coh60] measures the agreement between two raters (e.g., the actual label against the predicted label). The kappa score measures the classifier performance with the model compared to its performance using random assignments. The metric value ranges from 0 to 1. While a value of 1 means there is a complete agreement, a value of 0 means there is no agreement.
- The *Informedness* metric [You50] is a function of sensitivity and specificity that measures the overall effectiveness of a test. The metric value ranges from 0 to

1. While a value of 1 means the test is effective, a value of 0 means the test is as efficient as a classifier with a random assignment strategy.

- The *Matthews correlation coefficient* metric [Mat75] is the proportion of responses correctly classified. Unlike *Accuracy* and *F1 score*, this metric takes into account the balance ratios of the four confusion matrix categories. Therefore, this metric takes into consideration both positive and negative cases which include false positive and false negative cases.

6.1.5.3 Validation

To validate our ML classifer, the module must classify perception data with at least a metric score of 90% .

6.2 V2X-Sensor Correlation Module

In this section, we present our V2X-Sensor Correlation module. The module uses a perception sensor to detect abnormal V2X data.

Thus, subsection 6.2.1 defines the local sensors embedded on the vehicle. Then, subsection 6.2.2 describes our V2X model. Next, the third subsection 6.2.3 describes the correlation module.

6.2.1 Sensors Set

Vehicular perception relies on the data acquisition of several sensor types embedded on an automated vehicle named *ego* (Figure 6.4). In this chapter, we consider a perception architecture with a local sensors (radars and cameras) and an OBU defined in Appendix A. This perception system receives V2X messages (m) from surrounding CAVs and sensor acquisitions (a). Then, each detection (message or acquisition) is processed into the same temporal and spatial frame (Data alignment). After alignment, each source processes each detection to a multiple object tracker. At the reception of each detection, the tracker will update the list of objects detected by the source within a time slot. Finally, each tracker provides its updated list of detected objects to our Sensor correlation module.

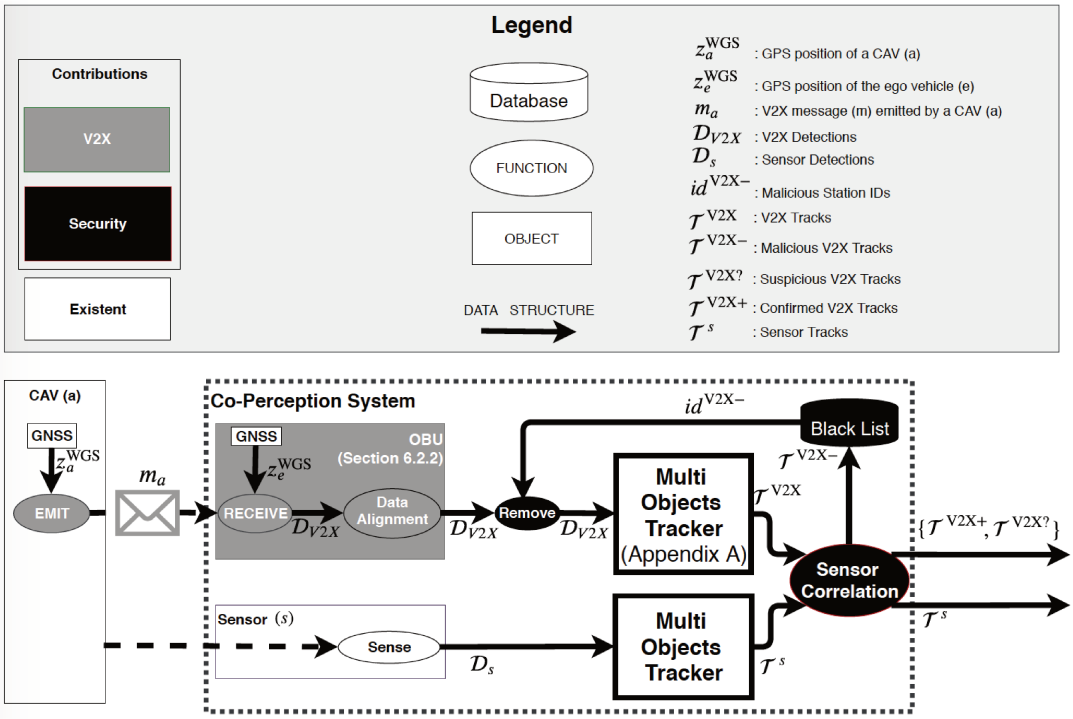


FIGURE 6.4: Resilient Perception System

Each sensor (s_i) senses all CAVs ($\mathcal{A}$) in its Field of View (FoV. Then, the sensor outputs several detected objects ($\mathcal{D}$) at a time (t):

$$\underline{sense}(\mathcal{A}, s, t) = \mathcal{D}_s(t) \tag{6.1}$$

As depicted in Figure 6.4, we will a single sensor model to mimic a real sensor [19a].

6.2.2 OBU Model

A CAV emits its GPS position ($\mathbf{z}_a^{WGS}$) to other CAVs. We model an OBU to simulate the transmission of a safety message($\mathbf{m}$) from a CAV ($\mathbf{a}$) to our ego vehicle thanks to the following functions:

$$\underline{emit}(\mathbf{a}, \mathbf{z}_a^{WGS}, t) = \mathbf{m}_a(t) \tag{6.2}$$

$$\underline{receive}(\mathcal{M}, \mathbf{z}_e^{WGS}, t) = \mathcal{D}_{V2X}(t) \tag{6.3}$$

In the next sections, we give the details of the safety message and the detected V2X objects.

Table 6.7 lists the variable used through this section.

Variable	Meaning
e	ego vehicle
a	an emitting CAV (a)
$\mathbf{m}_a$	a safety message sent by a CAV (a)
$\mathcal{M}$	V2X safety messages
WGS	The WGS-84 reference coordinate system used by the GPS
f	Coordinate system of reference
$\mathbf{z}_a^f$	Mobility data (z) of a CAV (a) defined in f
$D_{V2X,a}$	Detected CAV (a)
$\mathcal{D}_{V2X}$	Detected V2X CAVs
$D_{s,i}$	Detected sensor object(i)
$\mathcal{D}_s$	Detected sensor objects
id_a	station ID of a CAV (a)
C_a	Station Type (e.g., vehicle, moto, pedestrian) of a CAV (a)
$\mathcal{C}_a^f$	V2X rectangular models of a CAV (a)
$\mathbf{p}_a$	OBU Position of an emitter (a)
$\mathbf{oa_i}$	Object attributes of a track (i)
$\mathcal{P}_i$	Measurement Parameters of a tracked sensor object (i)
$\mathcal{P}_a$	Measurement Parameters of a tracked CAV (a)
$\mathbf{R}_i$	Covariance of the measurement noise associated to a tracked sensor object (i)
$\mathbf{R}_i$	Covariance of the measurement noise associated to a tracked CAV (a)
θ_a	Angle between the CAV head (a) and the magnetic North
$\psi_{s,i}$	Angle centered on a sensor (s) between the North and a tracked object (i)
$\psi_{V2X,a}$	Angle centered on an OBU ($V2X$) between the North and a tracked CAV (a)

TABLE 6.7: Variables within the Correlation Module

6.2.2.1 Message Model

We model a safety message as the following vector:

$$\mathbf{m}_a(t) = \begin{bmatrix} id_a & \mathbf{z}_a^{\text{WGS}} & C_a & \mathcal{C}_a \end{bmatrix} \tag{6.4}$$

We define a V2X Object Model as a rectangular model with four corners (Figure 6.5). Moreover, each corner position depends on the vehicle width (w_a) and length (l_a).

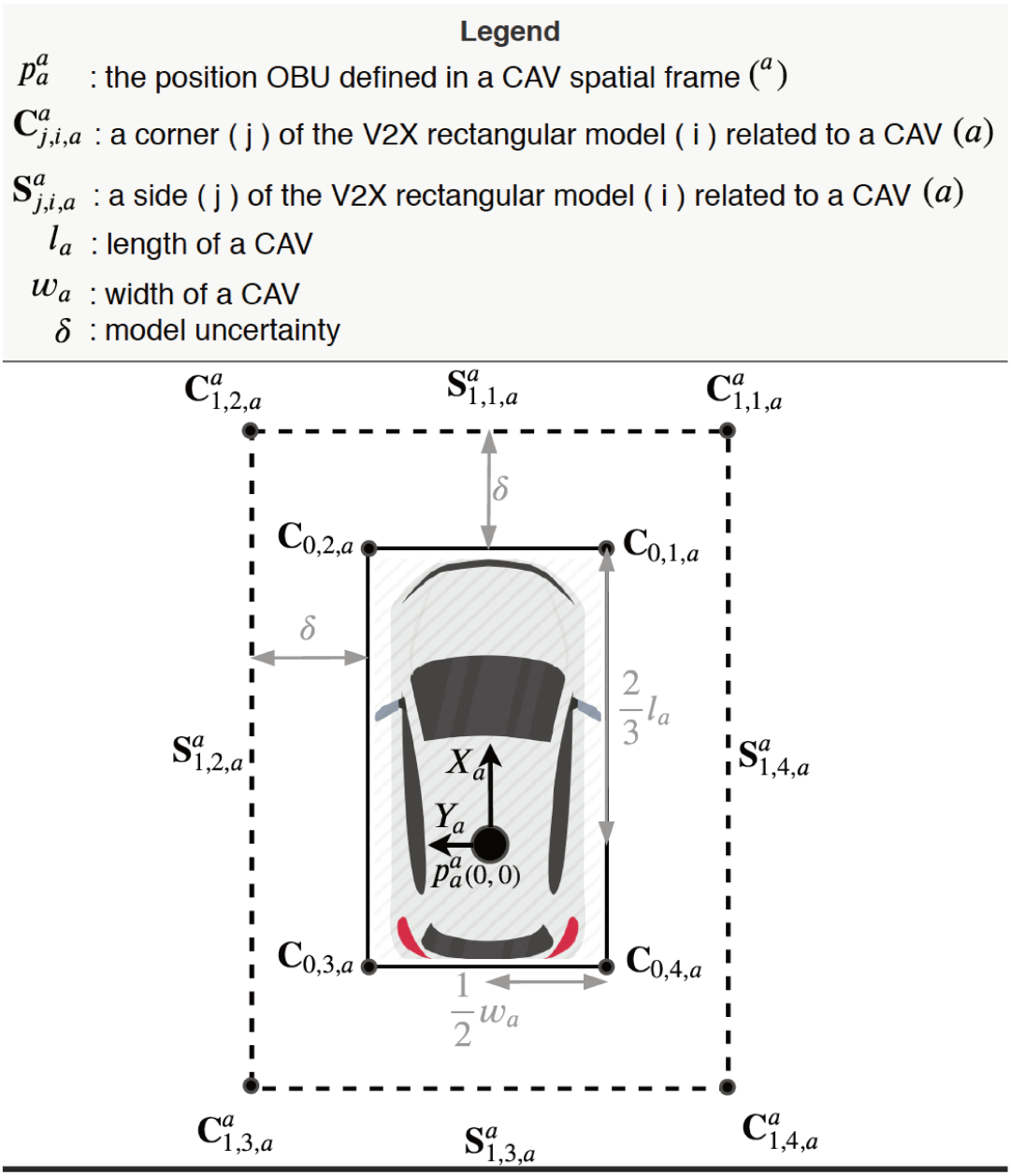


FIGURE 6.5: V2X object models

6.2.2.2 Data Alignment

Before data fusion, each perception source must define each detected object in terms of data structure, spatial frame, temporal frame and classification semantic.

For an OBU, a detected object is modeled as the following set:

$$D_{V2X,a} = \{\mathbf{z_a^e}, \mathbf{R}_a, \mathcal{P}_a, \mathbf{oa}_a\} \tag{6.5}$$

$\mathbf{z_a^e}$ is the object state in the spatial frame of the ego vehicle (e). Besides, **oa** is a set of attributes unused by a tracker such as the source bearing angle (ϑ):

$$\mathbf{oa}_a = \{V2X, id_a, \mathcal{C}_a^e, \psi_{V2X,a}\} \tag{6.6}$$

Regarding temporal alignment, our work assumes temporal synchronization and constant acquisition frequency for all sources.

6.2.2.3 Spatial alignment

For spatial alignment, we perform two spatial transformation. First, we transform the V2X state from the GNSS system to the East North Up (ENU) system [19b]:

$$\mathbf{z}^{\text{ENU}} = \underline{geodetic2enu}(\mathbf{z}^{\text{WGS}}, \mathbf{0}^{\text{WGS}}, \mathcal{P}^{\text{WGS}}) \tag{6.7}$$

Where $\mathbf{0}^{\text{WGS}}$ is the location of the *WGS-84* origin. Finally, $\mathcal{P}^{\text{WGS}}$ is the spheroid settings [19b]. Then, we position our V2X object coordinates into the ego vehicle frame with a second transformation ($\mathbf{z}_a^e = \mathbf{z}_e^{ENU} - \mathbf{z}_a^{ENU}$). The same transformation is performed for the object model $\mathcal{C}_a^e$.

6.2.3 Correlation Module

We check the plausibility of a V2X object by verifying if the camera and OBU have detected this object. In Table 6.8, we collet the variables used in the correlation module.

Variable	Meaning
$\mathcal{S}_s$	Sensor settings
$\mathcal{T}^s$	Tracked sensor objects
T_i	Tracked V2X object (i)
$\mathcal{T}^{\text{FoV}}$	V2X tracks in camera FoV
$\mathcal{T}^{\text{V2X}}$	Tracked V2X objects
$\mathcal{T}^{\text{V2X}+}$	Real V2X tracks
$\mathcal{T}^{\text{V2X?}}$	V2X tracks with unconfirmed existence
$\mathcal{T}^{\text{V2X}-}$	Malicious V2X tracks
$\rho_{s,max}$	Maximal detection range of the sensor
$\rho_{s,i}$	Distance between the position of a sensor (s) and the position of a track (i)
$\phi_{s,max}$	sensor azimuth (FoV)

TABLE 6.8: Variables within the Correlation Module

Algorithm 1 is composed of three algorithms used in our correlation module. The next sections describe each algorithm.

Algorithm 1: V2X-Sensor Correlation (checkCorrelation (...))

Input : $\mathcal{T}^{\text{V2X}}, \mathcal{T}^{s}, \mathcal{S}_s$
Output: $\mathcal{T}^{\text{V2X+}}, \mathcal{T}^{\text{V2X?}}, \mathcal{T}^{\text{V2X-}}, \mathcal{T}^{\int}$

if $\mathcal{T}^{V2X}$ *is not empty* **then**
  $\mathcal{T}^{\text{FoV}}$ = checkFoVTracks($\mathcal{T}^{\text{V2X}}, \mathcal{S}_s$)
  if $\mathcal{T}^{FoV}$ *and* $\mathcal{T}^{s}$ *are not empty* **then**
    { $\mathcal{T}^{\text{V2X?}}, \mathcal{T}^{\text{V2X+}}$ } = checkExistence($\mathcal{T}^{s}$, $\mathcal{T}^{\text{FoV}}$)
    { $\mathcal{T}^{\text{V2X-}}, \mathcal{T}^{\text{V2X?}}$ } = checkNLoS($\mathcal{T}^{\text{V2X?}}, \mathcal{T}^{\text{V2X+}}$)
  else
    $\mathcal{T}^{\text{V2X?}} \leftarrow \mathcal{T}^{\text{V2X}}$
    $\mathcal{T}^{\text{V2X+}} = \emptyset$
    $\mathcal{T}^{\text{V2X-}} = \emptyset$
  end
else
  $\mathcal{T}^{\text{V2X?}} \leftarrow \mathcal{T}^{\text{V2X}}$
  $\mathcal{T}^{\text{V2X+}} = \emptyset$
  $\mathcal{T}^{\text{V2X-}} = \emptyset$
end

6.2.3.1 FoV Check

Algorithm 2 verifies the presence of the tracked V2X object in the camera Field of View (FoV). To do so, the algorithm must compute the angle between the sensor (s) heading and the position (x_i^e, y_i^e) of the track (i) as defined in Equation 6.8. Besides, the algorithm computes the distance between the sensor location and the location of emitter OBU (Equation 6.9).

$$\psi_{s,i} = \arctan \frac{x_i^e - x_s^e}{y_i^e - y_s^e}; \tag{6.8}$$

$$\rho_{s,i} = \sqrt{(x_i^e - x_s^e)^2 + (y_i^e - y_s^e)^2}; \tag{6.9}$$

Algorithm 2: FoV Check (checkFoV(...))

