INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2025IPPAT022

Thèse de doctorat

Cyberattacks Detection in Cooperative Intelligent Transportation System (C-ITS)

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

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

Thèse présentée et soutenue à Palaiseau, le 3 juillet 2025, par

AYOUB WEHBY

Composition du Jury :

Houda LABIOD Professeure, HUAWEI	Présidente/Examinatrice
Oyunchimeg SHAGDAR Cheffe de Projet Innovation Connectivité, Renault Groupe (AMPERE)	Rapporteuse
Dominique GAITI Professeure, Université de Technologie de Troyes	Rapporteuse
Sherali ZEADALLY Professeur, University of Kentucky	Examinateur
Diogo Menezes FERRAZANI MATTOS Professeur, Fluminense Federal University	Examinateur
Rida Khatoun Professeur, Télécom Paris (INFRES)	Directeur de thèse
Ahmad FADLALLAH Professeur, University of Sciences and Arts in Lebanon (Computer Science Department)	Co-Encadrant de thèse
Pierre GACHON Expert Leader Cyber Sécurité, Renault Groupe (AMPERE)	Invité

Acknowledgments

The bigger the achievement, the greater the support behind it. This thesis is not an ordinary three-year journey, it is a one-of-a-kind experience, made fruitful by the support of many people.

First of all, I want to thank my supervisor, Prof. Rida KHATOUN, and my co-supervisor, Prof. Ahmad FADLALLAH, for their guidance and the experiences they have shared. Your invaluable support has influenced my path in the last couple of years and changed my perception of the future.

I want to extend my thanks to the jury members: Dr. Oyunchimeg SHAGDAR, Prof. Dominique GAITI, Prof. Houda LABIOD, Prof. Sherali ZEADALLY, Prof. Diogo Menezes FERRAZANI MATTOS, and Mr. Pierre GACHON, for their interest in my work and their insightful contributions.

I want to express my gratitude to Télécom Paris and Institute Polytechnique de Paris, administration and staff, for providing enriching experiences such as seminars, trainings, and opportunities for doctoral students to be ready for future endeavors.

I want to thank the Cybersecurity and Cryptography team members for the scientific and joyful moments we shared throughout this thesis. We have created memories that will stay with me forever.

A special thanks to the colleagues who became friends, Emma, Fadel, Ali, Hassan, Adonis, Ali, Mohammed Lamine, Nathan, Liam, Ismet, and Peter. I was blessed to share valuable moments with each one of you throughout this journey.

My special thanks go to my hometown friends with whom I share a lifetime of experiences, including this one, though far apart. Amjad, Bader, Jad, Rayan, and Waed, thank you for being you. I greatly acknowledge and appreciate the support of my colleague, friend, and brother, Fadi, who is always there through thick and thin. Your presence has shifted time zones and moved mountains.

Last but not least, the cause behind it all, the unwavering support. I am deeply grateful to my parents, brother, and sisters for their trust, support, and encouragement, which kept me pushing through challenging times. To my whole family and relatives, thank you for being a major part of this journey.

Abstract

The rapid advancement of Cooperative-Intelligent Transportation Systems (C-ITS) is transforming modern mobility by enabling seamless communication between cars, infrastructure, and other road users. These systems enhance road safety, optimize traffic flow, and support the development of autonomous driving technologies. The network within such systems is inherently unstable, as the number of cooperating cars fluctuates based on their location and movement direction. Consequently, these networks are susceptible to large-scale attacks due to the potential for a high number of participating cars.

A prevalent large-scale attack is the Distributed Denial of Service (DDoS) attack, which has demonstrated its capacity to disrupt systems for extended periods. In the C-ITS context, also known as connected cars, service disruption caused by DDoS can severely impact road users, ranging from faulty navigation prompts to catastrophic accidents. Therefore, DDoS attacks pose a significant risk to current and future services within the connected car domain, underscoring the critical challenge of ensuring security and reliability.

This research proposes an approach for a distributed Intrusion Detection System (IDS) designed to mitigate DDoS attacks in a connected car environment. Initially, we start by presenting a comprehensive overview of C-ITS, detailing its key components, architectural frameworks, communication protocols, and security mechanisms. We explore the core challenges faced by these systems, emphasizing their vulnerabilities to cyber threats. A discussion of various attack vectors, including intra- and inter-vehicle attacks, is provided, along with a review of existing countermeasures. This foundational understanding establishes the groundwork for investigating more advanced and targeted threats.

A primary focus is on DDoS attacks utilizing Cooperative Awareness Messages (CAMs), which are designed to facilitate awareness among cars. These attacks can overwhelm connected car networks, leading to disruptions or failures in critical safety message exchange. Through simulation-based experiments, we analyzed diverse attack scenarios to assess their impact on network performance, message dissemination, and overall road safety. These scenar-

ios revealed a decrease in the CAM broadcast rate from legitimate cars and an accumulation of invalid data.

Building on this analysis, we generated a CAM-DDoS dataset, reflecting realistic car traffic demand in Luxembourg City, to simulate DDoS attacks. This dataset, collected from Road-Side Units (RSUs), facilitated the development of three machine learning models, which serve as essential components of an IDS. To ensure consistent model performance, we employed a single dataset for training, validation, and testing and then validated the performance using two additional, distinct datasets. Given the challenge of generalizability in machine learning models, we evaluated the robustness of our DDoS detection approach against various morphing DDoS attacks.

We also investigate a particularly stealthy form of DDoS attacks leveraging Sybil identities. In these scenarios, attackers generate multiple fake identities to manipulate network behavior and disrupt cooperative awareness among cars. We examine the limitations of proposed detection mechanisms in addressing these complex threats and propose an optimized data-handling approach that serves as the foundation for our proposed ensemble hard voting detection model. The results obtained demonstrate an effective attack detection with reduced false alarms against Sybil DDoS attacks.

This thesis reflects on the broader implications of securing connected cars against evolving cyber threats. The need for continuous advancements in IDS technologies, dataset diversity, and adaptive security strategies to safeguard intelligent transportation systems in real-world deployments is emphasized. The research contributes to the anomaly module within an IDS for the connected cars architecture. This IDS can be situated in the Multi-access Edge Computing (MEC) node that provides computational services while maintaining the existing connected car infrastructure. By addressing key challenges and proposing innovative detection methodologies, this thesis contributes to the ongoing efforts to enhance the resilience and security of connected car networks.

Résumé

L'évolution rapide des systèmes de transport intelligent coopératif (C-ITS) transforme la mobilité moderne en permettant une communication fluide entre les véhicules, les infrastructures et les autres usagers de la route. Ces systèmes améliorent la sécurité routière, optimisent le flux du trafic et soutiennent le développement des technologies de conduite autonome. Le réseau au sein de ces systèmes est intrinsèquement instable, car le nombre de véhicules coopérants fluctue en fonction de leur localisation et de leur direction de déplacement. Par conséquent, ces réseaux sont vulnérables aux attaques à grande échelle, en raison du nombre potentiellement élevé de voitures participantes.

Une attaque de grande échelle fréquente est l'attaque par déni de service distribué (DDoS), qui a démontré sa capacité à perturber durablement les systèmes. Dans le contexte des C-ITS, également appelé véhicules connectés, les perturbations de service causées par les attaques DDoS peuvent gravement affecter les usagers de la route, allant de simples erreurs de navigation à des accidents catastrophiques. Les attaques DDoS représentent ainsi un risque majeur pour les services actuels et futurs dans le domaine des véhicules connectés, soulignant l'importance critique d'assurer leur sécurité et leur fiabilité.

Cette recherche propose une approche pour un système de détection d'intrusion (IDS) distribué, conçu pour atténuer les attaques DDoS dans un environnement de véhicules connectées. Nous débutons par une présentation exhaustive des C-ITS, détaillant leurs principaux composants, leurs architectures, leurs protocoles de communication et leurs mécanismes de sécurité. Nous explorons les défis fondamentaux auxquels ces systèmes sont confrontés, en mettant en évidence leurs vulnérabilités face aux cybermenaces. Une discussion sur divers vecteurs d'attaque, y compris les attaques intra- et inter-véhicules, est également proposée, accompagnée d'un examen des contre-mesures existantes. Cette compréhension fondamentale établit les bases pour l'étude de menaces plus avancées et ciblées.

L'axe principal de cette recherche se porte particulièrement sur les attaques DDoS exploitant les Cooperative Awareness Messages (CAMs), conçus pour faciliter la prise de conscience

mutuelle entre véhicules. Ces attaques peuvent saturer les réseaux de véhicules connectées, entraînant des perturbations ou des échecs dans l'échange des messages critiques de sécurité. À travers des expériences basées sur des simulations, nous avons analysé divers scénarios d'attaque afin d'évaluer leur impact sur les performances réseau, la diffusion des messages et la sécurité routière globale. Ces scénarios ont mis en évidence une diminution du taux d'émission des CAMs par les véhicules légitimes ainsi qu'une accumulation de données invalides.

Sur la base de cette analyse, nous avons généré un ensemble de données CAM-DDoS, reflétant la demande réaliste de trafic automobile dans la ville de Luxembourg, afin de simuler des attaques DDoS. Cet ensemble de données, collecté à partir des Unités en Bord de Route (UBR), a permis le développement de trois modèles d'apprentissage automatique, servant de composantes essentielles à un IDS. Pour garantir des performances cohérentes, nous avons utilisé un même ensemble de données pour l'entraînement, la validation et le test, puis validé les performances avec deux ensembles de données distincts. Face au défi de la généralisabilité des modèles d'apprentissage automatique, nous avons également évalué la robustesse de notre approche de détection face à diverses attaques DDoS morphologiques.

Nous nous intéressons également à une forme particulièrement furtive d'attaque DDoS exploitant des identités Sybil. Dans ces scénarios, les attaquants génèrent plusieurs fausses identités pour manipuler le comportement du réseau et perturber la prise de conscience coopérative entre les véhicules. Nous analysons les limitations des mécanismes de détection proposés pour faire face à ces menaces complexes et proposons une approche optimisée de gestion des données, qui constitue la base de notre modèle de détection par vote majoritaire. Les résultats obtenus démontrent une détection efficace des attaques, avec une réduction notable des fausses alertes contre les attaques Sybil DDoS.

Cette thèse s'inscrit dans une réflexion plus large sur la sécurisation des véhicules connectés face à l'évolution constante des cybermenaces. Elle souligne la nécessité de progrès continus en matière de technologies IDS, de diversification des ensembles de données et de stratégies de sécurité adaptatives pour garantir la protection des systèmes de transport intelligents dans des déploiements réels. Cette recherche contribue au module d'anomalies d'un IDS intégré à l'architecture des véhicules connectés. Cet IDS peut être déployé sur un nœud Multi-access Edge Computing (MEC), tirant parti de ses ressources de calcul tout en maintenant l'infrastructure existante. En relevant les principaux défis et en proposant des méthodologies innovantes de détection, cette thèse contribue aux efforts en cours pour renforcer la résilience et la sécurité des réseaux de véhicules connectées.

Acronyms

AA Authorization Authority

ACA Authorization CA

ARP Address Resolution Protocol

AT Authorization Ticket

ATs Authorization Tickets

AU Application Unit

AUC Area Under the Curve

BTP Basic Transport Protocol

BSM Basic Safety Message

CA Certificate Authority

CAs Certificate Authorities

CAM Cooperative Awareness Message

CAMs Cooperative Awareness Messages

CAN Control Area Network

CBR Channel Busy Ratio

CCH Control Channel

C-ITS Cooperative-Intelligent Transportation System

CPM Collective Perception Message

C-V2X Cellular-Vehicle-to-Everything

DCC Decentralized Congestion Control

DDoS Distributed DoS

DENM Decentralized Environmental Notification Message

DENMs Decentralized Environmental Notification Messages

DF Double Fault

DoS Denial of Service

DSRC Dedicated-Short Range Communication

EA Enrollment Authority

ECA Enrollment CA

ECC Elliptic Curves Cryptography

ECUs Electronic Control Units

ETSI European Telecommunications Standards Institute

FC Fog Computing

FN False Negative

FNs False Negatives

FP False Positive

FPs False Positives

HIDS Host-based IDS

HistGBC Histogram-based Gradient Boosting Classifier

HSM Hardware Secure Module

IDC Intermediate Data Center

IEEE Institute of Electrical and Electronics Engineers

IDS Intrusion Detection System

IDSs Intrusion Detection Systems

ITS Intelligent Transportation System

ITS-G5 Intelligent Transportation System operating at 5GHz

LDM Local Dynamic Map

LIN Local Interconnect Network

MA Misbehaviour Authority

MANETs Mobile Adhoc Networks

MCC Matthews Correlation Coefficient

MEC Multi-access Edge Computing

MOST Media-Oriented Systems Transport

NIDS Network-based IDS

OBU On-Board Unit

OBUs On-Board Units

OTA Over-The-Air

PAM Platoon Awareness Message

PKI Public Key Infrastructure

PseudoID Pseudonymous ID

Q-stat Q-statistics

RA Registration Authority

RCA Root Certificate Authority

RF Random Forest

ROC Receiver Operating Characteristic

ROI Region of Interest

RSSI Received Signal Strength Indicator

RSU Road-Side Unit

RSUs Road-Side Units

SCMS Security Credential Management System

SDN Software Defined Network

TN True Negative

TNs True Negatives

TP True Positive

TPs True Positives

UDP User Datagram Protocol

V2D Vehicle-to-Device

V2I Vehicle-to-Infrastructure

V2P Vehicle-to-Pedestrian

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

VANET Vehicular Adhoc Network

VANETs Vehicular Adhoc Networks

WAVE Wireless Access in Vehicle Environments

Contents

Acknowledgments ii

Abstract iv

Résumé vi

Acronyms x

List of Figures xvii

List of Tables xx

1 Introduction 1

1.1 Context of the Work 1

1.2 Problem Statement 2

1.3 Contributions 4

1.4 Outline 6

2 C-ITS Components, Communications, Services, and Challenges 7

2.1 Introduction 7

2.2 C-ITS Components and Architecture 8

2.2.1 Components . . . 8
2.2.2 Main Architecture . . . 8
2.2.3 Alternative and Advanced Architectures . . . 9
2.3 Communication Media and Standards . . . 13
2.3.1 Intra-vehicular Communications . . . 14
2.3.2 Inter-vehicular Communications . . . 15
2.4 C-ITS Services and Messages . . . 17
2.4.1 C-ITS Services . . . 17
2.4.2 C-ITS Messages . . . 18
2.5 Security in C-ITS . . . 19
2.5.1 ETSI Security Architecture . . . 19
2.5.2 IEEE Security Architecture . . . 20
2.5.3 Security Requirements . . . 21
2.6 Attacks on C-ITS . . . 22
2.6.1 Intra-vehicular Attacks . . . 22
2.6.2 Inter-vehicular Attacks . . . 23
2.7 Countermeasures in C-ITS . . . 24
2.7.1 Security by design . . . 24
2.7.2 Intrusion Detection Systems . . . 25
2.8 Conclusion . . . 26

3 How Does Distributed Denial of Service Affect the Connected Cars Environment? 27

3.1 Introduction . . . 27

3.2 Distributed Denial of Service Attacks . . . 28
3.2.1 Principle and Mechanism . . . 28
3.2.2 Types of DDoS Attacks . . . 28
3.2.3 Potential Targets . . . 30
3.2.4 Modeling . . . 31
3.2.5 Consequences of DDoS Attacks on Connected Cars . . . 32
3.3 Related Work . . . 33
3.4 Simulation Scenario Sets and Parameters . . . 35
3.4.1 Simulation Scenario Set I . . . 37
3.4.2 Simulation Scenario Set II . . . 37
3.4.3 Simulation Scenario Set III . . . 38
3.4.4 Simulation Scenario Set IV . . . 38
3.4.5 Performance metrics . . . 39
3.5 Results and Analysis . . . 39
3.5.1 Simulation Scenario Set I . . . 41
3.5.2 Simulation Scenario Set II . . . 42
3.5.3 Simulation scenario set III . . . 44
3.5.4 Simulation Scenario Set IV . . . 47
3.5.5 Discussion . . . 48
3.6 Conclusion . . . 49

4 Shielding the Connected Cars: A Dataset-Powered Defense Against DDoS 50

4.1 Introduction . . . 50

4.2 Related Work . . . 51
4.3 System Overview . . . 52
4.3.1 Communication and Attack Models . . . 52
4.3.2 Detection Model . . . 53
4.4 Dataset Generation . . . 56
4.4.1 Simulation Environment . . . 56
4.4.2 Dataset Description . . . 57
4.4.3 Dataset Contents . . . 58
4.5 Results and Analysis . . . 60
4.6 Conclusion . . . 62

5 Toward Adaptive Intrusion Detection System for Mitigating Morphing Distributed Denial of Service Attacks in Connected Cars 64

5.1 Introduction . . . 64
5.2 Related Work . . . 65
5.3 Proposed Intrusion Detection System Architecture . . . 66
5.3.1 Architecture and Deployment . . . 66
5.3.2 Models . . . 69
5.3.3 Protocols . . . 69
5.3.4 Rules . . . 70
5.3.5 Performance Evaluation and Potential Response Mechanism . . . 70
5.3.6 IDS Internal Workflow . . . 71
5.4 Performance Evaluation . . . 73
5.4.1 Testbed Configuration . . . 73

5.4.2 Datasets . . . 74
5.4.3 System Overview . . . 75
5.4.4 Performance Metrics . . . 76
5.5 Results and Analysis . . . 76
5.5.1 MCC Evaluation . . . 76
5.5.2 Evaluation of Confusion Matrices . . . 77
5.6 Conclusion . . . 79

6 Generalization and Limitations of DDoS Detection Models in Sybil-Based Attacks on Connected Cars 83

6.1 Introduction . . . 83
6.2 Sybil Attacks . . . 84
6.2.1 Definition of Sybil Attack . . . 84
6.2.2 Sybil Attack Types and Intended Effects . . . 84
6.2.3 Basis of a Sybil Attack Model . . . 85
6.2.4 Countermeasures of Sybil Attacks . . . 86
6.3 Related Work . . . 86
6.4 System Models . . . 89
6.4.1 Attack Model . . . 89
6.4.2 Detection Model . . . 90
6.5 Experimental Setup and Attack Methodology . . . 90
6.5.1 Simulation Environment . . . 91
6.5.2 Previous Dataset . . . 92
6.5.3 Attack Mechanism . . . 93

6.5.4 Performance Metrics . . . 94
6.5.5 Testing the Models' Generalizability . . . 95
6.6 Enhanced Dataset and Optimized Classification Models . . . 96
6.6.1 New Dataset . . . 96
6.6.2 Dataset Statistics . . . 98
6.6.3 Developing New Models . . . 99
6.7 Conclusion . . . 106

7 Conclusion and Perspectives **108**
7.1 Conclusion . . . 108
7.2 Future Perspectives . . . 109
7.2.1 Short-term Perspectives . . . 109
7.2.2 Long-term Perspectives . . . 110

8 List of Publications **112**

Appendices **114**

A CAM field description **114**

B DDoS general detection related works **115**

References **117**

List of Figures

2.1 Main Cooperative-Intelligent Transportation System (C-ITS) architecture 9

2.2 Vehicle-Road-Cloud [22] 10

2.3 Cloud-Vehicular Adhoc Network (VANET) [23] 11

2.4 5G-VANET [24] 12

2.5 Software Defined Network (SDN)-VANET [25] 13

2.6 Social-VANET [26] 14

2.7 In-vehicular network [27] 15

2.8 Vehicular communication protocol stack [10] 16

2.9 Automotive heterogeneous connectivity [29] 17

2.10 European Telecommunications Standards Institute (ETSI) protocol stack [35] . . . 19

2.11 ETSI Public Key Infrastructure (PKI) architecture [36] 20

2.12 Institute of Electrical and Electronics Engineers (IEEE) PKI architecture [39] 21

2.13 Example architecture of a distributed onboard Intrusion Detection System (IDS) [53]. 25

3.1 Distributed DoS (DDoS) attack with a rate 5 times the normal rate of transmission of Cooperative Awareness Message (CAM) messages by cars. 29

3.2 Replay DDoS attack with a rate 5 times the normal rate of transmission of CAM messages by cars. . . . 29
3.3 Scenario statistics . . . 36
3.4 Scenario sketch . . . 36
3.5 Simulation scenario set I metrics (1) . . . 42
3.6 Simulation scenario set I metrics (2) . . . 43
3.7 Simulation scenario set II metrics (1) . . . 44
3.8 Simulation scenario set II metrics (2) . . . 45
3.9 Simulation scenario set III metrics (1) . . . 46
3.10 Simulation scenario set III metrics (2) . . . 46
3.11 Simulation scenario set IV metrics (1) . . . 47
3.12 Simulation scenario set IV metrics (2) . . . 48

4.1 Ensemble learning methods [85] . . . 53
4.2 Modified Luxembourg [20] map . . . 57
4.3 CAM structure conforming to the ETSI standard [30]. . . . 59
4.4 Matthews Correlation Coefficient (MCC) evaluation . . . 61
4.5 Training and testing time (with model size) . . . 62

5.1 IDS architecture for connected cars. . . . 68
5.2 IDS internal workflow . . . 72
5.3 Matthews correlation coefficient values across seven scenarios for each model. . . . 77
5.4 Confusion matrices . . . 80
5.5 Confusion matrices (continued) . . . 81

6.1 Round Robin for n Pseudonymous ID (PseudoID)s 89

6.2 Sybil DDoS attack with a rate 5 times the normal rate of transmission of CAM messages by cars. 90

6.3 Modified Luxembourg [20] map 91

6.4 Common areas marked with orange. 102

6.5 Receiver Operating Characteristic (ROC) curve and the Area Under the Curve (AUC) of the voting classifiers. 104

List of Tables

3.1 Road-Side Unit (RSU) specifications according to the 5G Automobile Association (5GAA) [61], X: Not available. . . . 32

3.2 Comparison of related works . . . 34

3.3 Common simulation parameters . . . 37

3.4 Simulation scenario set I & II parameters . . . 37

3.5 Simulation scenario set III parameters . . . 38

3.6 Simulation scenario set IV parameters . . . 38

3.7 Comparison between simulation scenario sets . . . 40

4.1 Related works . . . 52

4.2 Model configuration . . . 55

4.3 Simulation parameters . . . 56

4.4 Time distribution . . . 58

4.5 Data distribution . . . 58

5.1 Summary of IDS related works . . . 66

5.2 Simulation parameters' comparison . . . 73

5.3 Time distribution . . . 75

5.4 Data distribution 75

6.1 Summary of related works 88

6.2 Confusion matrix data of Random Forest (RF) 95

6.3 Confusion matrix data of Adaboost 95

6.4 Confusion matrix data of Histogram-based Gradient Boosting Classifier (HistGBC) 96

6.5 Data distribution 98

6.6 Time distribution 98

6.7 Confusion matrix data of newly trained models 100

6.8 Confusion matrix data of RSU1 models 100

6.9 Confusion matrix data of RSU023 models 101

6.10 Relationship table between a pair of classifiers D_i and D_k 103

6.11 Confusion matrix data of voting classifier models 105

6.12 Confusion matrix comparison between hard voting and newly trained models . . . 106

6.13 Confusion matrix comparison between soft voting and newly trained models 106

A.1 Some dataset fields description 114

B.1 Comparison of DDoS detection approaches. 116

Chapter 1

Introduction

1.1 Context of the Work

The concept of connected cars is transforming our perception of the automotive landscape. Various partnerships, such as C2C-CC [1] and AUTOSAR [2], have been established, and numerous projects, including Caltrans [3], SCOOP@F [4], C-Roads and INDID [5], have been initiated or completed to further advance this field. The private sector also demonstrated significant interest, with several companies, like Fortran Traffic Systems [6] in Canada and Renault Group [7] in France, providing solutions for integrating advanced traffic management into infrastructure. Specifically, Fortran manufactures advanced traffic signals, while Renault develops intelligent connected vehicles. According to the National Safety Council, the number of fatalities resulting from car accidents reached 39,345 in 2024 [8]. This figure underscores the need for enhanced safety and traffic management, which could be provided through the services of connected cars. Trovao [9] offered an overview of future trends shaping the automotive industry, supported by statistics on revenue and environmental benefits already realized. These trends encompass the integration of digital technology in most electric vehicles, shared mobility concepts allowing short term access of multiple users to the same vehicle, increasing availability of autonomous self-driving vehicles, truck platooning, and enhanced connectivity of cars to the Internet of Things (IoT). Each of these trends is crucial for efficient resource utilization. However, their implementation necessitates the presence of certain foundational services, including cooperation and information exchange among vehicles to establish a well-connected environment before enabling additional functionalities.

The primary access technology connecting these vehicles is the Dedicated-Short Range

Communication (DSRC), based on the WiFi Alliance 802.11p standard, operating at 5.9 GHz. DSRC is standardized by the ETSI within the C-ITS as Intelligent Transportation System operating at 5GHz (ITS-G5), and by the IEEE as Wireless Access in Vehicle Environments (WAVE) [10]. Both standards (ITS-G5 and WAVE) support various communication modes in connected cars, including Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and extending to Vehicle-to-Everything (V2X). Furthermore, the integration of cellular technology (Cellular-Vehicle-to-Everything (C-V2X)) into the connected car environment is attracting considerable research interest. Nevertheless, DSRC technology alone has the potential to position connected cars as primary roadway users.

Currently, DSRC technology is being used to facilitate driver assistance applications in connected cars, such as Advanced Driver Assistance Systems (ADAS) and Cooperative Adaptive Cruise Control (CACC). Awareness services are central to cooperation and information exchange among cars; Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs) are employed for these services. Moreover, researchers are exploring next-generation cooperative services [11], proposed by the 5G Automotive Association (5GAA), to incorporate additional functionalities through new provisional use cases. These use cases, including coordinated cooperative driving maneuvers for lane merging or changing, cooperative curbside management for pedestrian pickup in crowded areas, cooperative traffic gap assistance for vehicles crossing busy roads safely, and other complex scenarios, underscore the need for connected cars to ensure enhanced driver safety and comfort.

Conversely, the advancements of connected cars increase the attractiveness of this domain to potential attackers. For instance, the 2014 Jeep Cherokee hack [12] demonstrated remote hijacking of the Control Area Network (CAN) bus. The stopping of robotaxis using traffic cones in San Francisco [13] exposed vulnerabilities in vehicle sensors. Furthermore, multiple instances of hacking autonomous car gear [14] reveal weaknesses in autonomous systems. All of these attacks pose significant threats, for example, to pedestrian safety by threatening lives or to their comfort by inducing traffic congestion. Consequently, there is a growing imperative for more robust security enhancements.

1.2 Problem Statement

Defense mechanisms are typically categorized into prevention, detection, and response. Prevention mechanisms involve proactive measures implemented to minimize the likelihood of attacks before they occur. This strategy helps reduce the attack surface and makes successful

exploitation more difficult for attackers. Detection mechanisms are employed to identify attacks when they are occurring or are about to happen. These act as early warning systems, enabling the recognition and defense against attacks in their initial phases, thereby limiting their impact. Finally, response mechanisms involve the actions and measures taken to mitigate attacks and minimize their effects on network resources and services. These actions include identifying attack sources through forensic studies to perform traceback for accurate reporting.

As C-ITS continues its evolution and deepens its integration with the surrounding environment, the potential for attacks will increase correspondingly. The inherent instability of these networks, arising from the dynamic number of cooperating vehicles based on their location and direction, further complicates security. Furthermore, the possibility of a large number of participating vehicles renders the network more vulnerable to large-scale attacks.

A cybersecurity report by Trend Micro Research [15] identified and qualitatively assessed 29 attacks on connected cars using the DREAD threat model [16]. These attacks include sniffing and spoofing V2X messages, remote hijacking and malware installation, and electronic jamming of wireless transmissions and safety systems, among others. This qualitative assessment ranked the attacks according to their severity scores. The DDoS attack, specifically targeting Intelligent Transportation System (ITS) infrastructure with the aim of disabling it or disrupting communication with nearby vehicles, received the highest score. The following research questions can be elaborated in this regard:

- **DDoS Impact:** How can one technically prove the impact of DDoS attacks with the current C-ITS infrastructure?
- **Deployment:** Where to deploy a DDoS detection solution that would take into consideration the unstable network topology of the C-ITS?
- **Interoperability:** How can we preserve interoperability with the existing infrastructures?
- **Detection Model:** How can Machine Learning be applied to detect DDoS attacks, and can ensemble learning improve detection?
- **Generalization:** How much is the solution generalizable to the evolving threat models and continuously changing/morphing attacks?

These questions lead to the contributions detailed in the next section, which aim to enhance cyberattack detection within C-ITS.

1.3 Contributions

This section outlines the contributions made to improve cyberattack detection within the connected car domain, with a specific focus on CAM-based DDoS attacks.

1- Distributed Denial of Service Impact Study: Prior to exploring countermeasures against DDoS attacks leveraging CAM messages, our initial aim was to evaluate their impact on the connected car environment. To achieve this, we investigated *"how does DDoS affect the connected cars environment"* [17] by performing various attack scenarios, adjusting the number of attackers, the use of encapsulation for messages, and changing the attackers' transmission power. We also defined our DDoS attack model that has exploited multiple On-Board Units (OBUs) to attack with a rate of 50 CAMs per second. In contrast to traditional DDoS attacks in Internet-based systems, which typically cause evident service disruptions and resource exhaustion, the impact of CAM-based DDoS attacks proved more subtle and stealthy. Specifically, we observed a phenomenon of fake congestion, where legitimate cars decreased their transmission rates in reaction to the attack. However, once the attack stopped, the system returned to normal operation, underscoring the attack's covert nature. Moreover, by monitoring the quantity and type of processed messages, we noticed that up to 92% of the traffic comprised attack-generated messages, effectively disrupting legitimate communications and filling storage resources with illegitimate data. This impact and the feasibility of carrying on such an attack motivated the subsequent research work.

2- Proposing an IDS Architecture Design: Following the assessment of DDoS attacks, a subsequent step involves proposing a suitable solution to reinforce the PKI infrastructure with an additional layer of security. This solution takes the form of a distributed IDS architecture, designed for deployment at the Multi-access Edge Computing (MEC) node within the state-of-the-art connected car architecture [18]. We further elaborate on how the IDS can leverage the MEC node's hosting capabilities and the key advantages associated with this deployment.

3- DDoS Dataset and Detection: To enhance the security of connected cars and *"shield their environment from DDoS attacks"* [18], another critical step involved creating a representative dataset of DDoS attacks to serve as the foundation for our proposed solution. We accomplished this by simulating a validated topology using SUMO software [19], derived from real-world traffic data from Google for Luxembourg City [20]. In our simulation, four Road-Side Units (RSUs) were strategically placed along a high-traffic highway section to collect CAMs. These messages were

subsequently transmitted to an IDS for analysis. For attack detection, we utilized three distinct machine learning algorithms: Random Forest, AdaBoost, and Histogram-based Gradient Boosting Classifier. Our findings indicated strong detection performance, achieving a MCC of 98%. Furthermore, we assumed all processed messages had valid certificates. Otherwise, these messages would be discarded at the RSU level during the signature verification process mandated by the PKI. To further evaluate the efficacy of the developed models under the same configuration, we expanded our study to detect more advanced DDoS attacks, specifically morphing DDoS attacks. These attacks, detailed in [17], included a JamDDoS attack, where attackers amplified their transmission power. Additionally, a Security DDoS attack was examined, where all cars securely encapsulate messages before transmission and securely decapsulate them upon reception. Furthermore, an attack combining both encapsulation and higher transmission concepts was investigated. The findings demonstrated the generalizability of our developed models against these advanced attacks, sustaining a high MCC score of 99%.

4- Challenging the Generalizability of our DDoS Detection Approach: As a final contribution to our work, we broadened our study to detect more sophisticated DDoS attacks, specifically those employing a Sybil-based model. Generally, a Sybil attacker fabricates multiple false identities to represent themself [21]; for instance, in an electronic voting system, an attacker can manipulate vote outcomes using numerous fake IDs. Particularly in connected cars, a Sybil attacker with multiple IDs can represent several cars, causing significant disruption through fake congestion or the sudden appearance of cars. In this study, we based our attack on the PseudoID change mechanism, utilizing a straightforward yet effective method to evade detection by our developed DDoS attack models. By altering only the station PseudoID field while leaving other message fields intact, the attacker can exploit this mechanism to conceal their actions and introduce a new Sybil-based DDoS attack model. Our findings indicated that after five PseudoID changes, our models became incapable of detecting the underlying attack. To mitigate this performance degradation reaching an MCC of 0.38%, we retrained our detection models and proposed an optimized data-handling strategy. By implementing an ensemble hard voting model, we achieved a balance between effective attack detection with an MCC of 73% and reduced false alarms by 53%.

1.4 Outline

This thesis comprises seven chapters.

- Chapter 2 discusses the background of C-ITS, including its components, communication media and technologies, as well as its services, security architectures, and attacks.
- Chapter 3 presents an impact study of a DDoS attack conducted using CAM messages within the connected car environment.
- Chapter 4 details the creation of a representative DDoS attack dataset and the development of three machine learning algorithms for detecting the underlying attack.
- Chapter 5 further evaluates the capabilities of our models in detecting morphing DDoS attacks and proposes an IDS architecture for connected cars.
- Chapter 6 depicts a more stealthy DDoS attack incorporating a Sybil concept that challenges our models and necessitates an optimized data-handling strategy.
- Chapter 7 concludes the work and outlines the future research directions stemming from this thesis.

Chapter 2

C-ITS Components, Communications, Services, and Challenges

2.1 Introduction

C-ITS is built upon Vehicular Adhoc Networks (VANETs), which in turn, evolved from Mobile Adhoc Networks (MANETs). VANETs introduced specialized services and technologies to the communication domain, consequently expanding the system cooperation, complexity, and associated challenges. As these services expanded, the term VANETs became inadequate to reflect the advancements, leading to the adoption of C-ITS. Throughout this manuscript, the terms C-ITS and "connected cars" are used interchangeably.

This chapter explores the various components of C-ITS and discusses its core architecture, along with advanced architectures proposed by researchers. It also examines the communication media and standards applied within C-ITS. Furthermore, it provides an overview of services offered by C-ITS and their corresponding messages. It describes the two primary security architectures and discusses the security requirements for C-ITS. Additionally, the chapter describes several inter- and intra-vehicle attacks within C-ITS, and presents countermeasures designed to prevent and detect these attacks.

2.2 C-ITS Components and Architecture

2.2.1 Components

Vehicular ad-hoc networks previously consisted of highly mobile nodes, namely vehicles, and fixed nodes located along the roadside. However, with the evolution of C-ITS, the network nodes have expanded beyond these entities. The following outlines the components within C-ITS:

- On-Board Unit (OBU): A device installed in vehicles, responsible for transmitting and receiving messages within the surrounding environment.
- Application Unit (AU): Software installed on vehicle owners' smartphones, enabling vehicle diagnostics via the device, operates outside the scope of the PKI architecture.
- RSU: A device positioned near traffic lights, responsible for transmitting and receiving messages within the surrounding environment. Its primary function is to advertise available services, such as awareness, to nearby nodes.
- Trusted Authorities: Entities (e.g., enrollment, authorization, registration, misbehavior, and certificate authorities) integrated into the infrastructure connected to RSUs, responsible for providing services such as security and cooperation within the core C-ITS network.

2.2.2 Main Architecture

All C-ITS components are integrated within the core network architecture of C-ITS, depicted in Figure 2.1. The OBU is connected to the AU via smartphone technologies (e.g., Bluetooth). Additionally, it connects to RSUs using DSRC technology and may utilize other technologies, such as 3G, 4G, or 5G and beyond, for services like Internet access. RSUs are also connected to the infrastructure for C-ITS-related service delivery.

In addition to this network architecture, a sophisticated in-vehicle network topology exists, comprising Electronic Control Units (ECUs), sensors, buses, and actuators, interconnected to ensure proper vehicle operation (Subsection 2.3.1).

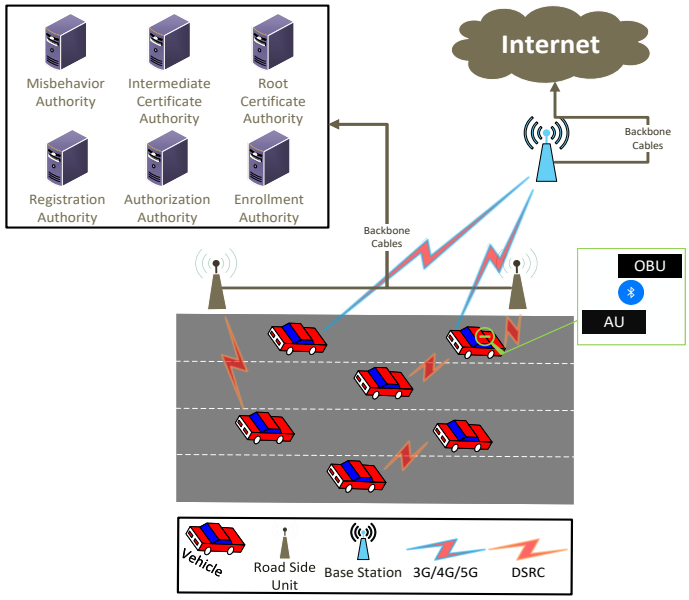


Figure 2.1: Main C-ITS architecture

2.2.3 Alternative and Advanced Architectures

In addition to the core C-ITS architecture, researchers have proposed several alternative and advanced architectures for interconnecting its various components. These proposals are characterized by a clear differentiation of architectural layers, each assigned distinct objectives. However, their practical deployment necessitates further research efforts. Detailed descriptions of the proposed architectures are mentioned below:

- Vehicle-Road-Cloud:

  Ji et al. [22] proposed a Vehicle-Road-Cloud architecture. This architecture aims to enhance the vehicular network's compatibility with evolving technologies and improve its data handling and computing capabilities. The proposed architecture (Figure 2.2) comprises four layers:

  - A security authentication layer identifies vehicles and RSUs connecting to the network.
  - A data acquisition layer collects and classifies network data, preparing it for secure transmission to the next layer.
  - An edge layer performs initial filtering and analysis of the collected data, formulating necessary decisions to minimize the load on the final layer.

– A cloud platform layer acts as the architecture's main controller, fabricating and implementing functions like data and connection management, intelligent navigation, and information security.

This proposal aims to achieve improved security, massive connectivity, and greater data throughput. However, no evaluation was conducted to validate these claims.

