People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed El Bachir El Ibrahimi University of Bordj Bou Arreridj
Faculty of Mathematics and Computer Science
Department of Computer Science

THESIS

Presented in fulfillment of the requirements for the degree of

Master in Computer Science

Speciality: Networking & Multimedia

THEME

Machine Learning for Misbehavior Detection in Next-Generation Vehicular Networks

Presented by:
MADI Ahmed Salah Eddine
MEKHFI Baya

Publicly defended on: 12/06/2025

In front of the jury composed of:
President: Dr. SAHA Adel
Examiner: Dr. BENAOUDA Nadjib
Supervisor: Dr. MOUSSAOUI Boubakeur

2024/2025

Acknowledgements

First and foremost, I thank Almighty God for granting me the strength and perseverance to carry out this work.

I would like to express my deepest gratitude to **Mr. Moussaoui Boubakeur**‘ , my thesis supervisor, for his guidance, insightful advice, and continuous support throughout this project.

I also extend my sincere thanks to all my teachers for the quality of their instruction, and to my colleagues for their support and the enriching exchanges we shared.

Dedication

To my loving family,

for their unwavering support, love, and sacrifices throughout this journey.

To my parents, who always believed in me,

To my sisters, for their encouragement and patience

— Madi Ahmed Salah Eddine

Dedication

To the soul of my little brother,

"AbdElBasset"

I dedicate this graduation to you.

To my dear family,

To my father, who worked tirelessly to support my education,

To my mother, who stood with me through every step,

To my sister and her husband, ***Hayet and Chaouki****,*

always encouraging and supportive,

To ***my grandmother****, whose prayers were my greatest support,*

And to ***Alladin and Ghizlane****, the joy of my days,*

I am deeply grateful to you all.

To my beloved friends, ***"SABANIKH "****, Especially* ***Maria****, the closest to my heart. thank you for your constant support, sincerity, and love.*

With heartfelt thanks also to my university friends ***Hanan, Marwa, and Nadine****, for their beautiful friendship and kind companionship. And to everyone who stood by me, Thank you sincerely and wholeheartedly.*

Every fall was a lesson, and every tear was fuel. And here I am today, standing tall a graduate worthy of victory.

— Baya Mekhfi

Abstract

Connected vehicles have great potential to enhance road safety, reduce traffic congestion, and play a vital role in green engineering by reducing pollution and fuel consumption. By enabling more efficient traffic flow, eco-routing, and optimized driving behaviors, connected vehicles contribute to a cleaner environment. However, when a vehicle is compromised, it can pose a serious threat to the entire network due to the potential harm it can cause. One of the major challenges in vehicular networks is the detection of misbehaving vehicles, which should then be blacklisted or their certificates revoked. In this work, we propose a novel scheme that leverages machine learning to accurately detect and classify vehicle behavior, enabling effective identification and management of misbehaving vehicles. To assess the effectiveness of our approach, a comprehensive comparative analysis was performed. The results demonstrate that our model outperforms existing methods in accurately classifying vehicle behaviors, highlighting its potential for real-world deployment in securing vehicular networks.

Index Terms : Misbehavior detection, Security, Machine learning, Classification, Eco-Friendly Mobility.

Résumé

Les véhicules connectés offrent un fort potentiel pour améliorer la sécurité routière, réduire les embouteillages et jouer un rôle essentiel dans l'ingénierie écologique en limitant la pollution et la consommation de carburant. En facilitant un flux de circulation plus efficace, un routage éco-responsable et des comportements de conduite optimisés, ces véhicules contribuent à un environnement plus propre.

Cependant, lorsqu'un véhicule est compromis, il peut représenter une menace sérieuse pour l'ensemble du réseau en raison des dommages qu'il peut causer. L'un des défis majeurs dans les réseaux véhiculaires est la détection des véhicules malveillants, qui doivent être mis sur liste noire ou voir leurs certificats révoqués.

Dans ce travail, nous proposons un nouveau mécanisme basé sur l'apprentissage automatique permettant de détecter et de classer avec précision le comportement des véhicules, facilitant ainsi leur identification et leur gestion. Pour évaluer l'efficacité de notre approche, une analyse comparative approfondie a été réalisée. Les résultats montrent que notre modèle surpasse les méthodes existantes en matière de classification des comportements, soulignant ainsi son potentiel d'intégration dans des environnements réels pour renforcer la sécurité des réseaux véhiculaires.

Mots-clés : Détection de comportements malveillants, sécurité, apprentissage automatique, classification, mobilité écologique.

ملخص

تمتلك المركبات المتصلة إمكانيات كبيرة لتعزيز السلامة على الطرق، والحد من الازدحام المروري، ولعب دور حيوي في الهندسة البيئية من خلال تقليل التلوث واستهلاك الوقود. ومن خلال تحسين تدفق حركة المرور، وتوجيه المركبات بطرق صديقة للبيئة، وتحسين سلوكيات القيادة، تُسهم هذه المركبات في تحقيق بيئة أنظف. ومع ذلك، فإن تعرّض مركبة واحدة للاختراق قد يشكل تهديدًا خطيرًا للشبكة بأكملها بسبب الأضرار المحتملة التي يمكن أن تتسبب بها. ويُعد كشف المركبات ذات السلوك السيئ من التحديات الرئيسية في شبكات المركبات، حيث يجب إدراجها في القوائم السوداء أو إلغاء شهاداتها الرقمية.

في هذا العمل، نقترح آلية جديدة تعتمد على تقنيات التعلم الآلي لرصد وتصنيف سلوك المركبات بدقة، مما يتيح التعرف الفعّال على المركبات المخالفة وإدارتها. ولتقييم فعالية هذا النهج، تم إجراء تحليل مقارن شامل. أظهرت النتائج أن النموذج المقترح يتفوق على الطرق الحالية من حيث دقة تصنيف سلوك المركبات، مما يؤكد قدرته على التطبيق الفعلي لتعزيز أمان شبكات المركبات.

الكلمات المفتاحية: كشف السلوكيات غير المتعاونة، الأمن، التعلم الآلي، التصنيف، التنقل الصديق للبيئة.

Contents

List of Figures **2**

List of Tables **3**

List of Abbreviations **4**

Introduction **7**

1 Vehicular Ad Hoc Networks (VANETs) **10**

1.1 Introduction 10

1.2 Definition 10

1.3 Architecture 11

1.3.1 On Board Unit (OBU) 11

1.3.2 Road Side Unit (RSU) 12

1.3.3 Public Key Infrastructure(PKI) 12

1.3.4 Trusted Authority (TA) 12

1.3.5 Certificate Authority(CA) 12

1.3.6 Difference 13

1.4 Standardized Communication Framework 13

1.5 Communication modes 15

1.5.1 Vehicle to Vehicle(V2V) 15

1.5.2 Vehicle to Infrastructure(V2I) 15

1.5.3 Vehicle to Pedestrian(V2P) 16

1.6 Applications of VANETs 16

1.6.1 Safety Applications 17

1.6.2 Traffic Efficiency Applications 17

1.6.3 Infotainment and Commercial Applications 17

2 AI and its Role in ITS **19**

2.1 Definition 19

2.2 Historical Overview of AI 19

2.2.1 Initial Establishment (1960s–1970s) 19

2.2.2 Symbolic AI and Education 20

2.2.3 Prologue and Logic (1980s) 20

2.2.4 Transition to Machine Learning (2010–2020) 21

2.3 AI: Domains and Applications 21

2.3.1 Basic Areas of Artificial Intelligence 21

2.3.1.1 Machine Learning and Deep Learning 21

2.3.1.2 Natural Language Processing (NLP) 22

2.3.1.3 Computer Vision 22

2.3.1.4 Robotics and Autonomous Systems 23

2.3.2 Practical Applications 23

2.3.2.1 Online Commerce and Personal Services 23

2.3.2.2 Cybersecurity 23

2.3.2.3 Healthcare 23

2.3.2.4 Agriculture 23

2.3.2.5 Financial Services 23

2.3.2.6 Urban and Transport Planning 24

2.4 Role of AI in ITS 24

2.5 Machine Learning 25

2.5.1 Categories of Machine Learning 26

2.5.1.1 Supervised Learning 26

2.5.1.2 Unsupervised Learning 26

2.5.1.3 Reinforcement Learning 27

2.5.2 Common Classification Algorithms 27

2.6 Deep Learning 29

2.6.1 Definition and Key Concepts 29

2.6.2 Deep Learning and Neural Networks 29

2.7 Deep learning methods 30

2.8 Comparative Overview: ML vs DL in VANETs 30
2.9 Local and Regional Large Language Models (LLMs) 31
2.9.1 DziriBERT 31
2.9.2 DarjaBERT 32

3 Security in Vehicular Networks **33**
3.1 Introduction 33
3.2 Security Characteristics 33
3.2.1 Authentication 34
3.2.2 Confidentiality 34
3.2.3 Availability 34
3.2.4 Integrity 35
3.2.5 Non-repudiation 35
3.3 Security Threats 35
3.3.1 Denial of Service (DoS) 36
3.3.2 Sybil Attacks 36
3.3.3 Eavesdropping 37
3.3.4 Linkability attack 37
3.3.5 Forward/Backward Attack 37
3.4 Security requirements 37
3.4.1 Digital Certificates and Public Key Infrastructure (PKI) 38
3.4.2 Cryptographic Algorithms 38
3.4.3 Secure Communication Protocols 39
3.4.4 Intrusion Detection Mechanisms 39
3.5 Conclusion 40

4 False Position Attack **41**
4.1 Introduction 41
4.2 Background and Related Work 41
4.2.1 Overview of False Position Attacks 41
4.2.2 Historical Context and Evolution 42
4.2.3 Related Attacks and Research Efforts 42
4.2.4 Attacker Capabilities 43

4.3 Characteristics of False Position Attack in VANETs 44
4.3.1 How the Attack Works 44
4.3.2 Goals and Immediate Impacts 45
4.3.3 Long-Term Impacts and Challenges 47
4.4 Machine Learning Algorithms for Misbehavior Detection 48
4.4.1 Random Forest (RF) 48
4.4.2 Support Vector Machine (SVM) 49
4.4.3 Extreme Gradient Boosting (XGBoost) 50
4.5 Detection and Preventing Techniques 51
4.5.1 Addressing False Position Attacks 51
4.6 Conclusion 52