Input : $\mathcal{T}^{\text{V2X}}, s_1$
Output: $\mathcal{T}^{\text{FoV}}$

$\psi_s = (\theta_s + \phi_s) \div 2$
for *each* $T_i^{V2X} \in \mathcal{T}^{V2X}$ **do**
  $\rho_{s,i} \leftarrow$ Equation 6.9
  $\psi_{s,i} \leftarrow$ Equation 6.8
  if T_i^{V2X} *is within sensor max range and bearing* **then**
    $\mathcal{T}^{\text{FoV}} \leftarrow T_i^{\text{V2X}}$
  end
end

Besides, the algorithm computes a rectangle model based on the width and the length values contained in the received V2X message.

6.2.3.2 Existence Check

Algorithm 3 verifies if there are objects simultaneously tracked by the camera and the OBU tracks in the camera FoV. Therefore, the algorithm verifies if the location of a camera track (p_j^e) is within the object model of the V2X track ($\mathbf{C}_i^e$). To do so, Equation 6.10 determines if a point is within a polygon [HA01].

$$test = \underline{inpolygon}(p_j^e, \mathbf{C}_i^e) \tag{6.10}$$

Algorithm 3: Existence Check (checkExistence(...))

Input : $\mathcal{T}^{\int}, \mathcal{T}^{\text{FoV}}$
Output: $\mathcal{T}^{\text{V2X?}}, \mathcal{T}^{\text{V2X+}}$

```
for each T_i^FoV ∈ T^FoV do
    for each T_j^∫ ∈ T^∫ do
        test ← Equation 6.11
    end
    if test then
        T^V2X+ ← T_i^FoV
    end
end
```

6.2.3.3 LoS Check

The second mandatory condition of the correlation module is to understand the non-detection of all V2X objects inside the camera FoV. An explanation for that is the presence of physical objects (e.g. other vehicles) which prevent the detection of these V2X object by the camera. The goal is to verify if there are objects occluding the Camera Line of Sight (LoS). Algorithm 4 verifies the intersection between two segments. The first segment (*segmentLoS*) extremities are the OBU location of the ego vehicle and of the V2X object. The second segment is iteratively each side of each physically confirmed object within the sensor FoV. The intersection algorithm outputs a binary value which confirms the intersection or the non intersection of the two segments.

In case of intersection, it means the tracked V2X object is in sensor NLoS. Therefore, the sensor cannot confirm its physical existence. Thus, the tracked V2X object is classified as an *unconfirmed track*.

However, if the V2X object is in sensor LoS. Then, the tracked V2X object should be detected by the camera. Thus, the tracked V2X object is classified as an *unconfirmed track*. Thus, it means the sensor correlation module is unable to classify the V2X track as malicious or non malicious.

$$test = \underline{intersect}(p_j^e, \mathbf{C}_i^e) \tag{6.11}$$

At this point, the V2X tracks classified as *unconfirmed* or *real* are sent to the fusion algorithm.

Algorithm 4: check LoS
(checkLoS (...))

Input : $\mathcal{T}^{\text{V2X?}}, \mathcal{T}^{\text{V2X+}}$
Output: $\mathcal{T}^{\text{V2X-}}, \mathcal{T}^{\text{V2X?}}$

for *each* $T_i^{V2X?} \in \mathcal{T}^{V2X?}$ **do**
  for *each* $T_j^{V2X+} \in \mathcal{T}^{V2X+}$ **do**
    $segmentLoS = [p_i^e p_e^e]$ ▷ Segment between ego OBU and object OBU
    for *each side* $\in \mathbf{S}_{(1,j)}$ **do**
      test= intersect($S_{(1,u,j)}, segmentLoS$)
    end
    if *test is true* **then**
      $\mathcal{T}^{\text{V2X-}} \leftarrow T_i^{\text{V2X?}}$
    else
      $T_i^{\text{V2X?}} \leftarrow$ remove
    end
  end
end

6.2.3.4 Revocation

The detected malicious tracks ($\mathcal{T}^{\text{V2X-}}$) are added to a black list and removed from the perception flow. The information contained in the black listed tracks are used to remove safety messages with the same station ID.

6.3 Conclusion

In this chapter, we present two modules of our FRPA to detect malicious V2X message.

The first module is a framework for ML based Failure Classifier. Our frameworks fits to any dataset (simulated or collected online) and any type of supervised ML algorithms (Random forest, KNN, and MLP).

The second module uses sensor measurements to detect V2X anomalies and prevent perception failures. Our module works with different type of sensors (radar and camera) and perception algorithms (Kalman Filter, Extended Kalman Filter, and Particle Filter). In the next chapter, we provide an evaluation of the described modules.

Chapter 7

Evaluation & Analysis

In this chapter, our contribution is two folds. First, we evaluated our ML based Failure Classifier. Secondly, we assessed and analyzed the performance of our V2X-Sensor Correlation Module. Finally, Section 7.3 concludes this chapter.

7.1 ML based Failure Classifier

This section presents the experiment setup, results for four failures classifiers and a discussion on the integration of our classifiers in our resilient perception architecture (GPA).

7.1.1 Experimentation Settings

We used an *Intel Core i5* with 3.3 GHz laptop. We implement four classifiers, named *MinMax*, *MLP*, *Adaboost*, and *Random Forest*, on Python using *numpy* [Dev13], *panda* [McK10], and *scikitlearn* [Ped+11] libraries. Each classifier was designed to classify the dimension data contained in a V2X safety message as malicious or non malicious. For this experiment, we used a dataset composed of three types of ITS-Station. A difficult task during experimentation is to define the settings for the *training* step to obtain a well-trained model with optimized hyper-parameters. For this purpose, we use the sci-kit class named *GridSearchCV* [Ped+11]. For *Cross-Validation*, we use the default *k*-fold cross-validation method with *k* set to 5 through empirical testing. For this case, the *training database* is split into 5 datasets where four sets serve as the *training dataset* and one as the *validation dataset*. Therefore, for a given set of hyper-parameters values, the *Cross-Validation* iterates this process five times until each one of the five datasets is used as a *validation dataset* for model validation. For each iteration, the *Cross-Validation* process computes a score defining the model performance of the *validation dataset*. Once the five iterations passed, a mean score value is computed that reflects the average performance of the model given a set of hyper-parameters. We choose *Matthews Correlation Coefficient* as a scoring metric. Then, the *GridSearch* method tests the next combination of hyper-parameters values according to Table 6.3. Once all the combinations tested, *GridSearchCV* outputs the hyper-parameters that got the highest mean score during the *Cross-Validation*. The process duration was 37 hours long just for *MLP*. At last, we test the trained model on the *testing dataset* and we obtain the results collected inTable 7.1.

7.1.2 Results Analysis

Table 7.1 displays the computed score for each performance metric as follows.

Method	TPR	FPR	TNR	FNR	ACC	F1	J	k	MCC
MinMax	**1**	0.452	0.547	**0.0**	0.942	**0.96**	0.54	0.67	0.71
MLP	0.88	**0.002**	**0.997**	0.11	0.982	0.92	0.87	0.91	0.91
AdaBoost	0.89	0.006	0.993	0.13	0.977	0.90	0.85	0.89	0.89
R Forest	0.90	**0.002**	**0.997**	0.09	**0.985**	0.94	**0.90**	**0.93**	**0.93**

TABLE 7.1: Classifiers Evaluation

7.1.2.1 Metrics Analysis

Another important task was to choose a suitable metric to assess the overall performance of a classifier model for the *training* and the *testing phases*. Based on Table 7.1, we study several metrics (Table 6.6).

The *F1 score* is not a reliable metric to give an overall performance for our framework. Indeed, the metric focuses only on the performance of the positive cases. For instance, the *MinMax* classifier that includes all the positive cases (non-misbehaving instances) has the highest score among all studied classifiers. However, we see that its *TNR* and *FPR* scores are the worst among classifiers. Thus, if we defined positive cases as misbehaving instances, the F1-score will have the lowest score among all classifiers. Therefore, in our context, it is a good metric performance for detecting positive cases only.

Overall, *Accuracy*, *Informedness*, *Cohen Kappa*, and *Matthews Correlation Coefficient* metrics converge towards the same ranking that is that Random Forest is the best classifier for the chosen dataset.

Accuracy is the most optimistic performance metric to assess the overall classifier performance. For instance, the *Naive* classifier has a high accuracy score (94%) because the classifier detected all the positive and half of the negatives cases (maximal *TP* and mean *TN* values). However, its *accuracy* value does not reflect its poor *FNR* score. *Informedness* is the most pessimistic performance metric to assess the overall classifier performance. Although *Cohen Kappa* and *Matthews Correlation Coefficient* scores are closely tied to *Informedness* score, both metrics give an in-between value of the overall performance of our classifier model. While *Cohen Kappa* metric fits a binary classification problem like in this chapter, *Matthews Correlation Coefficient* metric fits multiple classification problems. Therefore, it suits as an overall performance metric for our framework.

7.1.2.2 Classifiers Analysis

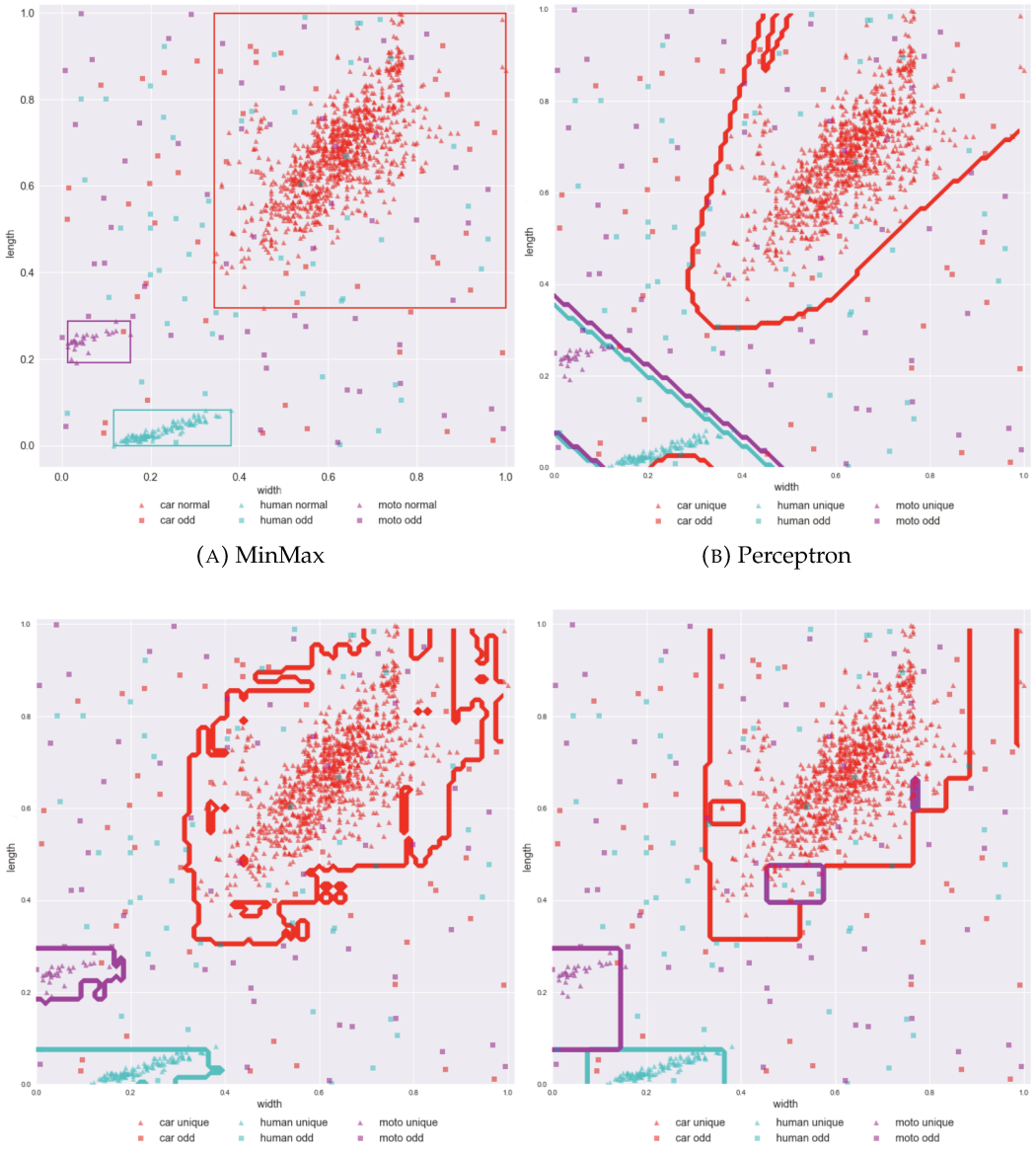
Each classifier has three colored areas of dimension plausibility where each area represents a type of ITS-Station. The inner area is the non-misbehaving domain and the outer area is the misbehaving area (Figure 7.1).

The *MinMax* method classifies all the non-misbehaving instances correctly. The reason behind this behavior is the design of *MinMax*. Indeed, the definition of each plausibility area is set on the lowest and highest values of width and length of its non-misbehaving dataset. Thus, the *MinMax* method reaches the maximal *TPR* value (100%) and the minimal *FNR* value (0%) (Table 7.1). However, Figure 7.1a shows that the plausibility boundaries do not fit each cluster well. Indeed, the error margin that is the area between the cluster and its plausibility boundaries is high. As a result, the *MinMax* method has a higher risk to classify a misbehaving instance as a non misbehaving one (rectangle point within the area). Thus, its *TNR* score is

mediocre (54%). Overall, the classifier performance has a MCC score of 71% which is below our requirement score (90%) defined in Section 6.1.5.3.

Each machine learning classifier (*MLP*, *AdaBoost*, and *Random Forest*) has a plausibility area that fits more the instances cluster of each ITS-Station type. The error margin is smaller than the margin of *MinMax* method (Figures 7.1b, 7.1c, and 7.1d). The risk to classify a misbehaving instance as non misbehaving is smaller. As a result, each classifier has a high *TNR* score (99%) which means the detection of misbehaving instances is accurate. However, by fitting the instances cluster, their *TPR* score is lower than the *TPR* score of the *MinMax* method. Therefore, our machine learning classifiers have a higher risk to classify a non-misbehaving instance as a misbehaving instance and a lower chance to classify it as a non-misbehaving instance. Overall, each machine learning classifier has a MCC score above or close to our requirement score.

To summarize, Random Forest with its default model is the best classifier in our context (*MCC*= 93%) despite having suboptimal hyper-parameter values. Therefore, our experiment highlights the importance of the classifier choice to detect. For instance, Figure 7.1d shows two moto plausibility areas within the car plausibility area. Therefore, a malicious CAV can pretend to be a moto.



(A) MinMax (B) Perceptron

(C) Random Forest (D) AdaBoost

FIGURE 7.1: Dimension Plausibility Boundaries for three ITS-Station types per tested Classifier

7.2 V2X-Sensor Correlation Module

This section presents our evaluation of our correlation module used to detect malicious V2X message. First, we will define the experimentation settings. Then, we present and analyze the obtained results.