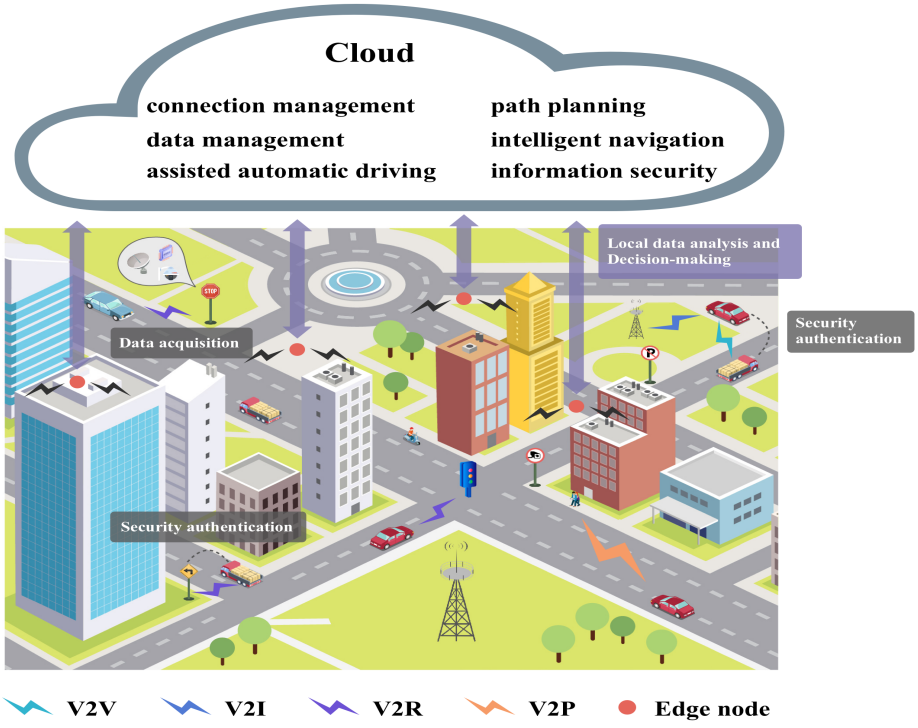


Figure 2.2: Vehicle-Road-Cloud [22]

- Cloud-VANET:

  The research in [23] addresses the limitations of inter-vehicular communication regarding low geographical coverage and network size restrictions. Specifically, it aims to mitigate broadcast storms and network disconnection issues. To this end, a Cloud-VANET architecture (Figure 2.3) is proposed, connecting vehicle resources through a cloud interface to overcome these limitations. The proposed architecture includes four layers/networks:

  – The Vehicular Cloud Network (VCN): Containing vehicles sharing physical resources.

  – The Digital Content Network (DCN): Digitizing road signboards and traffic signal lights to be transmitted to vehicles only.

  – The Infrastructure Cloud Network (ICN): Representing RSUs or enodeBs providing cloud services to vehicles.

– The Server-to-Cloud Network: Consisting of servers offering services like bandwidth management and data storage sharing.

Simulations, conducted using the SUMO traffic simulator and OMNeT++, analyzed the architecture's performance in terms of data packet loss ratio and Signal-to-Noise Interference Ratio (SNIR) loss. The results demonstrated stable network connectivity across various scenarios, effectively addressed the broadcast storm problem, and improved mobility management.

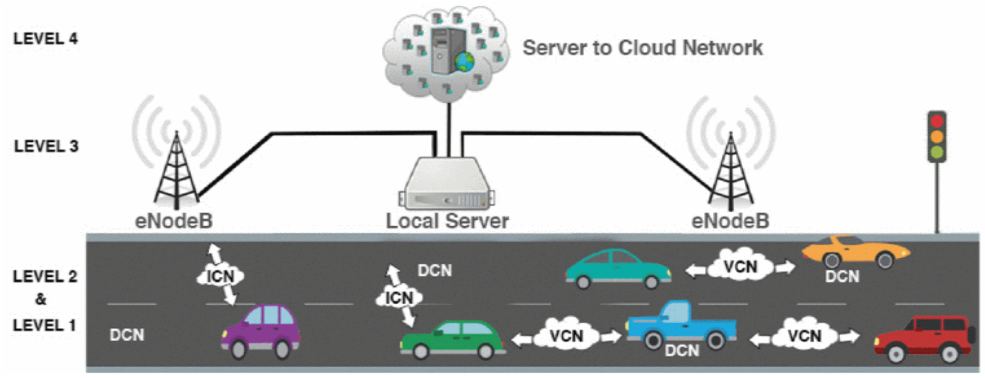


Figure 2.3: Cloud-VANET [23]

- 5G-VANET:

  Khan et al. [24] proposed an architecture (Figure 2.4) that integrates Fog Computing (FC) in a broad concept, where the network topology is divided into distinct zones, each managed by a network of SDN controllers. The architecture is detailed across three layers:

  – The Infrastructure layer: Encompassing the connected vehicle infrastructure (FC-Zone Controllers as RSUs, FC-Cluster Heads, and FC-Vehicles as the Vehicles).

  – The Cloud-RAN layer: Featuring FC-Baseband Unit Controllers that oversee FC-Zone Controllers across zones, thereby connecting the Infrastructure layer to the upper layer.

  – The Control layer: The architecture's final layer, which includes SDN controllers for fog orchestration and resource management of the entire topology.

  This distributed architecture offers advantages in flexible, programmable, and efficient resource allocation, as well as minimizing controller overhead and improving network throughput. To validate the advantages of the proposed architecture, MATLAB simulations were done to show the improved performance against other approaches in terms of throughput, delay, and the number of packets received by controllers.

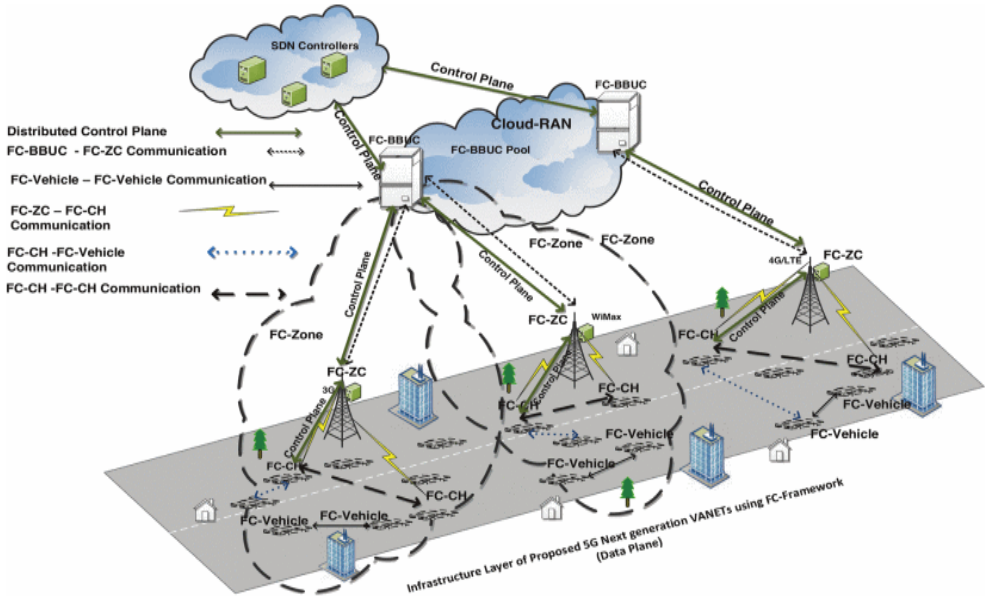


Figure 2.4: 5G-VANET [24]

- SDN-VANET:

  Liu et al. [25] designed an architecture (Figure 2.5) to integrate SDN and FC concepts into the Internet of Vehicles. This architecture consists of four layers:

  – The Data layer: Consisting of vehicles and wireless communication interfaces (RSUs, LTE base stations, and 5G small cells).

  – The Virtualization layer: The architecture's core, which abstracts various services (networking, computation, communication, and storage).

  – The Control layer: Responsible for managing the Virtualization layer by providing services from the Application layer.

  – The Application layer: The architecture's top layer, which offers applications like Data Sensing Management, ITS Management, and Road Safety Management.

  This architecture aims to enable logically centralized control and resource abstraction to maximize utilization. Two evaluation scenarios were implemented and analyzed to show the improvement in terms of end-to-end data packet delay.

- Social-VANET:

  Kerrache et al. [26] proposed an architecture (Figure 2.6) that incorporates social network profiles to enhance trust establishment within the vehicular network. This architecture is structured into three layers:

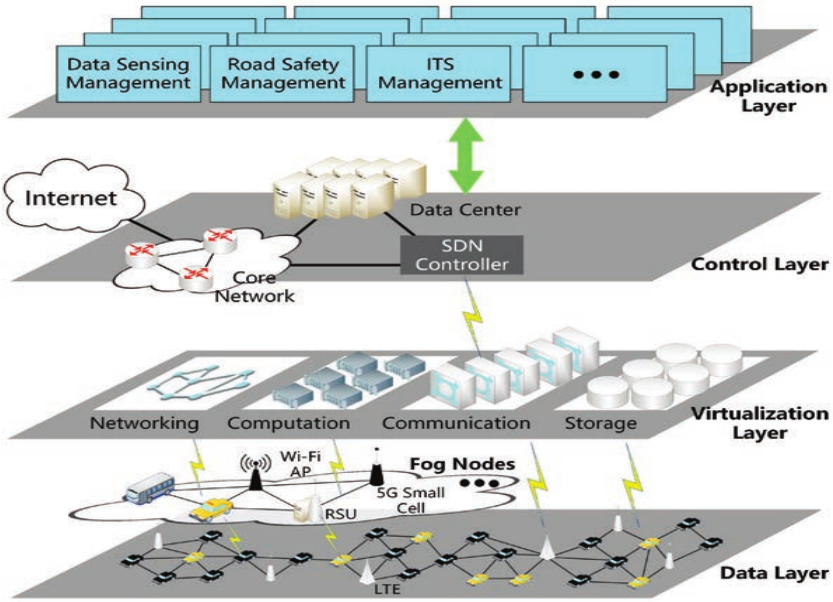


Figure 2.5: SDN-VANET [25]

- The Infrastructure layer: Consisting of vehicles and RSUs, as well as devices belonging to vehicle owners and passengers that communicate with social networks.
- The Trust layer: Consisting of trusted entities that ensure trust between devices, vehicles, and trusted authorities.
- The Social layer: Responsible for interfacing with Online Social Networks to enhance trust using human factors and location-based honesty.

While this architecture improves misbehavior detection, it introduces additional delays that could negatively impact safety applications. Evaluation was performed using the NS-2.35 simulator, focusing on comparing misbehavior detection performance with and without human factors, and benchmarking against similar approaches.

2.3 Communication Media and Standards

C-ITS employs two communication media: the internal vehicular communication medium for intra-vehicular communication and the external vehicular communication medium for inter-vehicular communication. The subsequent sections detail these mediums, specifying the various communication types within each.

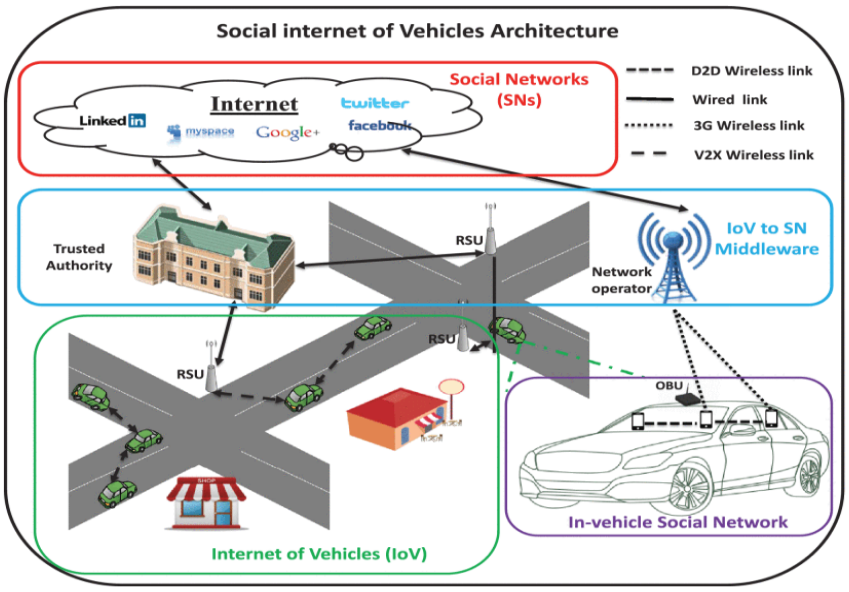


Figure 2.6: Social-VANET [26]

2.3.1 Intra-vehicular Communications

Intra-vehicular communication interconnects the various devices and units within a vehicle. These devices range from small motors for window operation to electronic control units that relay messages to perform tasks like vehicle locking or trunk opening (Figure 2.7). Such communication is essential for proper vehicle functionality and passenger safety and comfort. Furthermore, the integration of these functionalities underscores the importance of their organized structure. The various intra-vehicular communications are:

- CAN [27]: This communication is used within vehicles, with varying transmission rates for different applications (low-speed CAN for seat and window controls, high-speed CAN for engine management, brakes, and cruise control).
- Local Interconnect Network (LIN) [27]: A time-triggered, master-slave communication used for door locks, rain sensors, and door mirrors, offering enhanced security compared to CAN.
- Media-Oriented Systems Transport (MOST) [27]: A fiber-optic network communication designed for high-volume data streams in automotive multimedia.
- FlexRay [27]: A fault-tolerant, high data-rate communication enabling real-time data transmission with bounded message latency, suitable for body and powertrain systems.

- Automotive Ethernet [28]: An adaptation of the Ethernet technology, tailored to meet automotive reliability and low-latency requirements. This communication is considered the next-generation in-vehicle networking solution, offering cost, weight, and energy consumption advantages. However, its widespread implementation remains under research investigation.

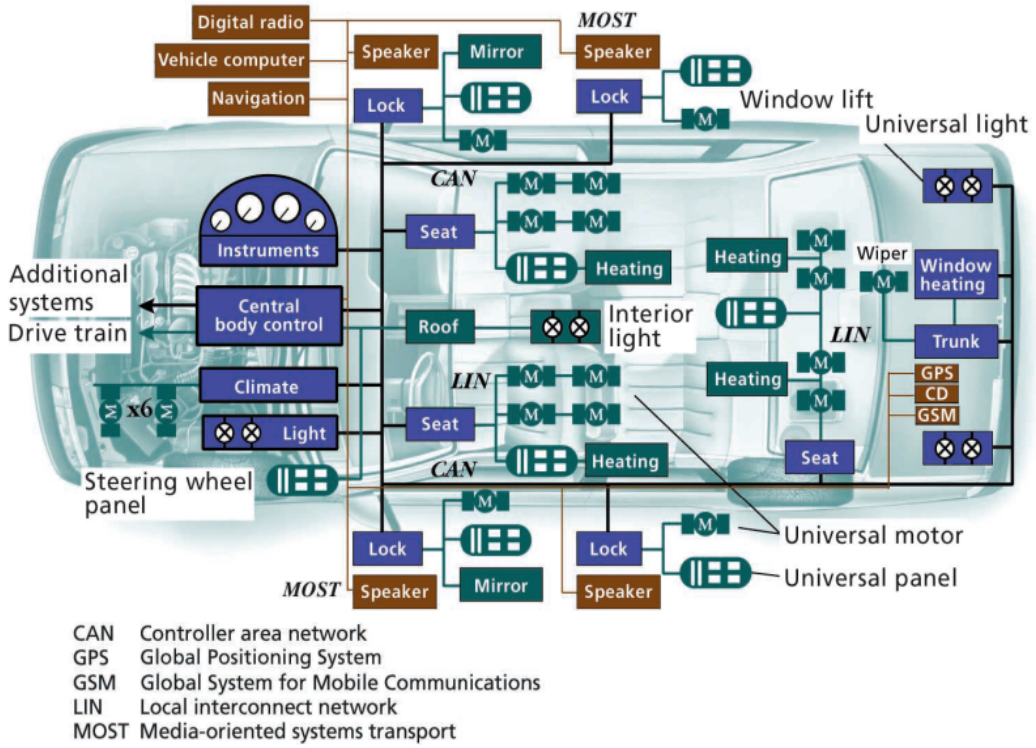


Figure 2.7: In-vehicular network [27]

2.3.2 Inter-vehicular Communications

Inter-vehicular communication facilitates the exchange of messages and information between vehicles and their surrounding environment, consisting of various communication types and technologies.

The communication types include the vehicle from one side connected to the other side that could include: vehicles as V2V, infrastructure as V2I, pedestrians via their devices as V2P/Vehicle-to-Device (V2D), and everything as V2X.

The two primary communication technologies for C-ITS are DSRC and C-V2X [10]. DSRC serves as the primary access technology, based on the WiFi Alliance 802.11p standard, enabling vehicular network communication at 5.9 GHz. It supports V2V, V2I, V2P/V2D, and most

V2X communication within the vehicular network. In contrast, C-V2X facilitates vehicle connections to cellular networks, operating at 10 MHz and 20 MHz, and supports cellular-based V2X communication, including 3G/4G/5G and beyond.

These technologies adhere to predefined standards for interoperability purposes. ETSI and IEEE have standardized DSRC in Europe and America, respectively, as ITS-G5 and WAVE standards. While each standard has unique messages and security services, they share common components in their protocol stack, as shown in Figure 2.8.

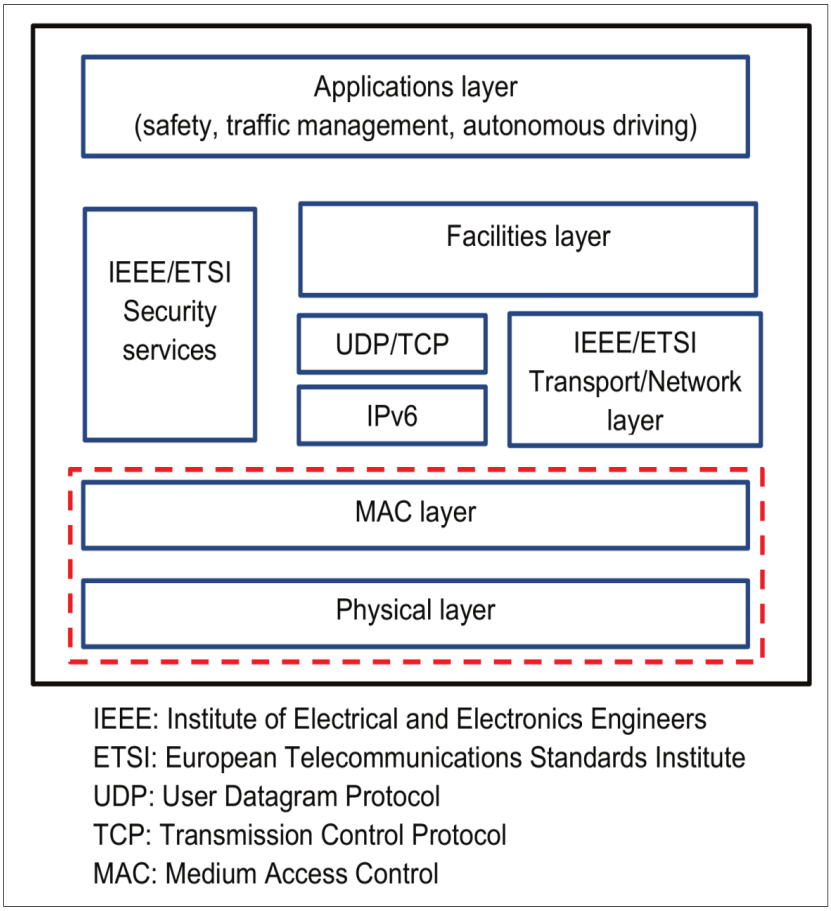


Figure 2.8: Vehicular communication protocol stack [10]

In summary, Figure 2.9 illustrates available connectivity methods for connected vehicles, emphasizing heterogeneous connectivity as a future trend. This connectivity enables infotainment services like integrated digital cockpits and seat entertainment. It also encompasses connections via Wi-Fi and cellular networks and includes V2X communications.

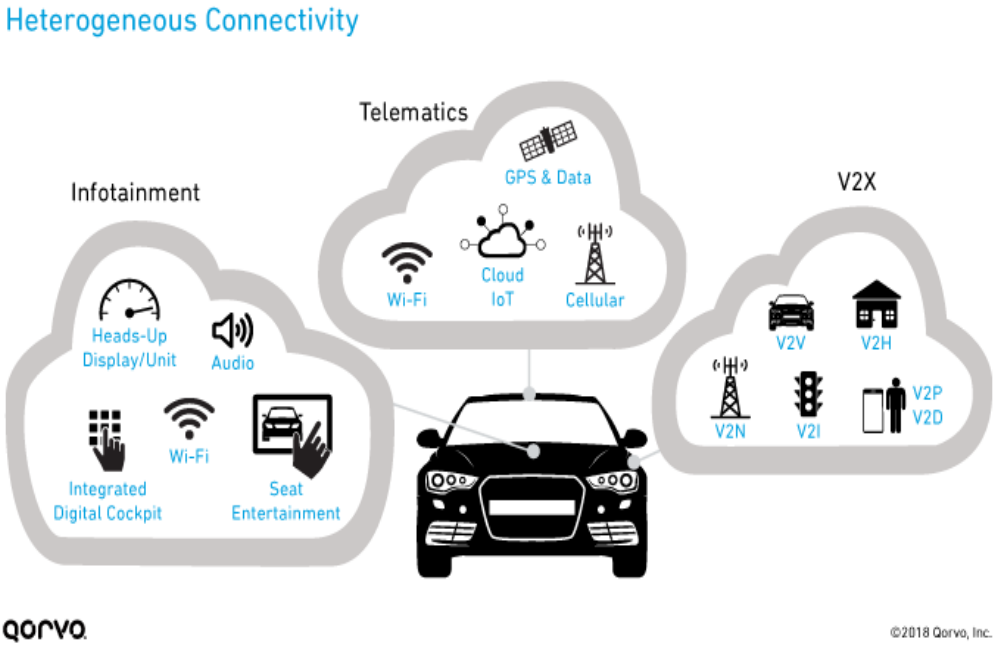


Figure 2.9: Automotive heterogeneous connectivity [29]

2.4 C-ITS Services and Messages

C-ITS provides the transportation system with a variety of advanced services, enhancing passenger comfort and safety. These services are delivered via standardized messages from ETSI/IEEE, transmitted through the vehicle communication network. The following subsections detail several of these services and their corresponding messages.

2.4.1 C-ITS Services

The name Cooperative-ITS implies a system where services are shared among vehicles to facilitate cooperation within the transportation network. We highlight several of these services, which are either implemented or under development:

- Awareness: This is central to cooperation in C-ITS, providing vehicles with essential information for safe road navigation and environmental awareness, including potential hazards. It is typically offered directly from one entity (OBU or RSU) to another.
- Perception: This collaborative service offers enhanced insights for vehicles in complex road scenarios, such as intersections or merging points. For example, an RSU could relay

messages from the opposite side of an intersection.

- Platooning: Another collaborative service, platooning links a group of vehicles, typically in a line, where the lead vehicle is in charge of guiding the following vehicles along their path.
- Infotainment: This service primarily connects the vehicle to various systems, providing both information and entertainment. For example, it enables the car to receive text messages and connect to social networks.

2.4.2 C-ITS Messages

The aforementioned services are delivered through specifically formatted messages standardized by either ETSI or IEEE. We focus in our work on the messages under the ETSI standard that have corresponding equivalents in IEEE standards.

- CAM[30]: A beacon message sent with a variable period of 0.1 seconds to 1 second, controlled by the Decentralized Congestion Control (DCC) mechanism in ETSI TS 102 687 [31]. It includes information about the transmitting vehicle, such as position and kinematics, to facilitate awareness services. It follows the format specified in ETSI EN 302 637-2 [30], and is equivalent to a Basic Safety Message (BSM) in IEEE standards.
- Decentralized Environmental Notification Message (DENM) [32]: An event-driven message triggered by detected events like road hazards or abnormal traffic conditions, also contributing to awareness services.
- Collective Perception Message (CPM)[33]: A message containing perceptions from stationary sensors mounted on C-ITS entities (RSUs or OBUs), supporting perception services.
- Platoon Awareness Message (PAM)[34]: An extension of CAM, sent with a period of 0.05 seconds to 0.1 seconds, contributing to platooning services and currently under research.

These messages are constructed, formatted, and encapsulated within the facilities layer of the ETSI protocol stack [35], as illustrated in Figure 2.10.

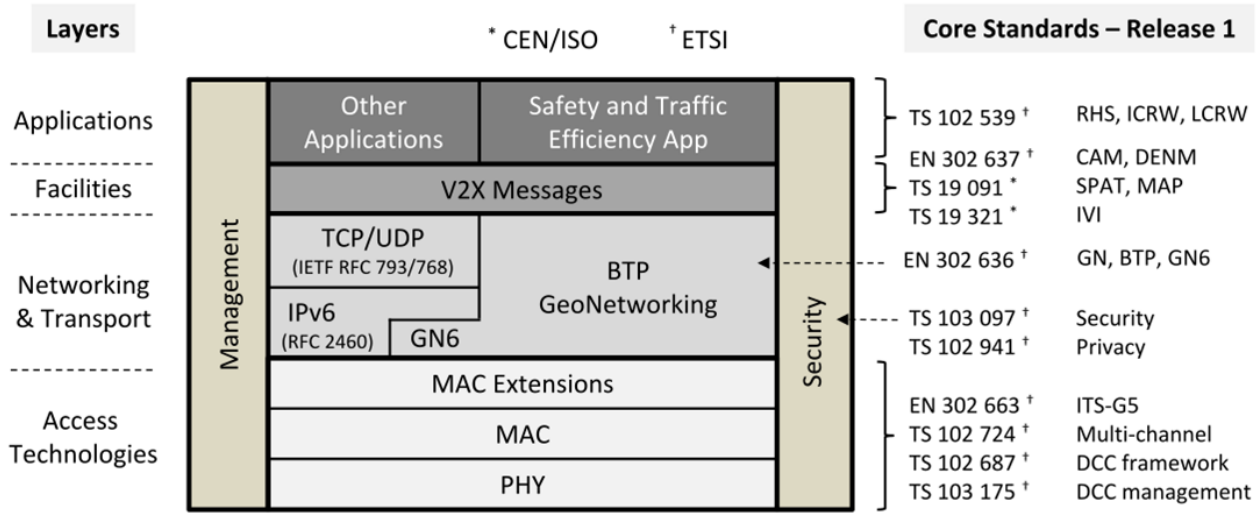


Figure 2.10: ETSI protocol stack [35]

2.5 Security in C-ITS

Given the range of services provided by C-ITS, security services are essential for a complete system. To this end, ETSI and IEEE offer distinct security architectures that manage message security and authenticate C-ITS entities through certificate issuance. The following subsections detail these architectures and discuss the security requirements for C-ITS. In relation to the C-ITS components discussed earlier (Subsection 2.2.1), this security architecture corresponds to the "Trusted Authorities" interconnected to form the PKI architecture.

2.5.1 ETSI Security Architecture

Figure 2.11 illustrates the PKI architecture proposed by ETSI, as detailed in their Technical Specification (TS 102 940) report [36], which includes:

- Enrollment Authority (EA): Responsible for enrolling new vehicles into the system and issuing their enrollment security certificate.
- Authorization Authority (AA): Responsible for issuing and verifying authorization tickets, which permit C-ITS entities to participate in specific services (awareness, perception, platooning, etc.).

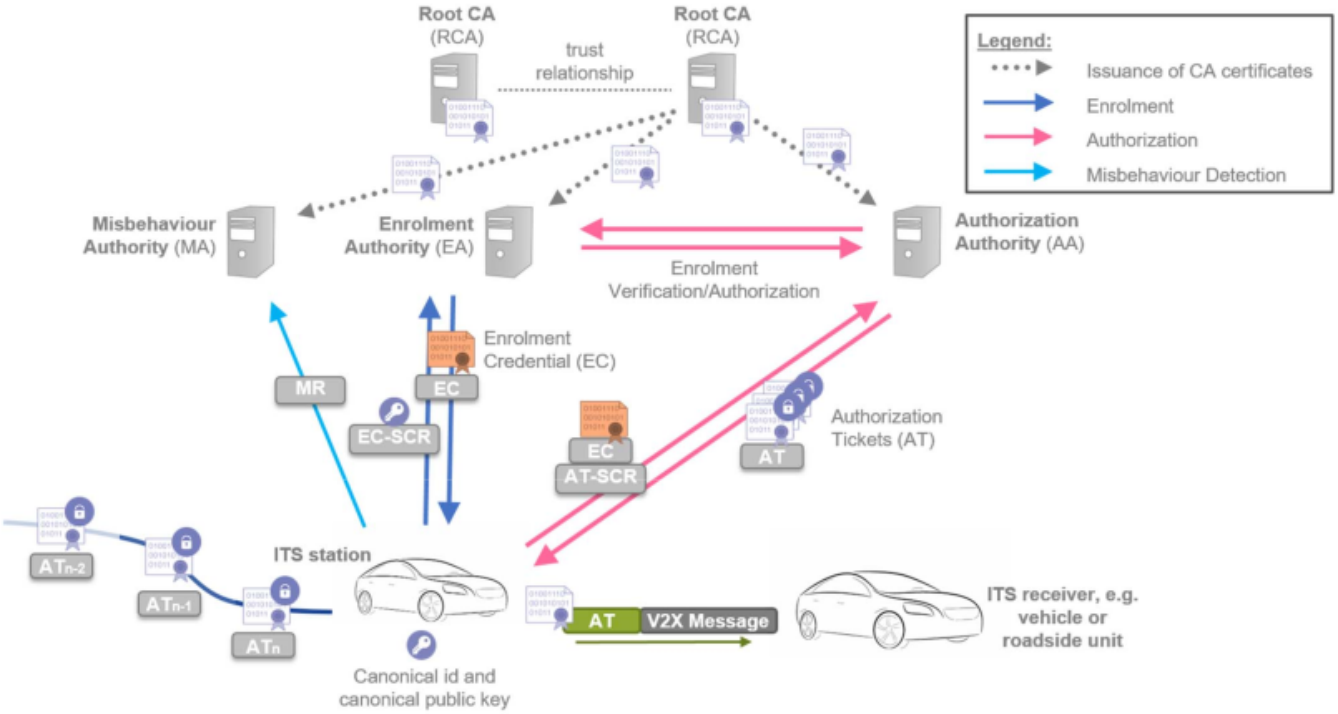


Figure 2.11: ETSI PKI architecture [36]

- Root Certificate Authority (RCA): Responsible for issuing certificates to intermediate certificate authorities, including EA and AA. This entity primarily operates offline due to the significant cost of its compromise to network administrators.
- Misbehaviour Authority (MA): Responsible for processing misbehavior reports from C-ITS entities and taking appropriate actions against misbehaving entities (e.g., reporting their certificates for PKI revocation). This entity was not addressed in release 1 [37] but is under development in release 2 [36], with the reporting service detailed in TS 103 759 [38].

2.5.2 IEEE Security Architecture

In contrast, IEEE proposes a different security architecture, illustrated in Figure 2.12, which includes the PKI as described in [39]. This architecture includes entities similar to those in ETSI's architecture:

- Security Credential Management System (SCMS) manager: The primary manager responsible for defining and enforcing security policies applied by Root Certificate Authorities (CAs).
- Enrollment CA (ECA): Responsible for issuing enrollment certificates.

- Authorization CA (ACA): Responsible for issuing authorization certificates.
- Root Certificate Authority (CA): Similar to the ETSI PKI, responsible for issuing certificates to other Intermediate CAs, including ECA and ACA, and operating primarily offline.
- Registration Authority (RA): Responsible for processing certificate requests from C-ITS entities and forwarding them to ECA and ACA.
- Misbehavior Authority: Responsible for receiving and analyzing misbehavior reports from C-ITS entities to identify misbehaving entities. This authority may request the RA to revoke certificates of misbehaving entities.

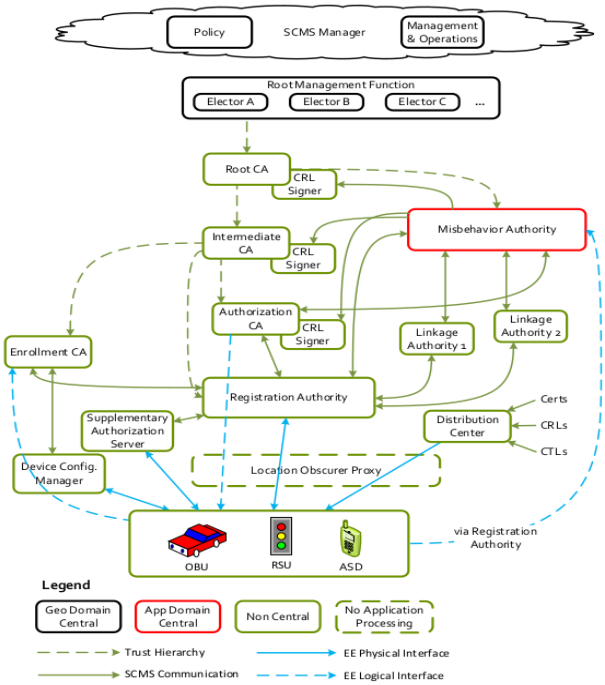


Figure 2.12: IEEE PKI architecture [39]

2.5.3 Security Requirements

The proposed security architectures aim to protect C-ITS from unauthorized entities participating in specific C-ITS services. However, not all C-ITS services necessitate adherence to the security architecture. For example, Internet access may only require a standard browser certificate, rather than vehicle authorization certificate. Conversely, like any system, C-ITS is vulnerable to insider attacks where entities with valid certificates act maliciously. Consequently, the correct functioning of C-ITS necessitates various security requirements [40], including:

- Availability: Ensuring system responsiveness and availability during peak and off-peak periods.
- Integrity: Maintaining the unmodified state of information from source to destination.
- Confidentiality: Restricting information access to authorized users.
- Authentication: Verifying that received messages originate from valid senders.
- Non-repudiation: Enabling receivers to prove message originality and preventing senders from denying message transmission.
- Replay Protection: Ensuring the uniqueness of information to prevent malicious replays.
- Privacy: Limiting communication to reveal only intended information.

2.6 Attacks on C-ITS

Publicly accessible systems are inherently more vulnerable to various attacks. C-ITS is not an exception, given that it inherently involves public use of its products and network. These attacks can originate from external sources, such as when a vehicle provides Internet access, or from internal sources, where a malicious vehicle using a C-ITS service transmits false data to nearby vehicles. Attacks can also occur onboard, such as when an embedded vehicle device starts acting maliciously. While attacks are generally classified as active or passive, global or local, and internal or external, we categorize C-ITS attacks into intra-vehicular and inter-vehicular attacks, which are detailed in the following subsections.

2.6.1 Intra-vehicular Attacks

Intra-vehicular attacks target vulnerabilities within the vehicle's internal systems, particularly its in-vehicle network (Figure 2.7). Recent examples include:

- Attacks on the CAN [41]: For instance, Lee et al. [42] describe an attack that can disrupt CAN bus signal transmission by using a resistor to lower differential voltage. Arapantonis [43] also demonstrated data tampering on the CAN bus.

- Malware attacks [44]: These involve injecting malicious code (viruses, Trojans, or backdoors) into a vehicle's system, potentially through direct access, Over-The-Air (OTA) updates, or mobile device-to-vehicle communication, with motives ranging from financial gain and vandalism to vehicle theft.

2.6.2 Inter-vehicular Attacks

Inter-vehicular attacks originate from outside the vehicle, either from other vehicles or external sources. Examples include:

- Vehicle sensor attacks [41]: Targeting sensors like Light Detection and Ranging (LiDAR) or Radar, these attacks aim to spoof the presence of physical objects, as demonstrated in [45] (LiDAR) and [46] (Radar).
- Remote hacking: Attackers seek remote access to vehicle platforms, exemplified by the 2014 Jeep Cherokee hack [12].
- Jamming: Attackers disrupt communication channels with noise, overloading resources. This can severely impact applications like cooperative adaptive cruise control, as shown in [47].
- Denial of Service (DoS): Attackers overload C-ITS network resources (communication, storage, processors, etc.). This can escalate to a DDoS if multiple attack sources are controlled, as detailed in [17].
- Black/Grey hole attack: Attackers fully or selectively drop forwarded messages, as described in [48].
- Sybil attack: Attackers impersonate multiple identities, as discussed in [21]. In C-ITS, this involves creating virtual vehicles by sending messages with different certificates.

Research laboratories and companies continuously seek countermeasures against these attacks. While some attacks seem complex to implement in a vehicular context, rapid technological advancements make it uncertain when or where severe vehicular network attacks may occur.

2.7 Countermeasures in C-ITS

Initially, vehicles were designed solely for transporting people and goods. However, with the advent of connected cars and the need for C-ITS networks to provide services like traffic optimization and safety, implementing countermeasures to protect C-ITS from attacks and prevent their occurrence became essential. We will explore these countermeasures in two primary forms: security by design, which aims to proactively prevent attacks, and intrusion detection systems, which aim to detect attacks as swiftly as possible. Naturally, a comprehensive countermeasure strategy includes the accurate reporting of detected attacks. While this can be achieved through traceback and forensics studies applied to the generated reports, this specific area of investigation falls outside the scope of our work.

2.7.1 Security by design

Security by design includes precautions and measures taken to protect a system from external and internal threats. Examples include:

- Access control: In C-ITS, PKI helps to provide access control through delivering and managing digital certificates, which defines vehicle roles and authorizes access to services.
- Usage of encryption: Applied to sensitive data within the C-ITS network, such as Authorization Ticket (AT) requests. However, not all data is encrypted, as some, like CAM messages, require accessibility without decryption.
- Cyber physical systems: ETSI has released technical specifications [36] for C-ITS, outlining the use of Hardware Secure Module (HSM) in vehicles for key storage and cryptographic functions.
- Zonal in-vehicle network segmentation: Zonal architecture [49] groups ECUs by functionality, simplifying communication and isolating zones.
- Conceptual/Theoretical frameworks: Ongoing research explores security frameworks for in-vehicle networks. For example, Piao et al. [50] proposed an Elliptic Curves Cryptography (ECC)-based framework to protect FlexRay networks from cyberattacks, and Nilsson et al. [51] developed a framework for the firmware updates over the air (FOTA) for the ECUs, which verifies update integrity and detects tampering attempts.