5 Contribution **53**
5.1 Introduction 53
5.2 Dataset and Attack Scenarios 53
5.3 Proposed approach 55
5.3.1 Preprocessing 55
5.3.1.1 Data Integrity Check and Initial Feature Reduction 56
5.3.1.2 Trust Feature Engineering 56
5.3.1.3 Dataset Subsetting for Model Training 57
5.3.1.4 Final Feature Selection for Modeling 57
5.3.2 Model Development 58
5.3.2.1 XGBOOST 58
5.3.2.2 RF 59
5.3.2.3 SVM 59
5.3.3 Training 59
5.3.3.1 Model Configurations and Training 60
5.3.4 Testing 62
5.4 Performance Evaluation 63
5.4.1 Evaluation Metrics 63
5.4.2 Results and discussion 64
5.5 Conclusion 65

General Conclusion **66**

References **67**

Conference Paper **74**

List of Figures

1.1 Architecture of Vanet 11
1.2 WAVE Architecture and Protocol Stack. [IEEE 1609.0-2013 14
1.3 DSRC Spectrum Allocation 14
1.4 VANET's communication modes 16

2.1 AI vs. Machine learning vs. Deep learning 22
2.2 Role of AI in ITS 24
2.3 Types of Machine Learning 26
2.4 Layered Structure of a Deep Neural Network 29

3.1 VANET security requirements 34
3.2 Threat model in a VANET 36
3.3 How a PKI system works in a VANET scenario 38
3.4 Cryptography 39

5.1 machine learning framework for misbehavior detection 55
5.2 Performance Comparison by Attack 64

List of Tables

2.1 Overview of common classification algorithms 28

4.1 Different attack types and their parameters. 44

5.1 Models Evaluations 65

5.2 Formulas for key evaluation metrics 65

List of Abbreviations

ACC Accuracy.

AI Artificial Intelligence.

C-ITS Cooperative Intelligent Transportation Systems.

CA Certificate Authority.

CAM Cooperative Awareness Message.

CNN Convolutional Neural Network.

DAE Denoising Autoencoder.

DBN Deep Belief Network.

DL Deep Learning.

DoS Denial of Service.

DSRC Dedicated Short Range Communication.

F1-score Harmonic Mean of Precision and Recall.

FN False Negative.

FP False Positive.

GLOSA Green Light Optimal Speed Advisory.

GPS Global Positioning System.

I2I Infrastructure to Infrastructure.

ITS Intelligent Transportation Systems.

LSTM Long Short-Term Memory.

MANET Mobile Ad Hoc Network.

ML Machine Learning.

OBU On-Board Unit.

PKI Public Key Infrastructure.

RF Random Forest.

RNN Recurrent Neural Network.

RSU Road Side Unit.

SVM Support Vector Machine.

TA Trusted Authority.

TN True Negative.

TP True Positive.

V2I Vehicle to Infrastructure.

V2P Vehicle to Pedestrian.

V2V Vehicle to Vehicle.

V2X Vehicle to Everything.

VANET Vehicular Ad hoc Network.

WAVE Wireless Access in Vehicular Environment.

XGBoost Extreme Gradient Boosting.

General Introduction

Cooperative Intelligent Transport Systems (C-ITS) promise, on the one hand, to enhance road safety by reducing accidents. On the other hand, they play a crucial role in creating a greener, more sustainable transportation ecosystem by reducing pollution and promoting environmentally conscious travel. These goals can be achieved through real-time communica- tion between vehicles (V2V), infrastructure (V2I), and traffic management systems. C-ITS improve traffic flow and fuel efficiency, significantly reducing emissions. These systems offer eco-driving advice, such as optimal acceleration and braking patterns, and suggest routes that minimize congestion and fuel consumption. C-ITSs also support green engineering objectives by facilitating the integration of electric vehicles, prioritizing public transport and non-motorized modes like bicycles, and promoting smart mobility planning. In addition, they contribute to long-term environmental goals by reducing traffic-related energy use and enabling infrastructure design that aligns with sustainable development principles. By combining safety improvements with pollution reduction and energy efficiency, C-ITS are a cornerstone of future green transportation systems.

C-ITS are built upon a complex network of communication types that go far beyond simple car-to-car interaction. These systems facilitate diverse forms of real-time communication, such as interactions between vehicles (V2V), links connecting vehicles with roadside infrastructure (V2I), exchanges between infrastructure components (I2I), and communication with pedestrians (V2P). Altogether, these interactions fall under the umbrella of vehicle-to-everything, or V2X, highlighting the broad connectivity that defines modern smart transportation environments. This integrated communication approach enables different elements of the transport ecosystem to coordinate efficiently, adapt to real-time situations, and enhance overall system responsiveness.

Despite technological advances, vehicles especially those operated manually remain a key risk factor on roads. However, the landscape is evolving rapidly with the introduction of intelligent transport solutions, particularly the cooperative capabilities offered by C-ITS. The vision for the near future is that every connected vehicle will function as a dynamic data node, gathering and sharing information anonymously with central traffic control systems. These systems will then analyze the inputs immediately and issue targeted notifications to only those road users who are directly affected by potential hazards. Whether it's alerting a driver to a traffic jam ahead or warning of a pedestrian stepping into the street, the aim is to deliver only the most relevant and timely information ultimately making mobility safer, smarter, and more efficient.

In order to meet the objectives mentioned above, C-ITS enable smooth data exchange between mobile nodes through wireless communication technologies such as Dedicated Short- Range Communications (DSRC) and 5G. C-ITS applications share sensitive data (such as identity, location, speed, direc- tion, etc.) for informed decision-making. The accuracy of a node's location information is especially critical, as it directly affects the reliability of the system. Unfortunately, vehicles are exposed to various cyber-security threats, including Sybil attacks, where an attacker generates multiple fake identities to manipulate the network, denial of service (DoS) attacks that disrupt communication by overloading the network, as well as identity spoofing and eavesdropping, which compromise data integrity and confidentiality.

Intrusion detection in Vehicular Ad Hoc Networks (VANETs) has been extensively studied using centralized, decentralized, and distributed approaches to combat various attacks such as Sybil, black-hole, DoS, and position falsifica- tion. Centralized systems rely on trusted authorities to collect and analyze data, while decentralized approaches use local nodes like Road Side Units (RSU) or cluster heads to monitor behavior using game theory, neural networks, or filtering techniques. Distributed systems allow vehicles to detect threats collaboratively through consistency checks, signal analysis, or trajectory comparisons. Recently, machine learning-based methods have gained traction, particularly for detecting posi- tion falsification attacks. These models use various features such as signal strength, velocity, trajectory plausibility, and movement patterns.[1]

In this study, we focus exclusively on machine learning- based approaches to address position falsification attacks. We propose a novel detection model that uses the XGBOOST framework,

which is designed to improve accuracy, robustness, and real-time performance. Our approach demonstrates the potential of advanced machine learning techniques in securing location-based services against malicious manipulation.

This work is organized into five main chapters:

- The first chapter addresses Vehicular Ad Hoc Networks (VANETs).
- Chapter 2 details the key concepts related to Intelligent Transportation Systems (ITS) and highlights the technologies that contribute to optimizing traffic management, road safety, and transportation efficiency in urban environments.
- Chapter 3 focuses on security issues in VANETs.
- Chapter 4 is dedicated to the analysis of position spoofing attacks.
- Chapter 5 presents the proposed solution for identifying malicious behaviors in vehicular networks. It outlines the experimental context, the implemented methods, the obtained results, and provides a discussion on the performance of the proposed model.
- Finally, a general conclusion summarizes the contributions of this research, highlights the encountered limitations, and suggests directions for future work.

Chapter 1

Vehicular Ad Hoc Networks (VANETs)

1.1 Introduction

"Intelligent transportation system (ITS) is an advanced application that aims to provide services relating to different modes of transport and traffic management and enable users to be better informed and make safer, more coordinated, and 'smarter' use of transport networks".[2]

ITS adds intelligence to either the vehicle, the infrastructure or the VRU. Vehicle ITS systems, which were shown to have a high safety potential,Infrastructures can be made more intelligent through sensors near traffic lights, which for example allow providing safe crossing to elderly pedestrians and persons with special needs, and providing green waves for cyclists. Today the importance of ITS increase rapidly where Cities face urgent transport challenges. Many are starting to tackle them by implementing new intelligent transport systems, and some have achieved impressive benefits. ITS technology can contribute to the safety and mobility vulnerable road users by enhancing the vehicle and transportation infrastructure. This will enable road users, including pedestrians and drivers, to safely utilize transportation networks.[3]

1.2 Definition

VANETs are an emerging technology derived from Mobile Ad Hoc Networks (MANETs), where the mobile nodes are intelligent vehicles equipped with high-end technologies (computers, radars,GPS , various types of sensors, and network devices). VANETs enable inter-vehicle communications (V2V) and vehicle-to-infrastructure communications (V2I). The different nodes can exchange alerts or useful information to improve road traffic safety. They can also share

data (music, videos, advertisements, etc.) to make the time spent on the road more pleasant and less boring. VANETs are based on two types of applications: those that form the core of an ITS aimed at improving road safety, and those deployed for the comfort of passengers.[4]

1.3 Architecture

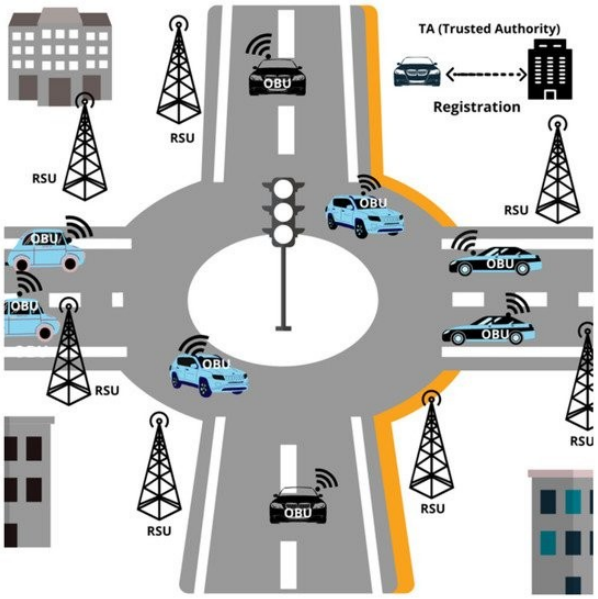

Figure 1.1: Architecture of Vanet[5].