7.2.1 Experimentation Settings

We defined the experimentation into two steps: the definition of the experimental setup and the driving scenario.

7.2.1.1 Experimental Setup

Table 7.2 describes our experimental setup. The core of the simulation platform was developed in Matlab (2019b). Besides, simulations were performed on a laptop that fits in terms of size in a vehicle.

Hardware	Processor	2,8 GHz Intel Core i5
	R.A.M.	8 Go 1600 MHz DDR3
Software	OS	macOS Mojave
	IDE	Matlab v2019b
	Libraries (Toolboxes)	Automated Driving [19a]
		Mapping [19b]
		Sensor Fusion & Tracking[19d]

TABLE 7.2: Definition of the simulation environment

7.2.1.2 Simulated Scenario

We evaluated our module considering a standardized safety scenario named *AEB Pedestrian Child Nearside* [ENC18]. Figure 7.2 depicts the simulation area ($50 \times 60m^2$)scenario. The scenario includes a pedestrian crossing the road (blue dots) behind a parked vehicle (yellow rectangle). Our ego vehicle (red rectangle) is in the middle of the road driving towards the pedestrian. Table 7.3 depicts the mobility behavior of each road user during the simulation.

		Starting Point	Ending Point	Color
Pedestrian	Position [X Y] (m) Velocity (km/h)	[32.5 -4] 0	[32.5 0] 5	Blue
Ego vehicle	Position [X Y] (m) Velocity (km/h)	[8.3 0] 30	[28.8 0] 30	Red
Parked vehicle	Position [X Y] (m) Velocity (km/h)	[27.4 -2.8] 0	[27.4 -2.8] 0	Yellow

TABLE 7.3: Scenario Settings

For the simulation, we used several sensors (GNSS receiver, camera, and radar) and several multiple objects trackers (MOT). For clarity sake, the reader may refer to Appendix A for details on sensors and trackers settings.

In this evaluation, we compare two types of architectures:

- one with a front Long Range Radar and V2X,
- one with a front Camera and V2X.

For evaluation fairness, we defined a bigger Field of View (FoV) for the radar. Indeed, each sensor must have the same area of detection. However, by default, a long-range radar model has a narrower FoV than a vision sensor model. This is why the default radar model was changed.

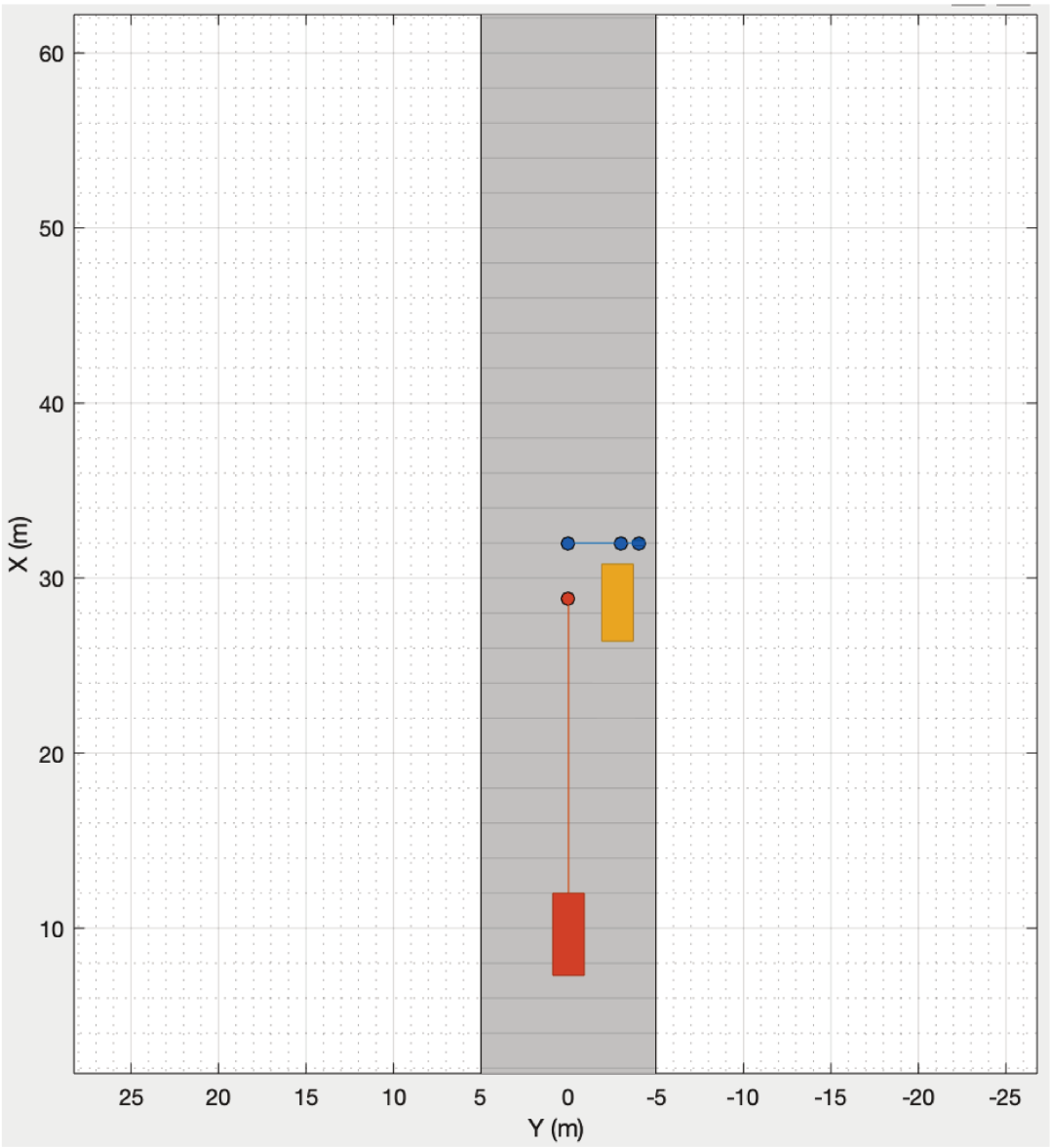


FIGURE 7.2: Safety Scenario: Automated Emergency Braking with a crossing child

7.2.2 Evaluation results

The evaluation has two goals:

- demonstrate the feasibility of our V2X-Sensor Correlation module,
- compare the radar and the camera performances.

For now, we make the following assumption: an absence of GPS error and constant emission frequency of safety messages. Then, we perform two evaluations with two values of emission frequency (1 Hz and 10 Hz), that are the standardized maximal and minimal values. Finally, we iterated 100 times the simulation to compensate for the randomness of the sensor model.

For each evaluation, we analyze three parameters:

- the latency of a perception cycle,
- the detection performance in the whole simulation area, and

- the detection performance inside the sensor FoV.

During this evaluation, we measure the run time of our sensor correlation module and of our two multiple objects trackers using the matlab script named *tic toc*. Then, we compute the run time of the perception life cycle that is equal to the sum of the two previously measured run times. Finally, we verify if the run time of our implementation is below latency requirements of 100ms (horizontal red line).

Also, we measure the detection performance of our sensor correlation module using four metrics for two areas of evaluation (the simulation area and the sensor field of view). The metrics are the mean real object detection score, ghost object detection score, uncertain object score, and detection accuracy score. The real object detection score is the ratio between the number of real object detected by our sensor correlation module and the number of real object in the evaluation area. We use the same computation for the ghost object detection score by using ghost object instead. The uncertain object score is equal to the number of object labeled as uncertain over the total number of object (ghost and real objects) in the evaluation area. Finally, the detection accuracy score is the inverse of the uncertainty metric.

7.2.2.1 Evaluation with a 1 Hz Emission Frequency of V2X Safety Message

Except at the first perception cycle, we observe that the latency of the perception cycle is below the threshold of 100 ms (Figure 7.3). Indeed, the multi-object tracker requires to perform some extra-processing during the initialization phase. Regarding the latency of our V2X-Sensor Correlation module (Figure 7.3b), we observe a 0,03 ms latency difference between the camera and the radar during the second and third perception iteration.

Then, we evaluated the detection performances of our V2X-Sensor Correlation module inside the whole simulation area. As seen in Figure 7.4a, the correlation module detects at least one of the two real objects (the parked vehicle and the crossing pedestrian) for both sensors. At the first iteration, our correlation module detects the parked vehicle but cannot detect the hidden pedestrian. However, the correlation module detects and classifies correctly the newly appeared pedestrian but cannot detect the parked vehicle (outside sensor FoV).

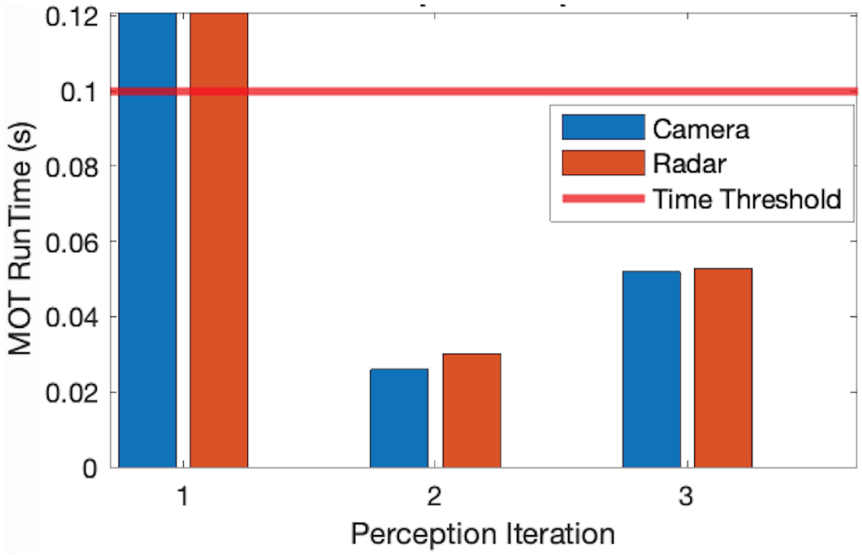
Moreover, the camera-based correlation module has a better detection score than the radar to detect V2X ghost objects (Figure 7.4b). For iterations 2 and 3, the ghost objects are outside sensor FoV. Therefore, the mechanism has no ghost object to detect.

Regarding detection uncertainty, the number of uncertain V2X objects increases because most of the real and ghost objects are moving outside the sensor FoV (Figure 7.4c). Therefore, our V2X-Sensor Correlation module is unable to detect and classify these objects (ghost or real). Thus, the detection accuracy decreases over time (Figure 7.4d).

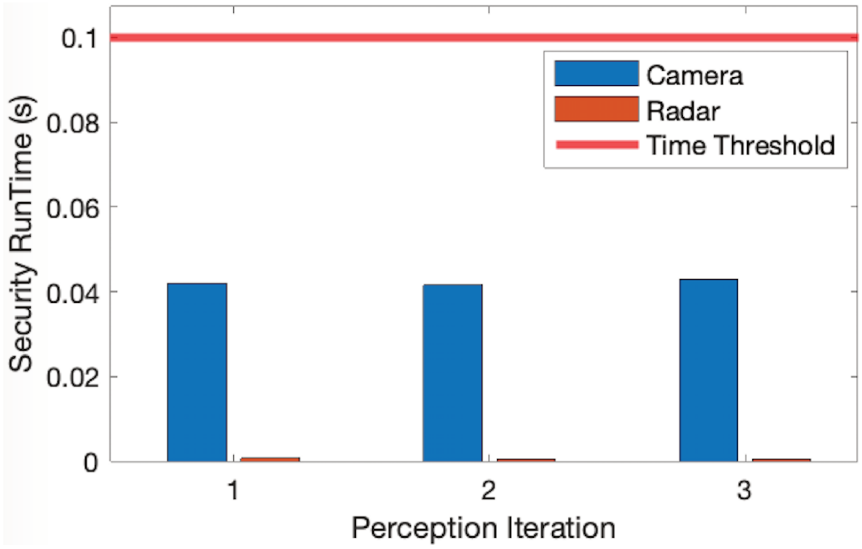
Then, we restrict the evaluation to the sensor FoV. During the first perception iteration, each sensor detects only one of the two real objects (Figure 7.5a). This result is caused by the pedestrian hidden behind the parked vehicle. For the second and last iteration, the pedestrian is the only real object remaining in the sensor FoV. Thus, both sensors can classify the V2X object correctly as real. In Figure 7.5b, 80% of the total number of ghost objects are detected at the first perception iteration. Overall, the detection score is good regarding the other security methods. However, we believed that such score can be improved by analyzing each played scenario instance. Regarding the uncertainty score (Figure 7.5c), there is two factors that can explain

such score. The first one is the hidden pedestrian that cannot be classified as a real object by our correlation module. Indeed, the sensor LoS is occluded by the parked vehicle. Thus, the correlation is unable to detect and classify the pedestrian at this time of the simulation. Besides, the second cause is the absence of sensor detection that leads to a mis classification of the parked vehicle. Sometimes, the camera may not detect the parked vehicle due to harsh operational conditions (e.g., sunlight). Therefore, the consequence of this undetection is the classification of the parked vehicle as a ghost vehicle. Thus, this failed detection raises the uncertainty score from 0.25 (perfect detection) to 0.4. Now, if we compare the radar and the camera, then the radar has a higher uncertainty rate than the camera. We observed this result because the radar track needs several radar detection before being confirmed and used. Therefore, overall, the parked vehicle requires more time to be tracked by the radar. Then for iterations number 2 and 3, the pedestrian is in sensor LoS and, thus, correctly detected. Besides, the V2X attacks are not located inside sensor FoV. Thus, the uncertainty score is almost null. Accordingly, the accuracy score increases over time as seen in Figure 7.5d.

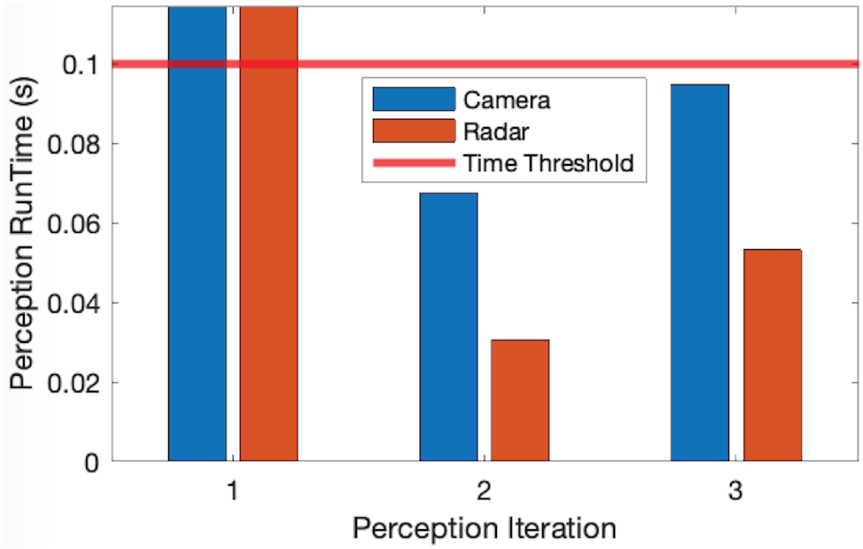
As shown, the detection system can detect ghost and real objects during the first perception iteration. However, the emission frequency of V2X safety messages is too low to collect pieces of evidence to prove the existence of the V2X object. Thus, we decide to perform the same evaluation with a higher frequency (10 Hz).