2.7.2 Intrusion Detection Systems

Intrusion Detection Systems (IDSs) are implemented in two primary forms, complementing the security by design measures in C-ITS. These forms are Host-based IDS (HIDS) and Network-based IDS (NIDS) [52]:

- HIDS: Monitors network traffic on individual vehicles, potentially distributed within the in-vehicle network to protect specific zones. An example is outlined in the AUTOSAR technical report [53], detailing distributed IDS requirements for vehicles, shown in Figure 2.13.
- NIDS: Monitors network traffic between vehicles, providing a broad network perspective, distributable across geographic domains, and capable of centralized network behavior aggregation.

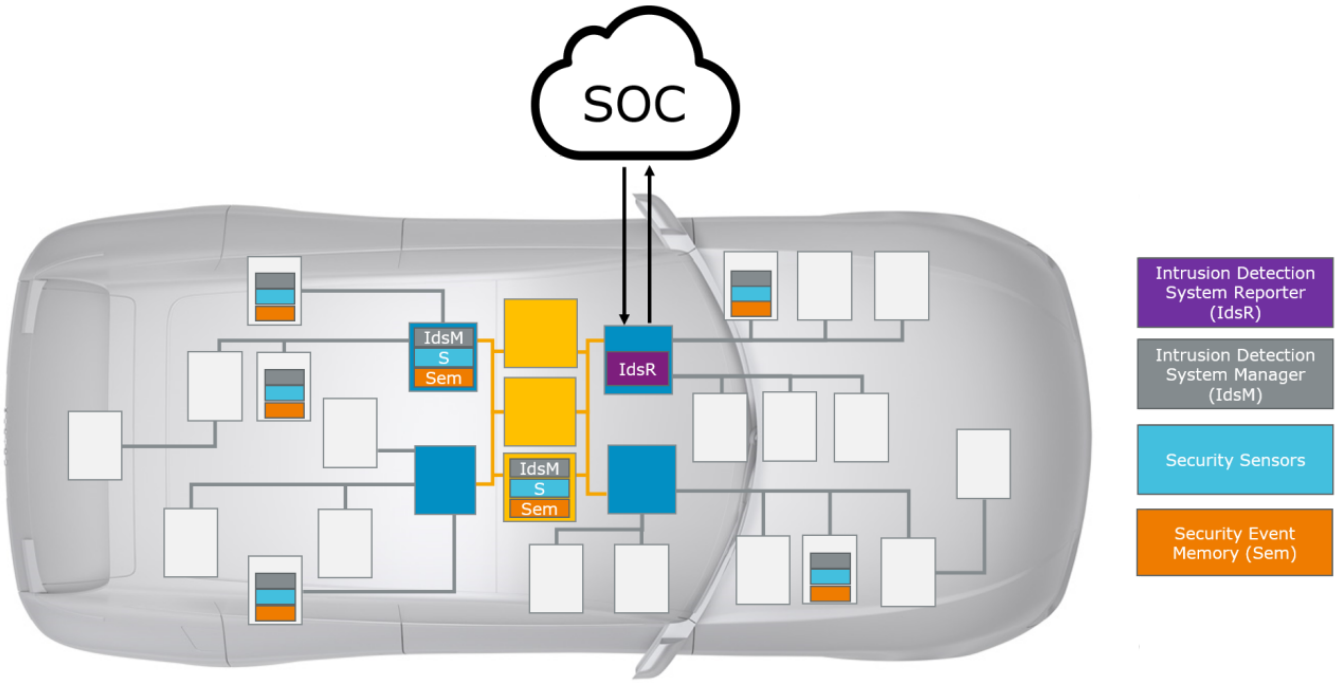


Figure 2.13: Example architecture of a distributed onboard IDS [53].

Companies are increasingly focusing on HIDS development due to the rapid advancement of C-ITS and growing consumer demand for Internet-related connectivity services in vehicles. Conversely, NIDS remains largely in the research phase, as cooperative vehicle implementation (e.g., platooning [54]) is still under investigation. This focuses our work on NIDS in the C-ITS network.

2.8 Conclusion

This chapter has provided an overview of the C-ITS domain, detailing its components and core architecture. It has also examined the communication media that facilitate C-ITS services through the messages they employ. Furthermore, the chapter has presented the security architectures governing the C-ITS domain, alongside its security requirements. Finally, it has discussed potential inter- and intra-vehicle attacks within the system, in addition to possible countermeasures.

C-ITS has undergone significant evolution over the past decade. With technological advancements, services are now widely accessible across diverse platforms, including smartphones, laptops, and other connected devices. This broad connectivity exposes C-ITS to large-scale attacks. While research has extensively addressed numerous attack types, large-scale attacks have received comparatively limited attention, partly due to the complexity of their implementation in vehicular networks. Nevertheless, we believe that proactive security measures are crucial to mitigate potential consequences. To address the challenge of large-scale attacks, the subsequent chapter will analyze the impact of DDoS on vehicular networks before proposing novel solutions. We propose a simulation-based evaluation study to assess the effects of large-scale attacks under various scenarios and conditions.

Chapter 3

How Does Distributed Denial of Service Affect the Connected Cars Environment?

3.1 Introduction

The DDoS attack is primarily characterized by its distributed nature, originating from multiple, geographically dispersed sources, which significantly complicates detection by IDSs. Furthermore, these attacks can be executed across diverse communication domains. Within the C-ITS domain, this can involve either Internet-specific or C-ITS-specific messages. A review of recent studies [55, 56, 57, 58] shows that while the effectiveness of DDoS attacks targeting Internet traffic, particularly User Datagram Protocol (UDP) packets associated with infotainment services, has been demonstrated, there is limited research on the impact of these attacks on safety services using critical safety messages. Therefore, a thorough analysis is needed to show the effects of DDoS attacks on safety services. To address this, we simulated various DDoS attack scenarios using CAM messages to gain a deeper understanding of their impact on connected car systems. This analysis facilitates the identification of system vulnerabilities and the evaluation of system responses to such attacks.

This chapter provides an overview of DDoS attacks. It also reviews related work and highlights the contributions of this study. Furthermore, it details the simulation scenarios and analyzes the obtained simulation results.

3.2 Distributed Denial of Service Attacks

3.2.1 Principle and Mechanism

Cyberattacks can be categorized as either passive or active. Passive attacks involve the collection of system information without disrupting normal operations, whereas active attacks directly interfere with system functionality. DDoS attacks, characterized by the overwhelming of target systems with excessive traffic, are generally categorized as active attacks. However, they frequently commence with a passive reconnaissance stage, during which attackers gather intelligence on the target system's vulnerabilities. This understanding of system weaknesses enables attackers to launch more effective and targeted DDoS attacks. This dual nature, initiating with passive reconnaissance and progressing to active disruption, positions DDoS attacks as a hybrid form of cyberattack.

A potential scenario for executing a DDoS attack within the connected car domain, which presents a more chaotic nature compared to classical targeted DDoS attacks, can be outlined in the following steps:

1. The attacker exploits vulnerabilities in specific OBU models, as exemplified by the case of the 2014 Jeep Cherokee hack [12].
2. The attacker deploys malware onto these devices, potentially through OTA updates or via a backdoor from the AU.
3. The attacker initiates the DDoS attack, either through a remotely executed command or a scheduled mass attack, using the exploited OBU models.

3.2.2 Types of DDoS Attacks

DDoS attacks can be executed through various methods and at different protocol layers. For instance, in Internet-based attacks [59], these layers include the Application, Transport, Network, Data Link, and Physical layers. Within the connected cars' protocol stack (Figure 2.10), the corresponding layers are the Application, Facilities, Networking and Transport, and Access Technologies [35]. In our work, we focus on the Facilities layer, where a range of message types can be employed for attacks. While we primarily consider CAM messages, alternative messages like CPM, which share information about sensor-detected objects in a car's vicinity, can also

be used. For both message types, DDoS attacks could be differentiated but not limited to the following types:

- **Normal DDoS**: known simply as DDoS, where the attacker modifies its sending rate to be higher than allowable rates without interfering with the message contents (Figure 3.1).
- **Varying DDoS**: the attacker keeps varying its sending rate, and it is considered an advanced version of a DDoS attack.
- **Random DDoS**: the attacker modifies the content of the messages in a random way.
- **Replay DDoS**: the attacker replays old messages received from different or specific cars (Figure 3.2).
- **Sybil DDoS**: the attacker modifies the identity field in the message before sending it, either to impersonate another car or only to mask their identity.
- **Hybrid DDoS**: the attacker could choose to alternate between different types making its attack more advanced.

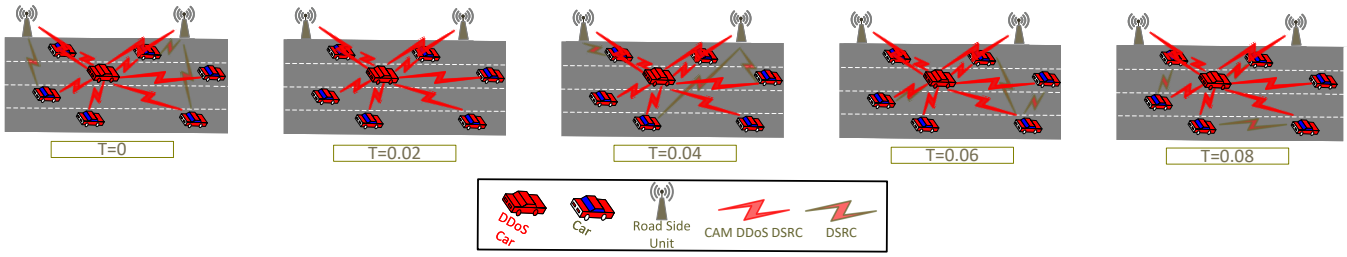


Figure 3.1: DDoS attack with a rate 5 times the normal rate of transmission of CAM messages by cars.

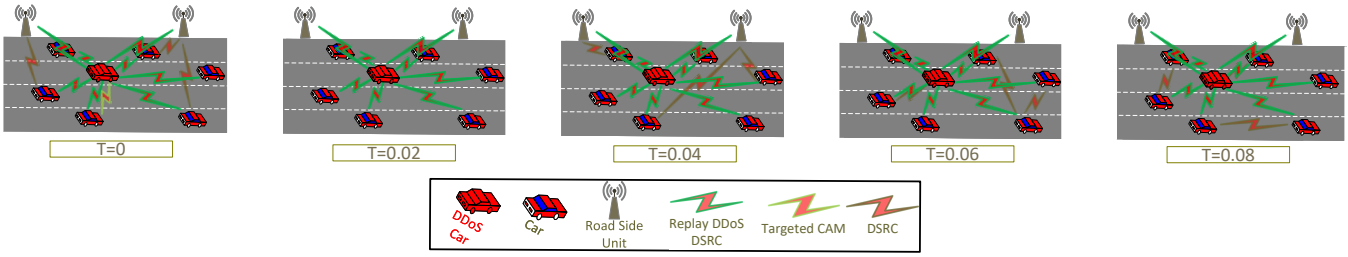


Figure 3.2: Replay DDoS attack with a rate 5 times the normal rate of transmission of CAM messages by cars.

3.2.3 Potential Targets

Like every cyberattack, DDoS attacks can be tailored to target different entities, network layers, services, or resources. Some potential targets in the domain of connected cars include:

- Entities:
  - Cars which include the software and hardware parts.
  - Infrastructures which includes cellular (3G/4G/5G and beyond) or DSRC network provider.
  - Manufacturers of cars, mostly for competitive intentions.
- Network layers:
  - Application layer, affecting infotainment and navigation systems.
  - Network layer, disrupting network routers to slow down communications and data transfers.
  - Data link layer, affecting physical devices by jamming wireless signals.
- Services:
  - Cooperative awareness services, affecting the cooperation between cars.
  - Infotainment services, disrupting their delivery.
  - Navigation services, affecting navigation devices.
- Resources:
  - Car's computing resources, degrading car's responsiveness for safety-critical applications.
  - Network bandwidth, affecting data transfer.
  - Storage resources, filling it with illegitimate data.

In our case, our DDoS attacks focus on the DSRC-related infrastructure of the connected cars, targeting the application layer that provides the cooperative awareness services. This affects storage resources that will be filled with illegitimate attacker messages.

3.2.4 Modeling

To model a DDoS attack performed on the infrastructure, i.e., on the RSU from a botnet of cars in a connected car environment, we make the following assumptions:

- The attack is performed in an environment where cars communicate with the RSUs for various services.
- The RSUs are responsible for handling and processing messages from cars.
- The attacker has already compromised certain cars by hacking into their OBUs or exploiting some vulnerabilities to obtain access to the so-called botnet of compromised cars.

Using any type of DDoS attack, the attacker's objective remains to overwhelm the RSU by flooding it with a large volume of messages causing a disruption or denial of service. Considering the assumptions mentioned above, and the overall objective of degrading or denying a service, we can define the attack parameters and derive the following mathematical model.

$$A_i = N * \lambda \tag{3.1}$$

$$A_v = A_i * T \tag{3.2}$$

where

$A_i =$ Attack intensity
$N =$ Number of cars in the botnet
$\lambda =$ Attacking rate (number of messages per second)
$A_v =$ Attack volume
$T =$ Attack duration in seconds

At some specific time, the attacker triggers the compromised cars to initiate the attack for a duration of T seconds. According to the number N of compromised cars and considering the pre-set rate of attack λ, we can compute the attack intensity A_i, using eq. (3.1), as the number of messages per second. Considering the intended attack duration T, we obtain the attack volume (A_v by eq. (3.2)) that the botnet would generate.

On the other hand, a RSU has limited processing capacity [60], and can only handle a maximum number of messages per second, denoted as μ. This value depends on the specific hard-

ware and software of the RSU and cannot be generalized as Table 3.1 shows with the varying characteristics of RSUs. In the domain of connected cars, it is sufficient to have a degradation of service to consider an attack successful specifically when considering safety-critical services. However, a DDoS attack is fully successful when the attack intensity A_i exceeds μ.

Table 3.1: RSU specifications according to the 5G Automobile Association (5GAA) [61], X: Not available.

RSU company	**Capacity of verifying messages per second**	**Capacity of signing messages per second**	**Reference**
YUNEX	4000	130	[62]
Unex Technology Corp	>2500	>110	[63]
Commsignia	>2000	X	[64]
NeoGLS	X	X	[65]
V2ROADS	X	X	[66]
Q-Free	X	X	[67]

3.2.5 Consequences of DDoS Attacks on Connected Cars

The inherently chaotic nature of a DDoS attack targeting connected cars can manifest in several direct and indirect consequences, including:

- Compromised Cooperative Functionality: A platoon subjected to a DDoS attack may transition from cooperative adaptive cruise control to adaptive cruise control, thereby losing cooperative advantages.
- Physical Safety Risks: Accidents may occur due to compromised control of connected cars under DDoS attacks.
- Social Trust Erosion: Public confidence in connected car technology may diminish due to perceived risks.
- Economic Impact: Corporate stock values may decline as a result of product recalls, legal liabilities, and reputational damage.

The following section presents related work that has analyzed DDoS attacks in various environments and using different metrics. Subsequently, we position our work to highlight our specific contributions.

3.3 Related Work

In [55], the authors explored the effect of DDoS attacks on the Software-Defined Internet-of-Vehicles (SD-IoV) control plane. The DDoS attack model included an attacker sending malicious flows generated by the "hping3" traffic generator, using random source and destination IPs, simulated in a highway scenario. Their attacks neutralized the average throughput of legitimate stations in certain cases, and were able to overload the computing and storage resources with attack messages. However, the messages used corresponded to the infotainment service, and the evaluation metrics were limited to the throughput of legitimate stations and the load on the controller.

Hosny et al. [56] demonstrated DDoS attacks within the fifth Generation-Internet of Vehicles (5G-IOV) domain. The simulated scenario represented an urban road in Alexandria, Egypt; however, traffic was randomly generated using the SUMO [19] simulator. Although the evaluation of the attack impact encompassed various metrics, the study was limited to 20 cars, with three of them designated as attackers flooding UDP messages. Consequently, the study lacked representation of real-world traffic demand and was constrained by the limited number of cars.

Malik et al. [57] conducted an experimental study on a platoon of connected cars in an urban scenario to investigate the effects of DoS/DDoS attacks. They quantified the attack's impact by measuring the number of received broadcast messages per node, the total packet loss rate, and the number of affected connected cars. However, the study did not provide detailed insights into the attack's impact on the specific types of messages received.

Baharlouei et al. [58] analyzed the influence of factors that enhance the realism of connected car simulations, including the importation of network topologies from real cities, the implementation of realistic communication settings (e.g., transmission range and node count), and the utilization of empirical mobility patterns. These factors were evaluated under DDoS attacks. While the authors demonstrated the impact of topology variations across four cities, the study focused on Address Resolution Protocol (ARP) requests, which are not utilized for safety services within connected car environments.

In [68], the authors demonstrated a DDoS attack utilizing advanced vehicular botnets capable of autonomous relocation to amplify attack effectiveness. The investigated DDoS attack targeted congestion response messages, which are transmitted upon requests for congestion information on a designated urban road segment. The evaluation metrics primarily focused on physical effects, including average speed, number of cars on the targeted road, and overall trip time per car. However, the study did not incorporate metrics to assess the network impact.

The authors of [69] analyzed a DDoS case study through a simulation of a highway street in Abu Dhabi, United Arab Emirates. Their analysis focused on transmitted and received broadcast messages, as well as the busy time of each node. While the study utilized broadcast messages, a characteristic of safety message transmission, the specific message type was not explicitly identified. However, the attack did not yield significant packet loss, due to the limited number of attacking cars (three).

Our work validates DDoS attacks through simulations on a highway scenario, primarily by observing elevated Channel Busy Ratio (CBR) values (representing the fraction of time a radio channel is occupied with transmissions, as defined in [70]) compared to normal scenarios. Furthermore, we analyze the composition of received messages to quantify the proportion of attack messages and the denial of legitimate messages, thereby confirming the attacks' efficacy. The subsequent section details the simulation scenarios and parameters employed to demonstrate the impact of DDoS attacks on connected car environments.

Table 3.2 summarizes the related works in terms of objective, metrics used, results, and limitations.

Table 3.2: Comparison of related works

Reference	Goal	Metrics used	Results	Limitations
[55]	Exploring the effect of DDoS attack using Internet Protocol (IP) packets on the control plane of SD-IoV	• Throughput • Controller load	• Neutralization of the throughput of benign nodes • Overloading of the computing and storage resources with attack messages	• Evaluated two performance metrics • Car's traffic was randomly generated • Studied a highway scenario
[56]	Demonstrating a DDoS attack in 5G-IoV using UDP packets	• Number of sent packets • Number of received packets • Packet loss • Packet delay • Queue length of gNodeBs	• High level of packet loss • Increase in packet delay • Increase in the number of packets in the queues	• The work did not explore different number of studied cars • Car's traffic was randomly generated • Studied an urban scenario
[57]	Conducting an experimental study of the effect of DoS/DDoS attack on the platoon of smart cars	• Number of received messages • Total packet loss rate • Number of cars affected by the attack	• High level of packet loss • More than half of the cars where affected by the attack	• No detailed analysis about the type of received messages • Studied an urban scenario
[58]	Analyzing the effect of changing topologies studied, communication settings, and mobility patterns on the simulation of a DDoS attack using ARP packets	• Average number of ARP requests sent • Number of dropped packets	The formation of the topologies studied had a higher effect on the effectiveness of the attack compared to communication settings and mobility patterns.	• Evaluated two performance metrics • Car's traffic was common across all topologies
[68]	Demonstrating the effect of a VANET-based botnet attack using congestion response messages on the usability of the targeted road	• Average speed of the targeted road • Number of cars on the targeted road • Trip time of each car	• Significant decrease in the average speed of the targeted road • Increased the trip time of normal cars	• No evaluation for network impact • Studied an urban scenario
[69]	Analyzing a DDoS study using messages that are broadcasted	• Number of sent broadcasts • Number of received broadcasts • Busy time of each node	No significant effect of the attack was presented	• No packet loss induced • Studied a manually made and a highway scenario
Our proposed approach	Demonstrating the impact of DDoS attacks performed using CAMs	• Rate of received messages • Rate of processed messages • Rate of unprocessed messages • Percentage of denied messages • Percentage of attacker messages • CBR • Sending rate	• The storage resources are filled with illegitimate data • Participating cars are tricked to contribute less to the services provided	Studied a highway scenario

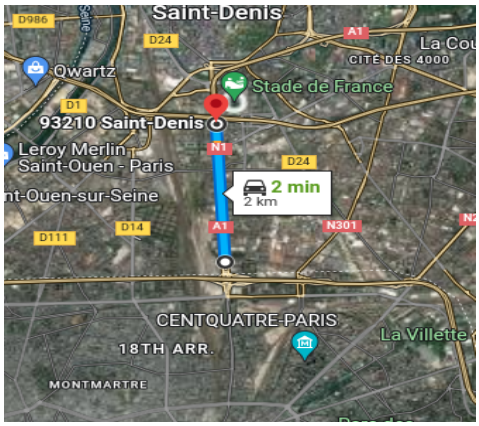
3.4 Simulation Scenario Sets and Parameters

This section presents the simulation environment for connected cars, the use case under study, and the various simulation scenario sets employed.

Simulating a real-world environment necessitates the accurate representation of its key characteristics through appropriate software. In the context of connected cars, these characteristics include the traffic model, network manager, and standard message formats. For our simulations, SUMO [19] was utilized for traffic modeling, given its widespread adoption in connected car simulations. OMNeT++ [71] was selected as the network manager due to its ability to provide synchronized coupling between SUMO and Artery [72], the software chosen for standard message formatting. Artery is a vehicular simulation framework based on ETSI ITS-G5 protocols, such as GeoNetworking and Basic Transport Protocol (BTP), ensuring that transmitted CAM messages conform to the standard format [30].

The use case considered in this work corresponds to the traffic demand of the $A1$ highway between Paris and Lille in France [73], specifically a two-kilometer section originating from the Paris side. This section is considered a potential target for attackers due to the high volume of traffic (Figure 3.3), particularly during peak hours. For this use case, several simulation scenario sets were defined, each comprising multiple simulation scenarios. These scenarios share a common concept, as depicted in Figure 3.4, where an attacker (represented by a red car) broadcasts CAM messages on the ITS-G5 control channel, bypassing the implemented DCC mechanism [31]. The selected DCC mechanism (Reactive 1ms, limiting normal cars to a 10 CAMs/sec transmission rate) monitors the CBR and imposes new transmission rate restrictions when a CBR threshold (defined in [31], illustrated in Figure 3.12b) is exceeded. This mechanism reduces message transmission rates to prevent network congestion. For the duration of the attack, the DCC mechanism was disabled for attackers, enabling them to sustain attacks successfully. Their transmission rate was set to five times the normal rate (50 CAMs/sec), a rate chosen to balance attack effectiveness with stealth. The transmission power was set to the default $200mW$ [74], providing a 1 km DSRC range. To capture all transmitted messages, the RSU was positioned at the midpoint of the two-kilometer highway, equidistant from both ends. The Control Channel (CCH) in the ITS-G5A frequency band was used, consistent with the transmission of safety messages [75]. Table 3.3 summarizes the common simulation parameters.

The following subsections detail the various simulation scenario sets, including their conditions, parameters, objectives, and performance metrics.



(a) Section view (Google maps)



(b) Color-Coded A1 car statistics

Map legend.	
In black	More than 150,000 vehicles per day.
In red	Between 100,000 and 150,000 vehicles per day.
In orange	Between 80,000 and 100,000 vehicles per day.
In yellow	Between 60,000 and 80,000 vehicles per day.
In green	Between 40,000 and 60,000 vehicles per day.
In light gray	Missing data.

(c) Color codes

Figure 3.3: Scenario statistics

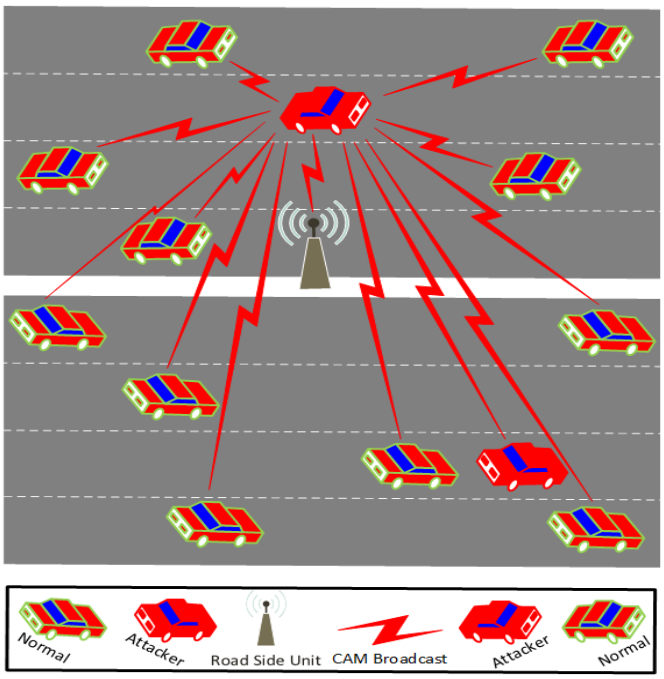


Figure 3.4: Scenario sketch

Table 3.3: Common simulation parameters

Parameter	Value
Simulation time	180 sec
Attack duration	30 sec (from 90 to 120)
Normal message rate	10 CAMs/sec
Attacker message rate	50 CAMs/sec
Normal transmission power	200 mW (default)
DSRC range	1 km
DCC mechanism	Reactive (1 ms)
Channel name	CCH
Lane length	2 km
Speed limit	110 km/h
Departure speed	Speed Limit
RSU position	1 km from both ends

3.4.1 Simulation Scenario Set I

This simulation scenario set comprises three simulations designed to differentiate between two DDoS attack types and a normal scenario. The first simulation models a DDoS attack aimed at overwhelming the RSU with CAM messages at a rate of 50 messages per second from 20 attackers. The second simulation, termed "JamDDoS," introduces an advanced attack that not only floods the RSU with messages at the same rate but also utilizes a transmission power of 2 $Watts$ (the maximum allowed value according to [74], without performing the classical jamming concept). This attack seeks to exploit the radio model implemented in Artery, where radios prioritize signals with higher transmission power. The simulations included 110 cars, with 20 designated as attackers in the attack scenarios. Table 3.4 details the parameters for this and the subsequent simulation scenario set.

Table 3.4: Simulation scenario set I & II parameters

Parameter\Case	Normal	DDoS	JamDDoS
Average normal cars	90	90	90
Average attack cars	20	20	20
Attack transmission power	200 mW	200 mW	2 W

3.4.2 Simulation Scenario Set II

In this simulation scenario set, the parameters and simulations from the preceding set (Section 3.4.1) were retained, with the addition of the security option provided by Artery [76]. While

this option does not implement a full PKI in our simulation, it accounts for the security overhead associated with sending and receiving secure messages. The certificates transmitted by the connected cars are pre-trusted and validated. However, security encapsulation is performed prior to each transmission, and security decapsulation upon message reception. Consequently, the introduction of this overhead increases the occupancy of the communication medium due to the augmented packet size, thereby amplifying the effectiveness of the DDoS attack.

3.4.3 Simulation Scenario Set III

This simulation scenario set includes five simulations designed to evaluate the impact of varying the number of hijacked cars relative to a normal scenario. Across all simulations within this set, the total number of connected cars was maintained at approximately 110, adhering to the traffic demand statistics from [73]. To uphold the hijacking paradigm for each attack scenario, the number of hijacked cars was incrementally increased while proportionally decreasing the number of normal cars. Table 3.5 details the corresponding parameters.

Table 3.5: Simulation scenario set III parameters

Parameter\Case	Normal	DDoS20	DDoS30	DDoS40	DDoS50
Average normal cars	90	90	80	70	60
Average attack cars	20	20	30	40	50

3.4.4 Simulation Scenario Set IV

This simulation scenario set examined the impact of increasing the number of attackers within the studied road segment. Consequently, the total number of connected cars varied across simulations. The number of normal cars was consistently maintained at approximately 110, while the number of attackers was incrementally increased. Alternatively, this scenario can be interpreted as an increase in the attackers' transmission rate, assuming a constant number of attacker cars. Table 3.6 details the corresponding parameters.

Table 3.6: Simulation scenario set IV parameters

Parameter\Case	Normal	DDoS20	DDoS30	DDoS40	DDoS50
Average normal cars	110	110	110	110	110
Average attack cars	0	20	30	40	50

3.4.5 Performance metrics

The following performance metrics were used to evaluate the simulation scenario sets conducted in this work:

- RSU level:
  - **Rate of received messages:** number of received messages per second.
  - **Rate of processed messages:** number of processed messages per second.
  - **Rate of unprocessed messages:** number of unprocessed messages per second.
  - **Percentage of denied messages:** the ratio of the number of unreceived messages in the attack scenario to the number of received messages in the normal scenario.
  - **Percentage of attacker messages:** the ratio of the number of received attack messages to the number of total received messages in each scenario.
  - **CBR:** this represents the fraction of time when a single radio channel is busy with transmissions [70].
- Cars level:
  - **Sending rate:** number of sent messages per second.

Table 3.7 presents a summary of the objective of each simulation scenario set in addition to the performance metrics used per set.

3.5 Results and Analysis

The simulation scenario sets were evaluated following a 70-second "warm-up" period, employing various metrics at multiple levels. At the RSU level, the number of received and processed messages was monitored, enabling the calculation of received, processed, and unprocessed message rates. Additionally, the analysis of processed message types during attacks facilitated the evaluation of the percentage of denied normal messages relative to normal scenarios, using eq. (3.3) for simulation scenario sets I, II, and IV. Furthermore, the percentage of attacker messages relative to total received messages during the attack period was calculated using eq. (3.4) for simulation scenario sets I, II, and III. In addition, the CBR (provided by an Artery module) was recorded at the RSU to provide an overview of the attack's impact on the surrounding environment.

Table 3.7: Comparison between simulation scenario sets

Simulation scenario set number	Goal	Performance metrics used
I	Compare the effect of DDoS and JamDDoS* attacks	• Rate of received messages • Rate of processed messages • Rate of unprocessed messages • Percentage of denied messages • Percentage of attacker messages
II	Enable the security feature and investigate its impact under DDoS and JamDDoS attacks	• Rate of received messages • Rate of processed messages • Rate of unprocessed messages • Percentage of denied messages • Percentage of attacker messages
III	Analyze the effect of having different numbers of hijacked cars	• Rate of received messages • Rate of processed messages • Rate of unprocessed messages • Percentage of attacker messages
IV	Analyze the effect of sending more attacker cars	• Percentage of denied messages • CBR • Sending rate

*JamDDoS: Attackers' transmission power increased to 2 Watts

At the car level, the sending rate of nearby cars was monitored to assess the attack's impact on the DCC mechanism. Consequently, in each simulation scenario set, we evaluated specific metrics that highlight the effect of the studied attack.

$$Rn = (1 - \frac{N_x}{N_{Normal}}) * 100 \tag{3.3}$$

$$Ra = \frac{A_y}{T_y} * 100 \tag{3.4}$$

where

$Rn =$ Percentage of denied normal messages

$Ra =$ Percentage of attack messages

$N =$ Number of normal messages

$A =$ Number of attack messages

$T =$ Total number of messages

$$x = \begin{cases} DDoS, JamDDoS & \text{Scenario sets I\&II} \\ DDoS(20, 30, 40, 50) & \text{Scenario set IV} \end{cases}$$

$$y = \begin{cases} Normal, DDoS, JamDDoS & \text{Scenario sets I\&II} \\ Normal, DDoS(20, 30, 40, 50) & \text{Scenario set III} \end{cases}$$

3.5.1 Simulation Scenario Set I

DDoS and JamDDoS attacks were initiated between 90 and 120 seconds of simulation time. To evaluate their impact, Figure 3.5 illustrates the rates of messages received, processed, and unprocessed by the RSU. In the normal scenario, the number of received messages (Figure 3.5a) averages approximately 1000 $msgs/sec$, corresponding to roughly 110 cars broadcasting CAM messages at a variable rate of 10 CAMs/sec (or less), regulated by the DCC mechanism. Conversely, during the attack period in both DDoS and JamDDoS scenarios, the number of received messages significantly increased, attributed to 20 attackers broadcasting CAM messages at 50 CAMs/sec and 90 normal cars broadcasting CAM messages at rates determined by the DCC mechanism, which depends on the CBR value measured by each car's transmitter.

Notably, the RSU received fewer messages in the JamDDoS scenario compared to the DDoS scenario. This is attributed to the more pronounced effect of JamDDoS on reducing the transmission rate of all surrounding normal cars, coupled with the RSU prioritizing attackers' transmissions. For instance, in the DDoS scenario, the average of $1450\ msgs/sec$ is derived from attackers (averaging $20 * 50 = 1000\ msgs/sec$) and normal cars (averaging $90 * 5 = 450\ msgs/sec$). In the JamDDoS scenario, the average is slightly lower, as attackers' messages are more reliably received, whereas normal messages are not. The final simulation scenario set further demonstrates the DCC mechanism.

Figure 3.5c depicts the number of unprocessed messages. In the normal scenario, this number fluctuates around 75 $msgs/sec$, reflecting the inherent loss of some transmitted messages due to interference or bit errors. However, the DDoS attack scenario exhibited a maximum of 220 unprocessed $msgs/sec$, as expected due to the high transmission rate inducing interference and bit errors. Conversely, the JamDDoS attack scenario showed fewer unprocessed messages, even compared to the normal scenario. This outcome is anticipated, as messages with higher transmission power are less susceptible to interference and, consequently, less likely to be unprocessed.

Figure 3.6 presents the additional metrics (Rn, Ra) analyzed during the attack period. From Figure 3.6a (depicting Rn), it is observed that the DDoS attack resulted in the denial of 36% of normal messages that would have been transmitted in a normal scenario, while JamDDoS demonstrated a greater impact, denying 52%. From Figure 3.6b (illustrating Ra), the percentage of attack messages in the DDoS scenario was 61%, indicating that over half of the received messages during the attack were attack messages. Furthermore, Ra for the JamDDoS scenario reached 71%, approximating 75% of the received messages. Concerning the radio model employed, it was anticipated that the JamDDoS attack would yield more substantial results, as

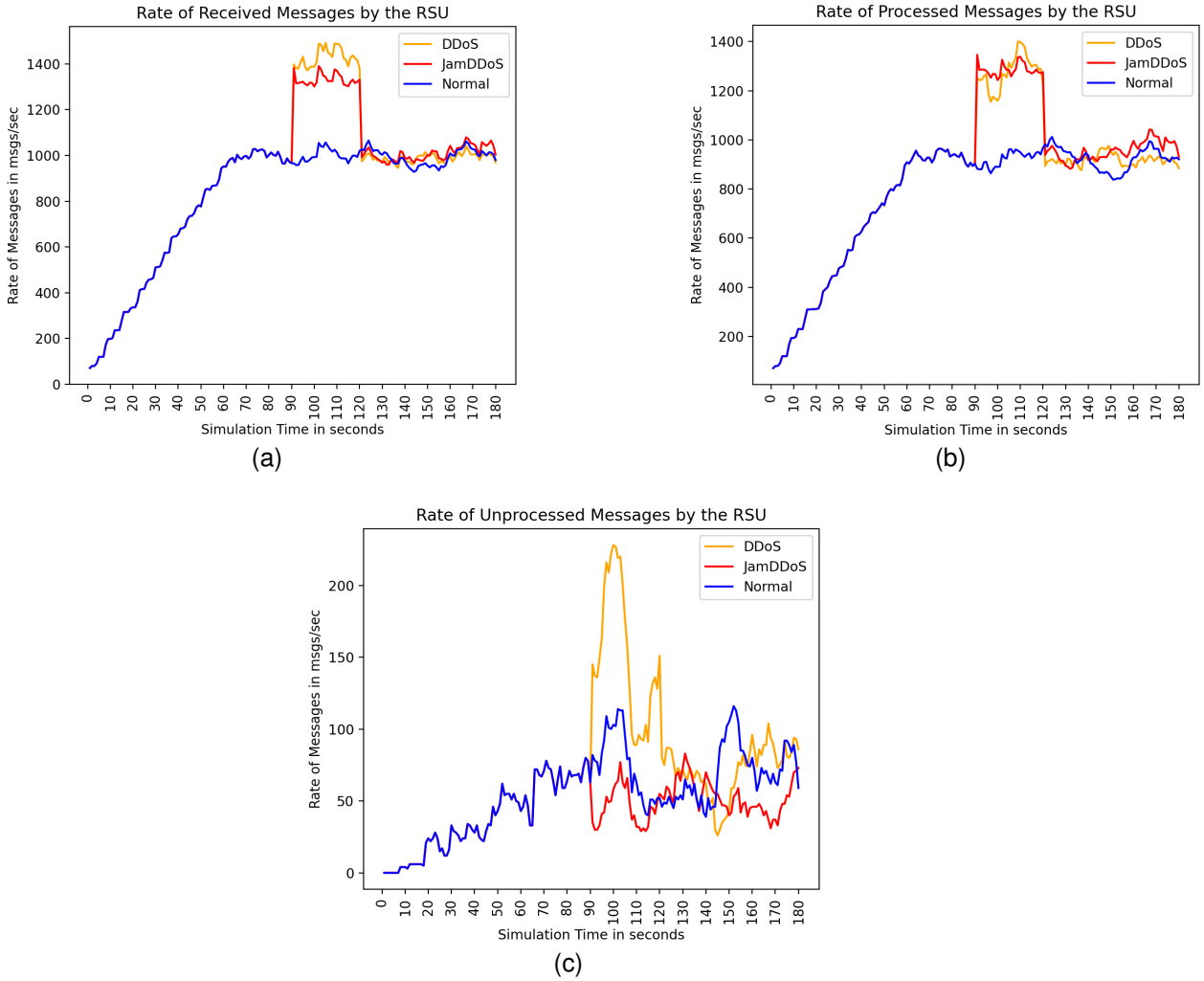


Figure 3.5: Simulation scenario set I metrics (1)

the radio prioritizes signals with higher transmission power, which in this case are the attacker messages. In conclusion, this simulation scenario set demonstrates that a DDoS attack not only denies the reception of normal messages but also saturates storage resources (e.g., temporary storage devices used for congestion level calculations) with attacker messages. Moreover, the use of higher transmission power by attackers (JamDDoS) amplifies the attack's effectiveness and severity.