A VANET Network primarily consists of three entities:

1.3.1 On Board Unit (OBU)

A vehicle in a VANET is equipped with an On Board Unit (OBU), enabling communication with RSUs (V2I) and with other vehicles (V2V). It periodically broadcasts safety messages containing sensitive information such as location, timestamp, velocity, identity, and more. Our misbehavior detection module, based on machine learning, will be installed on each vehicle to ensure a rapid response to any suspicious or abnormal situations. The module continuously monitors incoming messages and classifies them in real time, reporting any detected anomalies to the Trusted Authority (TA) immediately.

1.3.2 Road Side Unit (RSU)

These entities operate under the authority of the Trusted Authorities (TAs). They are deployed along the roadside and may include devices such as traffic lights, street lamps, or other infrastructure components. Their primary role is to support the TA in managing vehicle movement and overall traffic flow. Additionally, they serve as access points to the network and to various types of traffic-related information.[6]

1.3.3 Public Key Infrastructure(PKI)

PKI is a set of physical components (computers, cryptographic software tools, or hardware), human procedures (such as checks and validation), and software (both system and application) designed to manage the public keys of users within a system.[7]

1.3.4 Trusted Authority (TA)

The Trusted Authority (TA) serves as the unique trusted entity in the system. It can act as the Public Key Infrastructure (PKI) or as another authority authorized by the PKI. It manages all critical operations such as vehicle registration, key management, verification, and more. Due to its sample storage and computational resources, the TA securely handles these essential transactions. Additionally, if a vehicle exhibits malicious behavior, the TA has the authority to revoke its certificate to maintain system integrity.

1.3.5 Certificate Authority(CA)

The Certificate Authority (CA) is the most critical component of the Public Key Infrastructure (PKI), owing to its central role in the various communication and trust mechanisms within the system. The CA is responsible for the issuance and management of digital certificates. Specifically, it generates public key certificates and ensures both the integrity and authenticity of their contents by digitally signing them with its private key. In order to issue a certificate, the CA must first receive a certification request, which includes the public key of the requesting entity.[8]

1.3.6 Difference

In vehicular ad-hoc networks (VANETs), the Trusted Authority (TA) serves as the central entity, supervising and controlling the system's security framework [9]. The Certificate Authority (CA), under TA's oversight, manages the issuance and validation of digital certificates, ensuring trust in communications [10]. The Public Key Infrastructure (PKI), built on CA's certificate management, provides the infrastructure for secure vehicle-to-vehicle interactions [11]. This hierarchical flow TA to CA to PKI enables secure data exchange, with the TA revoking certificates to maintain integrity if needed [12]. Vehicles utilize the PKI to authenticate messages, relying on the CA's signed certificates and the TA's authority [9]. The TA's role extends beyond PKI to include registration and key management, distinguishing it from the CA's focused certificate duties [10]. The CA operates within the PKI framework, bridging the TA's trust anchor role with practical certificate deployment [11]. This structure ensures scalability and security, critical for dynamic VANET environments [12]. The TA's control over the CA prevents unauthorized certificate issuance, safeguarding the PKI's reliability [9]. Together, these components form a robust trust chain, essential for emergency message dissemination in VANETs,as it's illustrated in Figure 1.1 . [10].

1.4 Standardized Communication Framework

Dedicated standardization committees are actively refining a comprehensive suite of standards to govern Vehicular Ad Hoc Networks (VANETs). Prominent among these are the IEEE 1609.x family, the IEEE 802.11p standard for physical and MAC layers, and the overarching Wireless Access in Vehicular Environments (WAVE) architecture.

The WAVE suite itself establishes a layered communication model, specifically designed to enable devices compliant with the IEEE 802.11 standard to operate seamlessly over the Dedicated Short-Range Communication (DSRC) frequency band. Further detailing this ecosystem, the IEEE 1609 family of standards delineates the core architecture, along with the requisite protocols, services, and interfaces. These elements are crucial for ensuring that all WAVE-compliant stations can interoperate effectively within the dynamic VANET environment. Significantly, the WAVE architecture also incorporates foundational provisions that define the security mechanisms for message exchange between vehicular nodes.

Collectively, these WAVE standards provide a robust and essential platform for the development

and deployment of a wide spectrum of VANET applications. These applications span critical domains such as road safety and network security, enhanced navigation services, automated toll collection systems, and the dissemination of real-time traffic alerts. The interrelation of the various standards within the IEEE 1609 WAVE architecture, particularly their alignment with the OSI reference model[13], is further elucidated in Figure 1.2

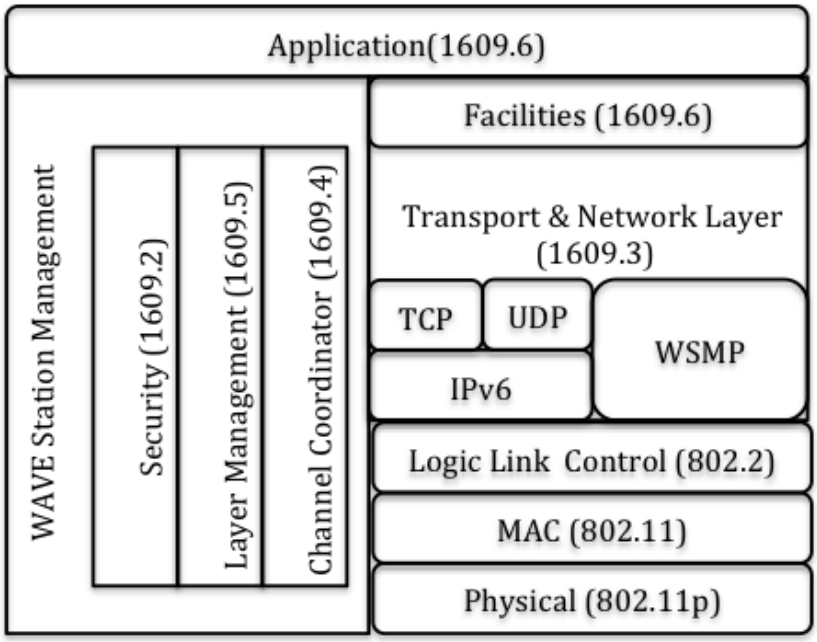

Figure 1.2:]

WAVE Architecture and Protocol Stack. [IEEE 1609.0-2013][13].

In the regulatory landscape of the United States, the Federal Communications Commission (FCC) has specifically allocated the DSRC spectrum, which ranges from 5.850 GHz to 5.925 GHz. This allocated band is meticulously structured into seven distinct 10 MHz wide channels, identified numerically as 172, 174, 176, 178, 180, 182, and 184. To promote broad interoperability and expedite the standardization process, the DSRC band operates on a license-by-rule basis. This approach means that while individual operational licenses are not required, the use of this spectrum is nevertheless subject to stringent regulations to ensure orderly and interference-free access[13]. As illustrated in Figure 1.3

Figure 1.3: DSRC Spectrum Allocation[13].

the centrally positioned channel, 178, is designated as the Control Channel (CCH), pivotal for system management and coordination. Flanking this, two channels at the spectrum's extremes, namely 172 and 184, are exclusively reserved for vital safety-of-life applications, including emerging technologies for accident avoidance. More specifically, channel 172 is optimized for applications that demand high availability and consistently low latency, while channel 184 is intended for higher-power transmissions and broader public safety communications. The remaining four channels operate as Service Channels (SCHs), offering flexibility for a combination of both safety-critical and non-safety-related applications, such as infotainment or general data services.
This strategic partitioning of the bandwidth, with its clear prioritization of safety-critical communications, unequivocally underscores the paramount importance of ensuring robust safety features within the entire VANET application ecosystem.[13]

1.5 Communication modes

In VANETs, there are primarily the fixed entities that make up the infrastructure (RSUs and TAs) and the vehicles. To exchange the various pieces of information and data related to road-user safety and comfort, these different entities must establish communications among themselves. For this reason, two types of communications are distinguished: vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I).

1.5.1 Vehicle to Vehicle(V2V)

This mode of communication operates using devices installed in vehicles called On-Board Units (OBUs), following a decentralized architecture. Communication between two vehicles takes place directly, in inter-vehicular ad hoc mode. They do not need to rely on infrastructure to communicate, as long as each vehicle is within the other's radio range. Otherwise, they rely on other vehicles to act as relays (intermediaries). This kind of transmission is known as multi-hop communication.

1.5.2 Vehicle to Infrastructure(V2I)

Vehicle-to-Infrastructure (V2I) communication is also referred to as infrastructure mode communication. This communication mode is made possible by the various entities within the

VANET network. In fact, the On-Board Units (OBUs) in vehicles, the Road Side Units (RSUs) placed along the roads, and even the Trusted Authorities (TAs) all work together to enable communication within the vehicular network. This mode of communication provides stronger connectivity compared to Vehicle-to-Vehicle (V2V) communication. It also ensures better utilization of network resources.also, it enables vehicles to benefit from additional features and services, such as Internet access and weather information.[14]

1.5.3 Vehicle to Pedestrian(V2P)

Vehicle-to-Pedestrian (V2P) communication refers to the exchange of information between vehicles and pedestrians. This type of communication can be used to warn drivers of a pedestrian crossing, thereby enhancing pedestrian safety—even under low-visibility conditions such as at night, in fog, or during heavy rain. Pedestrians' mobile or wearable devices can be used to facilitate V2P communication. This functionality plays a crucial role in ensuring the overall safety of all individuals present near or on the roadway.[15]

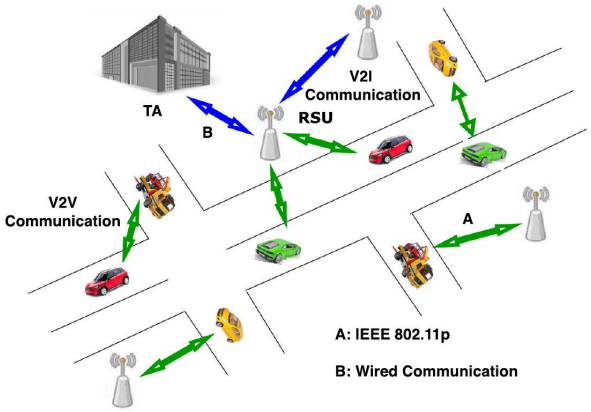

Figure 1.4: VANET's communication modes
[16]

1.6 Applications of VANETs

VANETs applications can be broadly classified into three main categories: safety applications, traffic efficiency applications, and infotainment or commercial applications. These applications aim to enhance road safety, optimize traffic flow, and provide comfort and convenience to drivers and passengers.