(A) Multiple Objects Tracker

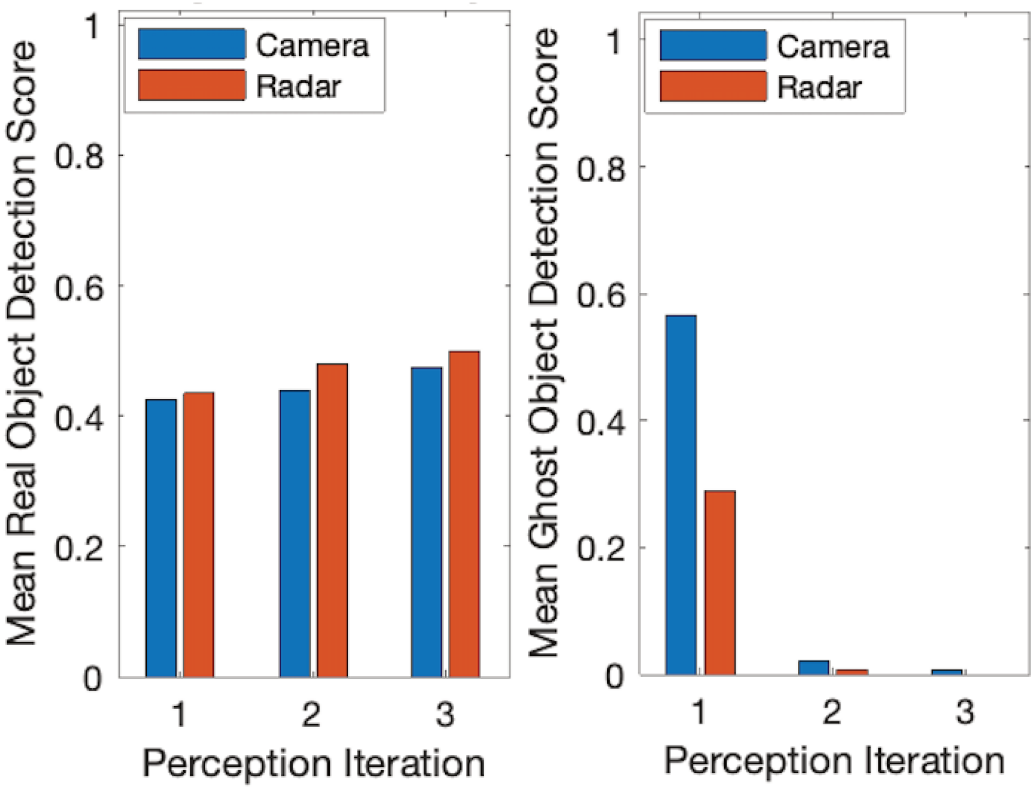


(B) V2X-Sensor Correlation Module

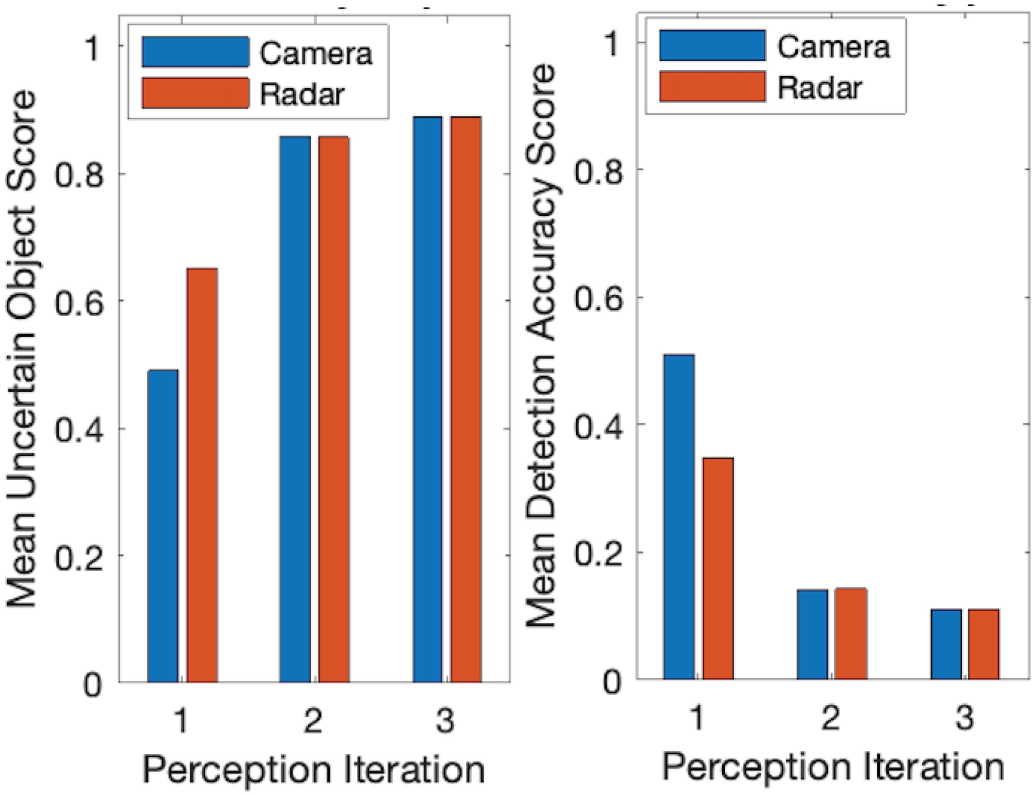


(C) Perception

FIGURE 7.3: Latency Evaluation (1 Hz)

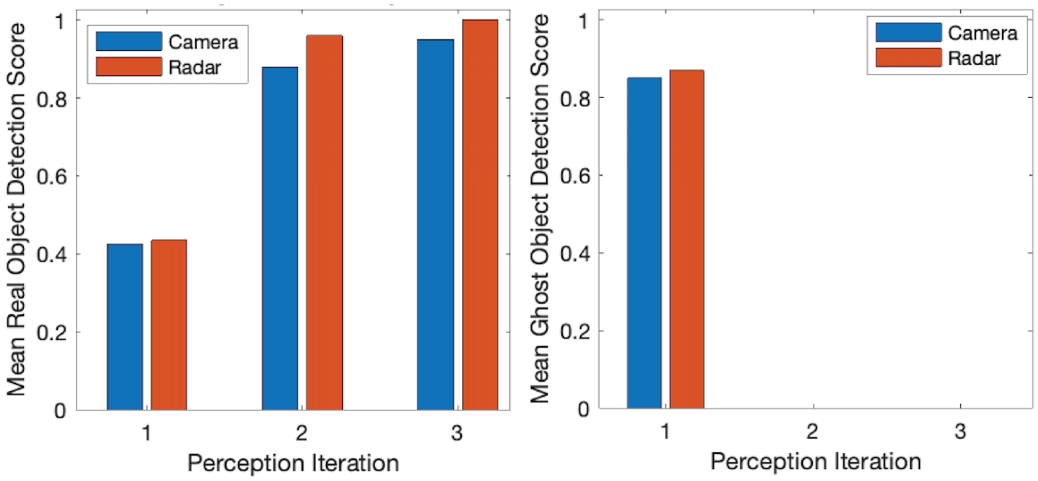


(A) Real Object Detection Rate (B) Ghost Object Detection Rate

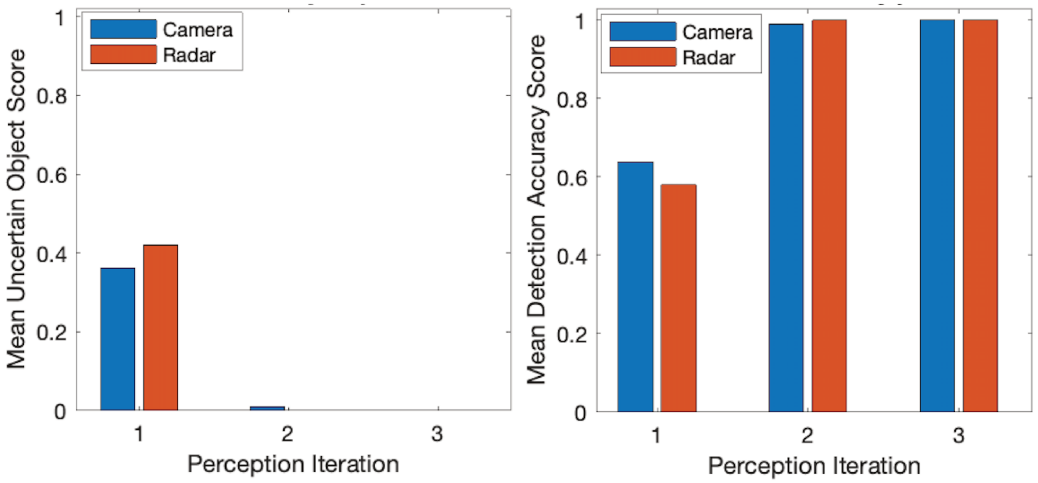


(C) Detection Uncertainty (D) Detection Accuracy

FIGURE 7.4: Evaluation: Simulation Area (1 Hz)



(A) Real Object Detection Rate (B) Ghost Object Detection Rate



(C) Detection Uncertainty (D) Detection Accuracy

FIGURE 7.5: Evaluation: Sensor FoV (1 Hz)

7.2.2.2 Evaluation with a 10 Hz Emission Frequency of V2X Safety Message

At 10 Hz (Figure 7.6), the perception latency has similar results than the ones observed at 1 Hz .

In Figure 7.7a, we observe an interesting result. From iteration 2 to 10, the radar detected two real objects. It means the radar detected the NLoS pedestrian. A similar result is observed from iteration 13 to 19.

Regarding Figures 7.7b, 7.7c, and 7.7d, the curves are similar to the one observed at 1 Hz.

In the sensor FoV, we observe the following results. Firstly, the radar is efficient in detecting NLoS Object (Figure 7.7a). However, the camera is unable to detect the NLoS pedestrian (iteration 1 to 19).

Besides, we observe a second interesting result (Figure 7.7b). From iteration 14 to 19, the camera detects more ghost vehicles than expected (the score is higher than 1). A limitation of our correlation module causes this result. Indeed, our correlation module classified real objects as uncertain because their OBU is outside the sensor FoV. Therefore, the NLoS object is classified as a ghost when an occluding real object OBU is outside the sensor FoV. Besides, Figures 7.8c and 7.8d showed the incapacity of the camera to detect the NLoS pedestrian.

To resume, a V2X safety message emitted at a high frequency increases the perception and detection capabilities of our sensor correlation module. Moreover, we observe that the radar is an efficient sensor to detect objects in NLoS.

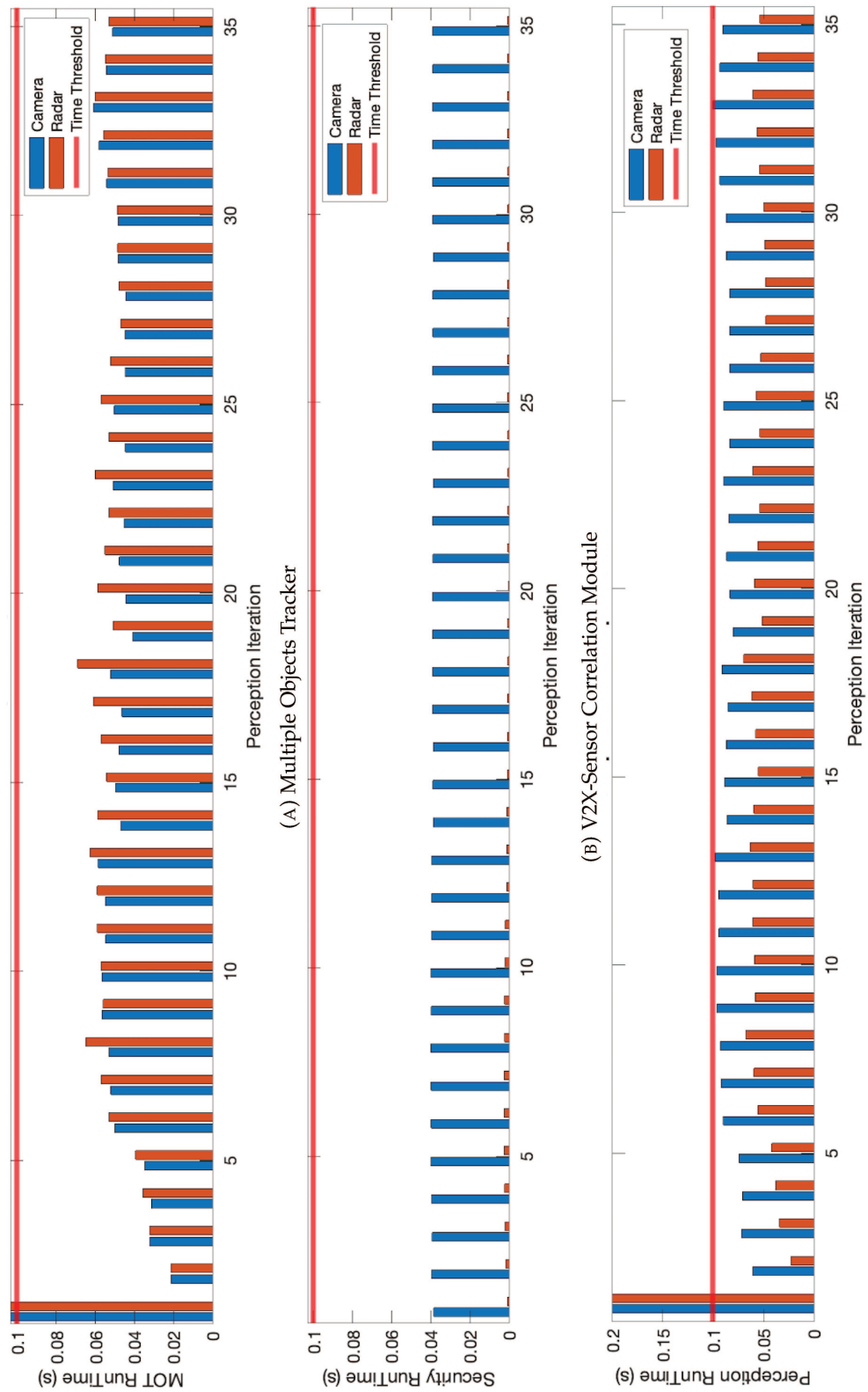
7.3 Conclusion

In this chapter, we evaluated two modules of our **FRPA** to detect perception anomalies: a ML based Failure Classifier and a V2X-Sensor correlation module.

In this first part of the chapter, we evaluated our ML classifier framework for different types of objects (pedestrian and vehicles). We implemented 4 classifiers such as *MinMax*, *MLP*, *Adaboost*, and *Random Forest*. Then, we evaluated these classifiers with 5 performance metrics. The first result of this evaluation is the choice of MCC as our performance metric instead of the commonly used accuracy metric. Indeed, MCC is the most balanced metric among all tested metrics. The second result is that Random Forest is the best algorithm to classify dimension data. Indeed, this classifier has a the best MCC score (93%) among all classifiers. Also, this unoptimized classifier has better results than the other optimized classifiers used in our evaluation. As a future work, it will be relevant to extend our evaluation to other types of data such as the emitter position, speed, and heading.

Then, we evaluated our V2X-Sensor correlation module with two types of sensors and with two different emission frequencies of a V2X safety message. First, we showed that our V2X-Sensor correlation module detects NLoS objects thanks to the combination of radar and V2X data. We observed that our V2X-Sensor correlation module has better detection capabilities at high frequency (10 Hz). As a consequence, our V2X-Sensor correlation module force the attacker to generate ghost outside sensor FoV or minimize the number of emitted messages. Overall, our V2X-sensor correlation module has a detection score close to 90%.

As further analysis, we will vary the GNSS noise and study its impact on the detection performance of our V2X-sensor correlation module. Besides, other parameters can include the traffic density and its impacts on the perception latency. Indeed, if we increase the number of CAV in the scenario, then the perception cycle must store and process more CAV data that may result in a DDoS of the perception system.