3.5.2 Simulation Scenario Set II

The inherent trade-off between security and performance presents a persistent challenge. In this simulation scenario set, the impact of enabling security for V2V/V2I communications is analyzed. It is observed that the number of received messages decreased by approximately 350

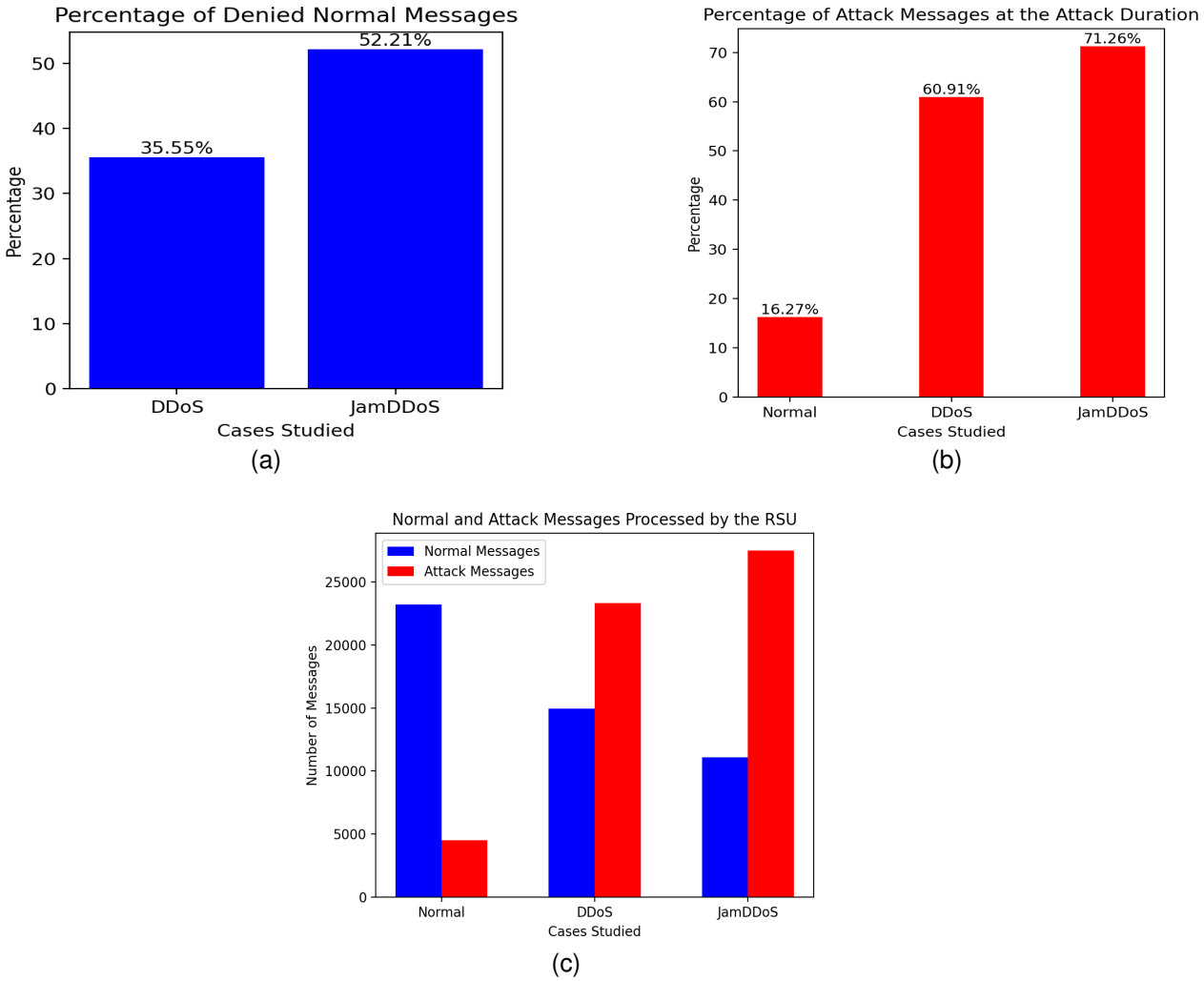


Figure 3.6: Simulation scenario set I metrics (2)

$msgs/sec$ (Figure 3.7a) for both normal and DDoS scenarios. However, for the JamDDoS scenario, the number of received messages decreased by 200 $msgs/sec$ during the attack. These reductions are attributed to the increased transmission time required for signed messages (message plus security overhead), resulting in higher CBR values. Consequently, cars are compelled to reduce their transmission rates, leading to a decrease in the number of messages received by the RSU. The 150 $msgs/sec$ drop difference (350-200) between the attack scenarios is justified by the JamDDoS attack's enhanced reliability in ensuring the reception of attack messages compared to the DDoS attack. Upon analyzing Figure 3.7c, we observe that, in the normal scenario, the introduction of security overhead augmented the likelihood of messages being unprocessed. This is due to the increased transmission time of signed messages, thereby elevating the probability of messages being affected by interference. While this likelihood is diminished in the DDoS scenario, the number of unprocessed messages remains a bit higher than in the absence of security. In the JamDDoS scenario, this likelihood is negligible, as the number of unprocessed

messages remains consistent (approximately 55 $msgs/sec$) during the attack. These findings indicate that the DDoS and JamDDoS attacks did not significantly exploit the overhead introduced by security measures.

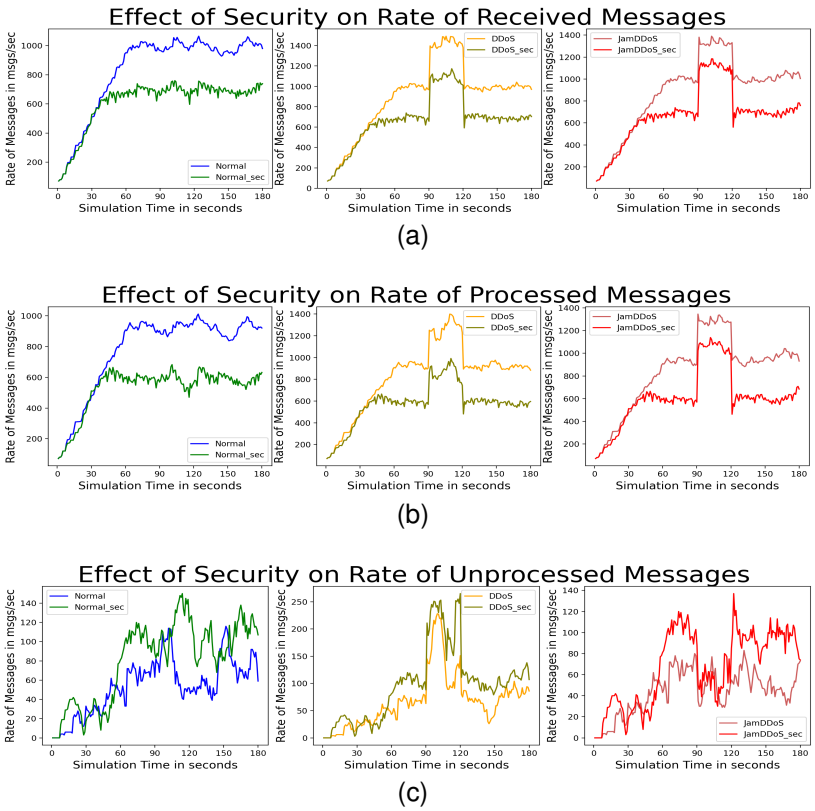


Figure 3.7: Simulation scenario set II metrics (1)

Upon analyzing the remaining two metrics depicted in Figure 3.8, it is observed that the DDoS attack resulted in the denial of 57% of normal messages (Figure 3.8a), contradicting previous observations and highlighting the enhanced impact of the DDoS attack. In the JamDDoS scenario, this effect is more detrimental, with 69% of normal messages denied. Figure 3.8b illustrates the Ra metric, indicating that approximately 75% of messages received by the RSU during the DDoS attack were attack messages. For the JamDDoS attack, this metric reaches 85%, rendering the security option ineffective against JamDDoS attacks. Furthermore, we argue that the increased percentage of attack messages is attributable to the JamDDoS attack exploiting the radio model's vulnerability. Consequently, the security overhead inadvertently facilitated a more pronounced impact of the attackers on the connected car environment.

3.5.3 Simulation scenario set III

The objective of this simulation scenario set was to investigate the impact of increasing the number of compromised or hijacked cars launching a DDoS attack. Figures 3.9a and 3.9b

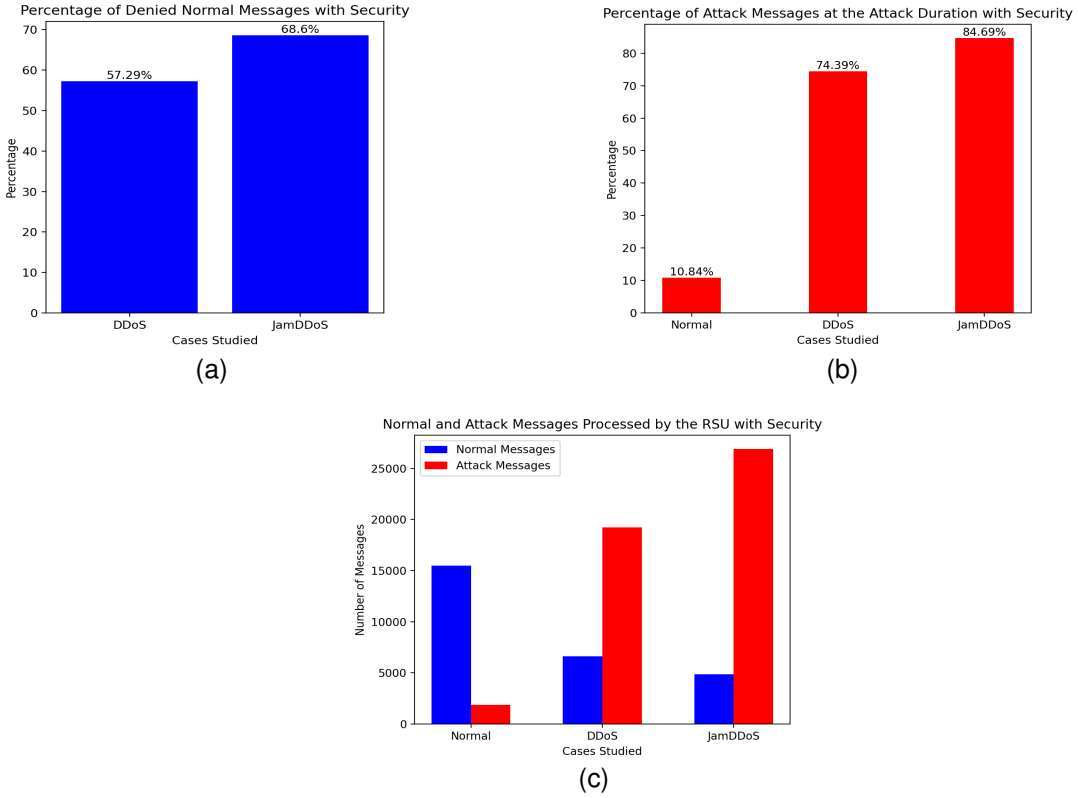


Figure 3.8: Simulation scenario set II metrics (2)

demonstrate that the number of received and processed messages increases in correlation with the number of hijacked cars. Nonetheless, the number of unprocessed messages (Figure 3.9c) exhibits a maximum of 100 unprocessed $msgs/sec$ in the normal scenario, which doubles to 200 $msgs/sec$ when 20 attackers are present. As the number of hijacked cars broadcasting CAM messages at a rate of 50 $msgs/sec$ increases, the unprocessed message rate reaches a maximum of 400 $msgs/sec$ with 50 hijacked cars.

To evaluate the severity of increasing the number of hijacked cars, the Ra metric was calculated for each scenario. Figure 3.10 demonstrates that, in a normal scenario, the percentage of messages originating from the 20 cars presumed to be hijacked would be 16%. However, this percentage rapidly escalates to 61% once these cars are hijacked. This indicates that over half of the received messages are from hijacked cars, potentially illegitimate data. Furthermore, an increment of 10 hijacked cars results in a 15% increase in illegitimate data, reaching 76%. Moreover, considering 50 hijacked cars (approximately half of the total cars), the result is catastrophic; 92% of received data being useless. These findings underscore the profound impact of hijacking legitimate cars on the utility and quality of received messages.

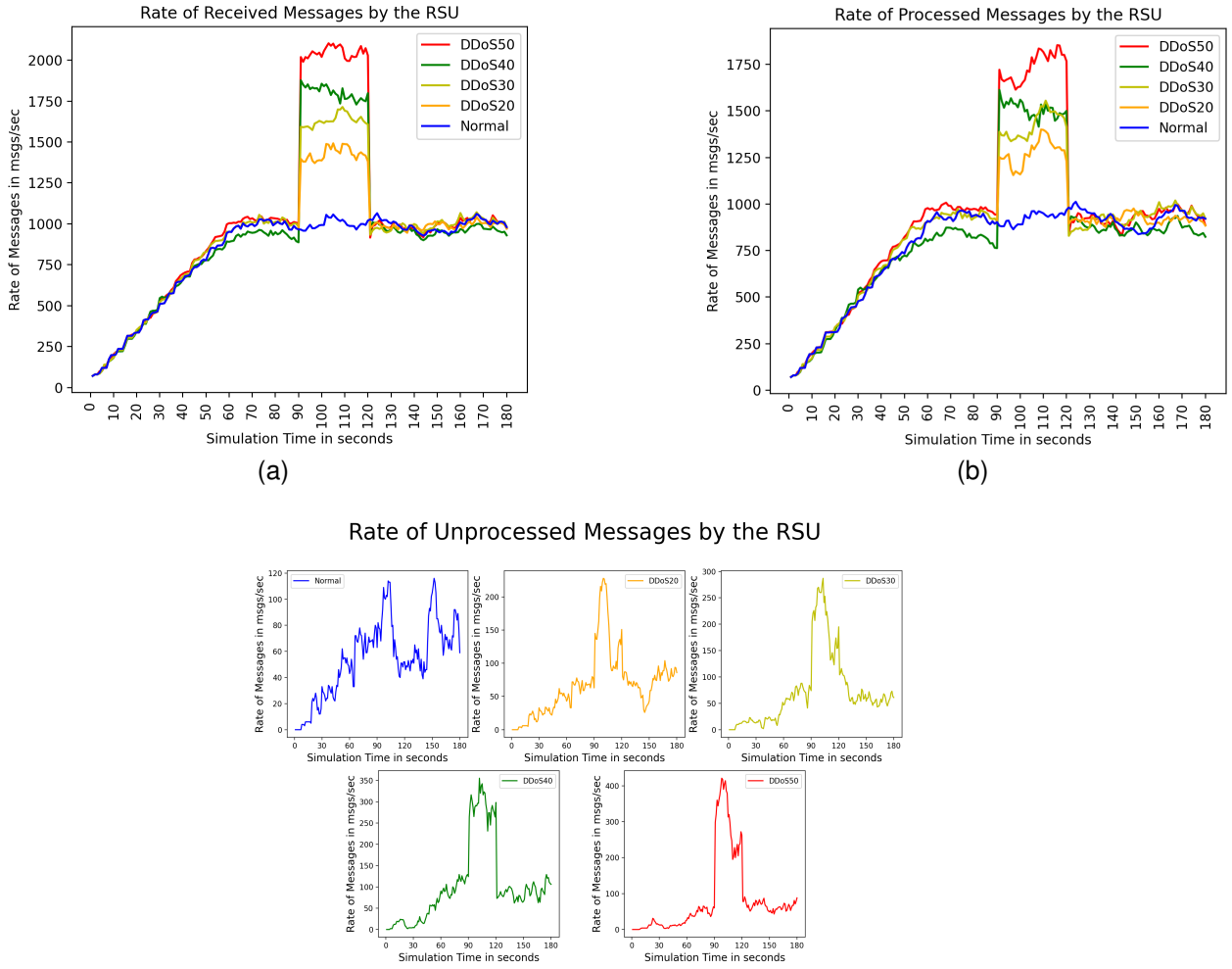


Figure 3.9: Simulation scenario set III metrics (1)

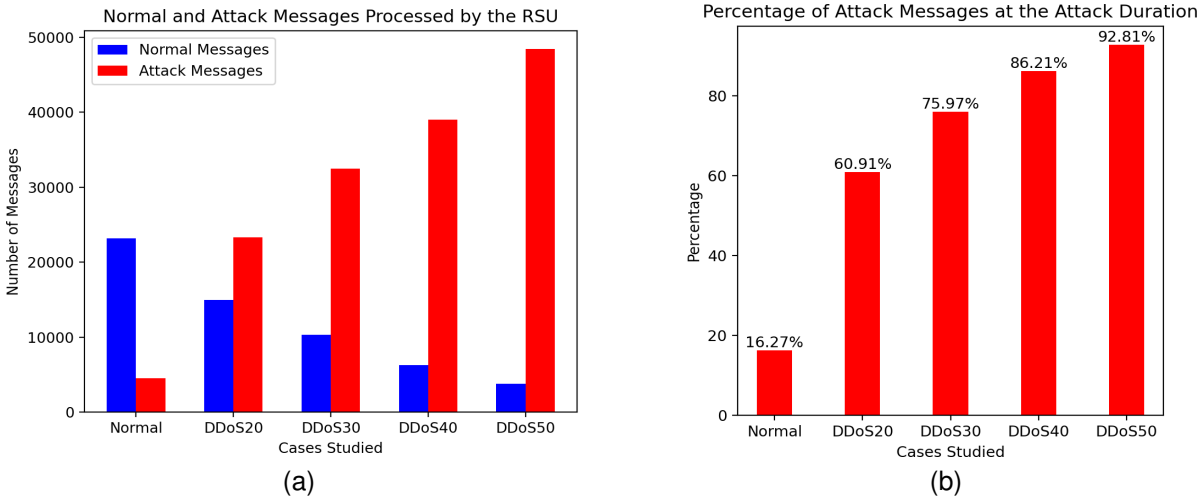


Figure 3.10: Simulation scenario set III metrics (2)

3.5.4 Simulation Scenario Set IV

As previously detailed, this simulation scenario set examined the impact of increasing the number of attackers within the studied road segment, while maintaining a constant number of normal cars. In terms of received, processed, and unprocessed messages, the outcomes were analogous to those presented in Figure 3.9, with a marginal increase in message volume due to the increased number of cars (normal traffic demand and additional attacker cars) in the scenarios. To quantify the attack's intensity, the Rn metric was calculated to determine the percentage of denied normal messages. Figure 3.11 demonstrates that the attack scenario with 20 attackers resulted in a 44% denial of normal messages. In essence, 44% of messages intended for the cooperative awareness service were suppressed. Furthermore, the presence of 30 attackers led to the denial of over half of the messages (61%) that would have been transmitted in the normal scenario. Additionally, increasing the number of attackers to 50 forced 80% of messages to be denied, leaving only 20% of normal messages received and effective for cooperative awareness.

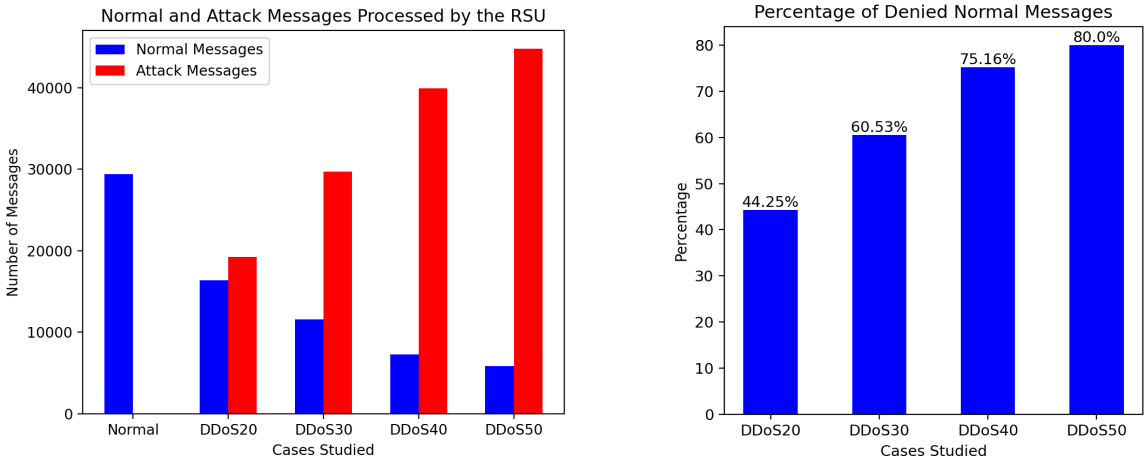
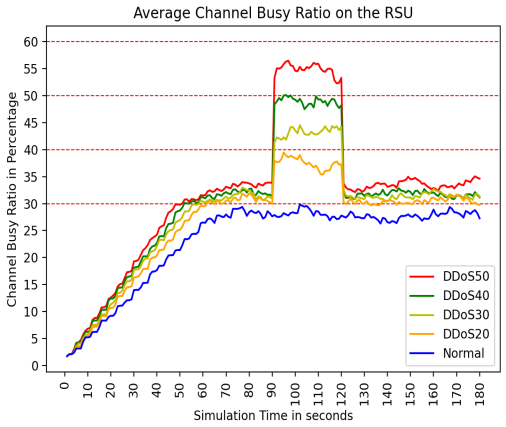


Figure 3.11: Simulation scenario set IV metrics (1)

To further extend our analysis, we examined the CBR, as defined in [70], to assess the DDoS attack's influence on the surrounding environment. The CBR at the RSU was chosen for evaluation due to the RSU's static nature, ensuring that all cars within its vicinity are affected uniformly. Figure 3.12a presents the CBR values at the RSU for each simulated scenario, alongside the four CBR thresholds (indicated by red dashed lines) that are listed in Figure 3.12b taken from [31], which dictate the entity's transmission state. While the RSU's transmission state is configured to transmit 1 msg/sec continuously, the observed CBR demonstrates that nearby cars are compelled to alter their transmission profiles in response to the elevated CBR. To support this, the transmission rate (Figure 3.12c) of a specific car entering the simulation at a predetermined time, ensuring its proximity to the RSU during the attack, was monitored. The findings reveal that

this car was forced to transition from a "Relaxed" to an "Active 1" transmission state under the attack from 20 attackers. Furthermore, the attack from 40 attackers induced a transition to "Active 2 & 3," whereas the attack from 50 attackers prolonged the car's "Active 3" state compared to the previous scenario. This illustrates that nearby cars are induced to reduce their message transmission rate. Based on the results obtained from this simulation scenario set, it is concluded that the DDoS attack's impact on the connected car environment was dual-faceted: the denial of normal messages resulted not only from the RSU processing more attack messages but also from normal cars being tricked into limiting their transmissions.

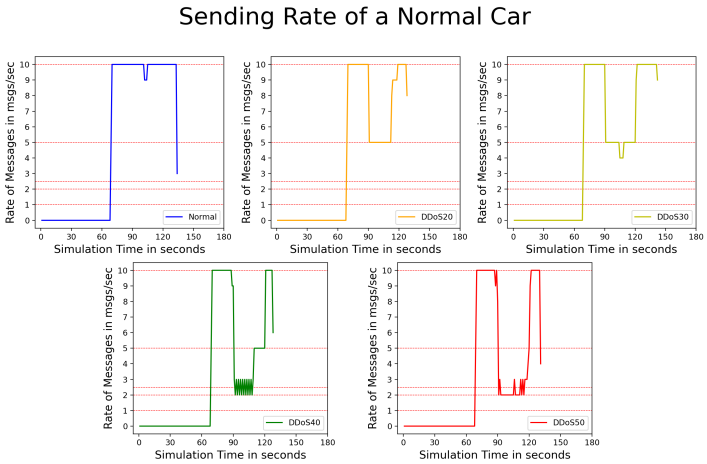


(a)

Table A.1: Mapping of CBR values to state and the currently allowed transmission opportunities per second when T_{on} is max 1 ms

State	CBR	Packet rate	T_{off}
Relaxed	< 30 %	10 Hz	100 ms
Active 1	30 % to 39 %	5 Hz	200 ms
Active 2	40 % to 49 %	2,5 Hz	400 ms
Active 3	50 % to 60 %	2 Hz	500 ms
Restrictive	> 60 %	1 Hz	1 000 ms

(b)



(c)

Figure 3.12: Simulation scenario set IV metrics (2)

3.5.5 Discussion

The simulation results underscore the efficacy of the DDoS attack. Notably, the observed effects diverge from those of conventional DDoS attacks that typically result in system outages.

Instead, the impact manifests as a denial of cooperative services, characterized by a reduced CAM broadcast rate from legitimate cars and the accumulation of invalid data, potentially containing null or meaningless values. This can be described as a silent attack, as the system interprets the false congestion as legitimate, prompting adaptive responses. Consequently, safety applications within connected car environments, ranging from periodic messages to urgent notifications, are vulnerable to disruption or denial under such DDoS attacks. Furthermore, traffic optimization algorithms, which are executed on backend servers and rely on CAM data, will be adversely affected by the acquisition of illegitimate data.

3.6 Conclusion

This chapter presented an evaluation of the impact of DDoS attacks on the Connected Cars environment through the use of distinct simulation scenarios. These scenarios included variations in hijacked car counts, attacker numbers, and attacker transmission power to demonstrate the effects of DDoS attacks. Additionally, a simulation scenario incorporating secured messages was used to examine the impact of security overhead. A comprehensive analysis of various metrics was conducted to assess the attack's impact. This study demonstrated that DDoS attacks can be executed using CAM messages, the fundamental safety messages for awareness. The results highlighted the severity of DDoS attacks on connected car entities and services, specifically the potential for storage resource saturation with illegitimate data and the reduction of service contribution from participating cars.

These findings emphasize the necessity for DDoS attack detection and message filtering prior to application layer processing, which supports safety and traffic optimization services. To develop a robust detection methodology, a crucial initial step is the simulation of a DDoS attack that closely approximates real-world conditions. Given that the current study focused on a single RSU and a limited highway segment, the subsequent chapter will present a more extensive simulation of a connected car network based on a realistic scenario. This will enable the simulation of a broader DDoS attack and the collection of CAM messages from multiple RSUs, facilitating the development of an enhanced detection approach.

Chapter 4

Shielding the Connected Cars: A Dataset-Powered Defense Against DDoS

4.1 Introduction

The quality of the representation of a problem significantly influences its solvability. Consequently, a fundamental step in proposing a solution approach is the creation of a scenario that closely mirrors the problem under consideration. To this end, this study employs the car traffic demand of Luxembourg City [20], validated against Google traffic data, and simulates a DDoS attack occurring during peak hours. Furthermore, data collection was conducted using simulated RSUs strategically positioned to cover a wide highway network, thereby emulating real-world data acquisition from a connected car environment. This approach facilitated the development of a detection methodology for the represented CAM-DDoS attack.

This chapter reviews related work on DDoS attacks targeting connected cars using safety messages. It outlines the system models, including communication, attack, and detection. Furthermore, this chapter details the simulation process and the generated dataset. Subsequently, it presents and analyzes the obtained detection results.

4.2 Related Work

The related works presented herein exclusively address DDoS attacks conducted using safety-related messages. Although prior studies on DDoS attacks involving diverse message types were investigated, the broader field was deemed adequately addressed, thus the scope of this review was limited to those detailed in Annex Table B.1, which are not further discussed in the subsequent sections.

In [77], the authors created their DDoS dataset based on an attack initiated by four simulated parked cars that were added to the default simulation of Veins [78]. XGBoost model [79] was the best in detecting the DDoS attack, achieving high scores in accuracy (99%) and F1-score (99%). However, the simulation of the DDoS attack from parked cars does not represent a realistic approach. Since DDoS attacks are not limited to parked cars and may be performed using any other car.

In [80], the authors created a novel DDoS dataset where they included different attacks that exploit the BSM that is equivalent to the CAM message. They simulated a DDoS attack done on the car traffic of two areas of Luxembourg City [20] that included an urban and a highway scenario. The authors evaluated different machine-learning algorithms to detect DDoS attacks, achieving high accuracy scores ($81\% \rightarrow 99\%$). The disadvantage of the work done was that the data collection mechanism included messages collected from each car, which is not a realistic approach. This gives more data to the models to learn on, leading to optimistic results. However, in a real scenario, collecting the messages received by all the cars is impractical, as it would necessitate cars transmitting their databases to the RSU, resulting in significant network overload.

The authors in [81] utilized the VeReMi extension dataset [82] as their basis for detection and assumed that the DoS/DDoS attacks were executed at random intervals for random cars. They conducted an analysis of various supervised machine-learning models, achieving promising results. This work shares the aforementioned disadvantage of collecting messages from all cars, which is unrealistic and leads to potentially optimistic performance evaluations.

Table 4.1 provides a summary of the mentioned related works.

The contributions of this work are twofold. Firstly, we present the development of a dedicated dataset for connected cars, simulating a DDoS attack using CAM messages. In contrast to related work, our approach involves collecting CAM messages at the RSU level, thereby emulating real-world deployment scenarios. This methodology is employed due to the impracticality of as-

Table 4.1: Related works

Reference	Goal	Metrics (Results)	Model(s)	Software	Message Content	Limitations
[77]	Creating a dataset and designing a misbehavior detection system based on a XGboost model	• Accuracy (99%) • Precision (X) • Recall (X) • F1-score (99%)	XGboost	• Omnet++ • Veins • SUMO	x and y coordinates, speed, acceleration, CO2 emission	• No detailed data presentation • Used parked cars for the attack • Studied a default scenario
[80]	Generating a DDoS attack dataset and testing several machine learning algorithms	• Accuracy (99%) • Precision (99%) • Recall (99%) • F1-score (99%)	• Multi-Layer Perceptron • Random Forest • K-Nearest Neighbor • Support Vector Classification • Decision Tree • Gaussian Naive Bayes	• Omnet++ • Veins • SUMO	Some BSM fields and other calculated ones	Collection of data is done at the level of the cars
[81]	Utilizing the VeReMi dataset to analyse the performance of different supervised machine learning models	• AUC-ROC (99%) • Accuracy (100%)	• Decision Tree • Random Forest • XGBoost • Logistic Regression • Naive Bayes • AdaBoost • K-Nearest Neighbor • Support Vector Machine	None	VeReMi based	Collection of data is done at the level of the cars
Our Work	Generating a CAM-DDoS dataset based on a representative connected cars scenario, collecting messages from RSUs and proposing a detection approach	• MCC (98%) • training time (3 to 8 sec) • testing time ($\sim$0.1sec)	• RF • Adaboost • HistGBC	• Omnet++ • Artery • SUMO	CAM ETSI Standard	Studied a highway scenario only

suming that on-vehicle message collection and processing for attack detection can currently be performed. Secondly, we evaluate the effectiveness of three supervised ensemble learning algorithms, within a dedicated IDS, against this simulated attack. The subsequent section details the system overview, including the communication, attack, and detection models.

4.3 System Overview

4.3.1 Communication and Attack Models

As our work employs CAM messages in the DDoS attack, our communication model exclusively utilizes DSRC technology, consistent with the study in Chapter 3. Regarding the attack model, we have maintained the same one from the previous chapter, where compromised cars transmit CAM messages at five times the normal rate. We introduced a necessary modification to incorporate a randomness factor wherein the attack is initiated at a random time, determined by the attacker (Table 4.4), and terminated at the end of the designated attack period. Crucially, the coordinated execution of the attack by the compromised cars is essential for its classification as a DDoS attack; otherwise, it would be considered a series of DoS attacks originating from random cars. Concerning pseudonym IDs and their utilization, as specified in Table A.1 of the ETSI standard [83] within the SCOOP@F project [4], our attacker model does not adhere to the specified criteria ($40,000$ signatures or 1 hour), thereby increasing its plausibility.

This configuration of the communication and attack models will be adopted in all studies within this thesis unless explicitly stated otherwise.

4.3.2 Detection Model

Machine learning has demonstrated its potential in addressing diverse and complex problems over the years. The emergence of ensemble learning has facilitated the solution of increasingly challenging tasks. Ensemble learning mainly includes three operational modes: bagging, boosting, and stacking [84]. Bagging (Figure 4.1a) employs the same base learner (A) in a parallel configuration, whereas boosting (Figure 4.1b) utilizes the base learner sequentially. On the other hand, stacking (Figure 4.1c) leverages distinct base learners to generate a derived meta dataset for a meta-learner, which ultimately forms the final ensemble model.

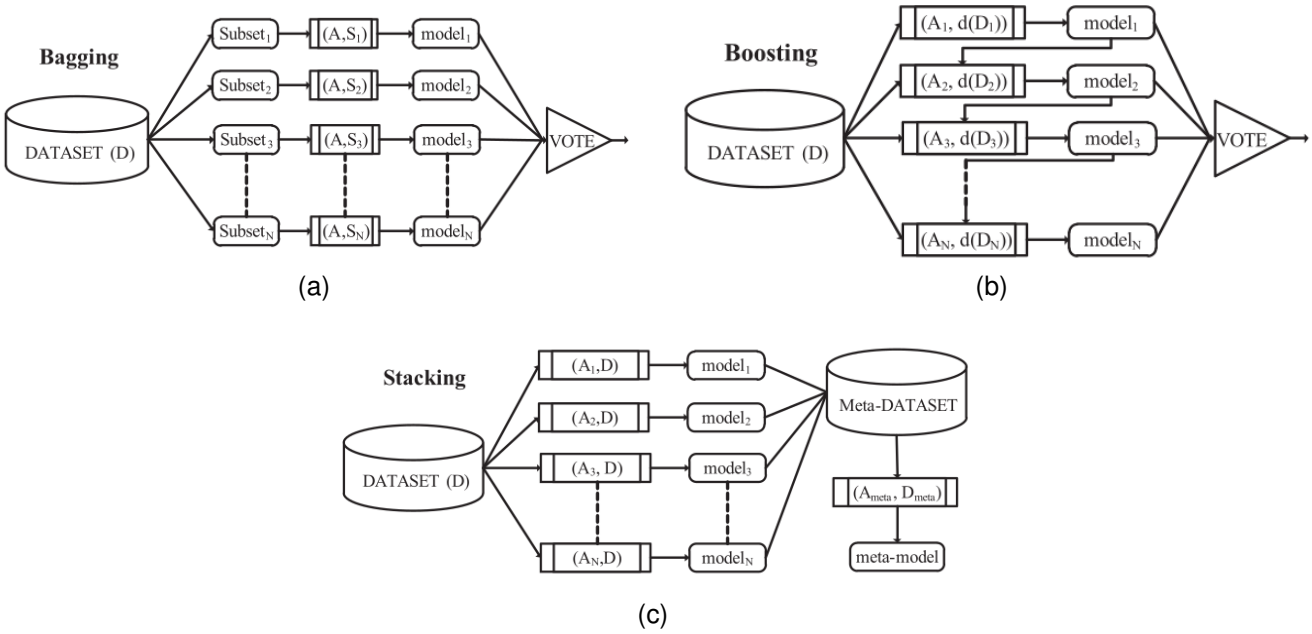


Figure 4.1: Ensemble learning methods [85]

Researchers have developed various standalone ensemble learning models, which have been integrated into well-established libraries such as scikit-learn [86]. For the present problem, Random Forest was selected as an advanced bagging method, given its widespread adoption. As a boosting ensemble, AdaBoost was chosen due to its methodology of adjusting the weights of misclassified instances for subsequent base learners. Additionally, HistGBC, a variant of the gradient boosting algorithm optimized for faster convergence on large datasets, as encountered in this work, was employed. While further models could be explored for specific trade-offs, the selected models yielded satisfactory results, leaving the topic open for future research on more detection algorithms. The following provides an abstract overview of each employed model:

Random Forest [87]: The RF algorithm is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. It leverages bootstrap aggregation (bagging) and randomly selects subsets of features to split nodes at each tree. The steps for training the RF model include:

(i) Sample a subset of the training data with replacement: $D_b = \{x_i, y_i\}$

(ii) Build a decision tree $h_b(x)$ on D_b using a random subset of features at each node.

(iii) Aggregate predictions from B trees for classification using majority voting for $h_b(x)$ where $b \in \{1, B\}$.

The final model is:

$$\hat{y}_{RF} = mode\{h_1(x), h_2(x), ..., h_B(x)\} \tag{4.1}$$

Adaboost [88]: Adaptive Boosting combines weak learners (e.g., decision trees) iteratively into a strong learner, focusing on examples that previous models misclassified. Each weak learner is assigned a weight based on its performance, and the final model is a weighted sum of the learners. The steps for obtaining an Adaboost model include:

Initialize weights of data points: $w_i = 1/N$ where $i \in \{1, N\}$.

Then, for each iteration t:

(i) Train a weak learner $h_t(x)$

(ii) Compute the learner's error: ϵ_t

(iii) Compute the learner's weight: $\alpha_t = \frac{1}{2}\ln(\frac{1-\epsilon_t}{\epsilon_t})$

(iv) Update weights $w_{t+1}(i)$ using $w_t(i)$ and α_t

(v) Repeat until all weak learners are trained

The final model is:

$$\hat{y}_{Adaboost} = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{4.2}$$

Histogram-based Gradient Boosting Classifier [89]: HistGBC is an optimized version of gradient boosting that discretizes feature values into bins (histograms) to speed up training and reduce memory usage. It builds decision trees iteratively, minimizing a loss function. The steps followed include:

Initialize the model: $F_0(x) = constant$

Then, for each iteration t:

(i) Calculate residuals: $r_{ti} = -\left[\frac{\partial L(y_i, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)}\right]$

(ii) Fit a weak learner $h_t(x)$ to the residuals

(iii) Update the model: $F_t(x) = F_{t-1}(x) + \mu h_t(x)$ where μ is the learning rate

(iv) Repeat until all weak learners are trained

The final model is:

$$\hat{y}_{Hist-GBC} = F_0(x) + \sum_{t=1}^{T} \mu h_t(x) \tag{4.3}$$

Following the generation of the three datasets, the models were trained and validated using the first dataset to determine the optimal hyperparameters, as presented in Table 4.2. These hyperparameters were obtained by a manual grid search without affecting the temporal statistics of the data. Subsequently, the generalizability of these models was assessed using the remaining two datasets.