1.6.1 Safety Applications

Safety applications are the primary motivation for VANETs, focusing on reducing the number of traffic accidents and enhancing road safety. These applications provide real-time information to drivers about road conditions, nearby vehicles, and potential hazards. Examples include:

- **Collision Avoidance**: Vehicles exchange information about their speed, position, and direction to detect potential collisions and warn drivers in advance.
- **Emergency Warning Messages**: In case of sudden braking, accidents, or road obstacles, vehicles broadcast warning messages to nearby vehicles to prevent chain collisions.
- **Road Hazard Notifications**: Alerts about dangerous road conditions, such as slippery roads, fog, or construction zones, are disseminated to improve driver awareness.

1.6.2 Traffic Efficiency Applications

Traffic efficiency applications aim to optimize traffic flow, reduce congestion, and improve travel times. By enabling vehicles to share information about traffic conditions and road status, these applications help drivers make informed decisions. Key examples include:

- **Traffic Management**: Real-time traffic data is collected and shared to guide drivers toward less congested routes, reducing travel delays.
- **Route Optimization**: Vehicles receive updates on road conditions, traffic signals, and alternative routes to minimize travel time and fuel consumption.

1.6.3 Infotainment and Commercial Applications

Infotainment and commercial applications focus on enhancing driver and passenger comfort by providing non-safety-related services. These applications leverage VANETs to deliver location-based and context-aware information. Examples include:

- **Location-Based Services**: Information about nearby restaurants, petrol stations, parking spots, or tourist attractions is provided to drivers.

- **Internet Access**: Vehicles connect to roadside gateways or other vehicles to provide passengers with internet access for entertainment or work.
- **Advertising and Commercial Services**: Businesses broadcast advertisements or promotions to vehicles passing through specific areas, such as discounts at nearby stores.

Chapter 2

AI and its Role in ITS

2.1 Definition

AI, being a specialized field, is concerned with the creation of systems, which, when called upon, can perform tasks that normally require human cognitive processes. It is, as Russell and Norvig (2010) put it, "the study of agents that receive percepts from the environment and perform actions."

This field ranges over many fields including logic, probability, learning, reasoning, and perception. Further, it poses significant societal concerns such as fairness, trustworthiness, and safety. Its uses range from microelectronic devices[17].

2.2 Historical Overview of AI

To better understand the current role and potential of AI, it is essential to explore its historical development, particularly in connection with education and human cognition[18]:

2.2.1 Initial Establishment (1960s–1970s)

- AI was connected to educational study from inception through Seymour Papert, a founder of constructivist thought and co-author of the Logo programming language.
- Seymour Papert created a new programming language for children in 1966 in collaboration with Wally Feurzeig while at BBN. The new language was Logo, based on Lisp but simplified for educational use.

- In 1969, Papert and Cynthia Solomon left BBN and formed the Logo Group at the MIT Laboratory, continuing to develop the language and consider educational uses.

2.2.2 Symbolic AI and Education

Some early AI projects included:

- Natural language processing
- Creation of music, poetry, and other arts experiences
- Simple games such as Tic Tac Toe

In 1971, Papert & Solomon argued that AI should play an important role in computer-based learning.

In 1977, Ken Kahn described three roles of AI in education, including:

- As a tool for creativity for learners
- As an interactive environment using AI
- As a theoretical model for planning learning experiences

2.2.3 Prologue and Logic (1980s)

The Prolog programming language was introduced into schools, for example Bob Kowalski in 1978.

Two lines of development evolved:

- A simple syntax to support simple deductive reasoning
- Interactive environments to engage learners in understanding inference mechanisms.

Students were using Prolog to build semantic networks and investigate educational topics using logic-based explanations.

2.2.4 Transition to Machine Learning (2010–2020)

Beginning in 2012, the field of AI research changed to promote deep neural networks for applications as it reached significant milestones in:

- Machine translation.
- Image and speech recognition.
- Playing games such as Go, chess, and Atari.

  In 2017, a new generation of tools for children was released, notably:

- Machine Learning for Kids (Dale Lane)
- Google Teachable Machine
- Cognimates (MIT Media Lab)
- Snap ! with TensorFlow.js (Kahn & Winters)

2.3 AI: Domains and Applications

AI is among the key 21st-century technologies that is already disrupting the entire spectrum from health care to farming to education. The most important branches of AI and its multiple applications, the opportunities and challenges that come with them are discussed below.

2.3.1 Basic Areas of Artificial Intelligence

2.3.1.1 Machine Learning and Deep Learning

Machine Learning (ML) enables learning from data without human programming, such as supervised, unsupervised, and reinforcement learning. Deep Learning is a subfield of ML that uses artificial neural networks to handle complex data, used in the area of image identification and language understanding.[19]

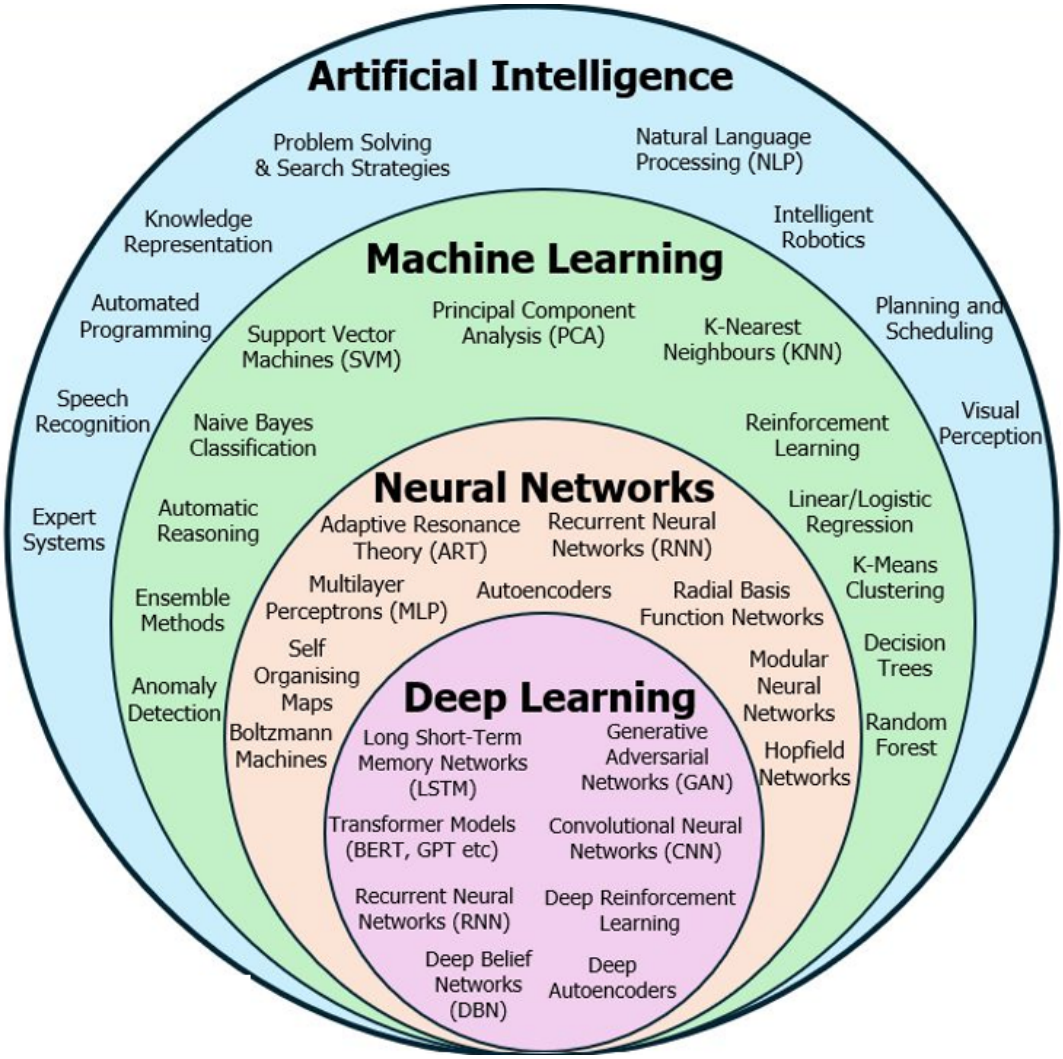

Figure 2.1: AI vs. Machine learning vs. Deep learning [20]

2.3.1.2 Natural Language Processing (NLP)

NLP enables machines to comprehend human language, and this is used to build applications like machine translation, sentiment analysis, and virtual assistants. Applications extensively use such models as GPT-4 and BERT. [21]

2.3.1.3 Computer Vision

Computer vision (CV) enhances ITS by processing visual data from cameras and sensors, improving safety and efficiency in vehicular networks like VANETs. Key applications include Automatic Number Plate Recognition (ANPR), traffic sign detection, vehicle/pedestrian detection, lane/obstacle detection, and anomaly detection for misbehavior in VANETs, using deep learning models like YOLO11 and Vision Transformers (ViTs) [22, 23]. Recent advance-

ments involve edge computing for real-time processing, self-supervised learning to reduce data needs, and 3D vision for autonomous navigation [24]. Challenges include adverse weather, privacy concerns, and computational demands, with future trends focusing on generative AI and multimodal models [25].