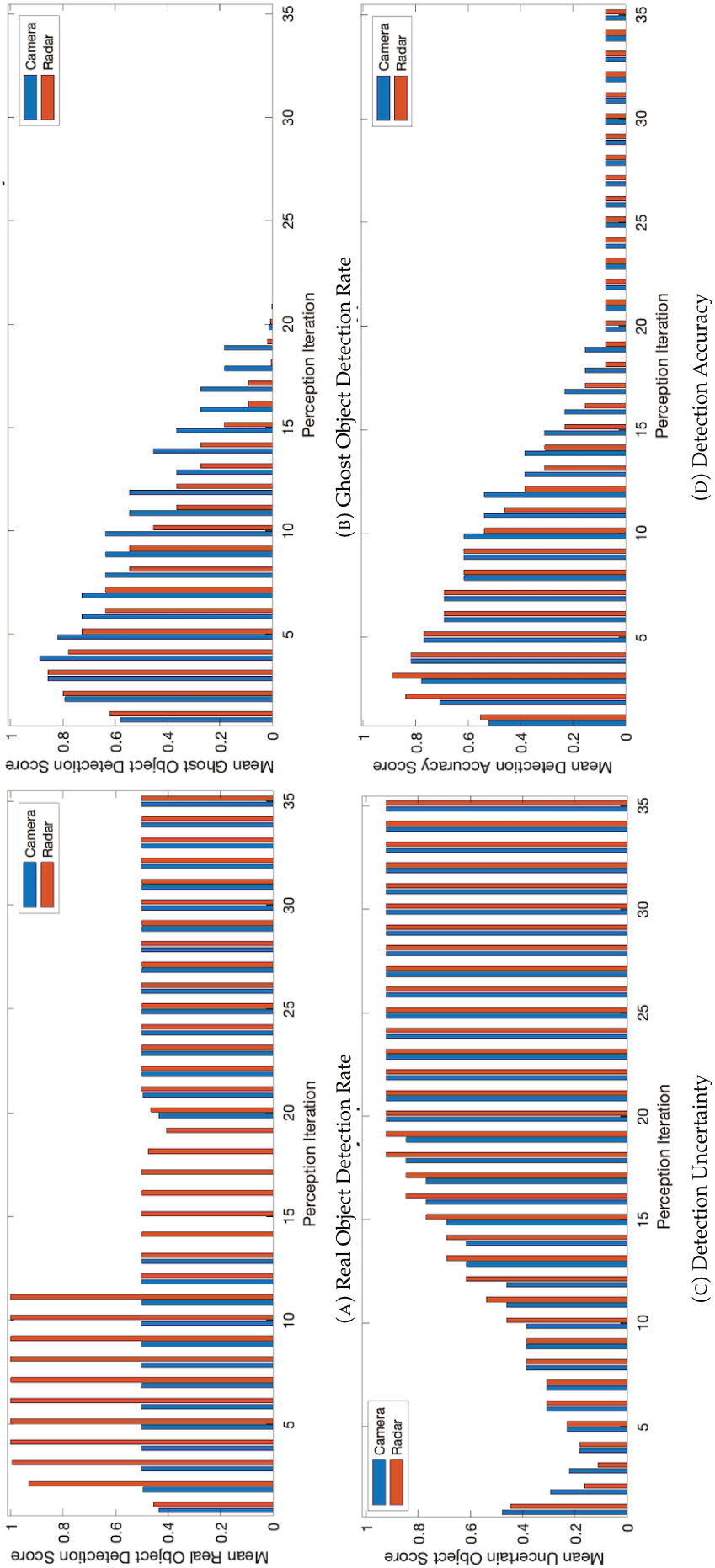


(A) Multiple Objects Tracker

(B) V2X-Sensor Correlation Module

(C) Perception

FIGURE 7.6: Latency Evaluation (10 Hz)



(A) Real Object Detection Rate (B) Ghost Object Detection Rate (C) Detection Uncertainty (D) Detection Accuracy

FIGURE 7.7: Evaluation: Simulation Area (100 ms)

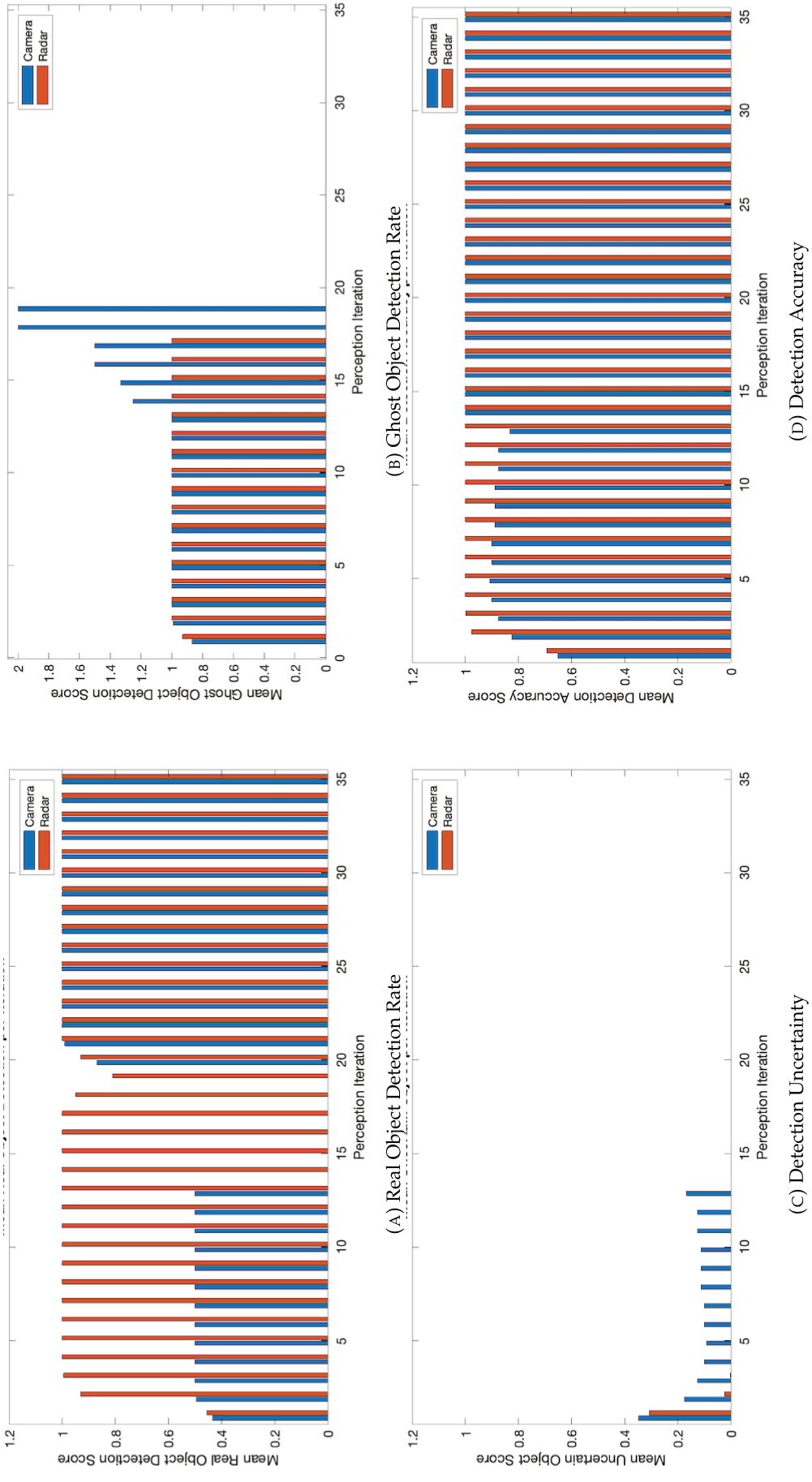


FIGURE 7.8: Evaluation: Sensor FoV (100 ms)

Chapter 8

Conclusion & Perspectives

The future of autonomous driving depends on our ability to prevent perception failures for any automated and connected vehicles. The integration of V2X communication in the perception system is very interesting. Nevertheless, it implies the presence of security attacks targeting all connected and automated vehicles. To face these attacks, it is mandatory to design an efficient failures-resilient perception system that fits any connected and automated vehicles. In our thesis, we propose a generic perception architecture (**GPA**). Based on this architecture, we defined a failures-resilient pseudo algorithm (**FRPA**) that prevents the presence of perception failures in any connected and automated vehicle.

To reach this solution, we focus on failures caused by V2X attacks. In the absence of TARA methods to identify the most threatening attacks, we proposed a threat analysis and risk assessment method named **SARA**. Our method helped us to assess the security risk level of attacks. Besides, we proposed an attacker model to identify where and how an attack may target our GPA. After identifying and assessing perception attacks, we investigate more in detail two critical perception security modules: anomalies classification and correlation. We proposed then a ML-based Failure Classification and a V2X-Sensor Correlation Modules. Then, we evaluated our contributions.

Finally, we summarize the thesis by outlining the contributions above and by discussing future research directions.

8.1 Conclusion

First of all, we analyzed works related to CAV failures in Chapter 2. Also, we compared TARA methods, security modules, and simulators. As a conclusion, we pointed out the need for a generic architecture to propose a generic proposal that is failures resilient.

In Chapter 3, we proposed a *Generic Perception Architecture* (**GPA**). Our architecture is capable of supporting several types of physical and logical CAV architectures, including vehicles with automation level 5. Alongside our **GPA**, we defined an associated *Perception Data Model* (**PDM**). The latter highlights the specific and common data among perception sources and their location in the perception lifecycle. Next, we designed a failure resilient pseudo algorithm (**FRPA**) to prevent failures. **FRPA** design takes into account our **PDM**. Indeed, each **FRPA** module is based on data properties (raw/processed, specificity/common, discrete).

We wanted to identify the attacks with the highest level of risk for our **GPA**. Knowing these attacks, we could prioritize the attacks to be solved. As no TARA methods exist for CAV, we proposed a new method named **SARA** in Chapter 4. Our method includes new threat analysis tools: a new attack to asset mapping as well

as a new attacker model. Besides, SARA includes new risk assessment metrics such as the degree of auto-pilot controllability and the degree of CAV observation. To prove its practicability, we applied our method on two use cases named *Comfortable Emergency Brake Failure* and *Vehicle Tracking*.

After identifying the attacks, we investigated how and when the perception attacks may occur in a driving scenario. Thus, in chapter 5, we proposed a new attacker model. Accordingly, we extended the security goals model from **SARA** and compared it with state of the art models to highlight our contribution.

After identifying the most threatening attacks, we proposed two modules: a framework for ML based failure and a V2X-Sensor Correlation Module. The first contribution detects abnormal data in a V2X message using supervised machine learningmethods. The second contribution detects abnormal data in a V2X message using a camera and a radar. We showed how a perception source could self-detect abnormal data to prevent failures. We also showed how to use another perception source to detect abnormal data.

In Chapter 7, we evaluated our two contributions. We tested our first contribution with abnormal classification data in a V2X message. During this evaluation, we tested optimized and unoptimized several ML models and compared them with a threshold-based model. Besides, we analyzed several metrics to assess the performance of a classifier model with an unbalanced dataset distribution (more normal data than abnormal data). Then, we tested our second contribution with abnormal position data in a V2X message. During this evaluation, we tested different frequencies of V2X message reception and different sensor types (radar and camera). Our contribution is highly accurate, but its performance is highly affected by the driving scenario (NLoS and frequency of the message reception). Besides, we showed that a radar outperforms a Camera in NLoS scenarios. Finally, our second contribution forces the attacker to avoid the sensor LoS and limits the frequency of message emission.

Overall, the thesis contributions resulted in the submission of ten french patents for Groupe PSA paving the way towards new research perspectives. Our contributions include the detection of sensor failures using V2X data. Others contributions include the computation of a machine learning features by using different methods. For instance, we propose to compute the distance between the V2X message emitter and receiver by using ToA, euclidean distance, RSSI. Then, we use the three distance values as inputs of our ML algorithm. Thus, the ML module verifies if the three computed distance values are similar meaning the emitter position in the V2X message is plausible. In addition to anomaly detection, we provided three secured cooperative ADAS applications such as autonomous railway crossings, autonomous boarding in a cargo ship, and autonomous navigation on an airport tarmac. Finally, two patents are related to our sensor correlation module.

8.2 Perspectives

As part of the thesis perspectives, it will be relevant to investigate the following improvements to our contributions in the short and long-terms.

8.2.1 Short-term perspectives

This section presents a potential extension for each thesis contribution.

8.2.1.1 SARA

We propose two extensions to SARA. Unlike existing safety risk assessment methods, SARA assumes that an attack may occur on a CAV under perfect road conditions. Thus, a first extension could be to extend it with safety road conditions. As a second future work, it will be relevant to link our attacker model to the attack to asset mapping to take into account the attacker location and capabilities. For instance, an attacker like a *communication deceiver* targets the vehicle through V2X communication. Being defined as an insider, he possess cryptography materials and could emit and sign message with incorrect data. Thus, this attacker category can be defined as an attacker with high capabilities in the SARA method. Therefore, we believe we can define a systemic approach linking our attacker model and the attack to asset mapping defined in SARA.

8.2.1.2 Framework for ML based Failure Classifier

Currently, the absence of complete and approved attacks datasets limits the proposal of new machine learning methods designed to detect and classify abnormal perception data. A perspective is to simulate these attacks and create peer-reviewed datasets. It will be also interesting to pursue other tests such as testing other classification algorithms or widening the range of hyper-parameters values.

8.2.1.3 V2X-Sensor Correlation

It will be relevant to test our module under harsh road conditions such as extreme weather conditions, urban environments, and high traffic density. Therefore, the implementation of these conditions will help to pursue extensive analysis to evaluate the resiliency and the robustness of our correlation module.

8.2.1.4 Failures Resilient Perception Algorithm Implementation

In our work, we implemented two FRPA modules using Matlab 2019b. To evaluate FRPA's resiliency and robustness, we must implement the remaining modules such as the cryptography, the physical, temporal and, fusion modules. In addition to FRPA modules, the implementation of a V2X model that accounts of signal attenuation, collision, and obstruction is mandatory. Finally, a weather model that alters the physical signal of sensors and V2X communication could be a valuable asset to evaluate FRPA resiliency under harsh environmental conditions.

8.2.2 Long term perspectives

This section highlights three long term research future works:

- **Privacy by design for secured automotive and cooperative perception**

Current perception trackers assume the absence of a privacy module. Indeed, cooperative perception systems rely on a unique lifetime station identifier to track surrounding connected and automated vehicles. However, vehicles have several identifiers changing over time to avoid being tracked. Therefore, an important open issue is to build a cooperative system scheme that identifies all surrounding vehicles within the perception system range. The first approach could be to investigate the data association mechanisms used in the local perception system. Indeed, a non cooperative perception must track a surrounding detected object without knowing its

identity while identification data contained in V2X messages could be used straightforward y to identify an object. The bias contained in mobility data may lead to an incorrect identification. Therefore, the conception of a pseudonym resilient tracker for V2X communication could be a new contribution to secure cooperative perception system. An other open issue is to define a privacy strategy to mislead the tracker. For instance, as mentioned in Chapter 3, a global voyeur can use the tracker to track the whole driving history of a target. A privacy mechanism could be defined to avoid the voyeur to track its target without affecting the efficiency of the local V2X tracker.

- **Distributed Geo-Sensing**

To enhance failure perception, an interesting approach is to use autonomous vehicles as distributed sensors providing collected information to other entities belonging to an intelligent transportation system. Indeed, a float of autonomous vehicles can drive and collect several data such as images, dot clouds, V2X information about a whole city.

- **Towards an open source test platform for secured CAV**

Currently, we find that the evaluation of algorithms for connected and autonomous vehicles is costly, incomplete, unoptimized, or proprietary. Thus, it is difficult to evaluate correctly security or safety solutions without an open-source dedicated for modeling and testing. The development fo such a research initiative will diminish the time spent in development and improve research proposals and industrial developments. Besides, this platform may help to provide realistic datasets.