Table 4.2: Model configuration

Model	**Hyperparameters***
Histogram-based Gradient Boosting	learning_rate: 0.01, max_depth: 8, min_samples_leaf: 5
Random Forest	n_estimators: 5, criterion: log_loss, max_depth: 10, min_samples_leaf: 2, max_features: 0.5, bootstrap: False, class_weight: {0: 1.0, 1: 1.5}
Adaboost	n_estimators: 4, learning_rate: 1.9, algorithm: SAMME, base_estimator:DTC, DTC_max_depth: 3

*Not mentioned parameters are kept as default values
DTC: Decision Tree Classifier
Corresponding scikit-learn version: 1.4.1

4.4 Dataset Generation

4.4.1 Simulation Environment

The simulation environment from the previous part (Chapter 3) was retained, utilizing the following software: SUMO [19] for traffic modeling, OMNeT++ [71] for network event management, and Artery [72] for standardized CAM message formatting.

To generate a CAM-DDoS dataset that closely replicates a potential real-world DDoS attack, the following simulation parameters were selected. The traffic model used in the previous part, based on the French highway (3.4), was replaced with the Luxembourg City model from [20], as it accurately reflects real-day traffic patterns. To provide a comprehensive proof-of-concept simulation including multiple network nodes (cars and RSUs), the Region of Interest (ROI) was selected to be the area covered by the four RSUs, as depicted by the green circles in Figure 4.2. The attacking cars were chosen by assigning a random number (0-100) to each car entering the ROI; cars with numbers less than 5 were designated as attackers during attack periods. As DDoS attacks are more prevalent during peak hours, the simulation was conducted around 08:10 am. To enhance the attack's randomness, the attack start times and durations were also randomized (detailed in Subsection 4.4.2). The communication range was set to 1 km, and the Reactive (1 ms) DCC mechanism was employed to regulate the transmission rates of normal cars based on the CBR with a maximum rate of 10 CAMs per second. A summary of the simulation parameters is provided in Table 4.3.

Table 4.3: Simulation parameters

Parameter	Value
Traffic model	LuSTscenario (dua-static)[20]
Region of interest	Highways (A1, A3, A4, & A6)
Number of RSUs	4
Attacker's percentage	5%
Simulation time	360 sec
Simulation interval	08h07 till 08h13
Attack duration	Random(Table 4.4)
Attacker message rate	50 CAMs/sec
DSRC range	1 km
DCC mechanism	Reactive (1 ms)

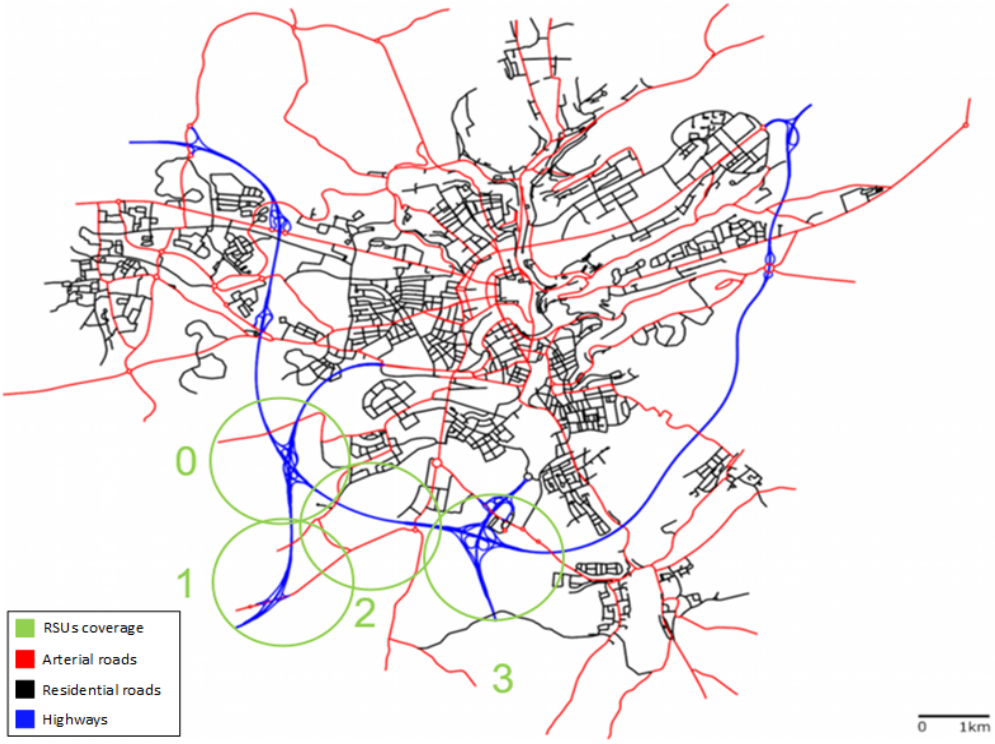


Figure 4.2: Modified Luxembourg [20] map

4.4.2 Dataset Description

This subsection describes the dataset's temporal and data distributions providing an explanation of the numerical values presented in Tables 4.4 and 4.5, respectively.

Firstly, the entries (1) and (2) correspond to the seed values provided to the "*srand()*" function within Artery, which initializes the "*rand()*" function, found in the "Data type" column. The entry (random) indicates that the seed was the timestamp of the simulation. This approach was implemented to generate two reproducible datasets and one non-reproducible dataset. As shown in the time distribution (Table 4.4), the first minute of the training data does not contain any attack. Subsequently, each subsequent minute features an attack whose starting time is determined by a pre-generated random number (0-29, column five) and whose duration is determined by another pre-generated random number (5-30, column six). Consequently, the training data comprises three attack intervals, while the validation and testing datasets each contain one interval. The testing data from the other two datasets (generated with seeds "2" and "random") was utilized to assess the model's generalizability.

Table 4.5 illustrates the data distribution in terms of attack and normal messages. The final column of this table demonstrates the varying percentages across different data chunks, thereby

confirming the randomness of the simulations. Message collection was performed exclusively at the RSU level, representing a practical data acquisition method in contrast to collecting messages directly from cars. These messages were then forwarded to a dedicated IDS for attack classification.

Table 4.4: Time distribution

Data type	Total time	Total normal time	Total attack time	Starting interval(s)	Attack interval(s)
Training (1)	4 min*	3 min 16 sec	44 sec	13;16;27 sec	6;17;21 sec
Validation (1)	1 min	54 sec	6 sec	25 sec	6 sec
Testing (1)	1 min	48 sec	12 sec	23 sec	12 sec
Testing (2)	1 min	42 sec	18 sec	21 sec	18 sec
Testing (random)	1 min	50 sec	10 sec	13 sec	10 sec

*First minute does not contain any attack, for the other three minutes refer to column "Starting Interval(s)".

Table 4.5: Data distribution

Data type	Total messages	Normal messages	Attack messages	Percentage of attack messages
Training (1)	664,976	609,281	55,695	8.3%
Validation (1)	164,828	156,783	8,045	4.9%
Testing (1)	170,051	154,586	15,465	9.1%
Testing (2)	171,224	150,850	20,374	11.9%
Testing (random)	168,377	157,470	10,907	6.4%

4.4.3 Dataset Contents

The datasets' content conforms to the structure of a CAM message, as specified in the ETSI standard [30]. As of the present, the developers of Artery have not incorporated all optional fields of the CAM message (illustrated in Figure 4.3); therefore, only the default fields are utilized, which would invariably be present in a real-world deployment scenario. In this study, the dataset, available on GitHub [90], undergoes a preprocessing stage to retain solely the features that vary according to simulation parameters, along with the timestamp and an engineered feature, comprising:

A. Standard Features: The following features (except $SimulationTime$) are directly extracted from the CAM messages:

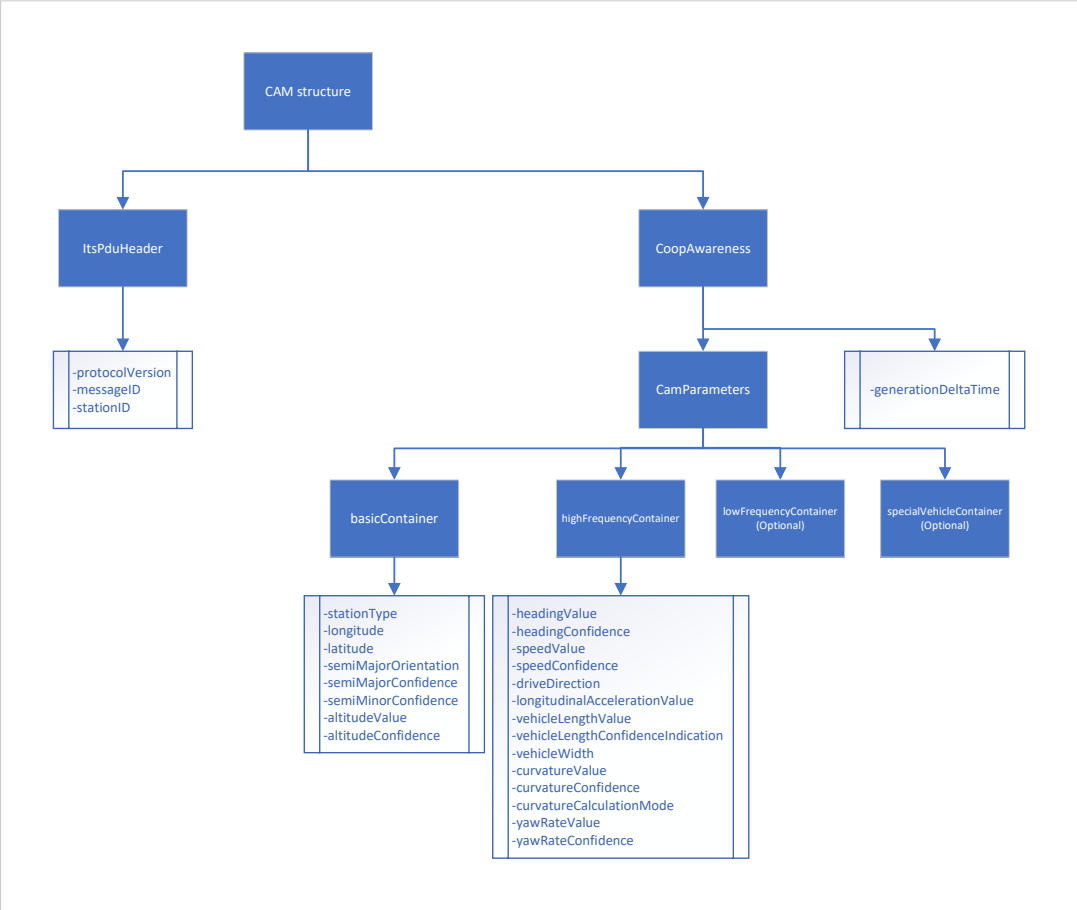


Figure 4.3: CAM structure conforming to the ETSI standard [30].

(i) $SimulationTime$: The recorded simulation time.

(ii) $StationID$: The unique identifier (PseudoID) assigned to the car.

(iii) $GenerationDeltaTime$: The differentiated timestamp of the reference position in the CAM, representing the message generation time.

(iv) $Longitude$: The car's longitudinal position.

(v) $Latitude$: The car's latitudinal position.

(vi) $Heading$: The car's orientation relative to the WGS84 coordinate system.

(vii) $Speed$: The car's speed.

(viii) $LongitudinalAcceleration$: The car's acceleration along the longitudinal axis.

(ix) $Curvature$: The curvature of the car's trajectory.

(x) $YawRate$: The rate of rotation around the car's vertical (z-axis) in the coordinate system centered at the car's center of mass.

B. Engineered Feature: In addition to the standard features, we introduced an engineered feature in the preprocessing phase of our work:

(xi) $TimeDifference$: The time interval between two consecutive CAM messages generated by the same $StationID$.

Detailed definitions of some fields, demonstrating their conformity with their respective standards, are provided in Table A.1 of the Annex.

4.5 Results and Analysis

To the best of our knowledge, the existing literature does not utilize standardized CAM messages within the context of a DDoS attack. While Gonçalves et al. [91] employed CAM messages, their study focused on alternative attacks. Consequently, we have evaluated our results to the fullest extent possible to demonstrate their potential.

The data distribution presented in Table 4.5 reveals an imbalanced dataset, necessitating either class balancing or the selection of appropriate evaluation metrics. Class balancing could be achieved through undersampling or oversampling techniques, which involve either reducing the majority class or augmenting the minority class, respectively. While exploring suitable sampling techniques is feasible, it would require data preprocessing. In this study, we opted to minimize data preprocessing, focusing instead on creating an engineered feature, applying a standard scaler to the dataset features, and removing static features. Consequently, we chose to evaluate the models' performance using the MCC, as calculated by eq. (4.4). This metric is robust for imbalanced datasets, as it considers the entire confusion matrix, thereby accurately assessing model performance across all classes, including the minority class [92].

It is noteworthy that classical evaluation metrics, such as accuracy and the F_1 score, yielded high scores for our models, reaching 99% and above. However, as demonstrated by Chicco and Jurman [93], these metrics are not recommended for imbalanced datasets. They also highlighted the advantages of the MCC metric over accuracy and the F_1 score in binary classification evaluation.

$$MCC = \frac{TP.TN - FP.FN}{\sqrt{(TP + FP).(TP + FN).(TN + FP).(TN + FN)}} \tag{4.4}$$

where

$$TP = \textit{True Positive}$$
$$TN = \textit{True Negative}$$
$$FP = \textit{False Positive}$$
$$FN = \textit{False Negative}$$

Figure 4.4 illustrates the MCC metric scores (in $\%$) for our models on the Validation (1) dataset, where all three selected models achieved scores approaching 99$\%$. However, the MCC scores decreased by approximately one percent on the Testing (1) dataset, although they remained high. Upon evaluating the models on the remaining two datasets, generated using distinct seeds, the performance remained consistently high, demonstrating the generalizability of our models in classifying unseen data.

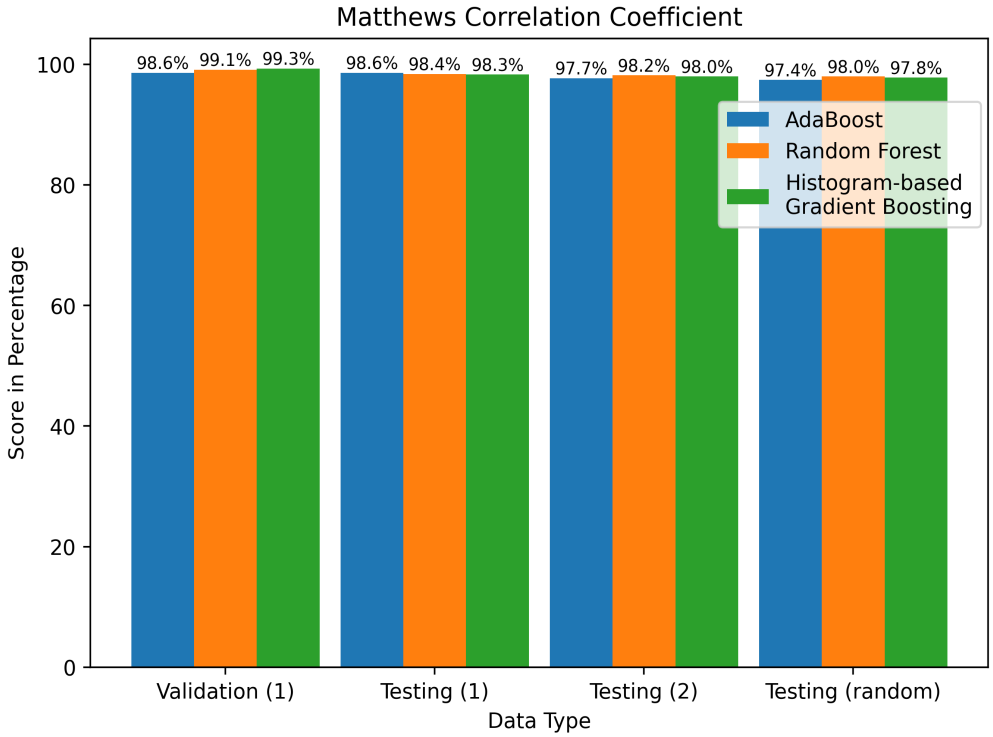


Figure 4.4: MCC evaluation

The training and testing times of each model, as depicted in Figure 4.5, were also evaluated. It is evident that the HistGBC model exhibits superior training times compared to the other models, however with a marginally longer message classification time, as observed from Testing (x), where x $\in 1, 2, random$. Random Forest demonstrated the longest training time, at 7.7 seconds, which is considered acceptable given the size of the training dataset. The overall training and

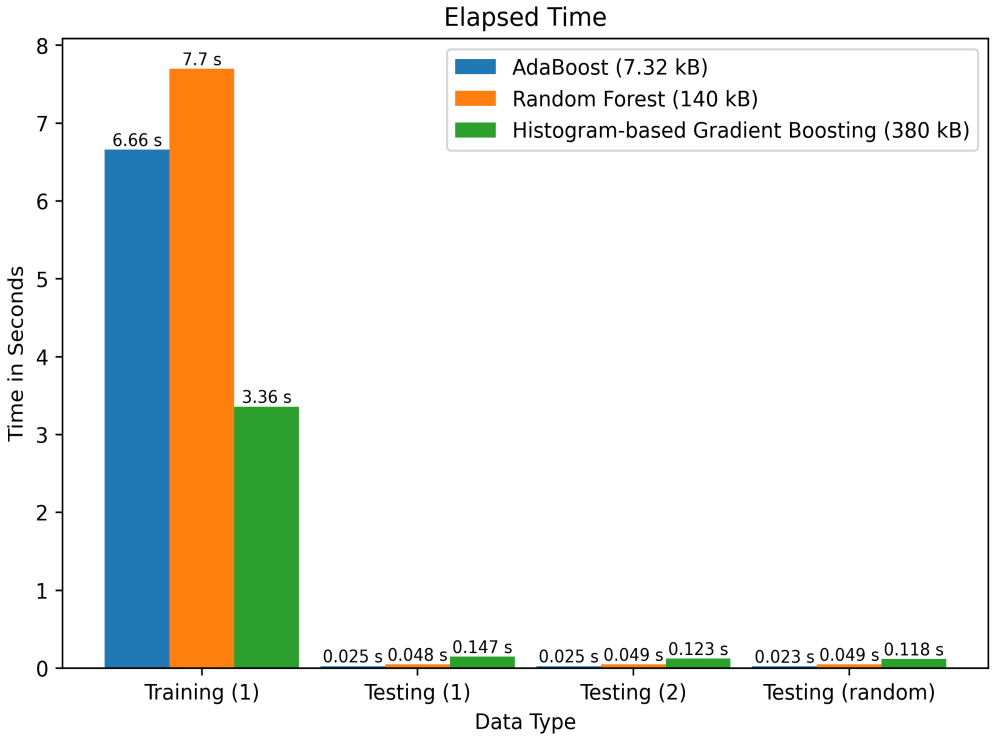


Figure 4.5: Training and testing time (with model size)

testing performance of the models is deemed satisfactory, considering the dataset and model sizes (indicated alongside the model names). This implies that one minute of new data collected by the IDS can be evaluated within one second, inclusive of preprocessing overhead, assuming negligible network latency between RSUs and the IDS.

4.6 Conclusion

This chapter presented three ensemble learning models, trained and evaluated on a dataset simulating a DDoS attack using CAM messages. The datasets[1] [90] were collected at the RSU level, mirroring real-world deployment scenarios. Our results demonstrated the models' capability to detect DDoS attacks using CAM messages within one second, and we further established their generalizability across two additional datasets featuring the same DDoS attack. This raises two key questions:

1. To what extent are these models generalizable, and can they detect diverse DDoS attack types?

[1] Available on our GitHub repository: https://github.com/A-Wehby/Connected-Cars-DDoS-Dataset

2. Where can an IDS be deployed within the existing C-ITS architecture to effectively manage various C-ITS attacks?

The next chapter addresses these questions and introduces our proposed C-ITS-dedicated IDS, discussing its characteristics. Furthermore, we evaluate the performance of our previously developed models against the diverse DDoS attacks presented in Chapter 3. Regarding the fact that DDoS attacks can be performed using CPM messages, our developed solution requires adaptation before being applied, as the CPM messages differ in structure from the CAM messages.

Chapter 5

Toward Adaptive Intrusion Detection System for Mitigating Morphing Distributed Denial of Service Attacks in Connected Cars

5.1 Introduction

As connected cars become increasingly prevalent, companies are inevitably concentrating on the development of IDS solutions tailored to this domain. Several studies and surveys [94, 95] have been conducted to better understand the state-of-the-art development of intrusion defense mechanisms in various domains and specifically in the automotive [95] domain. The fundamental consideration for any IDS is its deployment location. This is followed by an examination of the IDS's characteristics and roles within its operational context and, subsequently an assessment of its expected performance and attack mitigation capabilities.

This chapter proposes an IDS architecture for the connected car environment, designed to be integrated within the state-of-the-art C-ITS framework. Furthermore, as an example of an IDS functionality, it evaluates the generalizability of our DDoS detection approach, developed in Chapter 4, against other DDoS attack types, specifically JamDDoS and Security DDoS, as analyzed in Chapter 3.

5.2 Related Work

This section reviews existing research on IDS solutions proposed for connected cars, focusing specifically on those utilizing safety DSRC-related messages (though not exclusively DDoS attacks). Particular emphasis is made on the deployment location and training data employed in these solutions.

Alladi et al. [96] assessed the deployment of deep neural network models on a Raspberry Pi 3B, functioning as an IDS on an MEC server. Their models were trained using the VeReMi extension dataset [82], which was also employed by Behravan et al. [97] for RSU-based attack detection using ensemble learning techniques. A significant limitation of studies utilizing the whole VeReMi dataset is its comprehensive collection of data from all cars. This approach deviates from realistic scenarios, where only a subset of the CAM data is typically accessible. Consequently, models trained on the whole dataset may not be useful for actual deployments.

Yurkovich [98] proposed an RSU-based IDS that emphasizes location-specific data. The dataset was generated by modifying the misbehavior detection framework [99], concentrating on two distinct map sections and deploying RSUs within these areas. While their approach analyzed data collected by RSUs, the assumption that the IDS can be hosted directly on the RSU remains problematic due to the RSU's limited resources and the concurrent execution of multiple services.

The authors of [91] proposed a four-tiered hierarchical IDS architecture, wherein communication is secured through a VANET PKI and Attribute-Based Encryption. The architecture comprises four levels: single cars, car clusters, RSU clusters, and backend servers. Each level conducts detection of DoS and Fabrication attacks, tailored to its capabilities; for example, the single car level employs rule-based detection. Furthermore, lower levels transmit collected messages to higher levels, while the highest level generates and disseminates detection rules and trained machine learning models to the lower tiers. The approach demonstrates promising detection results; however, as acknowledged by the authors, further investigation is required to assess the impact of data communication overhead between the various levels.

Existing IDS-related research suggests attack detection at the RSU level. However, given the inherent capacity limitations of RSUs [60], this approach is deemed impractical. Therefore, we propose performing detection at a higher level, specifically within the MEC node. Furthermore, we utilize messages collected exclusively by RSUs, thus reflecting a realistic message acquisition strategy. Table 5.1 summarizes the reviewed IDS-related works.

The introduction of an IDS into the connected car environment is intended to complement,

Table 5.1: Summary of IDS related works

Ref.	Data used	Approach	Limitations
[96]	VeReMi extension dataset	Performing deep neural network prediction on Rasberry Pi as a RSU	Training data is collected per car and detection is on the RSU
[97]	VeReMi extension dataset	Performing ensemble learning on RSU	Training data is collected per car and detection is on the RSUs
[98]	Data collected on the RSUs in the simulation	Performing ensemble detection on RSUs	Detection is on the RSUs
[91]	Data collected from cars	Performing RF detection on RSUs	Training data is collected per car and a part of the detection is on the RSUs
Our proposed approach	Data collected on the RSU(s) in the simulation	Performing the detection on the IDS in the MEC node	Only highway scenario is considered

not replace, the existing PKI framework. Therefore, the IDS augments PKI security by providing an additional layer of defense, particularly against malicious insider threats within the C-ITS. This IDS should not be restricted to DDoS attack detection; rather, it should be designed to accommodate solutions for a wide range of attack vectors. Consequently, it may comprise various software modules operating in parallel to address diverse threats. A more detailed description of this system and our proposed approach is provided in the next section.

5.3 Proposed Intrusion Detection System Architecture

Given the substantial volume of data entering systems and core servers, it is imperative to ensure data usability and integrity prior to taking effect on them. This necessitates the verification of incoming data legitimacy. Consequently, IDSs are deployed to filter received data before using them. This filtration process encompasses multiple layers and adheres to diverse protocols, which are detailed in the subsequent subsections.

5.3.1 Architecture and Deployment

The architecture of an IDS can be categorized as horizontal, vertical, or standalone, or a combination thereof. A vertical architecture indicates the presence of a hierarchical structure within

the IDS. A horizontal architecture signifies the deployment of distributed IDSs. Furthermore, the architecture defines the operational mode of the IDS. Consequently, various IDS architectures can be derived, based on their operational modes:

- **Standalone**: A single IDS performing local detection.
- **Standalone horizontal**: Multiple distributed standalone IDSs, each performing local detection, commonly known as host-based IDS.
- **Standalone vertical**: A single IDS performing local detection and reporting to a superior/hierarchical IDS.
- **Standalone horizontal vertical**: Multiple distributed standalone IDSs, each performing local detection and reporting to a superior/hierarchical IDS.
- **Horizontal**: Multiple distributed IDSs cooperating with each other.
- **Vertical**: Multiple IDSs performing detection at different hierarchical levels.
- **Horizontal vertical**: Multiple distributed cooperating IDSs performing detection and reporting to superior/hierarchical IDSs, commonly known as network-based IDS.

Beyond architecture and operational mode, IDS placement is also critical. The state-of-the-art C-ITS framework can be split into three primary levels, from our perspective. These levels are outlined in Figure 5.1 for clarity and do not imply a separation akin to that found in SDN (e.g., separation of application, control, and data planes). An abstract description of these levels is as follows:

- **Infrastructure level:** Comprising cars, RSUs, and cellular base stations. These entities communicate via C-ITS communication technologies (DSRC and cellular communications). RSUs and base stations forward necessary received data to higher levels.
- **MEC level:** Consisting of multiple MEC nodes. Each node houses various components/ services specific to vehicular networks (e.g., PKI components) and cellular networks (e.g., mobile operator network components). These MEC nodes can communicate with each other using diverse protocols depending on the communicating components. This level receives and analyzes data from the lower level. It can perform actions independently or request policies from the higher level, in addition to reporting significant events and data to the higher level.

- **Road operator and core network level:** Encompassing road operator and automotive servers, as well as internet access services. This level receives reports and requests from the MEC level and may enforce policies or provide services for implementation by the MEC level.

Given the limitations of RSU-based IDS placement proposed in recent studies, which do not align with current RSU capabilities, we propose positioning the IDS at the MEC node level, as illustrated in Figure 5.1. This level offers computational power through its servers, along with services such as virtualization and containerization, which enable an IDS to perform the necessary tuning and retraining of its machine learning models.

An example of IDS functionality and communication at the MEC level is described below:

Consider distributed IDSs 1 and 2, each collecting data from associated RSUs and base stations. These IDSs perform local detection within their respective MEC nodes (e.g., MEC 1), reporting and cooperating with a superior/hierarchical IDS within the Intermediate Data Center (IDC) shown as IDCs of MEC 1. These IDCs also exchange information with IDCs in MEC 2. Finally, the IDCs from MEC nodes 1 and 2 report to the higher data center at the road operator and core network level shown as DCs. Further details regarding this reporting are discussed in the model, protocol, and rules sections.

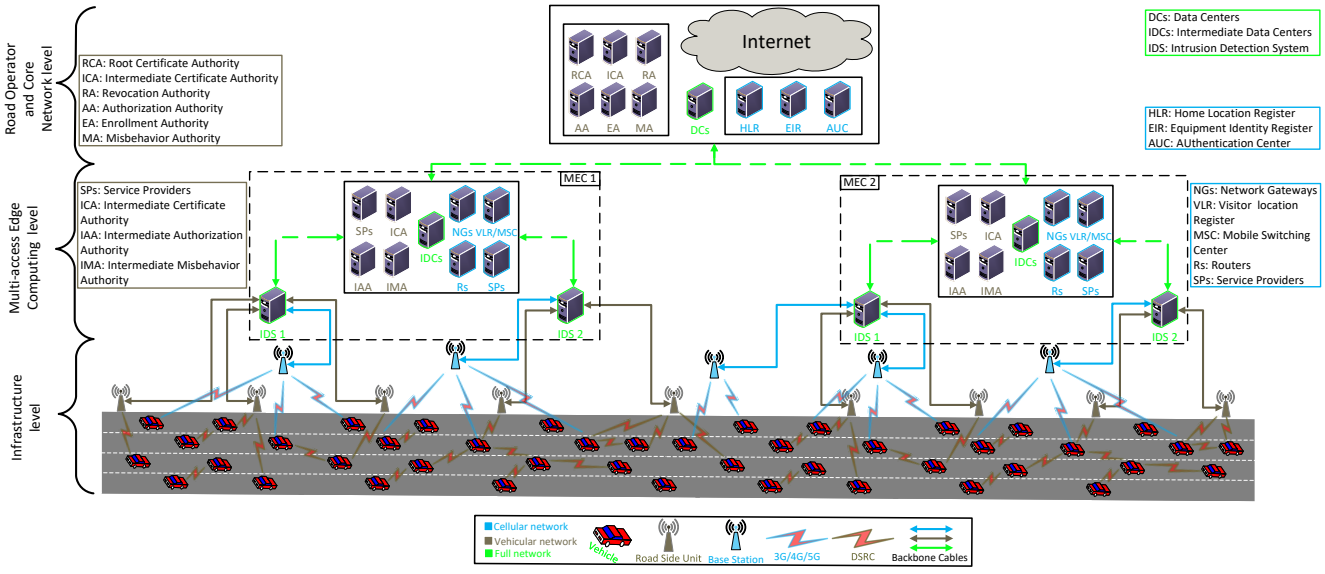


Figure 5.1: IDS architecture for connected cars.

5.3.2 Models

An IDS model can be developed based on several classification factors, as discussed in [94]. These factors include:

- **Type**: Network-based, host-based, or hybrid.
- **Detection technique**: Signature-based, anomaly-based, or hybrid.
- **State awareness**: Stateless or stateful.
- **Usage frequency**: Online or offline.
- **Response type**: Active or passive.

The selection of certain factors is strongly correlated with the architecture and placement of the IDS. For instance, a standalone IDS is typically host-based, performing local attack detection. Horizontal IDSs are generally network-based. Considering the architecture depicted in Figure 5.1, it is neither a purely horizontal nor standalone IDS, but rather a hybrid type. IDSs 1 and 2 are deployed to cover network areas, reflecting a network-based approach, while each MEC node hosts an IDS (within the IDCs), embodying a host-based approach. Regarding detection techniques, a hybrid approach is consistently recommended [100] to facilitate rapid detection of known attacks and enhance resilience against unknown threats. Concerning state awareness, a stateful approach enables the detection of multi-step attacks, a capability lacking in stateless approaches. Finally, to ensure prompt detection and response, online and active IDSs are indispensable. These IDSs continuously monitor network traffic and system activity, enabling immediate threat mitigation.

5.3.3 Protocols

Protocols for IDSs remain an open area of research within the Internet domain. Consequently, in the context of connected cars, the protocols employed will be based on existing research works. Examples include:

- **The Intrusion Detection Message Exchange Format** [101]: Standardizes the format of exchanged messages.
- **Intrusion Detection Message Exchange Requirements** [102]: Defines the high-level requirements for communication mechanisms.

- **The Intrusion Detection Exchange Protocol** [103]: Specifies the protocol for data exchange between IDSs.
- **The Syslog Protocol** [104]: Describes the protocol for transporting event notification messages.

Adherence to these research works in the development of a new IDS protocols, tailored specifically to connected cars, will ensure interoperability across different road operators and core network owners.

5.3.4 Rules

As previously mentioned, IDS protocols for connected cars are still under active research. Consequently, security and communication rules remain undefined. To mention some examples, IDSs should not process any message that has not been authenticated by the existing PKI. Another practical example, as illustrated in Figure 5.1, involves IDS 1 and 2 cooperating to enhance security. They can exchange information regarding compromised devices and anomalous traffic patterns to improve threat detection and response. For example, if IDS 1 detects a malicious car, it can share this information with IDS 2, enabling IDS 2 to take appropriate actions, such as discarding traffic originating from that car. Furthermore, both IDSs can transmit detection reports to their respective IDCs, which can then update the system's global threat perception. The IDCs can utilize this information to implement new security policies, deploy security patches, and notify misbehavior authorities of potential incidents.

5.3.5 Performance Evaluation and Potential Response Mechanism

IDS evaluation is evolving in line with attack vectors. While evaluating an IDS's accuracy and response time remains fundamental, validation of its generalizability across novel data and its ability to detect diverse attacks has become increasingly vital. This can be framed as assessing the "intelligence" of the developed IDS. With the increasing use of artificial intelligence algorithms, the evaluation now extends to determining the frequency of IDS retraining and whether learning occurs offline or online. This has elevated the standards for classifying an IDS as "smart", although effective and generalizable IDSs with certain limitations are still achievable.

Response mechanisms are pivotal to IDS efficacy. As an IDS is expected to respond to incidents to validate its functionality, responses can range from simple alerts to more impactful

actions. For instance, our IDS, upon detecting malicious messages and identifying the responsible cars with the assistance of superior IDSs (located at the road operator and core network level in Figure 5.1), can request the revocation of the corresponding car's certificate. This revocation request is directed to the revocation authority within the connected car architecture. Subsequently, the revoked car identity can be disseminated throughout the network, enabling other cars to discard messages originating from the malicious source.

A potential approach to initiating revocation requests involves establishing a reputation score for each car. Superior IDSs increment a car's reputation score for each reported malicious message until a predefined threshold is reached, triggering certificate revocation. However, mechanisms can be implemented to decrement the score, mitigating potential false positives. This could be based on a time-based approach (e.g., score decrement after a specified period) or more complex strategies for verifying the authenticity of well-behaved cars. For example, Abbes et al. [105] proposed a blockchain-based solution, incorporating smart contracts, to secure car reputation management. Furthermore, with the release of ETSI standards release 2 for connected cars, a misbehavior authority has been added to the higher-level authorities. This allows IDSs to submit reports according to the misbehavior reporting service standards [38] to this authority, which will then administer appropriate countermeasures.

5.3.6 IDS Internal Workflow

A final consideration regarding IDSs is their internal workflow. As previously mentioned, IDSs may incorporate various software modules operating in parallel to address diverse threats. Figure 5.2 illustrates the internal workflow of our IDS, representing the dataflow through the IDS from a software perspective. In this figure, we examine an IDS handling CAM messages from the RSUs. Upon data reception, the IDS may aggregate all data or partition it into groups, based on programmed rules. The subsequent stages involve data cleansing and preprocessing, which may include the creation of additional features for the detection modules. These modules may implement signature-based detection, anomaly-based detection, and a shared reputation database. Signature-based detection is known to identify known attack patterns. Anomaly-based detection, often developed using artificial intelligence techniques, identifies behaviors deviating from normal patterns. The shared reputation database maintains records of reported cars, enabling their direct identification upon subsequent message transmissions. After data processing through these three modules, a decision criterion is applied to classify the data.

This work contributes to the module of anomaly-based algorithms through the development of machine learning models capable of detecting morphing DDoS attacks. The decision criteria

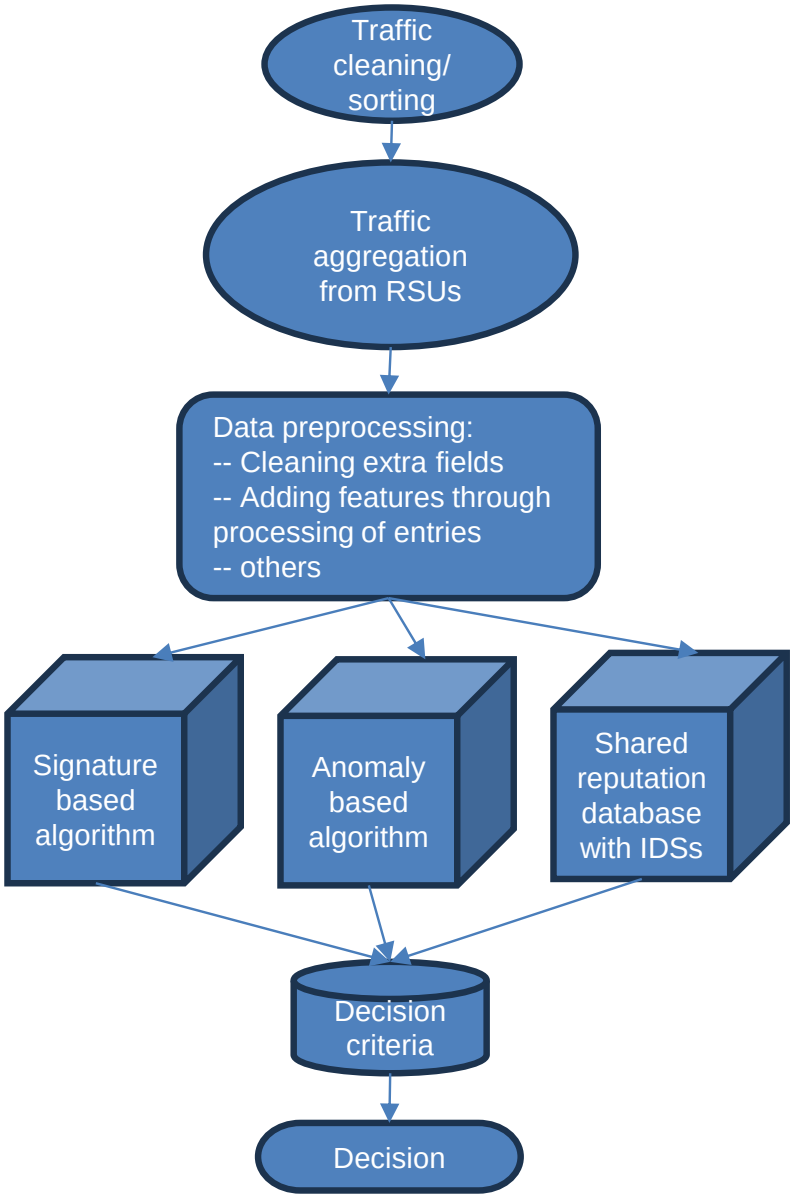


Figure 5.2: IDS internal workflow

and the integration of the three aforementioned modules remain future research directions.