2.3.1.4 Robotics and Autonomous Systems

Robotics combines mechanical engineering and AI for the purpose of developing technologies that can operate within their environment, such as industrial robots, drones, and autonomous cars.

2.3.2 Practical Applications

2.3.2.1 Online Commerce and Personal Services

Recommendation engines process user activity to offer personalized recommendations, while chat-bots improve customer service.[21]

2.3.2.2 Cybersecurity

ML models are able to find the unusual in network behavior and effectively defend against cyber threats.

2.3.2.3 Healthcare

Algorithms crunch medical records to predict which patients will get sick, and robots assist with delicate surgeries.

2.3.2.4 Agriculture

AI-based tools help monitor the health of agriculture and predict the optimal planting time and conditions to drive efficiency and sustainability.

2.3.2.5 Financial Services

Corporations are relying on neural networks to detect fraud and to make better credit decisions, and in algorithmic trading.

2.3.2.6 Urban and Transport Planning

Automatic driving algorithms accelerate traffic, and reduce wear and tear on infrastructure.

2.4 Role of AI in ITS

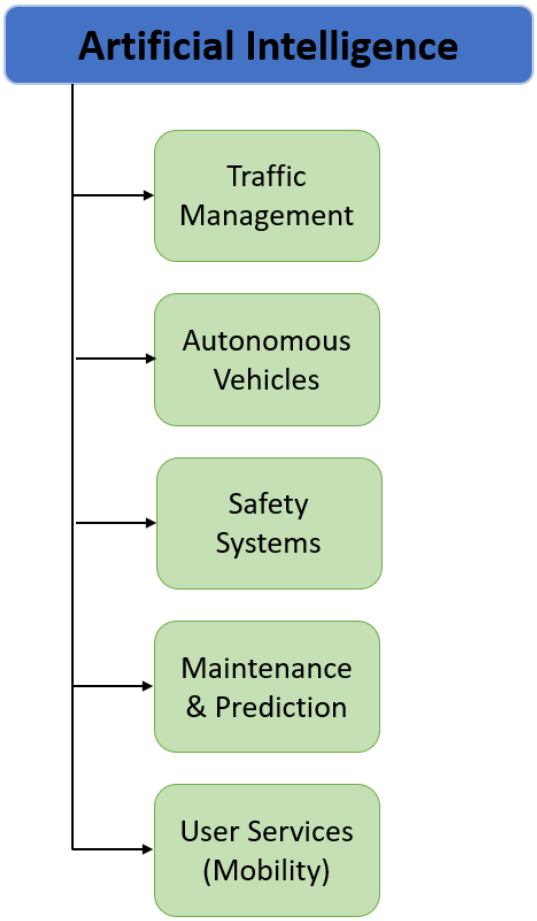

Figure 2.2: Role of AI in ITS

AI plays a foundational role in the construction and optimization of ITS. It enables adaptive, predictive, and autonomous functionalities that significantly enhance the efficiency, safety, and sustainability of transportation systems.

Figure 2.2 The diagram illustrates the main functional domains involved in Intelligent Transportation Systems.It highlights five core components: traffic management, autonomous vehicles, safety systems, predictive maintenance, and user services. Based on these components, this section presents three key areas in which Artificial Intelligence significantly contributes to ITS(Traffic Management and Optimization,Safety and Risk Detection,Anomaly Detection and Attack Prevention).

- AI serves as a central component in modern ITS development. It is crucial in traffic management, autonomous vehicle control, safety systems, predictive maintenance, and user services. These domains are interconnected through intelligent data analysis and decision-making capabilities enabled by AI technologies.
- A key application of AI is in traffic management. Intelligent systems process data from real-time sensors, GPS, and surveillance cameras to anticipate traffic volumes, optimize traffic light timings, and suggest alternative routes. These applications contribute to reducing travel time and air pollution—especially in densely congested urban areas.

- AI also drives the development of smart infrastructure. V2V and V2I communications help ensure coordinated movement, early warnings, hazard detection, and optimal use of road capacity.
- Moreover, personalized mobility services benefit from AI by leveraging user behavior and preferences to improve service delivery. This supports multi-modal route planning and fosters the design of innovative transport solutions.
- Reliable AI-driven analysis can be used for predictive diagnostics of vehicles and infrastructure components. It helps minimize downtime and costs associated with regular maintenance by identifying faults before they occur.
- AI greatly contributes to the improvement of safety in ITS. Smart systems can detect potentially dangerous behavior, obstacles, or pedestrians and trigger emergency responses without human intervention.
- Finally, AI supports the advancement of sustainable mobility. It optimizes fuel efficiency, enables electric vehicle management, and encourages eco-friendly driving behaviors, helping to reduce the environmental footprint of urban transportation.[26]

2.5 Machine Learning

Machine learning is a branch of Artificial Intelligence that focuses on developing models and algorithms that let computers learn from data without being explicitly programmed for every task. In simple words, ML teaches the systems to think and understand like humans by learning from the data.[27]

It can be broadly categorized into four types: Figure 2.3 shows the types of ML.

2.5.1 Categories of Machine Learning

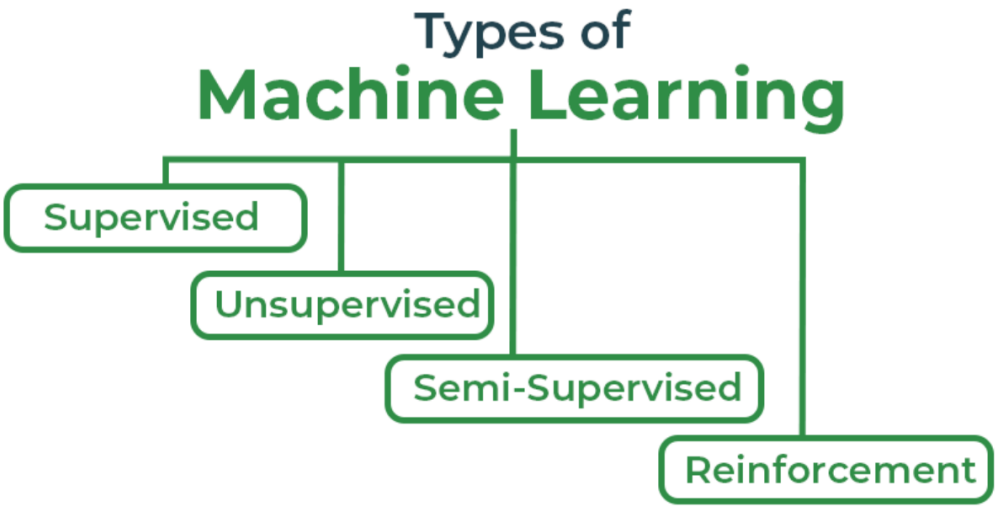

Figure 2.3: Types of Machine Learning [27]

2.5.1.1 Supervised Learning

Supervised learning is a branch of machine learning in which algorithms are trained using datasets where each input example is associated with a known output label. During training, the model makes predictions and is corrected by comparing them to the true labels, gradually adjusting its internal parameters to reduce errors. Once trained, the model can apply this learned mapping to infer labels for new, unseen data. Common applications include :

classification Where the output is a categorical variable (e.g., spam vs. non-spam emails, yes vs. no).

regression Where the output is a continuous variable (e.g., predicting house prices, stock prices).[28]

2.5.1.2 Unsupervised Learning

Unsupervised learning is a machine learning technique in which the algorithm is used on data that neither has labeled nor obvious output. The ultimate goal of exploratory data analysis is to find structures, patterns, groupings and trends in data. Instead of being trained with known targets, the model processes the input data and groups them according to similarities or statistical links. Such a strategy is usually considered for clustering, dimensionality reduction and pattern recognition purposes.[29]

2.5.1.3 Reinforcement Learning

Reinforcement learning is a machine learning paradigm where an agent learns to take decisions after interacting with the environment. The agent takes actions and observes rewards or punishments. It adapts its strategy over time to maximize its cumulative reward. Unsupervised learning is different from reinforcement learning because the latter does not need annotated input/output instances. Rather, the agent learns from, and is updated by, the consequences of its actions, which is particularly suited for sequential decision-making problems (e.g., robotics, game).[30]

- In the present work, we concentrate specifically on the classification task of supervised learning.the very component on which our research is built.

2.5.2 Common Classification Algorithms

Classification can be subdivided into several supervised learning approaches, each with distinct characteristics and use-cases. Below are some of the most widely adopted classification algorithms:The main algorithms used in this work are illustrated in Table 2.1 and are briefly described below[28].

Table 2.1: Overview of common classification algorithms

Algorithm	**Description**
Decision Tree	A decision tree is a tree-like structure that models decisions and their possible consequences. Internal nodes represent decision criteria, while terminal nodes denote the outcomes.
Random Forest	This method combines multiple decision trees trained on different subsets of data and features. The final prediction is typically based on the majority vote (classification) or average (regression) of all trees. Random Forests reduce overfitting and increase robustness.
Support Vector Machine (SVM)	The SVM algorithm finds the optimal hyperplane in an n-dimensional space that separates data into distinct classes. The data points used to define this hyperplane are called support vectors. SVMs are effective for high-dimensional and structured data.
K-Nearest Neighbors (KNN)	KNN classifies a new input by identifying the k most similar examples from the training dataset. The final prediction depends on the majority class or average value among these neighbors. Its accuracy is sensitive to the choice of k and the distance metric used.
Gradient Boosting	Gradient Boosting builds a strong predictive model by combining multiple weak learners, typically decision trees. Each new model is trained to correct the errors made by previous ones in a sequential and additive manner.

2.6 Deep Learning

2.6.1 Definition and Key Concepts

Deep learning models are computational structures trained by data scientists to perform specific tasks with minimal human intervention. These models consist of predefined sequences of algorithms that guide how data should be processed and interpreted. Through this training approach, deep learning systems are capable of identifying complex patterns in various data formats such as text, images, and audio.[31]

2.6.2 Deep Learning and Neural Networks

Deep learning uses a multi-layered structure of algorithms called neural networks[32]:

- **Input layer:** Data enters through the input layer.
- **Hidden layers:** Hidden layers process and transport data to other layers.
- **Output layer:** The final result or prediction is made in the output layer.