- **Data-driven for perception optimization**

The machine learning based perception needs to train the model by trying different kinds of combinations of hyper-parameters. The set of such hyper-parameters can be different from time to time, and from scenarios to scenarios. As a future work, it will be relevant to study context-aware methods to selectively utilize the suitable hyper-parameters based on environment's context, in a data-driven manner. For example, at an intersection with a high density of pedestrians, the perception algorithm settings should be more sensitive to false negative faults (actual real pedestrian perceived as non-existing). while with roads where there is few pedestrians, the perception algorithm settings should be more sensitive to the possibility of false positive faults (ghosts).

Appendix A

Simulation Model & Settings

A.1 Sensors

This section defines the mutual and exclusif sensor properties used in Section 6.2

A.1.1 Common

Property \ Type	Camera		Radar					
Identifier	1	2	3	4	5	6	7	8
Location (m)	$\begin{bmatrix}2.1\\0\end{bmatrix}$	$\begin{bmatrix}0.56\\0\end{bmatrix}$	$\begin{bmatrix}3.7\\0\end{bmatrix}$	$\begin{bmatrix}-1\\0\end{bmatrix}$	$\begin{bmatrix}0\\0.9\end{bmatrix}$	$\begin{bmatrix}0\\-0.9\end{bmatrix}$	$\begin{bmatrix}2.8\\0.9\end{bmatrix}$	$\begin{bmatrix}2.8\\-0.9\end{bmatrix}$
Maximum detection range (m)	150		174		30			
Height above ground plane (m)	1.1		0.2					
Yaw (°)	0	180	0	180	120	-120	60	-60
Pitch (°)	0							
Roll (°)								
Time between sensor updates (s)	0.1							
Add noise to measurements	Enable							
Maximum number of detections	50							
Coordinate system of detections	Ego Cartesian							
Actor profiles	Definition of Scenario Actors							
Sensor Specific	Table A.2		Table A.3					

TABLE A.1: Definition & Settings of Sensor Models

A.1.2 Camera

Sensor Identifier	1 | 2
Bounding box accuracy (pixels)	5
Noise intensity to filter position & velocity (m/s^2)	
Angular field of view of vision sensor ($\circ$)	43,60
Maximum detectable object speed (m/s)	50
Maximum allowed occlusion of an object ()	0.5
Probability of detection	0.9
Minimum image size of detectable object (pixels)	$\begin{bmatrix}15\\15\end{bmatrix}$
Number of false detections per image ()	0.1
Focal length (pixels)	$\begin{bmatrix}800\\800\end{bmatrix}$
Optical Camera center (pixels)	$\begin{bmatrix}320\\240\end{bmatrix}$
Image size produced by camera (pixels)	$\begin{bmatrix}480\\640\end{bmatrix}$
Radial distortion coefficients()	$\begin{bmatrix}0\\0\end{bmatrix}$
Tangential distortion coefficients()	
Skew angle of the camera axes ()	0

TABLE A.2: Properties specific to a Camera

A.1.3 Radar

Property \ Sensor Identifier	3	4	5	6	7	8
Azimuth Elevation fields of view (○)	$\begin{bmatrix}20\\5\end{bmatrix}$		$\begin{bmatrix}90\\5\end{bmatrix}$			
Min. Max. detection range rates (m)	$\begin{bmatrix}-100\\100\end{bmatrix}$					
P(detection) of a target ()	0.9					
False alarm rate ()	1e-6					
Ref. range given P(detection) (m)	100		50			
Ref. X-section given P(detection)	0					
Radar loop gain	RLG(x)					
Azimuthal resolution (○)	4		10			
Elevation resolution (○)	10					
Range resolution (m)	2.5					
Range rate resolution (m/s)	0.5					
Azimuth bias fraction () Elevation bias fraction ()	0.1					
Range bias fraction () Range rate bias fraction	0.05					
Enable to measure elevation Enable to measure range rate	Disable					
Create false alarm detections Enable line-of-sight occlusion	Enable					

TABLE A.3: Specific Radar Properties

A.1.4 GNSS

Property	Value (default)
Sensor Identifier	9
Update rate of receiver (Hz)	1
Origin of local NED reference frame	[0 0 0]
Horizontal position accuracy (m)	1.6
Vertical position accuracy (m)	3
Velocity accuracy (m/s)	0.1
Global position noise decay factor	0.999
Random number source	Global stream
Initial seed	67

TABLE A.4: Model of GNSS Receiver

A.2 Multi-Object Tracker

A.2.1 Multiple Objects Tracker (*MOT*)

Parameter	Value
Kalman filter initialization function	initcvkf(...)
Assignment Threshold	55
Confirmation parameters for track creation	$\begin{bmatrix}4\\5\end{bmatrix}$
Coasting threshold for track deletion	5
Maximum number of tracks	200
Maximum number of sensors	20
Enable cost matrix input	Disable

TABLE A.5: Parameters of Multi-Object Tracker

A.2.2 Tracker

Meaning & Variable	Model
Detection ($\vec{\mathbf{z}}_k$)	$\begin{bmatrix}x & y & v_x & v_y\end{bmatrix}'$
Transition Matrix ($\mathbf{F}$)	$\begin{bmatrix}1 & dt & 0 & 0\\0 & 1 & 0 & 0\\0 & 0 & 1 & dt\\0 & 0 & 0 & 1\end{bmatrix}$
Measurement Model ($\mathbf{H}$)	$\begin{bmatrix}1 & 0 & 0 & 0\\0 & 0 & 1 & 0\\0 & 1 & 0 & 0\\0 & 0 & 0 & 1\end{bmatrix}$
State ($\vec{\mathbf{x}}$)	$\mathbf{H}' \times \mathbf{z}_k$
Measurement Noise ($\vec{\mathbf{R}}_k$)	detection.MeasurementNoise
State Covariance ($\mathbf{P}_k$)	$\mathbf{H}' \times \vec{\mathbf{w}}_k \times \mathbf{H}$
Process noise Covariance ($\vec{\mathbf{Q}}_k$)	1

TABLE A.6: Filter Parameters

Appendix B

Use Cases for Perception Failure

B.1 Object Classification Failure

: We identified four scenarios related to the use case of object classification failure.

1. A CAV uses a revolving light to gain *road priority*
   - Camera classifies obstacle as *special vehicle*
   - V2X message classifies obstacle as "civilian vehicle"
2. Misbehaving CAV claims fake rights (e.g., police car with road priority)
   - V2X message from ITS-V classifies it as "special vehicle"
   - Camera classifies ITS-V as "civilian vehicle"
3. Altered Roadsign (90km/h limitation) is altered into a 10 km/h roadsign
   - Camera classifies obstacle as "10km/h roadsign"
   - V2X, roadmap classifies obstacle as "90 km/h roadsign"
4. CAV is misclassified by a Sensor.
   - physical alteration of the connected vehicle shape
   - faulty sensor classifier

B.2 Object Detection Failure

In this use case, a CAV is undetected by a sensor due to several cause of failures. We identified 5 categories of failures scenario.

1. destruction/alteration/removal of:
   - the sensor hardware
   - the sensor data (software, measurements, and settings)
   - the C-ITS station or its components (traffic light, brake light, traffic sign)
2. signal spoofing
   - sensor spoofing
   - GPS spoofing (malicious safety message)
3. signal attenuation/ reflection

- adversarial material on the vehicle (signal absorption or reflection)
- harsh weather conditions (Blizzard, Rain)
- replay attack

4. signal obstruction
    - The CAV and the ego vehicle are sensor jammed.
    - road topologies (hill, hairpin turns, superposed roads)
    - road obstacles (buildings and vehicles)
    - harsh weather conditions (blizzard, sand storm, mist or tropical rain).
5. Combination of malicious V2X data and a cause of sensor failure

B.3 Self-localization Failure

We identified a single scenario for this use case. In a tunnel, a CAV emits an incorrect position to surrounding CAVs.

- GNSS Data are unavailable
- IMU is faulty

Publications

[Mon+18a] Jean-Philippe Monteuuis et al. “"My autonomous car is an elephant": A Machine Learning based Detector for Implausible Dimension”. In: *3rd International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE. Shanghai, China, 2018, pp. 1–8.

[Mon+18b] Jean-Philippe Monteuuis et al. “Attacker model for Connected and Automated Vehicles”. In: *2nd Computer Science in Cars Symposium - Future Challenges in Artificial Intelligence & Security for Autonomous Vehicles*. ACM. Munich, Germany, 2018, pp. 3–14.

[Mon+18c] Jean-Philippe Monteuuis et al. “SARA: Security Automotive Risk Analysis Method”. In: *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*. ACM. Incheon, Republic of Korea, 2018, pp. 3–14.

[Mon+19] Jean-Philippe Monteuuis et al. “Sensor-V2X Correlation for Misbehavior Detection”. to be submitted to a journal. 2019.

Patents

[Mon+a] Jean-Philippe Monteuuis et al. “Détection erroné d’un faux à l’aide du système V2X”. Patent 2019P01099 FR (FR).

[Mon+b] Jean-Philippe Monteuuis et al. “Intégration des dimensions d’un véhicule dans un certificat d’authentification”. Patent 2019P01024 FR (FR).

[Mon+c] Jean-Philippe Monteuuis et al. “Procédé d’aide à la conduite fiable pour l’embarquement et débarquement d’un véhicule cargo communicant sécurisé”. Patent 2019P00380 FR (FR).

[Mon+d] Jean-Philippe Monteuuis et al. “Procédé d’aide à la conduite pour vérifier l’intégrité d’un panneau de signalisation perçu par caméra”. Patent 2018P02037 FR (FR).

[Mon+e] Jean-Philippe Monteuuis et al. “Procédé d’aide à la conduite sur un aéroport/port à l’aide d’une infrastructure communicante”. Patent 2019P00381 FR (FR).

[Mon+f] Jean-Philippe Monteuuis et al. “Procédé de détection de défaillances et limitations des capteurs à l’aide des communications Car2X”. Patent 2017P01574 FR (FR).

[Mon+g] Jean-Philippe Monteuuis et al. “Procédé de détection de la plausibilité des données V2X basé sur l’agrégation des techniques de mesures du signal radio”. Patent 2019P00379 FR (FR).

[Mon+h] Jean-Philippe Monteuuis et al. “Procédé et système pour traiter un message transmis à un véhicule automobile par une entité communicante distante”. Patent 2019P00531 FR (FR).

[Mon+i] Jean-Philippe Monteuuis et al. “Procédé sécurisé d’aide à une conduite fiable et sûre pour le franchissement d’un passage à niveau communicant”. Patent 2018P00608 FR (FR).

[Mon+j] Jean-Philippe Monteuuis et al. “Procédédé de gestion d’un message transmis par une station émettrice d’un système de transport intelligent”. Patent 2018P01119 FR (FR).

Award

[Mon18] Jean-Philippe Monteuuis. *Resilience by design & failures forecasting for a connected autonomous vehicle*. 2nd Prize, My Thesis in 3 minutes, European CyberWeek. Rennes, France, Nov. 2018.

Bibliography

[You50] William J Youden. "Index for rating diagnostic tests". In: *Cancer* 3.1 (1950), pp. 32–35.

[Coh60] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales". In: *Educational and Psychological Measurement* 20.1 (1960), p. 37. URL: http://epm.sagepub.com/cgi/reprint/20/1/37.

[Mat75] Brian W Matthews. "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". In: *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405.2 (1975), pp. 442–451.

[Fel94] Martin Fellendorf. "VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority". In: *64th Institute of Transportation Engineers Annual Meeting*. Vol. 32. Springer. 1994, pp. 1–9.

[Kra98] Stefan Krauß. "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics". PhD thesis. Dt. Zentrum für Luft-und Raumfahrt eV, Abt. Unternehmensorganisation und ..., 1998.

[ZBG98] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. "GloMoSim: a library for parallel simulation of large-scale wireless networks". In: *Proceedings. Twelfth Workshop on Parallel and Distributed Simulation PADS'98 (Cat. No. 98TB100233)*. IEEE. 1998, pp. 154–161.

[BP99] Samuel Blackman and Robert Popoli. "Design and analysis of modern tracking systems(Book)". In: *Norwood, MA: Artech House, 1999.* (1999).

[HA01] Kai Hormann and Alexander Agathos. "The point in polygon problem for arbitrary polygons". In: *Comput. Geom. Theory Appl.* 20.3 (2001), pp. 131–144. ISSN: 0925-7721. DOI: http://dx.doi.org/10.1016/S0925-7721(01)00012-8.

[HL02] Michael Howard and David LeBlanc. *The STRIDE Threat Model. From the Book Writing Secure Code*. 2002.

[LH02] David LeBlanc and Michael Howard. *Writing secure code*. Pearson Education, 2002.

[Fir04] Donald Firesmith. "Specifying reusable security requirements." In: *Journal of Object Technology* 3.1 (2004), pp. 61–75.

[GGS04] Philippe Golle, Dan Greene, and Jessica Staddon. "Detecting and correcting malicious data in VANETs". In: *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*. ACM. 2004, pp. 29–37.

[Kae+04] Nico Kaempchen et al. "Sensor fusion for multiple automotive active safety and comfort applications". In: *Advanced Microsystems for Automotive Applications 2004*. Springer, 2004, pp. 137–163.

[TYI04] Hiroomi Takizawa, Kenichi Yamada, and Toshio Ito. "Vehicles detection using sensor fusion". In: *IEEE Intelligent Vehicles Symposium, 2004*. IEEE. 2004, pp. 238–243.

[DW05] Michael Darms and Hermann Winner. "A modular system architecture for sensor data processing of ADAS applications". In: *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* IEEE. 2005, pp. 729–734.

[KBD05] Nico Kaempchen, Matthias Buehler, and Klaus Dietmayer. "Feature-level fusion for free-form object tracking using laserscanner and video". In: *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* IEEE. 2005, pp. 453–458.

[NF05] Karl Naab and Andreas Frey. "Efficient in-vehicle sensor and signal processing architectures for driver assistance and active safety systems". In: *12th World Congress on Intelligent Transport SystemsITS AmericaITS JapanERTICO*. 2005.

[Gru+06] Dominique Gruyer et al. "SiVIC and RTMaps, interconnected platforms for the conception and the evaluation of driving assistance systems". In: *Proceedings of the its world congress*. Vol. 10. 2006.

[Her+06] Shawn Hernan et al. "Threat modeling-uncover security design flaws using the stride approach". In: *MSDN Magazine-Louisville* (2006), pp. 68–75.

[Mah+06] Mirko Mahlisch et al. "Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection". In: *2006 IEEE Intelligent Vehicles Symposium*. IEEE. 2006, pp. 424–429.

[Flo+07] Nikos Floudas et al. "High level sensor data fusion approaches for object recognition in road environment". In: *2007 IEEE Intelligent Vehicles Symposium*. IEEE. 2007, pp. 136–141.

[RH07] Maxim Raya and Jean-Pierre Hubaux. "Securing vehicular ad hoc networks". In: *Journal of computer security* 15.1 (2007), pp. 39–68.

[Ray+07] Maxim Raya et al. "Eviction of misbehaving and faulty nodes in vehicular networks". In: *IEEE Journal on Selected Areas in Communications* 25.8 (2007), pp. 1557–1568.

[YOW08] Gongjun Yan, Stephan Olariu, and Michele C Weigle. "Providing VANET security through active position detection". In: *Computer communications* 31.12 (2008), pp. 2883–2897.

[Hen+09] Olaf Henniger et al. "Security requirements for automotive on-board networks". In: *Intelligent Transport Systems Telecommunications,(ITST), 2009 9th International Conference on*. IEEE. 2009, pp. 641–646.

[Pie+09] Sylvia Pietzsch et al. "Results of a precrash application based on laser scanner and short-range radars". In: *IEEE Transactions on Intelligent Transportation Systems* 10.4 (2009), pp. 584–593.