In the next sections, we present the performance evaluation of our study. We begin by detailing the testbed configuration used to generate the datasets, followed by a description of the system models employed in our work, and conclude with the presentation of our performance metrics. Subsequently, we present and analyze the obtained results.

5.4 Performance Evaluation

5.4.1 Testbed Configuration

Simulation Environment: This work builds upon our previous chapters. The message collection from the impact study (Chapter 3) was conducted at a single RSU located in the center of the French highway section under investigation. Consequently, the detection models developed in Chapter 4 are applied to these collected CAM messages. To emphasize the differences in experimental setups between these prior chapters, for which no additional simulations were performed in this study, we provide a comparison of simulation parameters in Table 5.2.

Table 5.2: Simulation parameters' comparison

Parameter	Value in training (ch.4)	Value in testing (ch.3)
Traffic model	LuSTscenario (dua-static)[20]	France traffic demand [73]
Region of interest	Highways (A1, A3, A4, & A6)	Highway (A1)
Number of RSUs	4	1
Percentage of attacking messages	8.3%	(16 to 33)%
Simulation time	240 sec	110 sec
Simulation interval	08h07 till 08h11	70s till 180s
Attack duration	Random(Table 5.3 row 1)	Constant(Table 5.3 row 2)
Attacker's message rate	50 CAMs/sec	50 CAMs/sec
Attack type(s)	DDoS	DDoS, JamDDoS, Security, & Security-JamDDoS
DSRC range	1 km	1 km
DCC mechanism	Reactive (1 millisecond)	Reactive (1 millisecond)

DDoS Attacks Studied: The DDoS attacks included in the testing data used in this work are:

- **JamDDoS**: Attacking cars increase their transmission power in the attack intervals to $2Watts$. (*Scenario Set I*, Subsection 3.4.1)
- **Security DDoS**: All cars encapsulate messages before sending and decapsulate messages upon reception. (*Scenario Set II*, Subsection 3.4.2)
- **Security-JamDDoS**: All cars encapsulate and decapsulate messages. In addition, attackers send with higher transmission power during the attack intervals. (*Scenario Set II*, Subsection 3.4.2)

- N **DDoS**: Several attackers (N) broadcast CAMs at a rate of 50 CAMs per second, where $N \in \{20, 30, 40, 50\}$. (*Scenario Set IV*, Subsection 3.4.4)

5.4.2 Datasets

Our training data is generated based on a representative traffic scenario for Luxembourg City, as detailed in [20]. The focus was on highway segments, where four RSUs were deployed to collect messages from passing cars. Conversely, the testing data was generated from a specific highway segment in France. While the attack duration and rate remained constant, the types of attacks varied, as described in subsection 5.4.1. This variation in attack scenarios enables us to evaluate the generalizability of our models.

The training and testing datasets [90], contain CAM messages compliant with the ETSI standard [30]. Each message includes 26 features related to the CAM structure, along with the simulation time upon message reception and a data label, totaling 28 features. Given that the SUMO software [19], used in our prior work, does not fill all features with dynamic data, data cleansing preprocessing is necessary to eliminate static features. Furthermore, the same feature engineering was applied to the testing datasets. Consequently, each final testing dataset contained the contents described in subsection 4.4.3.

Table 5.3 shows the details about the random attacking intervals present in the training data and specifies when the attack is occurring in the testing data. In the table, the *Total time* is the time of each simulation corresponding to each dataset. The *Total attack time* is the sum of all the attack intervals, so the *Total normal time* is the sum of all the remaining times in the simulation of the corresponding dataset. The *Starting interval(s)* specify the times at which the attack starts, occurring at 60, 120, and 180 seconds from the beginning of the simulation that generated the training dataset. This translates to attacks beginning at seconds 73, 136, and 207, with durations of 6, 17, and 21 seconds, respectively. For the testing dataset, the attack initiates 20 seconds into the simulation and lasts for 30 seconds.

As illustrated in Table 5.4, the testing data encompasses seven distinct datasets of varying sizes. Notably, these datasets exhibit class imbalance, characterized by a substantial difference between the number of normal and attack messages. Our approach to addressing this imbalance is detailed in Section 5.5.

Table 5.3: Time distribution

Data type	Total time (seconds)	Total normal time (seconds)	Total attack time (seconds)	Starting interval(s) (seconds)	Attack interval(s) (seconds)
Training	240 sec*	196 sec	44 sec	13;16;27 sec	6;17;21 sec
Testing	110 sec**	80 sec	30 sec	20 sec	30 sec

*First 60 seconds do not contain any attack, for the other 180 seconds we have attacks that start according to column "Starting Interval(s)".
**Warm-up period of 70 seconds was removed from each of the original datasets of 180 sec because the highway will not be filled with the corresponding traffic according to the France traffic demand [73].

Table 5.4: Data distribution

Data type	Total number of messages	Number of normal messages	Number of attack messages	Percentage of attack messages
Training	664,976	609,281	55,695	8.3%
Testing (20)	115,008	95,797	19,211	16.7%
Testing (30)	122,373	92,696	29,677	24.25%
Testing (40)	125,350	85,421	39,929	31.85%
Testing (50)	133,213	88,439	44,774	33.61%
Testing (Security)	72,545	53,326	19,219	26.49%
Testing (JamDDoS)	114,532	87,042	27,490	24.0%
Testing (Security-JamDDoS)	79,556	52,657	26,899	33.81%

5.4.3 System Overview

The attack model employed in this work has an amplified transmission power in the case of the JamDDoS attack, in addition to its increased transmission rate. Concerning the detection models, they are based on the models developed in Chapter 4, which include a Random Forest model ($140kB$), an AdaBoost model ($7.32kB$), and a HistGBC model ($380kB$). The hyperparameter configurations for all models remained unchanged (Table 4.2). Algorithm 1 outlines the logic of our detection function, which involved removing non-changing features, applying feature engineering, and implementing a standard scaler to normalize all features, ensuring data consistency for the imported models. Notably, training of the models can be performed offline, and subsequently imported. These models demonstrated robust generalizability across previously generated public datasets available on [90] (Section 4.5). Therefore, we aim to evaluate the extent to which these models maintain their generalizability on the new datasets presented in this work.

Algorithm 1: Detection function per model

Input : Data collected for detection (X_{test})
Output: Predicted results ($\hat{y}_{model}$)
$X_{test} \leftarrow DroppingStaticFeatures(X_{test})$;
$X_{test} \leftarrow FeatureEngineering(X_{test})$;
$scaler \leftarrow MinMaxScaler()$;
$X_{test_scaled} \leftarrow scaler.FitTransform(X_{test})$;
$model = ImportModel()$;
$\hat{y}_{model} \leftarrow model.predict(X_{test_scaled})$;

5.4.4 Performance Metrics

For evaluation, and to address the challenge of imbalanced datasets, we employed the following performance metrics in our tests:

- **MCC** [92]: A robust metric for imbalanced datasets, calculated by considering the complete confusion matrix according to eq. (4.4).
- **Confusion matrix**: A matrix presenting the True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP) resulting from the model's detection process.

5.5 Results and Analysis

Given the imported models from our previous chapter, the imbalanced data distribution present in the testing datasets, as shown in Table 5.4, is not a primary concern. Our trained models demonstrated strong performance on imbalanced testing datasets, indicating their suitability for the current testing datasets. Therefore, following data preparation via Algorithm 1, we evaluated the detection results using the MCC and confusion matrices.

5.5.1 MCC Evaluation

Figure 5.3 presents the MCC values for each model across the seven test scenarios. Notably, the AdaBoost model consistently outperformed the other two, with the exception of the $JamDDoS\ (N = 20)$ and $Security - JamDDoS\ (N = 20)$ scenarios, where performance was nearly equivalent. This aligns with the results from the original testing data (shown as horizontal lines, corresponding to the training dataset), indicating a slight advantage for AdaBoost.

The models demonstrated their capability to detect N-DDoS attacks with varying attacker counts, even under differing traffic conditions and geographical locations. Furthermore, they effectively detected $JamDDoS$, $Security\ DDoS$, and $Security-JamDDoS$ attacks, further demonstrating their generalizability to diverse attack scenarios beyond the original training data.

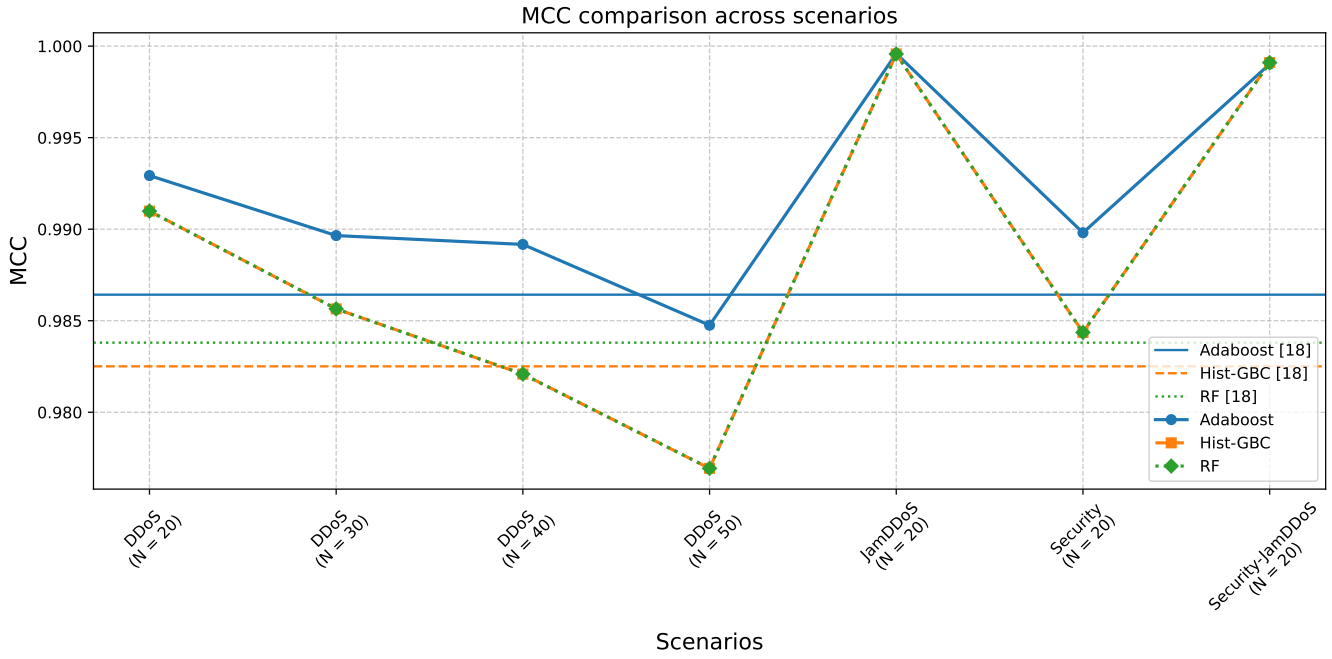


Figure 5.3: Matthews correlation coefficient values across seven scenarios for each model.

These results suggest that these models are strong candidates for a meta-learner, an additional detection layer that could enhance overall performance. By integrating the strengths of multiple models, a meta-learner could potentially achieve even higher detection rates, a promising direction for future research.

However, further investigation is necessary to determine the optimal model selection and the trade-offs between different performance metrics. It is crucial to examine whether a high MCC value invariably correlates with superior performance, or if other factors, such as False Positives (FPs) and False Negatives (FNs), should also be considered. Therefore, we analyzed the confusion matrices of all models in the subsequent section.

5.5.2 Evaluation of Confusion Matrices

While favorable MCC values suggest a well-balanced model in predictions over imbalanced datasets, the overlapping MCC values between the three models in two scenarios ($JamDDoS\ (N = 20)$ and $Security - JamDDoS\ (N = 20)$) in Figure 5.3, and particularly the

overlap between all MCC values of RF and HistGBC, necessitate a more granular analysis of the confusion matrices. Figures 5.4 and 5.5 present the confusion matrices for the three models across the seven scenarios. Examining these matrices provides insights into the specific strengths and weaknesses of each model, including their ability to accurately classify True Positives (TPs) and True Negatives (TNs), as well as their tendency for FPs and FNs.

Analyzing Figure 5.4a and 5.4b, we observe that the higher MCC value obtained from Figure 5.3 for scenario DDoS $(N = 20)$ of the AdaBoost model is attributed to the lower number $(215 < 283)$ of FNs, with a slightly higher number of FPs $(11 > 5)$. Evaluating these results within the domain of connected cars, we can tolerate a marginal increase in FPs, prioritizing the minimization of undetected attack messages, i.e., lower FNs. This analysis extends to the DDoS $(N = 30, 40, 50)$ and $Security$ $(N = 20)$ scenarios. However, for the $JamDDoS$ $(N = 20)$ and $Security-JamDDoS$ $(N = 20)$ scenarios, the trade-off is less noticeable, resulting in similar MCC values across the models, as depicted in Figure 5.3.

Figure 5.4a and 5.4c reveal instances where the RF and HistGBC models yield similar predictions. While this may seem counterintuitive, it can arise from several factors:

- *Similar decision boundaries:* Both models learn highly analogous decision boundaries due to the structured and relatively uncomplicated nature of the data.
- *Balanced hyperparameters:* The models' hyperparameters are finely tuned to the problem domain, enabling the detection of identical patterns.
- *Non-random data splits:* The training and testing data split is not random and particularly conducive to the models.
- *Highly informative features:* Certain features exhibit strong correlations with the target feature without inducing overfitting.

Considering the specific context of our study, we can exclude the possibility of "non-random data splits" because the training and testing data originate from distinct sources, as indicated in Table 5.2. Furthermore, given that the models were imported and demonstrated varying performance on the same testing data (horizontal lines, Figure 5.3), it is improbable that they possess "similar decision boundaries" or "balanced hyperparameters".

The most plausible explanation for the observed similarity in predictions is the presence of "highly informative features". Given the inclusion of a single engineered feature across all testing datasets, which significantly influences the classification process, the models rely heavily on this

feature for predictions. This can lead to convergent outcomes from different models. This hypothesis is supported by the consistent patterns observed in the confusion matrices of Figure 5.4 and 5.5. Examining the right and left columns of these figures reveals that the models tend to produce similar predictions, suggesting a shared reliance on the same underlying information.

Comparing model performance across the different scenarios, we observe a marginal decrease in performance for all three models when detecting DDoS $(N = 20, 30, 40, 50)$ in Figure 5.4. However, considering the significant variations in attacker numbers and their impact on message transmission (for *Scenario Set IV* Subsection 3.5.4), this decrease is not substantial. While a more in-depth analysis could explore this performance change, the high MCC values indicate successful model generalization to varying attacker counts.

Figure 5.5 compares model performance across the $JamDDoS$, $Security$, and $Security - JamDDoS$ scenarios, revealing strong performance in the $JamDDoS$ and $Security - JamDDoS$ scenarios. Conversely, a slight performance decrease is observed for AdaBoost, and a more pronounced decrease for RF and HistGBC, in the $Security$ scenario. Despite these decreases, the high MCC values affirm successful model generalization to these morphing DDoS attacks.

Elaborating on these three morphing DDoS attack scenarios, it appears that our model perceives them as similar. This is attributed to the processing of CAM messages at the facilities layer, adhering to the ETSI protocol stack (Figure 2.10). Consequently, irrespective of security overhead or increased transmission power, our models process these messages uniformly, explaining their robust performance against such morphing DDoS attacks.

5.6 Conclusion

In this study, we proposed an IDS internal workflow and placement that aligns with the state-of-the-art C-ITS architecture. We illustrated how cooperation can enhance IDS capabilities across geographically diverse deployments and outlined the protocols necessary for achieving interoperable IDSs. Furthermore, we evaluated the generalizability of our previously developed models in detecting morphing DDoS attacks, despite being trained on traditional DDoS attack data. The obtained MCC values demonstrated the robustness of these models in detecting such attacks. These developed models contribute to the anomaly-based component of the IDS. Additionally, within the IDS presentation, we addressed response mechanisms that can be implemented to counteract the effects of these attacks.

Given that our results indicated the models perceived the attacks as similar, attributed to a

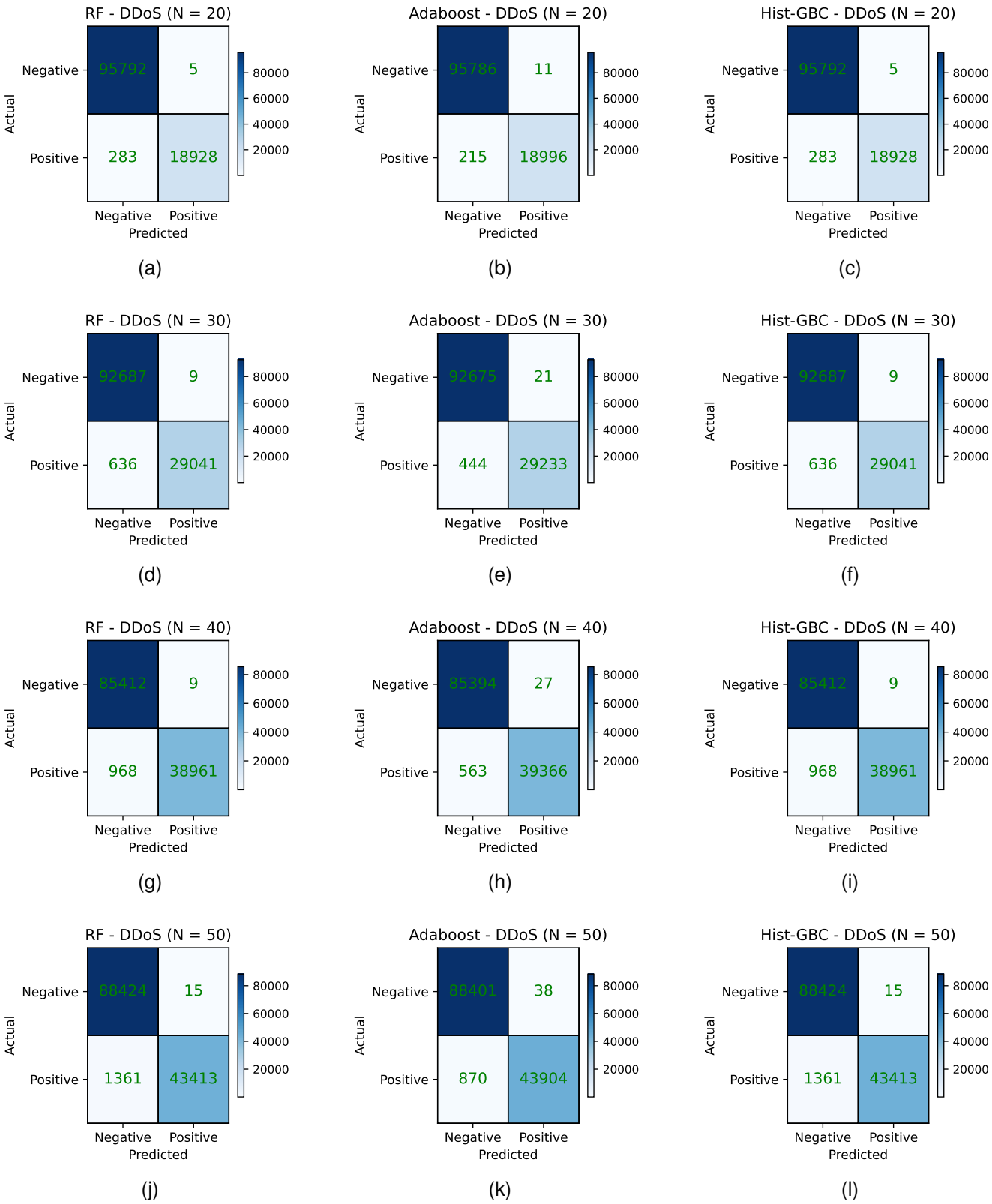


Figure 5.4: Confusion matrices

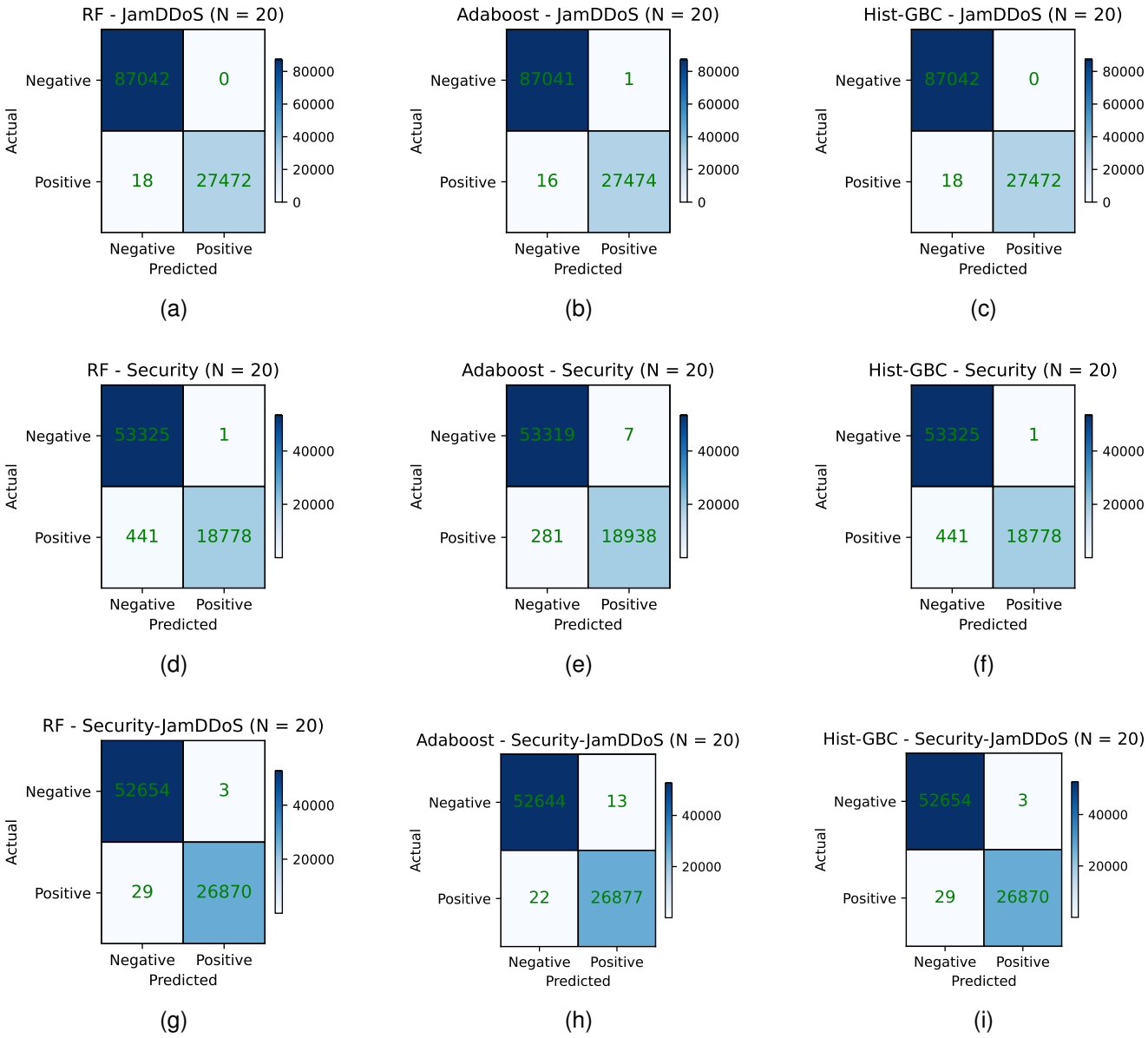


Figure 5.5: Confusion matrices (continued)

shared temporal characteristic effectively classified by the models, we further investigated the generalizability of these models to determine their limitations. To this end, and to challenge our detection approach, the next chapter introduces the integration of the Sybil concept into the DDoS attack model, while maintaining adherence to communication standards within C-ITS.

Chapter 6

Generalization and Limitations of DDoS Detection Models in Sybil-Based Attacks on Connected Cars

6.1 Introduction

Threat models are becoming increasingly sophisticated as we enter a rapidly evolving technological era. These models are no longer limited to single attack vectors but have expanded to encompass multi-vector attacks on the Internet, as reported by Cloudflare [106]. As transportation systems, particularly connected cars, represent one of the fastest-evolving technologies, they are increasingly vulnerable to such multi-vector attacks. For instance, a car performing a DoS attack while altering its identification information transforms its attack vector into a Sybil DoS attack. Furthermore, if attackers gain control over multiple models of OBUs in cars, they can launch a Sybil-based DDoS attack on the system. When applied in a safety-critical domain, these attacks pose a significant risk of compromising connected cars, potentially leading to severely detrimental consequences. The mitigation of such multi-vector attacks remains an open challenge, particularly since these attack models can circumvent detection models designed for single-vector attacks.

This chapter demonstrates how a DDoS attack can be transformed into a Sybil DDoS attack that bypasses DDoS detectors in a safety-related connected car environment. Subsequently, it presents an optimized data-handling approach that forms the basis of our proposed ensemble hard voting detection model.

6.2 Sybil Attacks

6.2.1 Definition of Sybil Attack

The general definition of a Sybil attack, introduced by [21] in 2002, refers to a user forging multiple fake identities to represent themselves. In the context of connected cars, this definition is slightly modified due to the presence of a PKI that manages the identities of all cars. Therefore, a Sybil attack in this domain is defined as an attacker using valid but compromised or stolen identities to impersonate legitimate cars. Fake or invalid identities, however, will not pass PKI signature verification and will be immediately dropped.

The use of multiple stolen or compromised identities is feasible due to the presence of Authorization Tickets (ATs) [107]. Cars typically request a batch of ATs (short-term certificates) from authorities for communication on the network, instead of using their real car IDs (long-term certificates). Each AT contains a unique PseudoID and relevant car information. These PseudoIDs help preserve the car's privacy by preventing global position tracking, as linking short-term certificates to long-term ones is required for location identification. However, these ATs can be leaked due to misuse by faulty OBUs or the compromise of OBUs, enabling an attacker to exploit them for a Sybil attack.

6.2.2 Sybil Attack Types and Intended Effects

In connected cars, Sybil attacks are typically executed using messages that contain identification fields. One primary candidate is the CAM message, which is designed to enhance situational awareness and is broadcast at a frequency of $10Hz$, as specified by the ETSI standard [30]. Given the high volume of CAM messages exchanged between cars, attackers can exploit them to generate multiple forged identities. Another potential target is the DENM [32], which is used to disseminate critical alerts regarding road conditions and traffic. As DENMs carry high-priority information that demands immediate attention, they present a valuable opportunity for attackers to manipulate traffic flow or cause disruptions. Moreover, various types of Sybil attacks can be executed using these messages, each with distinct objectives and impacts. The following outlines a non-exhaustive list of possible Sybil attack types within the domain of connected cars, along with their intended effects.

- **Untargeted Sybil**: A fundamental Sybil attack in which the adversary generates fake, self-signed PseudoIDs to overwhelm the signature verification system [108].

- **Random Sybil**: The attacker transmits messages containing randomized data but valid signatures [108, 109], thereby injecting meaningless information into applications relying on these messages, such as traffic optimization services. This disrupts real-time traffic assessments and alternative route recommendations.
- **Replay Sybil**: The attacker replays previously received messages while altering the PseudoID [108, 109], aiming to mislead the system regarding the actual identity and presence of a legitimate car.
- **Static Sybil**: The attacker broadcasts messages indicating stationary or slow-moving cars [109], fabricating artificial traffic congestion to manipulate road conditions.
- **Logical Sybil**: The attacker injects plausible but fraudulent data with valid certificates [110, 108], simulating nonexistent cars in motion to create fake moving congestion on the road.
- **Power-Controlled Sybil**: A more sophisticated variant of the Logical Sybil attack, where the attacker modulates transmission power based on the forged location data in the message [111, 112]. This enables the attack to evade Received Signal Strength Indicator (RSSI)-based detection while simultaneously introducing falsified data to compromise the system.
- **Hybrid Attacks**: Attackers may combine multiple Sybil techniques to enhance effectiveness. For example, a DoS Sybil attack can involve repeatedly using the same PseudoID to flood the network, leading to DoS conditions. Other advanced combinations may further increase stealthiness and impact.

6.2.3 Basis of a Sybil Attack Model

A Sybil attack is a sophisticated high-level attack in which the adversary has already compromised the operating system of the targeted devices. With privileged access, the attacker can:

- Modify application-layer messages, e.g., altering location data in a CAM message.
- Manipulate security credentials, e.g., sending messages with different certificates by cycling through PseudoIDs within the PseudoID pool.
- Adjust transmission power before sending messages to evade detection mechanisms.

These capabilities make the Sybil attack one of the most stealthy and impactful threats, allowing adversaries to inject falsified information using valid user certificates. As a result, countermeasures are being actively developed to mitigate such threats.

6.2.4 Countermeasures of Sybil Attacks

According to a recent short survey [113], countermeasures against Sybil attacks can be categorized as follows:

- **Trust-based**: Evaluates multiple parameters to compute a trust score, assessing the credibility of a node.
- **Machine learning-based**: Utilizes machine learning algorithms to classify and detect Sybil-generated messages.
- **Blockchain-based**: Leverages blockchain technology to secure the network of connected cars from Sybil attacks.
- **Cryptography-based**: Incorporates cryptographic mechanisms within the PKI framework to prevent unauthorized access and identity forgery.
- **Routing protocol-based**: Designs secure routing protocols to counteract Sybil attacks in multi-hop communications.

6.3 Related Work

The related works presented in this section are categorized based on the Sybil attack models addressed by previous studies. Since the datasets used in this work have been recently published, no prior studies have utilized them, making direct performance comparisons infeasible. Therefore, this section focuses on reviewing Sybil attack models and the most relevant proposed solution approaches, even though they do not explicitly address DDoS attacks.

In [110], the authors proposed an attacker model that analyzes the current road traffic conditions, such as the distance between adjacent cars and the average velocity of the lane under consideration. Using this data, the attacker generates virtual nodes by sending forged CAMs. For detection, the authors examined the driving patterns of cars by constructing a driving pattern matrix for each car over a specified time interval. Detection is based on the concept of similarity between these matrices, with a predefined threshold used to identify deviations. In an extended version of the work [114], the authors employed a Support Vector Machine (SVM) approach for detection, utilizing three different kernel functions. In [115], an alternative classification approach was introduced using K-Nearest Neighbors (KNN), which was optimized to address runtime complexity issues.

In [108], the authors introduced four distinct Sybil attack models:

1. Traffic Congestion Sybil: This model aims to create plausible ghost cars to simulate traffic congestion.
2. Data Replay Sybil: This attack replays position messages broadcasted by a victim car using fake IDs.
3. DoS Random Sybil: This model floods the network with random data, utilizing different fake IDs.
4. DoS Disruptive Sybil: This attack replays position messages from multiple victim cars to disrupt the network.

Additionally, the authors proposed a misbehavior authority solution consisting of three phases: (1) detecting misbehavior associated with a single PseudoID; (2) linking related PseudoIDs; and finally, (3) detecting the type of Sybil attack using a Recurrent Neural Network (RNN).

The work presented in [116] revisits the third attack model from [108], which they identify as a DDoS Sybil attack. The proposed detection architecture involves three key entities: a traffic processor that monitors and pre-processes all traffic data, an entropy calculator that computes entropy values of real-time traffic and sends them to the final entity, the detection module. The entropy calculation for a selected parameter (vehicle ID) is performed using a sliding window approach with incremental updates to ensure smoothness and quick response times. The detection module then applies a fast quartile deviation check algorithm to detect the presence of Sybil DDoS attacks.

In [109], the authors utilized a dataset extracted from their previous work [117], where they studied three Sybil attack models. The paper revisits a Sybil scenario with random values, in which the attacker fabricates random messages; a static value scenario, where the attacker simulates congestion by modifying the timestamp and signature of the messages and changing the speed and acceleration to zero; and a replayed value scenario, where the attacker replays messages by altering their timestamps and signatures. For detection, the authors used four machine learning models: Logistic Regression, Support Vector Machine (SVM), Random Forest, and Artificial Neural Network (ANN).

In [111], the authors proposed five scenarios, two of which introduced more advanced attack models. These advanced models involve an attacker modifying the signal transmission of its messages. This technique allows the attacker to bypass detection models that rely on RSSI measurements. Additionally, the paper presents a two-level detection model: the first level uses

RSSI measurements to detect conventional Sybil attacks, while the second level incorporates a trajectory analysis of the car, supported by blockchain security, to detect the more sophisticated attacks.

The authors in [112] revisited the smart Sybil attacker that employs power control during the attack. They considered two attack models: one involving power control and the other without. To counter these attack models, the authors proposed a solution based on a Longest Common Subsequence (LCSS) similarity algorithm to detect attackers without power control. For attackers utilizing power control, they employed a Change Point Detection method, which identifies changes in the mean value of the Received Signal Strength (RSS) time series.

Table 6.1 provides a summary of the related works discussed.

Table 6.1: Summary of related works

Reference	Sybil attack model	Description	Intended effects	Proposed solution
[110, 114, 115]	Virtual node creation	Sending forged CAM messages based on learnt traffic data	Fake congestion	Driving Pattern Matrix followed by similarity check, support vector machine, or K-Nearest Neighbors
[108]	1. Traffic congestion 2. Data replay 3. Random 4. DoS disruptive	1. Creating plausible ghost cars 2. Replaying position messages 3. Random message spamming 4. Replay multiple position messages	1. Fake congestion 2. Report legitimate cars as malicious 3. Overwhelm misbehavior detectors 4. Degradation of system's safety	A misbehavior authority solution based on local detectors and machine learning approaches
[116]	Random	Random message spamming, identified as a DDoS	Overwhelm misbehavior detectors	Entropy calculation followed by fast quartile deviation check algorithm
[109]	1. Random 2. Static 3. Replay	1. Broadcasting messages with random data fields 2. Broadcasting messages representing stopped cars 3. Replaying messages by valid signatures	1. Overwhelm misbehavior detectors 2. Fake congestion 3. Report legitimate cars as malicious	Machine learning based
[111]	Power control	Modifying the signal transmission	Bypass RSSI based detectors	RSSI based followed by a trajectory based detection supported by blockchain
[112]	Power control	Modifying the signal transmission	Bypass RSSI based detectors	Longest common subsequence and change point detection methods
Our work	PseudoID sweep	Cycle between authorized PseudoIDs	Deceive detectors	Ensemble learning based

6.4 System Models

6.4.1 Attack Model

The Sybil attack model adopted in this work focuses solely on changing the PseudoID using a Round-Robin algorithm (Figure 6.1). Each time the attacker sends a message (with a rate of 50 CAMs$/sec$), they alter their PseudoID according to the attack mechanism detailed later in subsection 6.5.3. It is assumed that the attacker has either deployed malware in specific OBUs or has direct control over compromised OBUs, enabling them to execute the attack. Consequently, all attack messages are transmitted with valid certificates, allowing them to bypass PKI checks. The attacker's primary goal is to evade DDoS detection mechanisms, while a secondary objective is to compromise advanced detection systems, potentially causing misclassification of legitimate messages as attack traffic. Figure 6.2 shows the impact of the combined Sybil-DDoS attack with five PseudoID changes, where after each transmission at instant T the car is represented with a new PseudoID, i.e. the five different colors.

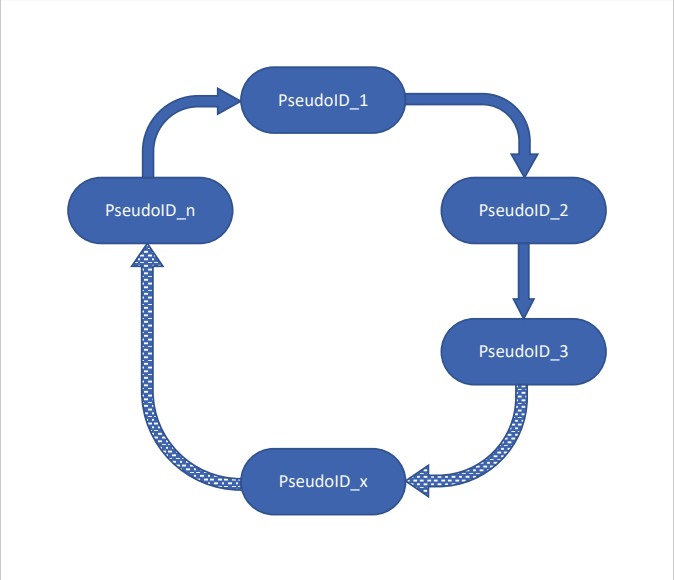


Figure 6.1: Round Robin for n PseudoIDs

The process of PseudoID change is regulated by specific rules, as outlined in Annex A of the ETSI standard [83]. For example, according to Table A.1 in [83], the SCOOP@F project [4] enforces a Round-Robin-based PseudoID change after $40,000$ signatures or one hour, whichever occurs first, within the validity period of the PseudoID pool, after which the pool itself is replaced. Similarly, the Car-2-Car Communication Consortium (Table A.2 in [83]) specifies a randomized change interval between 10 to 30 minutes, while the Society of Automotive Engineer (SAE) standard (Table A.3 in [83]) recommends a change every five minutes. Since the Sybil attack an-

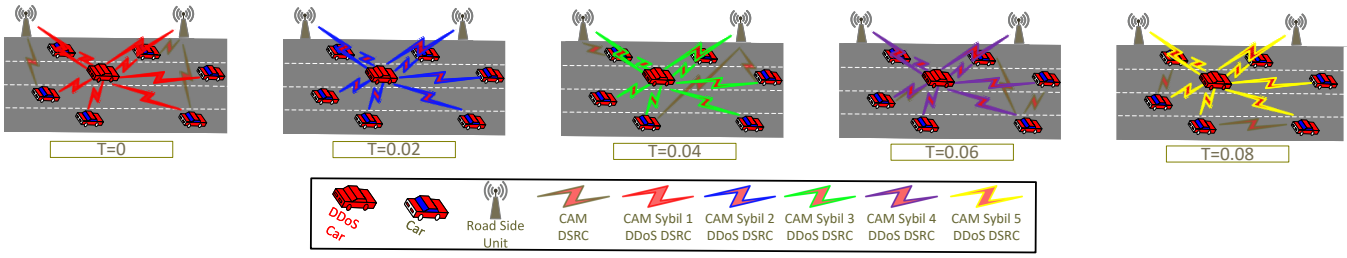


Figure 6.2: Sybil DDoS attack with a rate 5 times the normal rate of transmission of CAM messages by cars.

alyzed in this work occurs within a much shorter time frame (less than a minute), the attacker remains fully compliant with these standardized rules. Consequently, this exploitation highlights a vulnerability within the existing PseudoID change mechanisms, allowing malicious actors to evade detection while adhering to protocol specifications.