Figure 2.4 A general example illustrating how a deep learning model works through multiple interconnected layers.

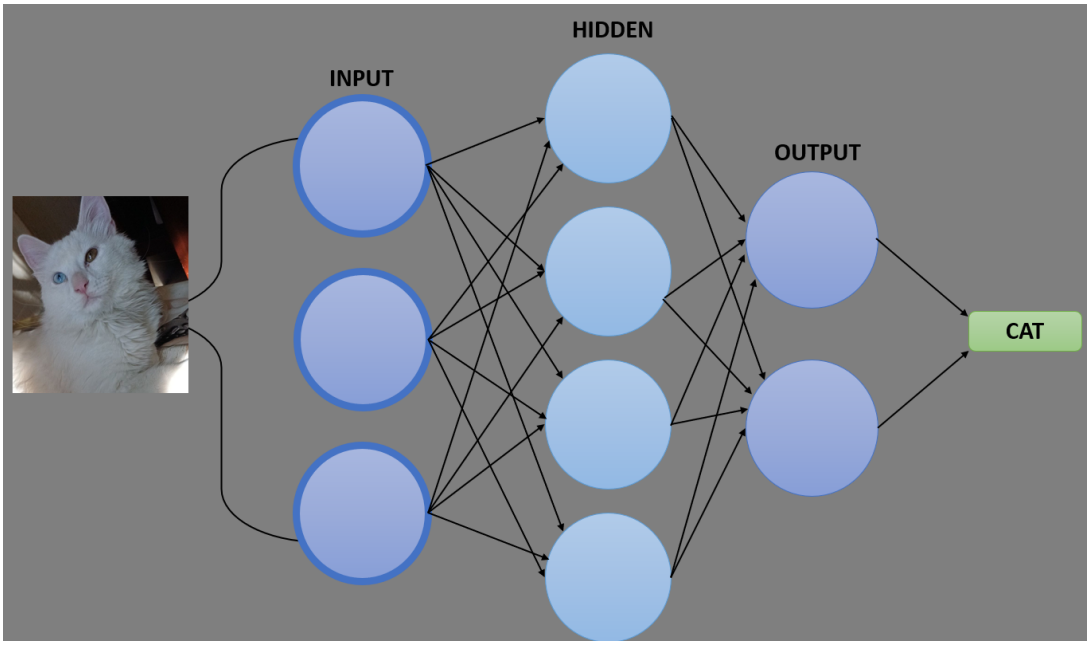

Figure 2.4: Layered Structure of a Deep Neural Network

2.7 Deep learning methods

Convolutional neural network (CNN) Recurrent neural network (RNN), De noising autoencoder (DAE), deep belief networks (DBNs), Long Short-Term Memory (LSTM) are the most popular deep learning methods have been widely used.[33]

- **Convolutional Neural Networks (CNNs)** are widely used for image processing and consist of convolutional, pooling, and fully connected layers, with training based on forward propagation and backpropagation.
- **recurrent Neural Networks (RNNs)** are designed to process sequential data like speech, handwriting, and text. They use cyclic connections that allow information to persist over time, storing previous inputs in hidden states to influence current outputs. Although RNNs are a newer deep learning technique, they are already applied in areas such as energy forecasting, hydrology, navigation, expert systems, and economics.
- **The Denoising AutoEncoder (DAE**) is a type of autoencoder designed to learn representations from noisy data. It consists of three layers: input, encoding, and decoding. An advanced version, the Stacked Denoising AutoEncoder (SDAE), enables nonlinear dimensionality reduction through a deep architecture and pre-training. DAE is increasingly applied in various fields such as energy forecasting, cybersecurity, finance, fraud detection, image classification, and speaker recognition.
- **Deep Belief Networks (DBNs)** are based on Restricted Boltzmann Machines, which are used to learn high-dimensional representations. Their applications include emotion detection, economic forecasting, medical image segmentation, and cancer diagnosis.
- **Long Short-Term Memory (LSTM)** is a variant of Recurrent Neural Networks designed to better manage long-term dependencies in sequential data. It is applied in fields such as geological modeling, air quality assessment, solar and wind energy forecasting, and seismic analysis.

2.8 Comparative Overview: ML vs DL in VANETs

ML and DL are two types of artificial intelligence techniques that are commonly used in VANETs. They differ mainly with respect to data needs, computational demands, model

complexities, and flexibility.

The typical machine learning only takes long for medium-sized datasets, but in addition requires manual feature selection. With advantages of quick training, low-cost resources and better interpretability, it is appropriate for embedded systems like on-board units in vehicles.

Deep learning, on the other hand, needs huge amounts of data (most of the time with no structure) and is based on advanced architectures like CNN or RNN/LSTM. These models can effectively extract the relevant features of the input on their own, and they perform really well on challenging tasks like image recognition, trajectory prediction, or online anomaly detection. But the application of traditional PKIs in VANETs is relatively difficult as the former impose great requirements on processing powers and storage space. To sum up ML is preferable for light weight and fast applications,while DL is more suitable for data-intensive scenarios that require deep analysis.[34]

2.9 Local and Regional Large Language Models (LLMs)

Several countries have recently initiated the development of localized large language models (LLMs) designed to align with their specific linguistic and cultural contexts. These efforts focus on enhancing AI systems' capabilities to understand and generate human language while incorporating regional dialects, expressions, and nuances. In Algeria, projects such as DziriBERT and DarjaBERT, tailored for Algerian Arabic (Darja), represent early advancements in this area.

2.9.1 DziriBERT

DziriBERT is a pre-trained language model developed to process Algerian dialectal Arabic. Built upon the BERT (Bidirectional Encoder Representations from Transformers) architecture, it was trained using Algerian Arabic text data collected from social media platforms, including Facebook and Twitter. The model was developed through a collaboration involving researchers from Meta AI and the Algerian research community. DziriBERT supports various natural language processing (NLP) tasks, such as:

- Text classification
- Sentiment analysis
- Named entity recognition

- Dialect identification

Although not as expansive as large-scale models like GPT-3, DziriBERT demonstrates the adaptability of transformer-based architectures for low-resource dialects like Algerian Darja [35].

2.9.2 DarjaBERT

DarjaBERT is another Algerian initiative focused on developing transformer models for Algerian spoken Arabic, particularly targeting informal dialectal texts found in social media and everyday conversations. It plays a significant role in:

- Chatbot development
- Informal language comprehension
- Transliteration tasks

While information on DarjaBERT remains limited, it is actively being researched and developed within Algerian universities as part of regional AI initiatives [36].

Chapter 3

Security in Vehicular Networks

3.1 Introduction

Security in VANETs is a fundamental pillar for ensuring the reliability, confidentiality, and integrity of communications between vehicles and infrastructure. With the increasing adoption of C-ITS, potential cybersecurity threats are also increasing. This chapter explores the specific security characteristics of VANETs, common types of threats, and the essential requirements needed to establish a secure and trustworthy environment.

3.2 Security Characteristics

To ensure trust and reliability in VANETs, various security requirements must be addressed during the design and implementation of communication protocols and systems. These requirements are essential for protecting the integrity, confidentiality, and availability of exchanged messages between vehicles V2V and V2I. The following are the key security requirements in VANETs:[37]

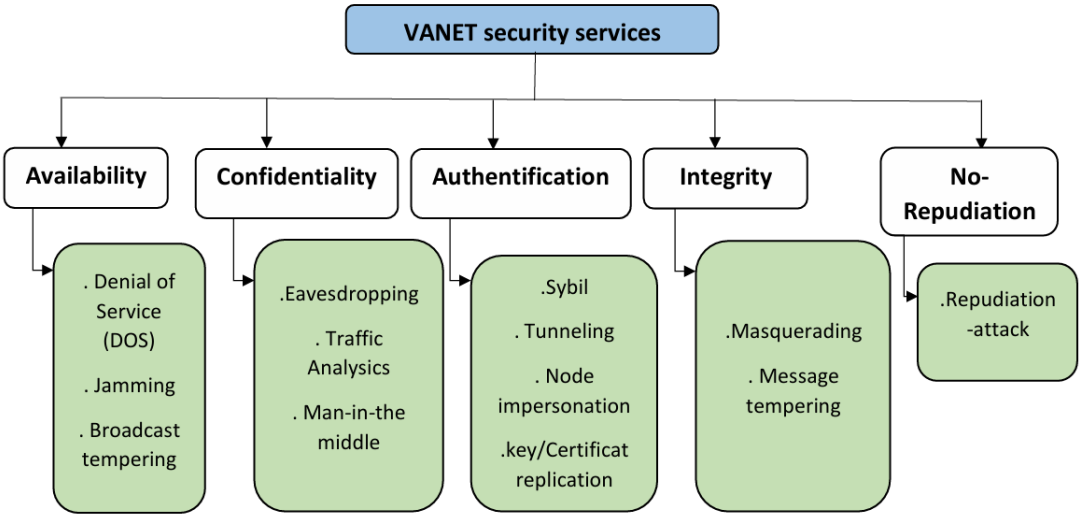

Figure 3.1: VANET security requirements

[38]

3.2.1 Authentication

Authentication ensures that entities participating in the network are truly who they claim to be. It allows nodes to verify the legitimacy of other nodes before accepting or acting on received messages. Two types of authentication are commonly considered: message authentication, which ensures that the source of a message can be traced, and entity authentication, which identifies the sender as a valid participant in the network. Authentication is critical for maintaining cooperation among vehicles and preventing impersonation attacks.

3.2.2 Confidentiality

Confidentiality refers to the protection of sensitive data from unauthorized access. Only authorized entities should be able to access and interpret messages transmitted over the network. This is typically achieved through symmetric or asymmetric encryption algorithms. In the context of VANETs, confidentiality helps prevent adversaries from tracking vehicle locations or eavesdropping on private communications.

3.2.3 Availability

Availability ensures that authorized users have continuous and timely access to network services. This requirement is vital for traffic management applications and safety-related services

that rely on real-time information. Maintaining availability requires robust communication infrastructure and protocols capable of withstanding Denial-of-Service (DoS) attacks or other disruptions. High availability contributes directly to the efficiency and reliability of intelligent transportation systems.