[McK10] Wes McKinney. "Data Structures for Statistical Computing in Python". In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 51–56.

[RH10] George F Riley and Thomas R Henderson. "The ns-3 network simulator". In: *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[Tia+10] Daxin Tian et al. "A vehicular ad hoc networks intrusion detection system based on BUSNet". In: *2010 2nd International Conference on Future Computer and Communication*. Vol. 1. IEEE. 2010, pp. V1–225.

[Var10] Andras Varga. "OMNeT++". In: *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

[Wie+10] Björn Wiedersheim et al. "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough". In: *Wireless On-demand Network Systems and Services (WONS), 2010 Seventh International Conference on*. IEEE. 2010, pp. 176–183.

[ETS11] TS ETSI. "102 165-1:" Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN)". In: *Methods and protocols* (2011), pp. 2011–03.

[Göh+11] Daniel Göhring et al. "Radar/lidar sensor fusion for car-following on highways". In: *The 5th International Conference on Automation, Robotics and Applications*. IEEE. 2011, pp. 407–412.

[GLG11] Jyoti Grover, Vijay Laxmi, and Manoj Singh Gaur. "Misbehavior detection based on ensemble learning in vanet". In: *International Conference on Advanced Computing, Networking and Security*. Springer. 2011, pp. 602–611.

[Gro+11] Jyoti Grover et al. "Machine learning approach for multiple misbehavior detection in VANET". In: *International Conference on Advances in Computing and Communications*. Springer. 2011, pp. 644–653.

[Ped+11] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *Journal of machine learning research* 12.Oct (2011), pp. 2825–2830.

[SGD11] Christoph Sommer, Reinhard German, and Falko Dressler. "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis". In: *IEEE Transactions on Mobile Computing* 10.1 (Jan. 2011), pp. 3–15. ISSN: 1536-1233. DOI: 10.1109/TMC.2010.133.

[WG11] Marko Wolf and Timo Gendrullis. "Design, implementation, and evaluation of a vehicular hardware security module". In: *International Conference on Information Security and Cryptology*. Springer. 2011, pp. 302–318.

[Bis+12] Norbert Bismeyer et al. "Assessment of node trustworthiness in vanets using data plausibility checks with particle filters". In: *Vehicular Networking Conference (VNC), 2012 IEEE*. IEEE. 2012, pp. 78–85.

[Jae+12] Attila Jaeger et al. "A novel framework for efficient mobility data verification in vehicular ad-hoc networks". In: *International Journal of Intelligent Transportation Systems Research* 10.1 (2012), pp. 11–21.

[Moa+12] Rim Moalla et al. "Risk analysis study of its communication architecture". In: *Network of the Future (NOF), 2012 Third International Conference on the*. IEEE. 2012, pp. 1–5.

[WS12] Marko Wolf and Michael Scheibel. "A Systematic Approach to a Qualified Security Risk Analysis for Vehicular IT Systems." In: *Automotive-Safety & Security*. 2012, pp. 195–210.

[BTC13] Sagar Behere, Martin Törngren, and De-Jiu Chen. "A reference architecture for cooperative driving". In: *journal of Systems Architecture* 59.10 (2013), pp. 1095–1112.

[DR13] Ameneh Daeinabi and Akbar Ghaffarpour Rahbar. "Detection of malicious vehicles (DMV) through monitoring in Vehicular Ad-Hoc Networks". In: *Multimedia tools and applications* 66.2 (2013), pp. 325–338.

[Dev13] NumPy Developers. "NumPy". In: *NumPy Numpy. Scipy Developers* (2013).

[DC13] Neelanjana Dutta and Sriram Chellappan. "A time-series clustering approach for Sybil attack detection in vehicular ad hoc networks". In: *Int. Conf. Adv. Veh. Syst., Technol. Appl*. 2013, pp. 21–26.

[Lef+13] Stéphanie Lefevre et al. "Impact of v2x privacy strategies on intersection collision avoidance systems". In: *2013 IEEE Vehicular Networking Conference*. IEEE. 2013, pp. 71–78.

[Ron+13] Michele Rondinone et al. "iTETRIS: a modular simulation platform for the large scale evaluation of cooperative ITS applications". In: *Simulation Modelling Practice and Theory* 34 (2013), pp. 99–125.

[Why+13] William Whyte et al. "A security credential management system for V2V communications". In: *Vehicular Networking Conference (VNC), 2013 IEEE*. IEEE. 2013, pp. 1–8.

[Bri14] Alex Brisbourne. "Teslas over-the-air fix: Best example yet of the internet of things?" In: *Wired Magazine February* (2014).

[ETS14] EN ETSI. "302 637-2 V1.3.2,"" in: *Intelligent transport systems (ITS)* (2014).

[Ghe+14] Branden Ghena et al. "Green Lights Forever: Analyzing the Security of Traffic Infrastructure." In: *WOOT* 14 (2014), pp. 7–7.

[KC14] Neeraj Kumar and Naveen Chilamkurti. "Collaborative trust aware intelligent intrusion detection in VANETs". In: *Computers & Electrical Engineering* 40.6 (2014), pp. 1981–1996.

[MV14] Charlie Miller and Chris Valasek. "A survey of remote automotive attack surfaces". In: *black hat USA* 2014 (2014), p. 94.

[OHR14] Marcus Obst, Laurens Hobert, and Pierre Reisdorf. "Multi-sensor data fusion for checking plausibility of V2V communications by vision-based multiple-object tracking". In: *Vehicular Networking Conference (VNC), 2014 IEEE*. IEEE. 2014, pp. 143–150.

[PFK14] Jonathan Petit, Michael Feiri, and Frank Kargl. "Revisiting attacker model for smart vehicles". In: *Wireless Vehicular Communications (WiVeC), 2014 IEEE 6th International Symposium on*. IEEE. 2014, pp. 1–5.

[SS14] Hichem Sedjelmaci and Sidi Mohammed Senouci. "A new intrusion detection framework for vehicular networks". In: *2014 IEEE International Conference on Communications (ICC)*. IEEE. 2014, pp. 538–543.

[Tom+14] Andreas Tomandl et al. "VANETsim: An open source simulator for security and privacy concepts in VANETs". In: *2014 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE. 2014, pp. 543–550.

[AGM15a] Khattab M Ali Alheeti, Anna Gruebler, and Klaus D McDonald-Maier. "An intrusion detection system against black hole attacks on the communication network of self-driving cars". In: *2015 sixth international conference on emerging security technologies (EST)*. IEEE. 2015, pp. 86–91.

[AGM15b] Khattab M Ali Alheeti, Anna Gruebler, and Klaus D McDonald-Maier. "An intrusion detection system against malicious attacks on the communication network of driverless cars". In: *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE. 2015, pp. 916–921.

[AGM15c] Khattab M Ali Alheeti, Anna Gruebler, and Klaus D McDonald-Maier. "On the detection of grey hole and rushing attacks in self-driving vehicular networks". In: *2015 7th Computer Science and Electronic Engineering Conference (CEEC)*. IEEE. 2015, pp. 231–236.

[Bou+15] Aymen Boudguiga et al. "RACE: Risk analysis for cooperative engines". In: *New Technologies, Mobility and Security (NTMS), 2015 7th International Conference on*. IEEE. 2015, pp. 1–5.

[EWS15] Karim Emara, Wolfgang Woerndl, and Johann Schlichter. "On evaluation of location privacy preserving schemes for VANET safety applications". In: *Computer Communications* 63 (2015), pp. 11–23.

[LJF15] Wenjia Li, Anupam Joshi, and Tim Finin. "Svm-case: An svm-based context aware security framework for vehicular ad-hoc networks". In: *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. IEEE. 2015, pp. 1–5.

[Mac+15] Georg Macher et al. "SAHARA: a security-aware hazard and risk analysis method". In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*. IEEE. 2015, pp. 621–624.

[Mag15] Leandros A Maglaras. "A novel distributed intrusion detection system for vehicular ad hoc networks". In: *International Journal of Advanced Computer Science and Applications (IJACSA)* 6.4 (2015), pp. 101–106.

[PS15] Jonathan Petit and Steven E Shladover. "Potential cyberattacks on automated vehicles". In: *IEEE Transactions on Intelligent Transportation Systems* 16.2 (2015), pp. 546–556.

[Pet+15a] Jonathan Petit et al. "Connected vehicles: Surveillance threat and mitigation". In: *Black Hat Europe* 11 (2015), p. 2015.

[Pet+15b] Jonathan Petit et al. "Pseudonym schemes in vehicular networks: A survey". In: *IEEE communications surveys & tutorials* 17.1 (2015), pp. 228–255.

[Pet+15c] Jonathan Petit et al. "Remote attacks on automated vehicles sensors: Experiments on camera and lidar". In: *Black Hat Europe* 11 (2015), p. 2015.

[SS15] Hichem Sedjelmaci and Sidi Mohammed Senouci. "An accurate and efficient collaborative intrusion detection framework to secure vehicular networks". In: *Computers & Electrical Engineering* 43 (2015), pp. 33–47.

[Abu+16] Mohammad Y Abualhoul et al. "Study and evaluation of laser-based perception and light communication for a platoon of autonomous vehicles". In: *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE. 2016, pp. 1798–1804.

[AM16] Khattab M Ali Alheeti and Klaus McDonald-Maier. "An intelligent intrusion detection scheme for self-driving vehicles based on magnetometer sensors". In: *2016 International Conference for Students on Applied Engineering (ICSAE)*. IEEE. 2016, pp. 75–78.

[AGM16] Khattab Ali Alheeti, Anna Gruebler, and Klaus McDonald-Maier. "Intelligent intrusion detection of grey hole and rushing attacks in self-driving vehicular networks". In: *Computers* 5.3 (2016), p. 16.

[BT16] Sagar Behere and Martin Törngren. "A functional reference architecture for autonomous driving". In: *Information and Software Technology* 73 (2016), pp. 136–150.

[Ber+16] Olga Berlin et al. "POSTER: Anomaly-based misbehaviour detection in connected car backends". In: *2016 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2016, pp. 1–2.

[Dom+16] Derrick Dominic et al. "Risk Assessment for Cooperative Automated Driving". In: *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy*. ACM. 2016, pp. 47–58.

[Fan+16] Qing Gang Fan et al. "VANET routing replay attack detection research based on SVM". In: *MATEC Web of Conferences*. Vol. 63. EDP Sciences. 2016, p. 05020.

[Hei+16] Rens W van der Heijden et al. "Survey on misbehavior detection in cooperative intelligent transportation systems". In: *arXiv preprint arXiv:1610.06810* (2016).

[Isl+16] Mafijul Md Islam et al. "A risk assessment framework for automotive embedded systems". In: *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*. ACM. 2016, pp. 3–14.

[Li+16] Zhiyi Li et al. "Assessing and mitigating cybersecurity risks of traffic light systems in smart cities". In: *IET Cyber-Physical Systems: Theory & Applications* 1.1 (2016), pp. 60–69.

[Mac+16] Georg Macher et al. "Threat and risk assessment methodologies in the automotive domain". In: *Procedia computer science* 83 (2016), pp. 1288–1294.

[MV16] Charlie Miller and Chris Valasek. "CAN Message Injection". In: *OG Dynamite Edition June* 28 (2016).

[Pla16] CITS Platform. "Platform for the Deployment of Cooperative Intelligent Transport Systems in the EU (E03188)(C-ITS Platform) Final report". In: *DG MOVE-DG Mobility and Transport, Brussels* (2016).

[Pon+16] Christoph Ponikwar et al. "Beyond the Dolev-Yao model: Realistic application-specific attacker models for applications using vehicular communication". In: *arXiv preprint arXiv:1607.08277* (2016).

[SAE16] SAE On-Road Automated Vehicle Standards Committee and others. *Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems*. 2016.

[Sch+16] Christoph Schmittner et al. "Using SAE J3061 for Automotive Security Requirement Engineering". In: *International Conference on Computer Safety, Reliability, and Security*. Springer. 2016, pp. 157–170.

[Ste+16] Marco Steger et al. "A security metric for structured security analysis of cyber-physical systems supporting SAE J3061". In: *Modelling, Analysis, and Control of Complex CPS (CPS Data), 2016 2nd International Workshop on*. IEEE. 2016, pp. 1–6.

[Wah+16] Omar Abdel Wahab et al. "CEAP: SVM-based intelligent detection model for clustered vehicular ad hoc networks". In: *Expert Systems with Applications* 50 (2016), pp. 40–54.

[WZ16] William Whyte and Zhenfei Zhang. "Quantum cryptanalysis, quantum-safe algorithms based on hard problems over lattices, and how we get there from here". In: , (2016).

[YXL16] Chen Yan, Wenyuan Xu, and Jianhao Liu. "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle". In: *DEF CON* 24 (2016).

[Aeb17] Michael Aeberhard. *Object-level Fusion for Surround Environment Perception in Automated Driving Applications*. VDI Verlag GmbH, 2017.

[AGM17] Khattab M Ali Alheeti, Anna Gruebler, and Klaus McDonald-Maier. "Using discriminant analysis to detect intrusions in external communication for self-driving vehicles". In: *Digital Communications and Networks* 3.3 (2017), pp. 180–187.

[AM17] Khattab M Ali Alheeti and Klaus McDonald-Maier. "An intelligent security system for autonomous cars based on infrared sensors". In: *2017 23rd International Conference on Automation and Computing (ICAC)*. IEEE. 2017, pp. 1–5.

[Alh+17] Khattab M Ali Alheeti et al. "An intrusion detection scheme for driverless vehicles based gyroscope sensor profiling". In: *2017 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2017, pp. 448–449.

[Cam17] Oliver Cameron. *Under the Hood of a Self-Driving Taxi*. July 2017. URL: https://news.voyage.auto/under-the-hood-of-a-self-driving-car-78e8bbce62a6.

[CSS17] K Chandan, Alvaro M Seco, and Ana Bastos Silva. "Real-time traffic signal control for isolated intersection, using car-following logic under connected vehicle environment". In: *Transportation research procedia* 25 (2017), pp. 1610–1625.

[Cri17] Common Criteria. "Common Methodology for Information Technology Security Evaluation, Evaluation methodology, V3.1, Revision 5". In: *Common Criteria* (2017).

[Dos+17] Alexey Dosovitskiy et al. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*. 2017, pp. 1–16.

[Els17] News from Elsewhere... *Man fined for painting road signs to aid his commute*. Nov. 2017. URL: http://www.bbc.com/news/blogs-news-from-elsewhere-42181263.

[Evt+17] Ivan Evtimov et al. "Robust Physical-World Attacks on Machine Learning Models". In: *CoRR* abs/1707.08945 (2017). arXiv: 1707.08945. URL: http://arxiv.org/abs/1707.08945.

[Gha+17] Fuad A Ghaleb et al. "An effective misbehavior detection model using artificial neural network for vehicular ad hoc network applications". In: *Application, Information and Network Security (AINS), 2017 IEEE Conference on*. IEEE. 2017, pp. 13–18.

[gro17] C-ITS Platform group. *Certificate Policy for Deployment and Operation of European Cooperative Intelligent Transport Systems (C-ITS)*. Tech. rep. C-ITS Platform, 2017. URL: https://ec.europa.eu/transport/sites/transport/files/c-its_certificate_policy_release_1.pdf.

[Gu+17a] Pengwenlong Gu et al. "k-Nearest Neighbours classification based Sybil attack detection in Vehicular networks". In: *2017 Third International Conference on Mobile and Secure Services (MobiSecServ)*. IEEE. 2017, pp. 1–6.

[Gu+17b] Pengwenlong Gu et al. "Support vector machine (svm) based sybil attack detection in vehicular networks". In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE. 2017, pp. 1–6.

[Han+17] Jun Han et al. "Convoy: Physical context verification for vehicle platoon admission". In: *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*. ACM. 2017, pp. 73–78.