6.4.2 Detection Model

We retained the detection models from our previous study, namely RF, AdaBoost, and HistGBC, to evaluate their resilience against the newly introduced Sybil attack model. Furthermore, to enhance model performance, we expanded the training and validation datasets following the balanced dataset recommendations outlined in [118]. To combine the outputs of these models, we used a voting classifier [119], which integrates multiple predictions to improve performance. This combination can be performed using either:

- Soft voting: Averages the predicted probabilities from each classifier to determine the final decision.
- Hard voting: Selects the majority class prediction based on the highest number of votes.

Additionally, weighting mechanism can be applied to assign different importance levels to each model within the voting classifier. In this study, we evaluated both hard and soft voting strategies to identify the most effective approach for detecting the Sybil-based DDoS attack.

6.5 Experimental Setup and Attack Methodology

This section starts by defining our simulation environment and outlining the necessary assumptions. Subsequently, we briefly review the dataset contents of the original training dataset

used in Chapter 4. We then present the attack mechanism employed in our attack model, alongside the performance metrics. Finally, we evaluate the generalization results of the models across various PseudoID sweep scenarios, highlighting their limitations.

6.5.1 Simulation Environment

Our simulation environment is structured into three key components: the software used, the studied scenario, and the vital assumptions considered.

Software: In this work, we kept using Artery [72] to generate CAM messages in the correct format, SUMO [19] for car traffic modeling, and OMNeT++ [71] as the network event simulator.

Scenario: We maintained the same scenario as in Chapter 4, studying a region of interest around the highways A1, A3, A4, and A6 in Luxembourg City, as shown in Figure 6.3. The attacker percentage was kept at 5% where attackers are chosen according to a randomly generated number.

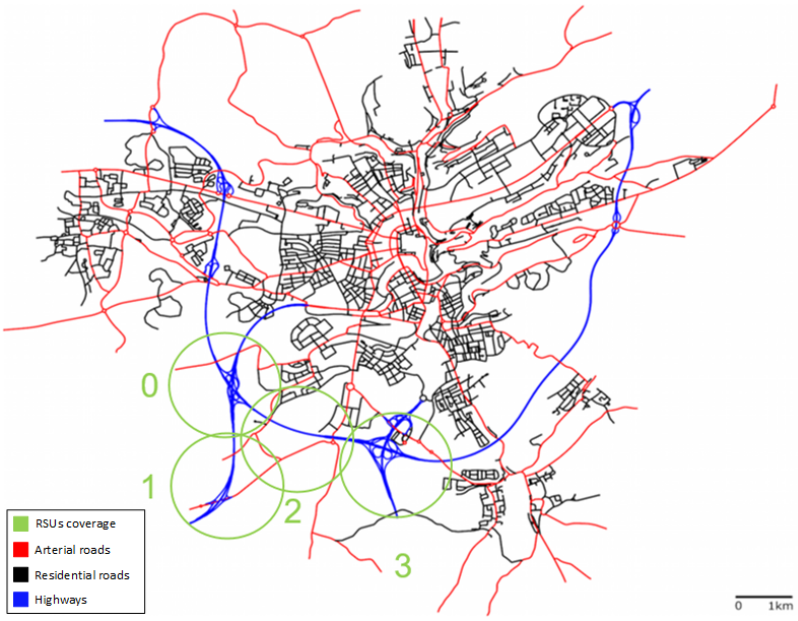


Figure 6.3: Modified Luxembourg [20] map

Assumptions:

(i) **Attacker Model**: The attacker model extends our previous work by adopting a more sophisticated approach. The attacker cycles through different PseudoIDs to resemble normal car behavior, sending messages at a normal rate. However, since Artery does not support dynamic PseudoID changes, we applied the attack mechanism (subsection 6.5.3) after generating the datasets. The PseudoID changes follow the Round-Robin algorithm used in the SCOOP@F project (subsection 6.4.1). The simulation remains within the validity period of the PseudoID pool, meaning the same pool is used for all attack periods.

(ii) **PseudoIDs of Normal Cars**: In alignment with the SCOOP@F project, normal cars do not change their PseudoIDs during the six-minute simulation, as the predefined conditions ($40,000$ signatures or 1 hour) are not met.

(iii) **Channel Busy Ratio Recording**: The new dataset includes a newly introduced $ChannelLoad$ field (subsection 6.6.1), collected at the RSU level. This field represents the CBR, which quantifies the fraction of time a radio channel is occupied with transmissions, as defined by the ETSI standard [70]. The standard specifies CBR calculation every $100\ ms$ to calculate channel load value. However, since attacks occur at the millisecond scale, this interval is insufficient to capture their precise impact. To enhance detection accuracy, we modified the CBR calculation interval to $1\ ms$ for RSUs only. This allows us to record CBR changes at finer granularity and associate them with each received CAM message.

6.5.2 Previous Dataset

The original training dataset used in Chapter 4, consists of CAM messages structured according to the ETSI standard [30]. This dataset includes the full structure of the CAM messages. We focus on a set of features that vary based on simulation parameters (full description in Subsection 4.4.3 of Chapter 4), as defined according to the ETSI standard [120], the brief contents are:

Standard Features: The following features (except $SimulationTime$) are directly extracted from the CAM messages that include: $SimulationTime$, $StationID$, $GenerationDeltaTime$, $Longitude$, $Latitude$, $Heading$, $Speed$, $LongitudinalAcceleration$, $Curvature$, and $YawRate$.

Engineered Feature: In addition to the standard features, we introduced an engineered feature in the preprocessing phase of our previous work:

(i) $TimeDifference$: The time interval between two consecutive CAM messages generated by the same $StationID$.

This engineered feature will serve as the basis for the attack mechanism presented in the subsequent subsection.

6.5.3 Attack Mechanism

To evaluate the robustness and generalizability of our previously proposed solution against different attack scenarios, we modified the collected datasets to simulate a Sybil DDoS attack, using a cyclic change of PseudoIDs (as described in algorithm 2). This modification was applied using the attack synthesis algorithm (algorithm 3) across the studied datasets.

Cyclic Change of Pseudonymous IDs: The algorithm presented in algorithm 2 implements a Round Robin approach to cycle through the pool of PseudoIDs available to the attacker in its ATs. Each time the algorithm is invoked, it receives the last used ID by the attacker ($oldID$), the cyclic value C to be applied to the $oldID$, as well as the original attacker ID ($attackerID$) to maintain reference to the specific attacker being simulated. Additionally, the $cyclic_limit$ is provided to ensure the algorithm does not exceed the allowed number of PseudoIDs. This algorithm is flexible and can be configured for any required number of cycles (n sweeps); in this work, we used it to perform 2, 3, 4, and 5 ID cycles. It is important to note that while this algorithm cycles through PseudoIDs, it does not guarantee the uniqueness of the generated IDs. Therefore, we performed an additional verification step to ensure the uniqueness of PseudoIDs after executing the attack.

Applying Cyclic Change of PseudoIDs to the Dataset: The previously described cyclic change algorithm is part of a larger process that emulates a Sybil DDoS attack by iterating through the entire dataset for each $attackerID$ present in the list of attacker IDs. During each attack period, when a message associated with the current attacker is encountered, the algorithm calculates the cyclic value based on the $TimeDifference$ field found in the dataset. This field is divided by 20, corresponding to the DDoS rate under investigation (i.e., 50CAMs$/sec$, which implies a 20 ms interval between messages). Then, the $messageID$ is changed. Before processing

the second message, the $oldID$ is updated to track the last used PseudoID of the attacker under study. This iterative process explains why the algorithm must loop through the entire dataset for each $attackerID$ in the list, ensuring that the attack is accurately simulated across all messages associated with the attacker.

Algorithm 2: Cyclic addition of sweep value (n)

Input : $oldID$: Last used ID, C: Cyclic value
Input : $attackerID$: AttackerID under study
Input : $cyclic_limit$: AttackerID plus n
Output: $sweepedID$: Sweeped AttackerID
$result \leftarrow (oldID + C)\%cyclic_limit$;
if $result \geq attackerID$ **then**
    $sweepedID \leftarrow result$;
else
    $result \leftarrow result\%n$;
    $sweepedID \leftarrow result + attackerID$;
end

Algorithm 3: Attack synthesis of sweep value (n)

Input : df: Dataset, $attackers_list$: List of attackers
Output: df: Modified dataset
for *(all IDs in* $attackers_list$*)* **do**
    $oldID \leftarrow attackerID$;
    for *(all messages in* df*)* **do**
        if *(in attack period and* $messageID == attackerID$*)* **then**
            $C \leftarrow TimeDifference/20$;
            $messageID \leftarrow Cyclic_addition(oldID, C, attackerID + n, attackerID)$;
            $oldID \leftarrow messageID$;
        end
    end
end

6.5.4 Performance Metrics

In terms of evaluation and considering the issue of imbalanced datasets presented in Chapter 4, we used the following performance metrics in our tests:

- **MCC**: is a robust metric against imbalanced datasets that is calculated by considering the whole confusion matrix according to eq. (4.4).

- **ROC curve**: is a visualization of the performance of a classification model for different values of classification threshold.
- **AUC**: is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve.
- **Confusion matrix**: a matrix presenting the TN, FP, FN, and TP of the model after the detection process.

6.5.5 Testing the Models' Generalizability

The previously developed models were utilized with the same configurations specified in Table 4.2. These models were trained on the same four-minute original training dataset collected in Chapter 4. To evaluate their generalizability and ensure precise comparison throughout the work, we used the testing dataset from this study, specifically the dataset labeled "Testing (random)" in Table 6.5. We then applied the attack synthesis algorithm (Algorithm 3) with $n = 2, 3, 4,$ and 5, generating four distinct testing datasets.

The confusion matrices for the previously developed models - RF, AdaBoost, and HistGBC - are presented in Tables 6.2, 6.3, and 6.4, respectively. The results indicate strong generalization performance (MCC $> 90\%$) for sweeps of 2, 3, and 4 PseudoIDs. Among these cases, AdaBoost consistently outperformed the other models by a margin of 1% to 3% in terms of the MCC metric. However, for the 5-PseudoID sweep, none of the models were able to effectively detect the Sybil DDoS attack, necessitating model retraining to address this scenario.

Table 6.2: Confusion matrix data of RF

Sweep number	TN	FP	FN	TP	MCC
2 IDs	894,467	22	6,169	114,320	97.06 %
3 IDs	894,467	22	11,706	108,783	94.39 %
4 IDs	894,467	22	16,279	104,210	92.15 %
5 IDs	894,467	22	120,477	12	0.42 %

Table 6.3: Confusion matrix data of Adaboost

Sweep number	TN	FP	FN	TP	MCC
2 IDs	894,451	38	3,388	117,101	98.38 %
3 IDs	894,452	37	4,497	115,992	97.85 %
4 IDs	894,451	38	8,449	112,040	95.96 %
5 IDs	894,452	37	120,475	14	0.34 %

Table 6.4: Confusion matrix data of HistGBC

Sweep number	TN	FP	FN	TP	MCC
2 IDs	894,473	16	6,056	114,433	97.12 %
3 IDs	894,473	16	11,256	109,233	94.61 %
4 IDs	894,473	16	16,270	104,219	92.16 %
5 IDs	894,473	16	120,480	9	0.37 %

These findings show that the models, initially trained on a DDoS attack with a rate of 50CAMs $/sec$, successfully generalized to sweeps of 2, 3, and 4 PseudoIDs. However, their limitations became evident when confronted with the 5-PseudoID sweep. In our previous dataset, the $TimeDifference$ feature played a crucial role in attack detection. Given the attack mechanism in this study, this feature has lost its effectiveness. To address this issue, we propose incorporating additional features into the new dataset (Subsection 6.6.1) to enhance the models' learning capabilities and improve detection performance against the new attack model.

6.6 Enhanced Dataset and Optimized Classification Models

This section is organized into three parts. The first part details the newly generated dataset. The second part presents statistical information, including the data and time distribution of the dataset. Finally, the third part introduces the newly developed models and analyzes their efficacy in enhancing detection performance.

6.6.1 New Dataset

In accordance with the assumptions regarding the more precise recording of the CBR, we implemented the necessary changes and re-ran the simulations to generate the updated datasets. Additionally, we recorded the corresponding RSU that received each CAM message. This led to the addition of two new features in the dataset, as detailed in [90].

New Features: The following features were introduced in the dataset to enhance the information captured during the simulation:

(i) RSU_X: The identifier of the RSU (denoted as X, where $X \in \{0, 1, 2, 3\}$) that received the corresponding CAM message.

(ii) $ChannelLoad$: The CBR recorded by the corresponding RSU at the time of reception of the CAM message.

Given that our new smart attack model bypasses the previously developed detection models, we incorporated additional engineered features in the dataset to mitigate the performance degradation. These new features were added during the preprocessing phase, which increased the time required for preprocessing.

Engineered Features: The following engineered features were introduced to enhance the dataset for more effective attack detection:

(i) $Message_Count$: The number of messages received from the corresponding $StationID$ in the last $0.495sec$ (selected instead of 0.5 to account for floating-point representation inaccuracies).

(ii) $Haversine_Distance$: The great-circle distance between two points (the car and the RSU) on a sphere, calculated using their longitudes and latitudes. This distance is computed using the formula presented in eq. (6.1).

(iii) New $TimeDifference$: The time difference between two consecutive messages generated by the same $StationID$ after the execution of the attack.

The counting interval for the $Message_Count$ feature was intentionally set to a small value ($0.495s$) to minimize computation time, as it was deemed unnecessary to use a full one-second interval as it can be deduced from the chosen interval. In addition, computation time can be further optimized under the cost of including multi-processing and increasing computation power. The $Haversine_Distance$ was selected due to its computational efficiency, as it requires less time to calculate compared to alternative methods such as the geodesic distance from the $geopy$ library, while still offering an acceptable margin of precision.

$$a = \sin^2(\frac{\Delta\phi}{2}) + \cos\phi_1 * \cos\phi_2 * \sin^2(\frac{\Delta\lambda}{2})$$

$$c = 2 * \arctan(\frac{\sqrt{a}}{\sqrt{1-a}})$$

$$d = R * c \tag{6.1}$$

where

$$\phi_1, \phi_2 = \textit{latitudes of points 1, 2 resp.}$$
$$\lambda_1, \lambda_2 = \textit{longitudes of points 1, 2 resp.}$$
$$\Delta\phi = \phi_2 - \phi_1$$
$$\Delta\lambda = \lambda_2 - \lambda_1$$

6.6.2 Dataset Statistics

The statistics of the raw updated dataset are presented in Tables 6.5 and 6.6, which detail the distribution of the number of messages and the intervals of the attacks. The values within parentheses in the tables represent the unsigned integer seed provided to the $srand$ function in C++ to ensure reproducibility. Specifically, two datasets (1 and 2) are generated based on predefined seeds, while a third dataset is produced with a random seed. In all datasets, the first minute consists solely of normal messages, while the remaining five minutes contain attack periods. These attack periods are characterized by random starting times (column 5 of Table 6.6) and random durations (column 6 of the same table).

Table 6.5: Data distribution

Data type	Total messages	Normal messages	Attack messages	Percentage of attack messages
Training (1)	999,909	920,682	79,227	7.92 %
Validation (2)	986,219	915,979	70,240	7.12 %
Testing (random)	1,014,978	894,489	120,489	11.87 %

Table 6.6: Time distribution

Data type	Total time	Total normal time	Total attack time	Starting intervals *	Attack intervals *
Training (1)	6 min	4 min 58 sec	62 sec	13;16;27; 25;23 sec	6;17;21; 6;12 sec
Validation (2)	6 min	5 min	60 sec	0;19;8; 5;21 sec	5;17;8; 12;18 sec
Testing (random)	6 min	4 min 26 sec	94 sec	0;17;20; 20;19 sec	17;20;6; 21;30 sec

*First simulation minute does not contain an attack

6.6.3 Developing New Models

Primary Training of Models: The effectiveness of the attack against our previously developed models highlighted the necessity of training new models. Given that three datasets were generated using different seeds, we allocated the first full dataset for training, the second for validation, and the third for testing. This dataset splitting strategy aligns with the recommendations of [118], which suggests maintaining balanced training and validation dataset sizes to improve model performance. Following this approach, each dataset provided six minutes of data, as presented in Table 6.6. The attack synthesis with $n = 5$ was applied to all of the three datasets.

After completing the training and validation phases, we evaluated the models on the test dataset. The results, summarized in Table 6.7, indicate a substantial improvement in performance. The MCC increased significantly from an average of 0.38% in the previous models (row 5 IDs, Tables 6.2, 6.3, and 6.4) to approximately 72.66% across the three newly trained models. This improvement can be attributed to the introduction of new and engineered features. Given the nature of the Sybil concept introduced in this study, it is expected that the models exhibit a higher rate of FNs, as the attacker's primary objective is to evade detection by making malicious messages appear legitimate. However, an excessive number of FPs must also be avoided, as this would lead to normal messages being misclassified as attack messages, thereby amplifying the perceived impact of the attack.

The results show that an overall MCC of approximately 72% is reasonable given the complexity of the attack. However, the number of FPs remains relatively high, considering the standard CAM message transmission rate of 10 CAMs$/sec$. A rough estimation, based on the RF model results, indicates that with 2510 FPs, approximately 251 normal cars $(2510/10)$ were misclassified as malicious at least once. To better understand the origin of these misclassified messages and determine where they were received, we extended our analysis by segmenting the data based on RSUs. This segmentation follows the principles of the Local Dynamic Map (LDM) framework [121], a conceptual data repository within cars and RSUs that stores information critical to road safety and traffic efficiency. The LDM includes data such as the number of nearby cars and RSUs, facilitating a more structured approach to analyzing message distributions. Additionally, we monitored the CBR at each RSU independently to inform the dataset split, as detailed in the following subsection.

Splitting the Data per Group of RSUs: To enhance the performance of the detection models, we analyzed the number of surrounding cars near each RSU, as defined in the LDM of the ETSI standard [121]. Our findings indicate that RSU 1 consistently maintains a lower car count (approximately 100 cars) in its LDM compared to RSUs 0 (167 cars), 2 (139 cars), and 3 (300 cars).

Table 6.7: Confusion matrix data of newly trained models

Model name	TN	FP	FN	TP	MCC
RF	891,979	2,510	48,584	71,905	73.71 %
Adaboost	892,120	2,369	51,907	68,582	71.87 %
HistGBC	892,440	2,049	51,340	69,149	72.39 %

Additionally, an independent study of the CBR at each RSU showed that RSU 1 experiences a lower CBR compared to RSUs 0, 2, and 3. These observations suggest that traffic patterns at RSU 1 differ significantly from those at the other RSUs. Given this discrepancy, we recognized that features added to the dataset, such as $ChannelLoad$ and $Message_Count$, could introduce bias in model training if car traffic patterns varied substantially. To mitigate this, we proposed splitting the dataset into two groups:

- RSU1 group: Data received exclusively by RSU 1.
- RSU023 group: Data received by RSUs 0, 2, and 3.

Following this segmentation, we retrained the three models separately on each subset and obtained the results presented in Table 6.8 for RSU1 and Table 6.9 for RSU023. From Table 6.8, it is evident that the models performed poorly on the RSU1 dataset, achieving an MCC of approximately 29%, despite being explicitly trained and validated on this subset. This reduced performance can be attributed to the low percentage of attackers within RSU 1's vicinity (only 5%). Consequently, the attack in this region remains hardly detectable, making it difficult for the models to distinguish malicious behavior from normal traffic even during training.

Table 6.8: Confusion matrix data of RSU1 models

Model name	TN	FP	FN	TP	MCC
RF	230,447	364	20,683	2,175	26.92 %
Adaboost	230,488	323	20,176	2,682	30.68 %
HistGBC	229,515	1,296	19,553	3,305	29.82 %

*Models in **bold** are the chosen ones

Examining Table 6.9, we observe a higher MCC score, indicating that the models performed better in detecting attack messages within the RSU023 dataset. Among the three models, the RF model demonstrated superior performance in identifying attacks, as reflected in its higher TP count. Conversely, the HistGBC model excelled in correctly classifying normal messages, achieving the highest TN count. These trade-offs suggest that each model captures distinct attack patterns, highlighting their complementary strengths. However, despite the dataset split, the number of FP did not significantly decrease across the models. The exception is the RF model,

where the total FP count across both splits (RSU1 and RSU023) was reduced ($364 + 1,485 = 1,849$) compared to its original count ($2,510$, Table 6.7). While this calculation is not entirely accurate and to further enhance detection performance, we conducted a final optimization step aimed at integrating the strengths of each model and improving overall classification accuracy.

Table 6.9: Confusion matrix data of RSU023 models

Model name	TN	FP	FN	TP	MCC
RF	669,011	1,485	28,611	70,026	81.49 %
Adaboost	668,423	2,073	33,962	64,675	77.51 %
HistGBC	669,179	1,317	29,746	68,891	80.86 %

*Models in **bold** are the chosen ones

Final Detection Model Optimization: To improve the final decision-making process of our models, which exhibit trade-offs between TN, FP, FN, and TP, we explored two possible ensemble learning solutions:

1. Stacking ensemble learning using a Random Forest as the final estimator.
2. A voting classifier that aggregates predictions from the trained models.

Random Forest was chosen as the final estimator for stacking because it previously demonstrated strong performance on the original dataset. However, the results of the stacking approach, MCC score of 81.6% on RSU023 data and 31% on RSU1 data, did not yield significant improvements compared to the individual models. Additionally, it introduced the overhead of training an extra model on the base estimators' predictions. Given these results, we opted for the voting classifier as our final optimization strategy. For each data split, we implemented a separate voting classifier based on the models trained on that respective split.

Before evaluating the performance of the voting classifier models, we needed to address the common coverage area shared by RSU 1 and the other RSUs (Figure 6.4). Since cars located within this overlapping region broadcast CAM messages to both RSU 1 and the other RSUs, some messages appear in both splits. This duplication affects the confusion matrices and inflates comparisons. Specifically, $7,824$ messages were repeated, consisting of $6,818$ normal messages and $1,006$ attack messages. To ensure a fair evaluation, we assigned all messages originating from the common coverage area to be evaluated by the models trained on the RSU023 split. This decision was based on the superior attack detection performance of the RSU023-trained models.

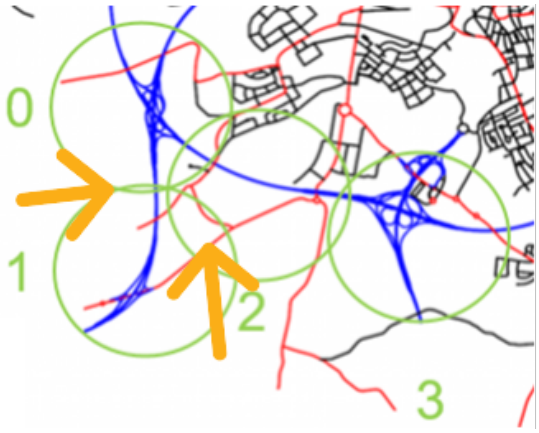


Figure 6.4: Common areas marked with orange.

Based on the results obtained for each of the three models on both splits (RSU1 and RSU023), we selected the following models for the final voting classifier:

- For RSU1: RF and Adaboost were chosen, as HistGBC contributed to a relatively high number of FPs.
- For RSU023: RF and HistGBC were chosen, since Adaboost did not perform as well as the other two models in detecting the attack class.

These selections were validated using the validation dataset before being applied to the test dataset to ensure their effectiveness.

Voting classifiers can be assigned weights to give more influence to stronger models. To determine whether weighting was necessary, we analyzed the pair-wise Q-statistics (Q-stat) and the Double Fault (DF) measure between the selected models, as defined in [122]. Q-stat ($Q_{i,k}$) measures the similarity between two classifiers, where values range from -1 to 1:

- 0 indicates statistical independence.
- Positive values mean both classifiers tend to classify the same instances correctly.

Double Fault ($DF_{i,k}$) represents the proportion of cases misclassified by both classifiers. Lower values indicate a lower likelihood of simultaneous misclassification. Considering a relationship table (Table 6.10) between a pair of classifiers D_i and D_k, we can compute $Q_{i,k}$ as in eq. (6.2) and $DF_{i,k}$ as in eq. (6.3). The computed values on the validation dataset were:

- Q-stat = 1 for both splits, indicating that the chosen models capture the same attack patterns.

- DF for RSU023 (RF & HistGBC) = 0.023
- DF for RSU1 (RF & Adaboost) = 0.049

Since the DF values are low, this suggests that the classifiers do not frequently misclassify the same instances. As a result, equal-weight voting is sufficient, and there is no need to assign additional weights to the models.

Table 6.10: Relationship table between a pair of classifiers D_i and D_k

	D_k correct(1)	D_k wrong(0)
D_i correct(1)	N^{11}	N^{10}
D_i wrong(0)	N^{01}	N^{00}

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \tag{6.2}$$

$$\text{DF}_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{6.3}$$

Before analyzing the confusion matrices of the voting classifiers, we first evaluated their ROC curves and the corresponding AUC to validate their overall performance. The results show that for the RSU023 split, which includes RF and HistGBC, the AUC values are:

- Soft Voting Classifier: 0.962 (Figure 6.5a)
- Hard Voting Classifier: 0.96 (Figure 6.5b)

These high and consistent AUC values confirm the reliability of the models in distinguishing between normal and attack messages.

It is important to note that for hard voting, the ROC curve was customized using the predictions from the base models. Unlike soft voting, hard voting does not support prediction probabilities, as defined in subsection 6.4.2. Therefore, to generate the ROC curve, we determined the majority vote among the base models and assigned the corresponding majority probability for the curve plotting.

On the other hand, the soft voting classifier on RSU1, which includes RF and Adaboost (Figure 6.5c), achieved an AUC of 0.676. This result aligns with expectations, as the individual base models (Table 6.8) did not perform well on their own. However, the AUC value still indicates that the classifier performs better than a random model. Meanwhile, the hard voting classifier

on RSU1 recorded an AUC of 0.603 (Figure 6.5d). Notably, its ROC curve intersects the random classifier line, suggesting a lower performance and a sensitivity to the probability threshold assigned to the classifier.

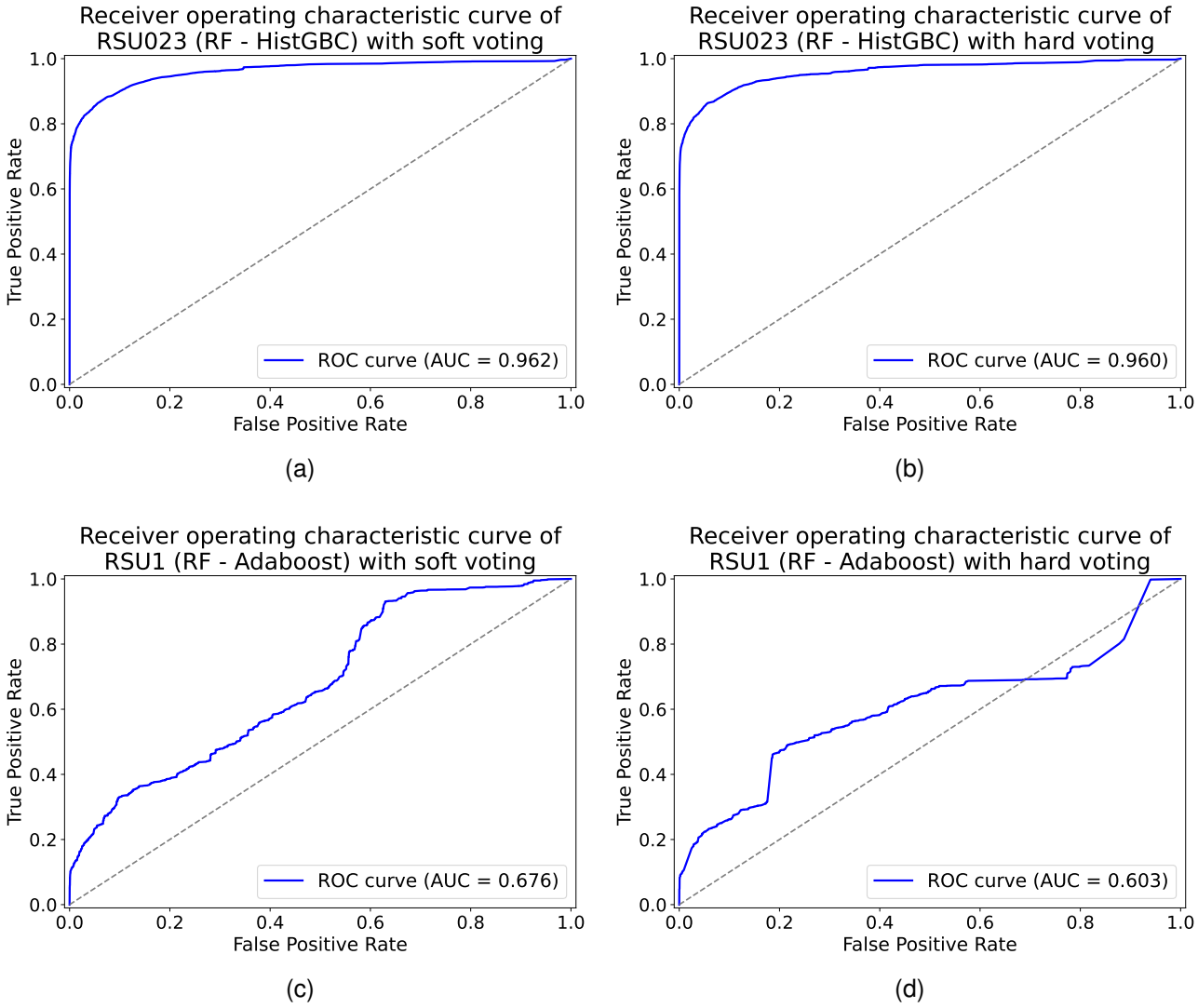


Figure 6.5: ROC curve and the AUC of the voting classifiers.

Our final evaluation focuses on analyzing the soft and hard voting confusion matrices to determine which approach is more suitable for our case.

Table 6.11 presents the soft and hard voting results for the models predicting the RSU1 and RSU023 data splits. Observing the "*RSU1*" and "*RSU023*" rows, we see that hard voting results in fewer TPs but also fewer FPs compared to soft voting. This confirms that the models disagree on some instances (aligning with the low $\text{DF}_{i,k}$), yet they still capture the same attack pattern globally ($Q_{i,k} = 1$). The difference arises because, in hard voting, when a tie occurs between

two models in a binary classification task, the classifier defaults to the class with the smaller label (i.e., the normal class, label 0). In contrast, soft voting averages the predicted probabilities of the models, leading to more calibrated voting results, in comparison to the selected models highlighted in bold in Tables 6.8 and 6.9.

To compare the overall impact of hard and soft voting, we summed the confusion matrices across both data splits and presented them in the $Total$ row of Table 6.11. Additionally, we computed the difference by subtracting the soft voting results from the hard voting results, which is shown in the last row. The key takeaway is that choosing hard voting over soft voting reduces the number of detected attack messages but also decreases the number of normal messages mistakenly classified as attacks. This means that hard voting mitigates the risk of falsely accusing normal messages, thereby lowering the escalated effect of the Sybil attack caused by excessive false positives in the IDS.

Table 6.11: Confusion matrix data of voting classifier models

Data region	Voting type	TN	FP	FN	TP	MCC
RSU1(RF,Adaboost)	hard	223,845	148	20,339	1,513	23.8%
	soft	223,774	219	19,929	1,923	26.6%
RSU023(RF,HistGBC)	hard	669,457	1,039	30,547	68,090	80.5%
	soft	669,192	1,304	29,429	69,208	81.1%
Total	hard	893,302	1,187	50,886	69,603	73.2%
	soft	892,966	1,523	49,358	71,131	73.9%
Difference of total	hard-soft	336	-336	1,528	-1,528	

To ensure a fair comparison of the MCC between hard and soft voting, which achieved 73.2% and 73.9%, respectively (as shown in the $Total$ row of Table 6.11), we compared these results with the newly trained models in Table 6.7. Specifically, we subtracted the confusion matrix data of each base model from the confusion matrix data of the hard and soft voting classifiers, and we summarized these comparisons in Tables 6.12 and 6.13. From these comparisons, we observe that RF performs best on the attack class, as it alone enables the detection of 774 more attack messages than soft voting and $2,302$ more attack messages than hard voting on the studied dataset. However, this improvement comes at the cost of an increase in FPs — with RF producing approximately $1,000$ additional FPs compared to soft voting and $1,300$ additional FPs compared to hard voting. On the other hand, when comparing Adaboost and HistGBC, the soft and hard voting classifiers outperform them in TPs or FPs. This is evident from the positive differences in TP counts (indicating improved attack detection) and negative differences in FP counts (indicating fewer misclassified normal messages) after the subtraction.

We conclude that both voting classifier methods effectively reduce the number of False Positives across all models, as indicated by the negative values in the FP column. This leads to the

Table 6.12: Confusion matrix comparison between hard voting and newly trained models

Compared model	TN	FP	FN	TP
RF	1,323	-1,323	2,302	-2,302
Adaboost	1,182	-1,182	-1,021	1,021
HistGBC	862	-862	-454	454

Table 6.13: Confusion matrix comparison between soft voting and newly trained models

Compared model	TN	FP	FN	TP
RF	987	-987	774	-774
Adaboost	846	-846	-2,549	2,549
HistGBC	526	-526	-1,982	1,982

key question: Which method should be chosen? From a security perspective, an IDS should ideally block the maximum number of attack messages to minimize potential threats. Under this criterion, RF alone performs best, as it achieves the highest detection rate for attack messages. However, given that we are dealing with a Sybil DDoS attack, which is designed to cause significant disruption while remaining partially undetected, there is an inherent risk of normal messages being misclassified as attack messages. To mitigate this false accusation effect, hard voting emerges as the preferred choice, as it achieves the lowest FP rate while still maintaining strong attack detection capabilities. The trade-off of accepting a lower true positive rate to reduce false alarms is considered acceptable here within the connected car domain, as undetected malicious cars are likely to be detected through their subsequent messages. Soft voting, on the other hand, represents a tradeoff between hard voting and RF alone, balancing attack detection and FP reduction.

6.7 Conclusion

This chapter introduced a DDoS Sybil threat model that exploits a vulnerability in connected cars, making the attack more stealthy. The model is based on the existence of a pool of PseudoIDs within OBUs, designed to preserve car privacy during communication. However, we demonstrated that simply changing PseudoIDs allows the attack to evade previously developed DDoS detection mechanisms. To address this, we retrained our detection models and proposed an optimized data-handling approach. By leveraging an ensemble hard voting model, we achieved a balance between effective attack detection and reduced false alarms.

Research is a continuously evolving domain with numerous avenues for investigation and exploration. As the connected car environment continues to develop, our research can be extended

in several directions to enhance security and improve performance. The thesis conclusion and future research perspectives are discussed in the final chapter.

Chapter 7

Conclusion and Perspectives

7.1 Conclusion

In this thesis, we focused on detecting cyberattacks within the C-ITS domain using AI-based approaches, specifically ensemble learning methods. We used the DDoS attack as the foundation of our attack model and investigated various forms of this attack, including the integration of a Sybil concept.

We began with an evaluation study to examine the impact of DDoS attacks conducted using CAM messages. The presented attack model, with a transmission rate five times the normal rate, demonstrated a subtle yet significant impact. This was shown through the disruption of communications, where legitimate vehicles reduced their CAM transmission rate, and the accumulation of invalid data, with up to 92% of the collected data potentially containing meaningless or null values.

As a reinforcement to the existing PKI infrastructure, we proposed a complementary solution in the form of a distributed IDS deployed across different geographical regions. This IDS is intended for deployment on the MEC node, leveraging its computational capabilities for training and validating machine learning-based models for attack detection. We also detailed the internal architecture of this IDS and positioned our contributions within the anomaly-based module.

The development of three machine learning models - Random Forest, AdaBoost, and HistGBC - was based on the creation of a representative DDoS dataset. To achieve this, we simulated a DDoS attack in Luxembourg City and collected CAM data from four strategically located RSUs. Our results yielded a high MCC score of 98%, indicating strong detection performance.

Furthermore, we evaluated our trained models against various morphing DDoS attacks, demonstrating their effectiveness by maintaining a high MCC score of 99%. These attacks included scenarios where an attacker transmitted with higher power and another sent fully encapsulated messages.

We also extended our evaluation to include a Sybil-based DDoS attack model, exploiting PseudoID changes to evade detection models. We demonstrated the generalization of our models with up to four PseudoID changes. However, with five PseudoID change, the detection performance of our models was compromised. To address this, we proposed an optimized data-handling strategy, improving detection performance to a 73% MCC score. This identified limitation of our models, coupled with their dependence on the PseudoID change strategy, opens directions for future research aimed at enhancing detection accuracy and protecting the connected cars domain.

7.2 Future Perspectives

Research is a constantly evolving field; as a short-term perspective, this work can be extended by proposing alternative solutions such as computational challenges for changing PseudoIDs or a novel detection method based on the LDM. A long-term extension involves developing a large-scale collaborative misbehavior detection system.

7.2.1 Short-term Perspectives

While our data-optimized handling strategy yielded improved detection results, we further propose alternative approaches to address the PseudoID changes that compromised our models.

PseudoID Change Challenge: Standards for PseudoID changes are still under development. Our presented Sybil model demonstrated that a strategically executed change in PseudoID allows attackers to perform Sybil-DDoS attacks. An alternative method to counter such attacks is to implement a computational challenge that vehicles must solve to change their PseudoID. This challenge should preserve vehicle privacy while ensuring robust performance and preventing abuse.

LDM-Based Anomaly Detection: Exploring the LDM update mechanism could provide insights into the conditions under which an LDM in a vehicle or RSU is updated. This understanding could potentially reveal methods to detect the sudden appearance of new PseudoIDs representing different vehicles in an entity's vicinity. This insight can be translated into extracting features related to the dynamic change of PseudoIDs.