3.2.4 Integrity

Integrity guarantees that messages have not been altered during transmission, either accidentally or intentionally. Ensuring data integrity involves detecting any unauthorized modification of data, which can be achieved using cryptographic hash functions or digital signatures. In wireless environments such as VANETs, data integrity can also be challenged by signal interference or transmission errors, which must be distinguished from malicious tampering.

3.2.5 Non-repudiation

Non-repudiation ensures that the origin of a message cannot deny having sent it. This requirement is crucial for accountability in vehicular communications. It allows for the identification of malicious actors and supports evidence gathering in case of disputes or legal investigations. Digital signatures are often used to achieve non-repudiation, particularly in safety-critical messages and traffic regulation enforcement scenarios.

3.3 Security Threats

VANETs are based on an open and decentralized architecture, which makes them particularly vulnerable to a wide range of security threats. The continuous exchange of critical data such as vehicle position, speed, and trajectory creates opportunities for malicious actors to exploit the network. These attacks may disrupt communication, compromise sensitive information, or alter system behavior in ways that pose safety risks to road users. This section outlines the main categories of attacks identified in the literature, with a focus on their impact on the core security requirements: confidentiality, integrity, availability, and non-repudiation [14].

Figure 3.2 :The diagram shows a malicious vehicle attempting to compromise the network through specific threats, such as eavesdropping on wireless communications, launching Dos attacks against the TA, or conducting linkability attacks to trace a vehicle's identity. Forward and backward attacks are also shown, indicating message manipulation between RSUs and

vehicles. This visual representation underscores the need for a multi-layered security strategy capable of defending each element of the system against diverse and coordinated threats.

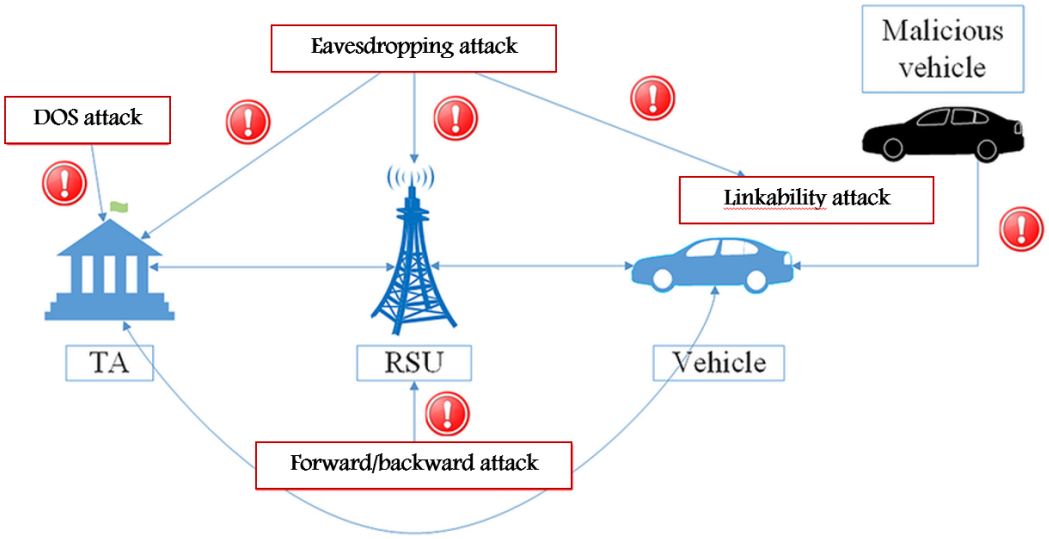

Figure 3.2: Threat model in a VANET [39].

3.3.1 Denial of Service (DoS)

DoS attacks aim to exclude legitimate users from network services by flooding the network with false or useless traffic. In VANETs, this may disrupt crucial applications like collision avoidance and emergency warnings by jamming channels or overwhelming them with bogus data. These attacks compromise system availability and may cause severe traffic congestion or accidents. Countermeasures include frequency hopping, channel switching, and intrusion detection systems.[40]

3.3.2 Sybil Attacks

In a Sybil attack, a malicious node generates multiple fake identities to deceive the system. This can disrupt traffic control by injecting false data, such as fake congestion alerts. Because of VANETs' decentralized and open nature, they are particularly vulnerable to this attack. Detection strategies include signal strength analysis and privacy-preserving pseudonym verification.[40]

3.3.3 Eavesdropping

This involves the unauthorized interception of messages. Attackers may passively monitor vehicle communications to extract sensitive data such as location, speed, or driver identity. This compromises confidentiality and poses serious privacy threats. Encryption using symmetric or asymmetric keys is a basic countermeasure.[40]

3.3.4 Linkability attack

Even if pseudonyms are changed regularly, linkability attacks aim to associate multiple messages with the same vehicle. This compromises user anonymity. Defenses include pseudonym management and analyzing signal patterns to detect correlations. [40]

3.3.5 Forward/Backward Attack

In this type of attack, adversaries relay or replay messages between vehicles and infrastructure (e.g., OBUs and RSUs) to bypass authentication or introduce false data. This threatens both the integrity and consistency of the system. Strong authentication protocols and secured message exchange mechanisms are necessary to mitigate this risk. [40]

3.4 Security requirements

In VANETs, ensuring a high level of security is essential due to the critical nature of the data exchanged and the real-time constraints of vehicular communication. The dynamic topology, high mobility, and open communication environment expose these networks to numerous vulnerabilities. To address these challenges, it is necessary to implement well-defined security requirements that translate core security principles into practical mechanisms. These requirements are designed to uphold properties such as confidentiality, integrity, authentication, availability, and non-repudiation across all network layers. This section presents the key components required to enforce these principles effectively within VANETs, including the use of digital certificates, cryptographic techniques, secure communication protocols, and behavior-based verification mechanisms.[41]

3.4.1 Digital Certificates and Public Key Infrastructure (PKI)

To guarantee authentication and trust within vehicular networks, a robust identity management system is essential. The implementation of a Public Key Infrastructure (PKI) enables the distribution and verification of digital certificates to all participating entities, such as vehicles, roadside units, and control centers. These certificates serve as proof of identity and authorization, allowing secure message signing and verification. With PKI, each message can be traced to a verified sender, ensuring that only legitimate actors contribute to network communication. Additionally, certificate revocation mechanisms can be established to exclude compromised nodes and maintain the integrity of the network. Figure 3.3: How a PKI system works in a VANET scenario. Each vehicle who generate two asymmetric keys and obtain digital certificate from CA. When a vehicle sends a digitally signed message, a CA (or Verification Authority) checks the signature to guarantee the integrity and origin of the data, and thus the secure exchange of data among vehicles.

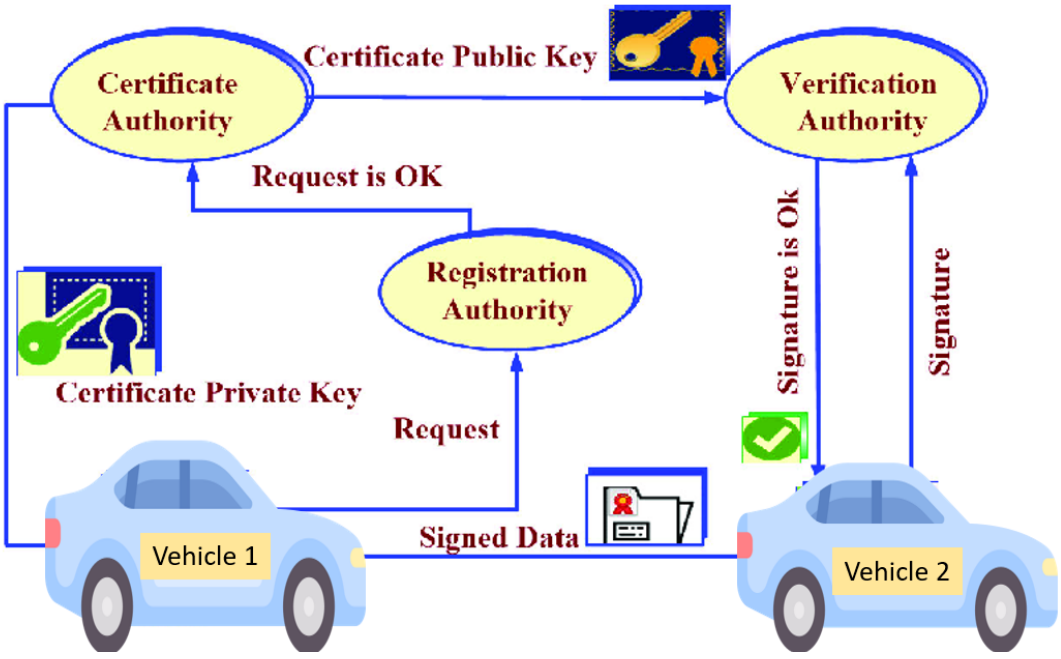

Figure 3.3: How a PKI system works in a VANET scenario

[42]

3.4.2 Cryptographic Algorithms

The protection of vehicular communication relies heavily on cryptographic algorithms capable of securing data with minimal computational delay. Asymmetric cryptographic schemes,

such as Elliptic Curve Cryptography (ECC), are well-suited to vehicular environments due to their efficiency and strong security. Digital signatures ensure message integrity and authentication, while encryption techniques protect the confidentiality of transmitted data. These algorithms prevent unauthorized access, tampering, and interception, thereby strengthening the resilience of the network against a wide range of attacks, including eavesdropping and message spoofing.

Figure 3.4: Cryptography

3.4.3 Secure Communication Protocols

Reliable and secure communication in VANETs depends on the integration of security features at the protocol level. Communication frameworks must incorporate mechanisms for mutual authentication, encryption, and secure session management to ensure data confidentiality and integrity. Whether based on Dedicated Short-Range Communications (DSRC), cellular networks, or hybrid architectures, protocols should support secure data exchange between vehicles (V2V) and between vehicles and infrastructure (V2I). Moreover, secure channels must be maintained between infrastructure nodes to protect management data and prevent unauthorized control of the system.