[17] "IEEE Standard for Wireless Access in Vehicular Environments–Security Services for Applications and Management Messages - Amendment 1". In: *IEEE Std 1609.2a-2017 (Amendment to IEEE Std 1609.2-2016)* (Oct. 2017), pp. 1–123. DOI: 10.1109/IEEESTD.2017.8065169.

[Jem+17] Ines Jemaa et al. "An Overview of Security Ongoing Work in Cooperative ITS". In: 2017.

[Kim+17] Myeongsu Kim et al. "Collaborative security attack detection in software-defined vehicular networks". In: *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE. 2017, pp. 19–24.

[MSN17] Pierre Merdrignac, Oyunchimeg Shagdar, and Fawzi Nashashibi. "Fusion of perception and v2p communication systems for the safety of vulnerable road users". In: *IEEE Transactions on Intelligent Transportation Systems* 18.7 (2017), pp. 1740–1751.

[Mon+17] JP Monteuuis et al. "Securing PKI Requests for C-ITS Systems". In: *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*. IEEE. 2017, pp. 1–8.

[Mur17] Charles Murray. *Automakers, Suppliers Ratchet Up Autonomous Car Programs*. https://www.designnews.com/automotive-0/automakers-suppliers-ratchet-autonomous-car-programs/189058092646554. May 2017.

[Pon17] Fabian de Ponte Müller. "Survey on ranging sensors and cooperative techniques for relative positioning of vehicles". In: *Sensors* 17.2 (2017), p. 271.

[Raw+17] Zaydounr Y Rawashdeh et al. *Comfortable Automated Emergency Brake for Urban Traffic Light Based on DSRC and On-Board Sensors*. Tech. rep. SAE Technical Paper, 2017.

[Sch17] Brandon Schoettle. "Sensor fusion: A comparison of sensing capabilities of human drivers and highly automated vehicles". In: *University of Michigan, Sustainable Worldwide Transportation, Tech. Rep. SWT-2017-12* (2017).

[SK17] S Sharanya and S Karthikeyan. "CLASSIFYING MALICIOUS NODES IN VANETS USING SUPPORT VECTOR MACHINES WITH MODIFIED FADING MEMORY". In: (2017).

[SLG17] Mingshun Sun, Ming Li, and Ryan Gerdes. "A data trust framework for VANETs enabling false data detection and secure vehicle tracking". In: *Communications and Network Security (CNS), 2017 IEEE Conference on*. IEEE. 2017, pp. 1–9.

[Tan+17] Heng Chuan Tan et al. "A non-biased trust model for wireless mesh networks". In: *International Journal of Communication Systems* 30.9 (2017).

[Tat17] Didi Kirsten Tatlow. *Scammers in China Fake Road Injuries, but Cameras Capture the Truth*. Feb. 2017. URL: https://www.nytimes.com/2017/02/21/world/asia/china-traffic-scam-fraud.html.

[Ulb+17] Simon Ulbrich et al. "Towards a functional system architecture for automated vehicles". In: *arXiv preprint arXiv:1703.08557* (2017).

[Yua+17] Ting Yuan et al. "Object matching for inter-vehicle communication systems—An IMM-based track association approach with sequential multiple hypothesis test". In: *IEEE Transactions on Intelligent Transportation Systems* 18.12 (2017), pp. 3501–3512.

[18a] 2018. URL: http://www.carqueryapi.com/.

[Ami+18] Hanane Amirat et al. "Fuzzy Clustering for Misbehaviour Detection in VANET". In: *2018 International Conference on Smart Communications in Network Technologies (SaCoNeT)*. IEEE. 2018, pp. 200–204.

[Bre+18] Benedikt Brecht et al. "A Security Credential Management System for V2X Communications". In: *IEEE Transactions on Intelligent Transportation Systems* (2018).

[ENC18] ENCAP. *Assessment Protocol-SA. Version 8.0.2*. ENCAP, Jan. 2018.

[18b] *Exploring Relationships in Body Dimensions*. 2018. URL: http://ww2.amstat.org/publications/jse/v11n2/datasets.heinz.html.

[Ezi+18] Elvin Eziama et al. "Malicious Node Detection in Vehicular Ad-Hoc Network Using Machine Learning and Deep Learning". In: *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2018, pp. 1–6.

[Fur18] Paola Furlan. *DATABASE MOTO*. 2018. URL: http://www.motocicliste.net/moto/db.asp.

[Hag+18] Sepand Haghighi et al. "PyCM: Multiclass confusion matrix library in Python". In: *Journal of Open Source Software* 3.25 (May 2018), p. 729. DOI: 10.21105/joss.00729. URL: https://doi.org/10.21105/joss.00729.

[HLK18] Rens W van der Heijden, Thomas Lukaseder, and Frank Kargl. "VeReMi: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs". In: *arXiv preprint arXiv:1804.06701* (2018).

[Jha+18] Saurabh Jha et al. "Avfi: Fault injection for autonomous vehicles". In: *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE. 2018, pp. 55–56.

[Kos+18] Dimitrios Kosmanos et al. "RF Jamming Classification using Relative Speed Estimation in Vehicular Wireless Networks". In: *arXiv preprint arXiv:1812.11886* (2018).

[Mon+18a] Jean-Philippe Monteuuis et al. ""My autonomous car is an elephant": A Machine Learning based Detector for Implausible Dimension". In: *3rd International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE. Shanghai, China, 2018, pp. 1–8.

[Mon+18b] Jean-Philippe Monteuuis et al. "Attacker model for Connected and Automated Vehicles". In: *2nd Computer Science in Cars Symposium - Future Challenges in Artificial Intelligence & Security for Autonomous Vehicles*. ACM. Munich, Germany, 2018, pp. 3–14.

[Mon+18c] Jean-Philippe Monteuuis et al. "SARA: Security Automotive Risk Analysis Method". In: *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*. ACM. Incheon, Republic of Korea, 2018, pp. 3–14.

[SAE18] Taxonomy SAE. "Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles". In: *SAE Standard J3016* (2018).

[18c] *Search new cars sorted by dimensions and boot capacity*. 2018. URL: https://www.automobiledimension.com/car-search-engine.php.

[SPL18] Prinkle Sharma, Jonathan Petit, and Hong Liu. "Pearson Correlation Analysis to Detect Misbehavior in VANET". In: *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE. 2018, pp. 1–5.

[Sin+18] Pranav Kumar Singh et al. "Misbehavior detection in C-ITS using deep learning approach". In: *International Conference on Intelligent Systems Design and Applications*. Springer. 2018, pp. 641–652.

[Sit+18] Chawin Sitawarin et al. "DARTS: Deceiving Autonomous Cars with Toxic Signs". In: *arXiv preprint arXiv:1802.06430* (2018).

[SSP18] Steven So, Prinkle Sharma, and Jonathan Petit. "Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 564–571.

[SBK18] Basant Subba, Santosh Biswas, and Sushanta Karmakar. "A game theory based multi layered intrusion detection framework for VANET". In: *Future Generation Computer Systems* 82 (2018), pp. 12–28.

[Teo18] Teoalida. *Motorcycle database*. July 2018. URL: http://www.teoalida.com/cardatabase/motorcycles/.

[Tra18] U.S. Department of Transportation Intelligent Transportation Systems Joint Program Office. *Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data: Department of Transportation - Data Portal*. Aug. 2018. URL: https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj.

[Van18] Rens Wouter Van der Heijden. "Misbehavior detection in cooperative intelligent transport systems". PhD thesis. Universität Ulm, 2018.

[Vuk+18] Vladimir Vukadinovic et al. "3GPP C-V2X and IEEE 802.11 p for Vehicle-to-Vehicle communications in highway platooning scenarios". In: *Ad Hoc Networks* 74 (2018), pp. 17–29.

[Xu+18] Hao Xu et al. "High Resolution Micro Traffic Data From Roadside LiDAR Sensors for Connected Vehicles and New Traffic Applications". In: *NDOT Research Report* (2018).

[ZF18] Jithin Zacharias and Sibylle Fröschle. "Misbehavior detection system in VANETs using local traffic density". In: *2018 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2018, pp. 1–4.

[Zen+18] Yi Zeng et al. "Senior2Local: A machine learning based intrusion detection method for vanets". In: *International Conference on Smart Computing and Communication*. Springer. 2018, pp. 417–426.

[19a] *Automated Driving Toolbox*. The MathWorks, Natick, MA, USA. 2019.

[ETS19] EN ETSI. *302 636-5-1 V2.2.1. Intelligent Transport Systems (ITS); Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol*. 2019.

[GQ19] Sohan Gyawali and Yi Qian. "Misbehavior Detection using Machine Learning in Vehicular Communication Networks". In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–6.

[Kaj19] Nevrus Kaja. "Artificial Intelligence and Cybersecurity: Building an Automotive Cybersecurity Framework Using Machine Learning Algorithms". In: (2019).

[Kam+19] Joseph Kamel et al. "CaTch: A Confidence Range Tolerant Misbehavior Detection Approach". In: 2019.

[19b] *Mapping Toolbox*. The MathWorks, Natick, MA, USA. 2019.

[Mon19] A Mond. *Developing Longitudinal Controls for a Self-Driving Taxi*. 2019. URL: https://fr.mathworks.com/company/newsletters/articles/developing-longitudinal-controls-for-a-self-driving-taxi.html.

[Mon+19] Jean-Philippe Monteuuis et al. "Sensor-V2X Correlation for Misbehavior Detection". to be submitted to a journal. 2019.

[19c] *PreScan*. Feb. 2019. URL: https://tass.plm.automation.siemens.com/prescan.

[19d] *Sensor Fusion & Tracking Toolbox*. The MathWorks, Natick, MA, USA. 2019.

[Sin+19] Pranav Kumar Singh et al. "Machine Learning Based Approach to Detect Position Falsification Attack in VANETs". In: *International Conference on Security & Privacy*. Springer. 2019, pp. 166–178.

[SPS19] Steven So, Jonathan Petit, and David Starobinski. "Physical layer plausibility checks for misbehavior detection in V2X networks". In: *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. ACM. 2019, pp. 84–93.

[Wan19] Le Wang. *VANET Toolbox: A Vehicular Network Simulator based on DES - File Exchange - MATLAB Central*. 2019. URL: https://fr.mathworks.com/matlabcentral/fileexchange/68437-vanet-toolbox-a-vehicular-network-simulator-based-on-des.

[Wan+19] Xiaoyang Wang et al. "Location Anomalies Detection for Connected and Autonomous Vehicles". In: *arXiv preprint arXiv:1907.00811* (2019).

[Zen+19] Yi Zeng et al. "DeepVCM: A Deep Learning Based Intrusion Detection Method in VANET". In: *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. IEEE. 2019, pp. 288–293.

[ETSa] TS ETSI. "102 165-1:" Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN)". In: *Methods and protocols* (), pp. 2017–10.

[ETSb] TS ETSI. "102 894-1 V1. 1.1: Intelligent Transport Systems (ITS)". In: *Users and applications requirements* ().

[USDa] USDOT. *Safety Pilot Model Deployment Data: Open Data*. URL: https://datahub.transportation.gov/Automobiles/Safety-Pilot-Model-Deployment-Data/a7qq-9vfe.

[USDb] USDOT. *Wyoming CV Pilot Basic Safety Message One Day Sample: Department of Transportation - Data Portal*. URL: https://data.transportation.gov/Automobiles/Wyoming-CV-Pilot-Basic-Safety-Message-One-Day-Samp/9k4m-a3jc.

[09] *Information technology – Security techniques – Evaluation criteria for IT security*. Standard. Geneva, CH: International Organization for Standardization, Aug. 2009.

[08] *Information Technology - Security Techniques - Methodology for IT Security Evaluation*. Standard. Geneva, CH: International Organization for Standardization, Aug. 2008.

[ISO11] ISO. *Road vehicles – Functional safety*. Norm. 2011.

[Int16] SAE International. *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*. Standard. SAE International, Mar. 2016.

ECOLE DOCTORALE

Titre: Résilience par conception & prévision de défaillances pour un véhicule autonome connecté

Mots clés: Perception,VAC,V2X,résilience,capteurs, défaillances

Résumé: Les véhicules autonomes dotés d'un niveau d'automatisation 5 conduiront de manière autonome dans tous les scénarios routiers tels que les autoroutes, les routes enneigées, les zones urbaines ou les embouteillages. L'intégration de la communication V2X, en tant que nouvelle source de perception du véhicule, pourrait supprimer les limitations de la perception locale en communiquant avec un piéton caché par un obstacle ou en détectant à l'avance la présence d'un véhicule caché par un brouillard épais. Cependant, cette communication V2X peut constituer une nouvelle source d'attaques menaçant la perception du véhicule. Les contre-mesures actuelles ne sont pas conçues pour toutes les architectures de véhicules autonomes, car elles requièrent l'assistance du conducteur ou fonctionnent avec un ensemble spécifique de capteurs. La thèse vise donc à proposer une architecture de perception générique et résiliante aux défaillances pour tous les types de véhicules connectés et autonomes. Dans cette thèse, nous proposons une architecture de perception générique nommée GPA avec son algorithme de perception résiliante aux défaillances (FRPA). Nous proposons une nouvelle méthode d'analyse de menaces et d'évaluation des risques nommée SARA, qui identifie et évalue le risque d'attaques ciblant les véhicules connectés et automatisés de niveau 5. Pour identifier où et comment ces attaques ont lieu, nous proposons un modèle d'attaquant et un modèle d'objectifs de sécurité pour tous les systèmes de perception automobile. Nous avons implémenté deux modules de notre algorithme FRPA: un module classification des défaillances basé sur une méthode de Machine Learning et un module de corrélation V2X-Capteur en considérant trois sources d'information: radar, caméra et V2X.
Nous avons mis en évidence plusieurs nouvelles attaques dans le cycle de perception et soulevé le besoin de nouvelles contre-mesures de sécurité centrées sur l'intégrité physique des infrastructures routières et sur les algorithmes de perception fiables. De plus, nos contre-mesures basées sur l'apprentissage automatique et la corrélation entre capteurs ont permis d'atteindre une très bonne précision pour détecter et classifier les défaillances de perception (score de précision supérieur à 90 %). Enfin, les idées développées dans la thèse ont abouti à 10 brevets déposés et à plusieurs publications.

Title: Resilience by design & failures forecasting for a connected autonomous vehicle

Keywords: Perception,CAV,V2X,Resilience,sensors, failures

Abstract: Autonomous vehicles with an automation level 5 will drive autonomously in any road scenarios such as highways, snowy roads, urban areas, or traffic jams. The integration of V2X communication, as a new source of perception for the vehicle could remove the limitations of local perception by communicating with an occluded pedestrian or by detecting in advance the presence of a vehicle under a heavy mist. However, this V2X communication may be a new source of attacks threatening the vehicle perception. Current countermeasures are not designed for all autonomous vehicles because these countermeasures require the driver assistance or work with a specific set of sensors. Therefore, the thesis aims to propose a generic failure resilient perception architecture for all types of connected and autonomous vehicles supporting different kinds of sensors. In this thesis, we propose a generic perception architecture named GPA with its failure resilient perception algorithm (FRPA). We propose a new threat analysis and risk assessment method named SARA that identifies and assess the risk of attacks targeting connected and automated vehicles with an automation level 5. To identify where and how these attacks occur, we propose an attacker and a security goal model for all automotive perception systems. We implemented two modules of our failures resilient perception algorithm (FRPA): a Machine Learning based Failure Classifier and a V2X-Sensor Correlation Module considering three kinds of source: camera, radar, and V2X.
We highlighted several new attacks in the perception pipeline and raise the need for new security countermeasures such as the physical integrity of road infrastructures and trustworthy perception algorithms. Besides, our countermeasures based on machine learning and sensor correlation showed very accurate results to detect and classifies perception failures (over 90% accuracy score). Finally, the ideas developed in the thesis resulted in 10 filled patents and several publications.

Institut Polytechnique de Paris
91120 Palaiseau, France