Other alternative methods, such as a trust module or a reputation score, could also enhance detection accuracy.

Trust Module for Voting: Instead of the voting ensemble used in our work, a further enhancement could involve implementing a trust module among the participating machine learning models. This module would replace the majority voting of the hard voting approach or the probability averaging in the soft voting approach. The basis for establishing trust in a particular model could be a set of performance metrics dynamically tested as detection progresses. This approach necessitates more labeled data to assess its effectiveness, thus requiring the generation of additional datasets. One example of such an approach is subjective logic, which can handle uncertainties, resolve conflicts, and update trust weights.

Reputation-Based Detection: Given that vehicles are represented by their PseudoIDs, a reputation score could be assigned to each PseudoID. This score could be shared among IDSs through the shared reputation database, as illustrated in the IDS's internal workflow (Figure 5.2). A further extension of this reputation-based approach would involve reporting the score to a misbehavior authority for comprehensive assessment and appropriate actions, such as certificate revocation.

7.2.2 Long-term Perspectives

The following perspectives present a broader view of our work and its potential implementation across countries as a large-scale distributed collaborative attack detection system. This can be visualized at the road operator and core network level of Figure 5.1, which depicts the IDS architecture for connected cars. First, we outline its architecture, followed by its potential mode of operation.

Architecture of Misbehavior Collaborative Detection: The architecture consists of hierarchical distributed misbehavior authorities, each covering different regions within a country. While these authorities are independent of road operators, they would have protocols for communication with local misbehavior authorities managed by road operators. An alternative model could be centralized, with a primary misbehavior authority overseeing all others. At the international level, misbehavior authorities would be distributed, with a central authority in each country facilitating communication with neighboring countries for cross-border information sharing.

Mode of Operation of Misbehavior Collaborative Detection: Within a country, local communication protocols and local digital certificates could be employed. However, for cross-border communication, the applied protocols should ensure interoperability between different entities. Blockchain technology could be leveraged here to guarantee the immutability of recorded actions, such as reported vehicles and revoked certificates, shared across countries.

Chapter 8

List of Publications

Published

- Ali El Attar, Ayoub Wehby, Fadlallah Chbib, Hassane Aissaoui Mehrez, Ahmad Fadlallah, Joel Hachem, and Rida Khatoun. "Analysis of machine learning algorithms for DDoS attack detection in connected cars environment". In: *2023 Eighth International Conference on Mobile and Secure Services (MobiSecServ)*. Miami Beach, FL, USA, 2023, pp. 1-7, doi: https://doi.org/10.1109/MobiSecServ58080.2023.10329152.
- Ayoub Wehby, Sherali Zeadally, Rida Khatoun, Mohammed Bouchouia, and Ahmad Fadlallah. “How Does Distributed Denial of Service Affect the Connected Cars Environment?”. In: *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*. Vallette, Malta, 2024, pp. 164-170, doi: https://doi.org/10.1109/CoDIT62066.2024.10708325.
- Ayoub Wehby, Rida Khatoun and Ahmad Fadlallah, "Shielding the Connected Cars: A Dataset-Powered Defense Against DDoS".In: *2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. Leeds, United Kingdom, 2024, pp. 1-7, doi: https://doi.org/10.1109/WINCOM62286.2024.10656993.

In Progress

- Ayoub Wehby, Sherali Zeadally, Rida Khatoun, and Ahmad Fadlallah. “Toward Adaptive Intrusion Detection System for Mitigating Morphing Distributed Denial of Service Attacks in Connected Cars”. (Submitted)
- Ayoub Wehby, Rida Khatoun, and Ahmad Fadlallah. “Generalization and Limitations of DDoS Detection Models in Sybil-Based Attacks on Connected Cars”. (Submitted)

Appendix A

CAM field description

Table A.1: Some dataset fields description

Field name	Type of data	Description	Range	Units
GenerationDeltaTime	int64	Time corresponding to the time of the reference position in the CAM, considered as time of the CAM generation [EN 302 637-2 V1.4.1]	0 - 65528	-
Longitude	int64	Longitudinal position of the vehicle [TS 102 894-2 - V1.3.1]	60781120 - 61418210	0,1 microdegree
Latitude	int64	Latitudinal position of the vehicle [TS 102 894-2 - V1.3.1]	495629380 - 495950540	0,1 microdegree
Heading	int64	Orientation of the direction of travel with regards to the WGS84 north [TS 102 894-2 - V1.3.1]	0 - 3600	0,1 degree
Speed	int64	Speed of the vehicle [TS 102 894-2 - V1.3.1]	0 - 4565	0,01 m/s
Flag	boolean	Indicator if it is a message from an attacker	True - False	-

Appendix B

DDoS general detection related works

Table B.1: Comparison of DDoS detection approaches.

Reference	Year	Studied messages	Deployment point	Mechanism	Metrics evaluated	Software used	Results	Weaknesses/Limitations
[123]	2020	UDP(MAC Layer, ARP protocol)	Host-based (local reputation list)	Statistical (based on the calculation of the median)	Detection Rate, False Positive Rate, False Negative Rate	SUMO, NS-3	DR(88% to 100%) FPR(3%) FNR(0%)	No global sharing of reputation list
[124]	2018	No specific data type	Centralized (RSU-based)	Statistical non-parametric (based on learning a baseline of the number of received messages)	average detection delay, false alarm probability, Receiver Operating Characteristic	OMNeT++, SUMO, Veins, MATLAB	low average detection delay versus false alarm probability, ROC graphically presented	Blocking attack locations leads to blocking normal messages also.
[125]	2019	No specific data type (Packets)	Host-based (building a trusted hierarchy for safe routing)	Trust (based on frequency value statistics, trust hypothesis statistics, residual energy, trust policy, data factor)	Average Latency, Packet Delivery Ratio, Detection Rate, Energy Consumption	NS-2.28	AL(30s) PDR(86%) DR(95.8%) EC(43J)	Studied only routed packets (no single hop messages)
[126]	2019	UDP	Central (reCAPTCHA controller)	Statistical (stochastic model)	Detection Rate, Packet Delivery Ratio, Average Latency, Energy Consumption	NS-2.28	DR(94.7%) PDR(85%)AL(33s) EC(47J)	Studies only routed packets (no single hop messages)
[127]	2021	TCP, UDP	Central (collection of real-time data and sending it to IDS)	Statistical (Random Forest is the trained classifier)	True Positive Rate, False Positive Rate, Receiver Operating Characteristic, Network Throughput	Mininet-Wifi	high detection accuracy low false positive rate throughput not affected (lightweight)	No specific safety VANET related messages.
[58]	2022	ARP Packets (UDP)	Host-based (uses MDASTI, the 1st reference)	Statistical (based on the calculation of the median)	Detection Rate, False Negative Rate, False Positive Rate	OMNeT++, SUMO, Veins	DR(>95%) FNR(near 0) FPR(no graph)	No global sharing of reputation list
[128]	2019	IP, TCP, UDP, etc (NSL-KDD and UNSW-NB15 Datasets)	Host-based (distributed collection and detection module)	Statistical (random forest based algorithm)	Accuracy, Recall, False Alarm Rate, F1-Measure	No software specific to VANET simulation	Acc(99.95%, 98.75%) Recall(99%) FAR(0.05%, 1.08%) F1-M(99%)	No specific safety VANET related messages.
[129]	2018	No specific data type (Packets)	Host-based (vehicle reads the network trace..)	Statistical (Multivariant Stream Analysis of four parameters)	Throughput ratio, Packet delay ratio, Packet delivery ratio, Packet drop ratio, Detection accuracy, Detection time	NS-2.34	Outperforming all compared baseline approaches (shown in the graphs)	Studied only routed packets (no single hop messages)

References

[1] *Car-2-Car Communication Consortium*. URL: https://www.car-2-car.org/ (visited on 09/05/2025).

[2] *AUTomotive Open System ARchitecture*. URL: https://www.autosar.org/ (visited on 09/05/2025).

[3] *Caltrans*. URL: https://dot.ca.gov/programs/traffic-operations/cav (visited on 09/05/2025).

[4] *SCOOP@F*. URL: https://www.scoop.developpement-durable.gouv.fr/en/ (visited on 09/05/2025).

[5] *C-ROADS FRANCE and Infrastructure Digitale de Demain*. URL: https://www.c-roads.eu/pilots/core-members/france.html (visited on 09/05/2025).

[6] *Fortan Traffic System*. (visited on: 07.03.2024). URL: https://www.fortrantraffic.com/.

[7] *Renault Group*. (visited on: 07.03.2024). URL: https://www.renaultgroup.com/en/news-on-air/news/renault-preparing-infrastructures-for-the-connected-car-of-the-future-with-scoop/.

[8] *NSC Announces 2025 Road to Zero Community Traffic Safety Grant Recipients*. URL: https://www.nsc.org/newsroom/nsc-announces-2025-road-to-zero-grants (visited on 09/05/2025).

[9] Joao P. Trovao. "The Vehicle Industry Is Moving Fast [Automotive Electronics]". In: *IEEE Vehicular Technology Magazine* 17.1 (2022), pp. 98–107. DOI: 10.1109/MVT.2021.3130399.

[10] Sherali Zeadally, Muhammad Awais Javed, and Elyes Ben Hamida. "Vehicular Communications for ITS: Standardization and Challenges". In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 11–17. DOI: 10.1109/MCOMSTD.001.1900044.

[11] 5GAA. *System Enablers and Best Practices for Next-Gen Cooperative Use Cases*. Tech. rep. Available online at: https://5gaa.org/system-enablers-and-best-practices-for-next-gen-cooperative-use-cases-5gaa-report/. 5G Automotive Association, Nov. 2022.

[12] Charlie Miller. "Lessons learned from hacking a car". In: *IEEE Design & Test* 36.6 (2019), pp. 7–9. DOI: 10.1109/MDAT.2018.2863106.

[13] *Stopping robotaxis with traffic cones*. URL: https://www.disconnect.blog/p/stopping-robotaxis-with-traffic-cones (visited on 09/05/2025).

[14] *Tesla Gear Gets Hacked Multiple Times in Pwn2Own Contests*. URL: https://www.darkreading.com/vulnerabilities-threats/tesla-gear-hacked-multiple-times-pwn2own-contests (visited on 09/05/2025).

[15] Numaan Huq et al. *Cybersecurity for Connected Cars Exploring Risks in 5G, Cloud, and Other Connected Technologies*. Tech. rep. Available online at: https://documents.trendmicro.com/assets/white_papers/wp-cybersecurity-for-connected-cars-exploring-risks-in-5g-cloud-and-other-connected-technologies.pdf. Trend Micro Research, 2021.

[16] *Threat Modeling with Microsoft DREAD*. Aug. 2022. URL: https://satoricyber.com/glossary/threat-modeling-with-microsoft-dread/.

[17] Ayoub Wehby et al. "How Does Distributed Denial of Service Affect the Connected Cars Environment?" In: *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*. 2024, pp. 164–170. DOI: 10.1109/CoDIT62066.2024.10708325.

[18] Ayoub Wehby, Rida Khatoun, and Ahmad Fadlallah. "Shielding the Connected Cars: A Dataset-Powered Defense Against DDoS". In: *2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM)*. 2024, pp. 1–7. DOI: 10.1109/WINCOM62286.2024.10656993.

[19] Pablo Alvarez Lopez et al. "Microscopic Traffic Simulation using SUMO". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938.

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

[21] John R Douceur. "The sybil attack". In: *International workshop on peer-to-peer systems*. Springer. 2002, pp. 251–260.

[22] Baofeng Ji et al. “Survey on the Internet of Vehicles: Network Architectures and Applications”. In: *IEEE Communications Standards Magazine* 4.1 (2020), pp. 34–41. DOI: 10.1109/MCOMSTD.001.1900053.

[23] Mohammad Syfullah and Joanne Mun-Yee Lim. “Data broadcasting on Cloud-VANET for IEEE 802.11p and LTE hybrid VANET architectures”. In: *2017 3rd International Conference on Computational Intelligence Communication Technology (CICT)*. 2017, pp. 1–6. DOI: 10.1109/CIACT.2017.7977321.

[24] Ammara Anjum Khan, Mehran Abolhasan, and Wei Ni. “5G Next Generation VANETs Using SDN and Fog Computing Framework”. In: *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 2018, pp. 1–6. DOI: 10.1109/CCNC.2018.8319192.

[25] Kai Liu et al. “A Hierarchical Architecture for the Future Internet of Vehicles”. In: *IEEE Communications Magazine* 57.7 (2019), pp. 41–47. DOI: 10.1109/MCOM.2019.1800772.

[26] Chaker Abdelaziz Kerrache et al. “TACASHI: Trust-Aware Communication Architecture for Social Internet of Vehicles”. In: *IEEE Internet of Things Journal* 6.4 (2019), pp. 5870–5877. DOI: 10.1109/JIOT.2018.2880332.

[27] G. Leen and D. Heffernan. “Expanding automotive electronic systems”. In: *Computer* 35.1 (2002), pp. 88–93. DOI: 10.1109/2.976923.

[28] Peter Hank et al. “Automotive Ethernet: In-vehicle networking and smart mobility”. In: *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2013, pp. 1735–1739. DOI: 10.7873/DATE.2013.349.

[29] Qorvo. *V2X in the Connected Car of the Future*. 2018. URL: https://www.qorvo.com/design-hub/blog/v2x-in-the-connected-car-of-the-future (visited on 09/05/2025).

[30] ETSI. *EN 302 637-2 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service*. Standard. Available online at: https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf. European Telecommunications Standards Institute (ETSI), 2019.

[31] ETSI. *TS 102 687 - Intelligent Transport Systems (ITS); Decentralized Congestion Control Mechanisms for Intelligent Transport Systems operating in the 5 GHz range; Access layer part*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/102600_102699/102687/01.02.01_60/ts_102687v010201p.pdf. European Telecommunications Standards Institute (ETSI), 2018.

[32] ETSI. *EN 302 637-3 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service*. Standard. Available online at: https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.03.01_60/en_30263703v010301p.pdf. European Telecommunications Standards Institute (ETSI), 2019.

[33] ETSI. *TS 103 324 - Intelligent Transport System (ITS); Vehicular Communications; Basic Set of Applications; Collective Perception Service; Release 2*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/103300_103399/103324/02.01.01_60/ts_103324v020101p.pdf. European Telecommunications Standards Institute (ETSI), 2023.

[34] ETSI. *TR 103 439 - Intelligent Transport Systems (ITS); Multi-Channel Operation study; Release 2*. Standard. Available online at: https://www.etsi.org/deliver/etsi_tr/103400_103499/103439/02.01.01_60/tr_103439v020101p.pdf. European Telecommunications Standards Institute (ETSI), 2021.

[35] Andreas Festag. "Standards für Fahrzeugkommunikation–von IEEE 802.11 p zu 5G". In: *e & i Elektrotechnik und Informationstechnik* 132 (2015), pp. 409–416.

[36] ETSI. *TS 102 940 - Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management; Release 2*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/02.01.01_60/ts_102940v020101p.pdf. European Telecommunications Standards Institute (ETSI), 2021.

[37] ETSI. *TS 102 940 - Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/01.03.01_60/ts_102940v010301p.pdf. European Telecommunications Standards Institute (ETSI), 2018.

[38] ETSI. *TS 103 759 - Intelligent Transport Systems (ITS); Security; Misbehaviour Reporting service; Release 2*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/103700_103799/103759/02.01.01_60/ts_103759v020101p.pdf. European Telecommunications Standards Institute (ETSI), 2023.

[39] "IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Certificate Management Interfaces for End Entities". In: *IEEE Std 1609.2.1-2020* (2020), pp. 1–287. DOI: 10.1109/IEEESTD.2020.9311462.

[40] Zeinab El-Rewini et al. "Cybersecurity challenges in vehicular communications". In: *Vehicular Communications* 23 (2020), p. 100214.

[41] Batuhan Gul and Fatih Ertam. "In-vehicle communication cyber security: A comprehensive review of challenges and solutions". In: *Vehicular Communications* (2024), p. 100846.

[42] Yousik Lee and Samuel Woo. “CAN Signal Extinction-based DoS Attack on In-Vehicle Network”. In: *Security and Communication Networks* 2022.1 (2022), p. 9569703.

[43] Elpidoforos Arapantonis. “Data tampering in Vehicle CAN Bus networks”. MA thesis. Lulea University of Technology, 2019.

[44] Abdulrahman Abu Elkhail et al. “Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses”. In: *IEEE Access* 9 (2021), pp. 162401–162437.

[45] Zizhi Jin et al. “PLA-LiDAR: Physical Laser Attacks against LiDAR-based 3D Object Detection in Autonomous Vehicle”. In: *2023 IEEE Symposium on Security and Privacy (SP)*. 2023, pp. 1822–1839. DOI: 10.1109/SP46215.2023.10179458.

[46] Yi Zhu et al. “Tilemask: A passive-reflection-based attack against mmwave radar object detection in autonomous driving”. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 2023, pp. 1317–1331.

[47] Amir Alipour-Fanid, Monireh Dabaghchian, and Kai Zeng. “Impact of Jamming Attacks on Vehicular Cooperative Adaptive Cruise Control Systems”. In: *IEEE Transactions on Vehicular Technology* 69.11 (2020), pp. 12679–12693. DOI: 10.1109/TVT.2020.3030251.

[48] P Remya Krishnan and P Arun Raj Kumar. “Detection and mitigation of smart blackhole and gray hole attacks in VANET using dynamic time warping”. In: *Wireless Personal Communications* 124.1 (2022), pp. 931–966.

[49] Felipe Carvalho Santos et al. *Zonal Architecture: A Way of Using of the CAN Protocol*. Tech. rep. SAE Technical Paper, 2024.

[50] Jin-Hui Piao, Yu-Jing Wu, and Yi-Nan Xu. “A security framework for in-vehicle FlexRay bus network”. In: *International Journal of Modeling and Optimization* 12.3 (2022).

[51] Dennis K Nilsson, Lei Sun, and Tatsuo Nakajima. “A framework for self-verification of firmware updates over the air in vehicle ECUs”. In: *2008 IEEE Globecom Workshops*. IEEE. 2008, pp. 1–5.

[52] Irmina Durlik et al. “Cybersecurity in Autonomous Vehicles—Are We Ready for the Challenge?” In: *Electronics* 13.13 (2024), p. 2654.

[53] AUTOSAR. *Requirements on Intrusion Detection System*. Tech. rep. Online: https://www.autosar.org/fileadmin/standards/R22-11/FO/AUTOSAR_RS_IntrusionDetectionSystem.pdf. AUTOSAR, 2022.

[54] Farah-Emma Braiteh, Francesca Bassi, and Rida Khatoun. “Platooning in Connected Vehicles: A Review of Current Solutions, Standardization Activities, Cybersecurity, and Research Opportunities”. In: *IEEE Transactions on Intelligent Vehicles* (2024).

[55] Abdul Jabbar Siddiqui and Azzedine Boukerche. “On the Impact of DDoS Attacks on Software-Defined Internet-of-Vehicles Control Plane”. In: *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. 2018, pp. 1284–1289. DOI: 10.1109/IWCMC.2018.8450433.

[56] Hamsa Mahmoud Hosny, Nayera Sadek, and Onsy A. Abdel-Alim. “Security of 5G-IOV Networks:DDOS Case Study”. In: *2022 International Telecommunications Conference (ITC-Egypt)*. 2022, pp. 1–5. DOI: 10.1109/ITC-Egypt55520.2022.9855708.

[57] Shahida Malik, Praveen Bandi, and Weiqing Sun. “An Experimental Study of Denial of Service Attack Against Platoon of Smart Vehicles”. In: *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*. 2021, pp. 23–30. DOI: 10.1109/MetroCAD51599.2021.00013.

[58] Hamideh Baharlouei, Adetokunbo Makanju, and Nur Zincir-Heywood. “Exploring Realistic VANET Simulations for Anomaly Detection of DDoS Attacks”. In: *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*. 2022, pp. 1–7. DOI: 10.1109/VTC2022-Spring54318.2022.9860624.

[59] Karanpreet Singh, Paramvir Singh, and Krishan Kumar. “Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges”. In: *Computers & security* 65 (2017), pp. 344–372.

[60] Jorge Pereira et al. “Assessing the reliability of fog computing for smart mobility applications in VANETs”. In: *Future Generation Computer Systems* 94 (2019), pp. 317–332.

[61] *2024: Updated List of C-V2X Devices*. (visited on: 04.12.2024). URL: https://5gaa.org/2024-updated-list-of-c-v2x-devices/.

[62] *RSU Specification of YUNEX*. (visited on: 04.12.2024). URL: https://www.westernsystems-inc.com/product/yunex-traffic-rsu-connected-vehicle/.

[63] *RSU Specification of Unex Technology Corp*. (visited on: 04.12.2024). URL: https://unex.com.tw/en/document/rsu-352_information-sheet/.

[64] *RSU Specification of Commsignia*. (visited on: 04.12.2024). URL: https://www.commsignia.com/products/rsu/.

[65] *RSU Specification of NeoGLS*. (visited on: 04.12.2024). URL: https://www.neogls.com/produits/ubr-et-uev/.

[66] *RSU Specification of V2ROADS*. (visited on: 04.12.2024). URL: https://v2roads.com/products/hardware/rsu/.

[67] *RSU Specification of Q-Free*. (visited on: 04.12.2024). URL: https://www.q-free.com/product/c-its-road-side-unit/.

[68] Mevlut Turker Garip et al. “Congestion attacks to autonomous cars using vehicular botnets”. In: *NDSS Workshop on Security of Emerging Networking Technologies (SENT), San Diego, CA*. 2015. DOI: 10.14722/sent.2015.23001.

[69] Wedad Ahmed Al-Dhuraibi and Mourad Elhadef. “Securing Vehicular Ad-Hoc Networks: A DDoS Case Study”. In: *2021 2nd International Conference on Computation, Automation and Knowledge Management (ICCAKM)*. 2021, pp. 112–117. DOI: 10.1109/ICCAKM50778.2021.9357733.

[70] ETSI. *EN 302 571 - Intelligent Transport Systems (ITS); Radiocommunications equipment operating in the 5 855 MHz to 5 925 MHz frequency band; Harmonised Standard covering the essential requirements of article 3.2 of Directive 2014/53/EU*. Standard. Available online at: https://www.etsi.org/deliver/etsi_en/302500_302599/302571/02.01.01_60/en_302571v020101p.pdf. European Telecommunications Standards Institute (ETSI), 2017.

[71] *OMNeT++*. URL: https://omnetpp.org (visited on 09/05/2025).

[72] Raphael Riebl et al. “Artery: Extending Veins for VANET applications”. In: *2015 International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. 2015, pp. 450–456. DOI: 10.1109/MTITS.2015.7223293.

[73] *A1 Traffic Statistics*. URL: https://routes.fandom.com/wiki/Autoroute_fran%C3%A7aise_A1_(Trafic) (visited on 09/05/2025).

[74] ETSI. *TS 102 792 - Intelligent Transport Systems (ITS); Mitigation techniques to avoid interference between European CEN Dedicated Short Range Communication (CEN DSRC) equipment and Intelligent Transport Systems (ITS) operating in the 5 GHz frequency range*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/102700_102799/102792/01.02.01_60/ts_102792v010201p.pdf. European Telecommunications Standards Institute (ETSI), 2015.

[75] ETSI. *TS 102 724 - Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems operating in the 5 GHz frequency band*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/102700_102799/102724/01.01.01_60/ts_102724v010101p.pdf. European Telecommunications Standards Institute (ETSI), 2012.

[76] Raphael Riebl et al. “Improved security performance for vanet simulations”. In: *IFAC-PapersOnLine* 49.30 (2016), pp. 233–238.

[77] Naam Alkadiri and Muhammad Ilyas. “Machine Learning-Based Architecture for DDoS Detection in VANETs System”. In: *2022 International Conference on Artificial Intelligence of Things (ICAIoT)*. IEEE. 2022, pp. 1–7.

[78] 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 (2011), pp. 3–15. DOI: 10.1109/TMC.2010.133.

[79] Tianqi Chen and Carlos Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[80] Hong Zhong et al. “Dataset for Evaluation of DDoS Attacks Detection in Vehicular Ad-Hoc Networks”. In: *International Conference on Wireless Algorithms, Systems, and Applications*. Springer. 2022, pp. 249–260.

[81] Ali EL Attar et al. “Analysis of Machine Learning Algorithms for DDoS Attack Detection in Connected Cars Environment”. In: *2023 Eighth International Conference On Mobile And Secure Services (MobiSecServ)*. IEEE. 2023, pp. 1–7.

[82] Joseph Kamel et al. “Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets”. In: *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE. 2020, pp. 1–6.

[83] ETSI. *TR 103 415 - Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management*. Standard. Available online at: https://www.etsi.org/deliver/etsi_tr/103400_103499/103415/01.01.01_60/tr_103415v010101p.pdf. European Telecommunications Standards Institute (ETSI), 2018.

[84] Thomas N Rincy and Roopam Gupta. “Ensemble learning techniques and its efficiency in machine learning: A survey”. In: *2nd international conference on data, engineering and applications (IDEA)*. IEEE. 2020, pp. 1–6.

[85] Tommaso Zoppi et al. “Meta-learning to improve unsupervised intrusion detection in cyber-physical systems”. In: *ACM Transactions on Cyber-Physical Systems (TCPS)* 5.4 (2021), pp. 1–27.

[86] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[87] Leo Breiman. “Random forests”. In: *Machine learning* 45 (2001), pp. 5–32.

[88] Yoav Freund and Robert E Schapire. “A decision-theoretic generalization of on-line learning and an application to boosting”. In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139.

[89] Guolin Ke et al. “Lightgbm: A highly efficient gradient boosting decision tree”. In: *Advances in neural information processing systems* 30 (2017).

[90] Ayoub Wehby. *Connected-Cars-DDoS-Dataset*. https://github.com/A-Wehby/Connected-Cars-DDoS-Dataset. Accessed: 22/02/2025. 2024.

[91] Fábio Gonçalves, Joaquim Macedo, and Alexandre Santos. “An intelligent hierarchical security framework for vanets”. In: *Information* 12.11 (2021), p. 455.

[92] Davide Chicco and Giuseppe Jurman. “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. In: *BMC genomics* 21.1 (2020), pp. 1–13.

[93] Davide Chicco and Giuseppe Jurman. “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation”. In: *BMC genomics* 21.1 (2020), pp. 1–13.

[94] Lionel N Tidjon, Marc Frappier, and Amel Mammar. “Intrusion detection systems: A cross-domain overview”. In: *IEEE Communications Surveys & Tutorials* 21.4 (2019), pp. 3639–3681.

[95] Brooke Lampe and Weizhi Meng. “Intrusion detection in the automotive domain: A comprehensive review”. In: *IEEE Communications Surveys & Tutorials* (2023).

[96] Tejasvi Alladi et al. “Artificial intelligence (AI)-empowered intrusion detection architecture for the internet of vehicles”. In: *IEEE Wireless Communications* 28.3 (2021), pp. 144–149.

[97] Mahshid Behravan et al. “Intrusion Detection Systems Based on Stacking Ensemble Learning in VANET”. In: *2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*. 2022, pp. 1–7. DOI: 10.1109/ICCSPA55860.2022.10019171.

[98] Peter Joseph Yurkovich. “RSU-Based Intrusion Detection and Autonomous Intersection Response Systems”. PhD thesis. Virginia Tech, 2022.

[99] Joseph Kamel et al. “Simulation framework for misbehavior detection in vehicular networks”. In: *IEEE transactions on vehicular technology* 69.6 (2020), pp. 6631–6643.

[100] Elijah M Maseno, Zenghui Wang, and Hongyan Xing. “A systematic review on hybrid intrusion detection system”. In: *Security and Communication Networks* 2022.1 (2022), p. 9663052.

[101] Benjamin Feinstein, David Curry, and Herve Debar. *The Intrusion Detection Message Exchange Format (IDMEF)*. RFC 4765. Mar. 2007. DOI: 10.17487/RFC4765. URL: https://www.rfc-editor.org/info/rfc4765.

[102] Michael A. Erlinger and Mark Wood. *Intrusion Detection Message Exchange Requirements*. RFC 4766. Mar. 2007. DOI: 10.17487/RFC4766. URL: https://www.rfc-editor.org/info/rfc4766.

[103] Gregory Matthews and Benjamin Feinstein. *The Intrusion Detection Exchange Protocol (IDXP)*. RFC 4767. Mar. 2007. DOI: 10.17487/RFC4767. URL: https://www.rfc-editor.org/info/rfc4767.

[104] Rainer Gerhards. *The Syslog Protocol*. RFC 5424. Mar. 2009. DOI: 10.17487/RFC5424. URL: https://www.rfc-editor.org/info/rfc5424.

[105] Slim Abbes and Slim Rekhis. "A blockchain-based solution for reputation management in IoV". In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2021, pp. 1129–1134.

[106] *Risk grows as multi-vector attacks become the norm*. URL: https://www.cloudflare.com/en-gb/the-net/multi-vector-threats/ (visited on 09/05/2025).

[107] ETSI. *TS 102 731 - Intelligent Transport Systems (ITS); Security; Security Services and Architecture*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/102700_102799/102731/01.01.01_60/ts_102731v010101p.pdf. European Telecommunications Standards Institute (ETSI), 2010.

[108] Joseph Kamel et al. "A Misbehavior Authority System for Sybil Attack Detection in C-ITS". In: *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. 2019, pp. 1117–1123. DOI: 10.1109/UEMCON47517.2019.8993045.

[109] Badis Hammi, Mohamed Yacine Idir, and Rida Khatoun. "A machine learning based approach for the detection of sybil attacks in C-ITS". In: *2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 2022, pp. 1–4. DOI: 10.23919/APNOMS56106.2022.9919991.

[110] Pengwenlong Gu et al. "Vehicle Driving Pattern Based Sybil Attack Detection". In: *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 2016, pp. 1282–1288. DOI: 10.1109/HPCC-SmartCity-DSS.2016.0182.

[111] Sarra Benadla et al. "Detecting Sybil Attacks in Vehicular Fog Networks Using RSSI and Blockchain". In: *IEEE Transactions on Network and Service Management* 19.4 (2022), pp. 3919–3935. DOI: 10.1109/TNSM.2022.3216073.

[112] S. Rakhi and K. R. Shobha. "LCSS Based Sybil Attack Detection and Avoidance in Clustered Vehicular Networks". In: *IEEE Access* 11 (2023), pp. 75179–75190. DOI: 10.1109/ACCESS.2023.3294469.

[113] Gideon Adele et al. "A Survey and Comparative Analysis of Methods for Countering Sybil Attacks in VANETs". In: *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*. 2024, pp. 0178–0183. DOI: 10.1109/CCWC60891.2024.10427979.

[114] Pengwenlong Gu et al. “Support Vector Machine (SVM) Based Sybil Attack Detection in Vehicular Networks”. In: *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. 2017, pp. 1–6. DOI: 10.1109/WCNC.2017.7925783.

[115] 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)*. 2017, pp. 1–6. DOI: 10.1109/MOBISECSERV.2017.7886565.

[116] Jiabin Li et al. “RTED-SD: A Real-Time Edge Detection Scheme for Sybil DDoS in the Internet of Vehicles”. In: *IEEE Access* 9 (2021), pp. 11296–11305. DOI: 10.1109/ACCESS.2021.3049830.

[117] Badis Hammi et al. “Is it Really Easy to Detect Sybil Attacks in C-ITS Environments: A Position Paper”. In: *IEEE Transactions on Intelligent Transportation Systems* 23.10 (2022), pp. 18273–18287. DOI: 10.1109/TITS.2022.3165513.

[118] Yun Xu and Royston Goodacre. “On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning”. In: *Journal of analysis and testing* 2.3 (2018), pp. 249–262.

[119] *Voting Classifier*. (visited on: 27.01.2025). URL: https://scikit-learn.org/stable/modules/ensemble.html#voting-classifier.

[120] ETSI. *TS 102 894-2 - Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary*. Standard. Available online at: https://www.etsi.org/deliver/etsi_ts/102800_102899/10289402/01.03.01_60/ts_10289402v010301p.pdf. European Telecommunications Standards Institute (ETSI), 2018.

[121] ETSI. *EN 302 895 - Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)*. Standard. Available online at: https://www.etsi.org/deliver/etsi_en/302800_302899/302895/01.01.01_60/en_302895v010101p.pdf. European Telecommunications Standards Institute (ETSI), 2014.

[122] Ludmila I Kuncheva and Christopher J Whitaker. “Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy”. In: *Machine learning* 51 (2003), pp. 181–207.

[123] Edivaldo Pastori Valentini, Rodolfo Ipolito Meneguette, and Adil Alsuhaim. “An Attacks Detection Mechanism for Intelligent Transport System”. In: *2020 IEEE International Conference on Big Data (Big Data)*. 2020, pp. 2453–2461. DOI: 10.1109/BigData50022.2020.9378309.

[124] Ammar Haydari and Yasin Yilmaz. “Real-Time Detection and Mitigation of DDoS Attacks in Intelligent Transportation Systems”. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 157–163. DOI: 10.1109/ITSC.2018.8569698.

[125] M. Poongodi et al. “DDoS Detection Mechanism Using Trust-Based Evaluation System in VANET”. In: *IEEE Access* 7 (2019), pp. 183532–183544. DOI: 10.1109/ACCESS.2019.2960367.

[126] M. Poongodi et al. “Intrusion Prevention System for DDoS Attack on VANET With reCAPTCHA Controller Using Information Based Metrics”. In: *IEEE Access* 7 (2019), pp. 158481–158491. DOI: 10.1109/ACCESS.2019.2945682.

[127] Mingyuan Zang and Ying Yan. “Machine Learning-Based Intrusion Detection System for Big Data Analytics in VANET”. In: *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. 2021, pp. 1–5. DOI: 10.1109/VTC2021-Spring51267.2021.9448878.

[128] Ying Gao et al. “A Distributed Network Intrusion Detection System for Distributed Denial of Service Attacks in Vehicular Ad Hoc Network”. In: *IEEE Access* 7 (2019), pp. 154560–154571. DOI: 10.1109/ACCESS.2019.2948382.

[129] Raenu Kolandaisamy et al. “A multivariant stream analysis approach to detect and mitigate DDoS attacks in vehicular ad hoc networks”. In: *Wireless communications and mobile computing* (2018). DOI: 10.1155/2018/2874509.

ECOLE DOCTORALE

Titre : Détection des cyberattaques dans les systèmes de transport intelligents et coopératifs

Mots clés : Apprentissage automatique, Cyberattaques, Déni de service, Comportement malveillant, Détection, Système de détection d'intrusions

Résumé : Les véhicules modernes deviennent de plus en plus intelligentes, communiquant entre elles pour améliorer la fluidité du trafic et renforcer la sécurité. Toutefois, cette connectivité les expose également aux cyberattaques. L'une des menaces les plus importantes est l'attaque par déni de service distribué (DDoS), où des acteurs malveillants submergent le réseau de messages frauduleux, perturbant ainsi la communication en temps réel entre les véhicules. Cela peut entraîner des problèmes allant de simples défaillances de navigation à des accidents graves. Cette recherche se concentre sur un type spécifique d'attaque DDoS exploitant les messages CAM (Cooperative Awareness Message), utilisés par les véhicules pour partager leur position et leur vitesse. Nous proposons une approche pour un système de détection d'intrusions (IDS) distribué, conçu pour atténuer les attaques DDoS dans un environnement de véhicules connectés. Dans un premier temps, Nous avons évalué l'impact des attaques CAM-DDoS sur les véhicules connectés à travers divers scénarios de simulation. Ces scénarios ont révélé une diminution du taux de diffusion des CAMs par les véhicules légitimes et une accumulation de données invalides. Par la suite, nous avons proposé une architecture distribuée pour un IDS et évalué l'efficacité de trois algorithmes supervisés d'apprentissage ensembliste contre ces attaques. Face au défi de la généralisabilité des modèles d'apprentissage automatique, nous avons également étudié la robustesse de notre approche de détection contre d'autres types d'attaques DDoS, comme le JamDDoS. Les résultats montrent que tous les messages DDoS sont traités de manière uniforme, ce qui explique la robustesse des modèles contre les attaques étudiées. En outre, nous avons démontré comment une attaque DDoS peut être transformée en une attaque Sybil DDoS, capable d'échapper aux détecteurs DDoS dans un environnement de véhicules connectés où la sécurité est critique. Pour y remédier, nous avons développé et évalué une approche optimisée de gestion des données, servant de base à notre modèle de détection ensembliste par vote majoritaire. Les résultats obtenus démontrent une détection efficace des attaques, avec une réduction des faux positifs dans le cas des attaques Sybil DDoS.

Title : Cyberattacks Detection in Cooperative Intelligent Transportation System (C-ITS)

Keywords : Machine Learning, Cyberattacks, Denial of Service, Misbehavior, Detection, Intrusion Detection System

Abstract : Modern vehicles are becoming smarter, communicating with each other to enhance traffic flow and improve safety. However, this connectivity also exposes them to cyberattacks, particularly Distributed Denial of Service (DDoS) attacks where malicious actors overwhelm vehicular networks with fake messages, disrupting real-time communication between cars. This can lead to issues ranging from navigation failures to serious accidents. This research focuses on a specific type of DDoS attack that exploits Cooperative Awareness Messages (CAMs), which cars use to share their location and speed. We propose an approach for a distributed Intrusion Detection System (IDS) designed to mitigate DDoS attacks in a connected car environment. Initially, we assessed the impact of CAM-DDoS attacks on connected cars through various simulation scenarios. These scenarios revealed a decrease in the CAM broadcast rate from legitimate cars and an accumulation of invalid data. Subsequently, we proposed a distributed architecture for an IDS and assessed the effectiveness of three supervised ensemble learning algorithms against these attacks. Given the challenge of generalizability in machine learning models, we also evaluated the robustness of our DDoS detection approach against other types of DDoS attacks, such as JamDDoS. The results show that all DDoS messages were processed uniformly, which explains the models' robust performance against studied DDoS attacks. Additionally, we demonstrated how a DDoS attack can be transformed into a Sybil DDoS attack that avoids DDoS detectors in a safety-critical connected car environment. To address this, we developed and evaluated an optimized data-handling approach that serves as the foundation for our proposed ensemble hard voting detection model. The results obtained demonstrate an effective attack detection with reduced false alarms against Sybil DDoS attacks.

Institut Polytechnique de Paris
91120 Palaiseau, France