3.4.4 Intrusion Detection Mechanisms

To complement preventive security measures, detection mechanisms must be implemented to identify abnormal or malicious behavior in real time. These mechanisms may include validation of vehicle positions, verification of message plausibility, and consistency checks based on expected behavior. By monitoring data patterns and verifying message authenticity against trusted credentials, the system can detect anomalies such as false position reporting, identity spoofing, or repeated message injection. These techniques provide an additional layer of defense, helping to maintain trust and stability within the network even in the presence of internal threats.

3.5 Conclusion

The security of Vanets plays a decisive role in ensuring the effectiveness and reliability of connected transportation systems. Throughout this chapter, the core security principles required to protect data exchanges in VANETs have been defined and contextualized. Practical implementation methods ranging from digital certification and cryptographic mechanisms to secure communication protocols and anomaly detection techniques have been outlined to demonstrate how these principles can be enforced in real vehicular environments. Moreover, the chapter has identified several prominent threats that compromise network integrity, availability, and user privacy. Understanding these challenges is essential for developing resilient architectures capable of responding to both known and evolving risks. The following chapter addresses one such critical threat in depth: the false position attack, and proposes a detection model based on machine learning approaches.

Chapter 4

False Position Attack

4.1 Introduction

VANETs rely heavily on the accuracy of location data to ensure the reliability of safety-critical applications, such as avoiding accidents and emergency messages. However, the open and dynamic nature of VANETs makes them susceptible to various cyber threats, one of which is the False Position Attack. This attack involves a malicious node broadcasting falsified location information, which can mislead other vehicles and disrupt network operations. The consequences of such attacks are particularly dangerous in VANETs, where inaccurate positioning can lead to hazardous situations, including traffic accidents or inefficient routing. This chapter aims to explore the nature of False Position Attacks within the context of VANETs, examining their characteristics, impacts, and potential solutions to enhance network security [43].

4.2 Background and Related Work

4.2.1 Overview of False Position Attacks

Position falsification attacks, also known as False Position Attacks, are a major issue in Vehicular Ad-hoc Networks (VANETs). These attacks happen when a malicious vehicle lies about its location to trick other vehicles in the network. This can disrupt services that depend on accurate location data, such as navigation or crash prevention systems [43]. For example, a vehicle might claim it is at a busy intersection when it is actually on a quiet road, causing confusion for other drivers who rely on this information.

Research has identified different types of position falsification attacks. In a *constant position attack*, a vehicle keeps broadcasting the same location, even if it moves. In a *random position attack*, the vehicle sends completely random locations, making its movements unpredictable [43]. These attacks are dangerous because VANETs use Cooperative Awareness Messages (CAMs) short messages that share a vehicle's position, speed, and direction—to make real-time decisions like avoiding collisions [44]. If the position data in these messages is fake, the whole system can fail.

4.2.2 Historical Context and Evolution

The problem of position falsification in VANETs has grown over time as vehicle communication technology has advanced. Early studies, like Grover et al. (2011), pointed out that attackers could use *GNSS spoofing* (tricking GPS systems) to fake a vehicle's location [45]. Over the years, attackers have become more advanced, using tools like software-defined radios (SDRs) to send fake signals with high accuracy [46]. This evolution makes the attacks harder to detect and stop.

The idea of faking locations originally came from military tactics, where GPS jamming was used to confuse enemies [47]. Now, these techniques have been adapted for civilian use in VANETs, creating new risks. For instance, the rise of the Internet of Vehicles (IoV)—where vehicles are connected to the internet—has made it easier for attackers to target multiple vehicles at once, spreading false position data across a wider area [48].

4.2.3 Related Attacks and Research Efforts

False Position Attacks are often linked to other threats in VANETs. For example, a *Sybil attack* involves creating fake identities, and when combined with position falsification, an attacker can pretend to be many vehicles in different locations, making the attack even worse [49]. Researchers have studied how these attacks spread in different settings, such as busy cities or highways, using simulations to understand their impact [50].

Recent studies have also looked at why attackers launch these attacks. Some might want to cause chaos, like redirecting traffic to create delays, while others might have financial motives, such as manipulating traffic data to affect toll systems or logistics [51]. Despite many studies, there is still no single solution that works for all situations in VANETs. The network's decentralized nature—where there is no central authority to monitor everything—makes it hard

to create a universal defense strategy [48]. This section shows that False Position Attacks are a complex and evolving problem, requiring ongoing research to protect VANETs effectively.

4.2.4 Attacker Capabilities

Understanding the capabilities of attackers in Vehicular Ad Hoc Networks (VANETs) is critical for developing effective misbehavior detection systems, particularly for position falsification attacks [49]. Attackers exploit VANETs' dynamic topology, wireless communication, and distributed nature to compromise safety-critical operations. The following outlines key attacker capabilities, drawing on recent classifications and their implications for network security [52]:

- **Message Impersonation**: Attackers manipulate safety messages, such as Basic Safety Messages (BSMs), to broadcast falsified data, including fake GPS coordinates in Random Offset or Constant Position Attacks. By impersonating legitimate vehicles, adversaries deceive nodes into accepting malicious messages, undermining authenticity [52]. Machine learning models, like XGBoost, detect these attacks by analyzing inconsistencies in message content and vehicle trajectories [53].

- **Generating Fake Identities**: Through Sybil attacks, attackers create multiple fake identities to amplify malicious broadcasts, simulating dense traffic or false road events. This capability exploits VANETs' lack of centralized control, enabling adversaries to overwhelm legitimate nodes [49]. Advanced detection systems, trained on datasets like VeReMi, identify such anomalies by cross-referencing movement patterns across identities [53].

- **Disrupting Network Availability**: Attackers launch availability attacks, such as black hole attacks, by dropping packets or redirecting traffic, disrupting ad hoc routing protocols like AODV. These actions, prevalent in Eventual Stop Attacks, create false congestion signals, degrading network performance [52]. Recent studies emphasize the need for secure routing protocols to counter these disruptions [54].

- **Exploiting Insider Privileges**: Insider attackers with valid certificates pose significant threats by sending authenticated yet false data, such as bogus road event alerts. These attacks bypass cryptographic defenses, requiring behavioral analysis for detection [49]. Machine learning approaches, as utilized in misbehavior detection systems, identify outliers in message delivery patterns to mitigate such threats.

- **Leveraging Computational Resources and Mobility**: Attackers use computational resources to execute sophisticated attacks, including real-time message forging, while exploiting VANETs' high mobility to launch transient attacks. Rapid topology changes enable adversaries to evade tracking, complicating defense mechanisms [52]. Dynamic trajectory modeling, integrated into detection systems, counters this by leveraging predictable mobility patterns [55].

4.3 Characteristics of False Position Attack in VANETs

4.3.1 How the Attack Works

False position attacks in Vehicular Ad Hoc Networks (VANETs) involve malicious vehicles broadcasting incorrect location data to deceive other nodes, compromising the reliability of Cooperative Intelligent Transport Systems (C-ITS). These attacks exploit the critical role of accurate positioning in safety and traffic management applications, leading to potential hazards such as collisions or traffic disruptions [53]. This section outlines the mechanisms of five distinct position falsification attacks, as identified in our study, each with unique characteristics and deceptive strategies, as shown in table 4.1:

Table 4.1: Different attack types and their parameters.

Attack ID	Attack Name	Parameters
1	Constant	$x = 5560$, $y = 5820$
2	Constant Offset	$\Delta x = 250$, $\Delta y = -150$
4	Random	Uniformly random in the simulation area
8	Random Offset	Δx, Δy uniformly random from $[-300, 300]$
16	Eventual Stop	Stop probability increases by 0.025 per update (10 Hz)

1. **Constant Position Attack**: In this attack, a malicious vehicle repeatedly transmits a fixed, pre-configured location, such as coordinates (5560, 5820), regardless of its actual movement. This static position can mislead nearby vehicles into perceiving the attacker as stationary at a specific point, potentially causing incorrect routing decisions or false hazard warnings.

2. **Constant Offset Position Attack**: Here, the attacker adds a fixed offset (e.g., $\Delta x = 250$,

$\Delta y = -150$) to its actual position coordinates. This creates the illusion of consistent movement along a shifted trajectory, such as changing lanes or deviating from the true path. The attack aims to obscure the vehicle's real location while maintaining a plausible but false presence on the road.

3. **Random Position Attack**: The attacker generates and broadcasts entirely random coordinates within the vehicular network's operational area, disregarding its true position. This erratic behavior can disrupt traffic flow by introducing unpredictable location data, confusing navigation systems and leading to inefficient or unsafe routing.

4. **Random Offset Attack**: Similar to the random position attack, this variant involves adding a random offset (e.g., Δx, Δy uniformly sampled from $[-300, 300]$) to the vehicle's actual coordinates. The offset is confined to a rectangular region around the vehicle, creating a more subtle deception compared to fully random positions. This attack simulates minor deviations, making detection challenging due to its proximity to legitimate movement patterns.

5. **Eventual Stop Attack**: In this sophisticated attack, the malicious vehicle behaves normally for a period, broadcasting accurate position data, before suddenly halting updates and repeatedly transmitting its last known position. With a stop probability increasing by 0.025 per second at a 10 Hz update rate, the attacker mimics a vehicle that has stopped, potentially causing other vehicles to misjudge road conditions or anticipate non-existent obstacles.

4.3.2 Goals and Immediate Impacts

The primary objective of false position attacks in Vehicular Ad Hoc Networks (VANETs) is to disrupt or manipulate the functionality of critical applications, such as navigation, traffic management, and safety systems like collision avoidance. By broadcasting falsified location data, attackers aim to undermine the trust and reliability of Cooperative Intelligent Transport Systems (C-ITS), leading to immediate operational, safety, and economic consequences [53]. The following outlines key impacts, illustrated with real-world simulations and experimental examples from recent studies.

- **Fake Traffic Jams**: Attackers broadcasting false positions can simulate traffic congestion,

prompting navigation systems to reroute vehicles unnecessarily. For instance, in simulations using the VeReMi dataset, a vehicle executing a Constant Position Attack (e.g., reporting a fixed location like (5560, 5820)) caused nearby vehicles to detour, increasing travel time by up to 15% and fuel consumption by 8% [53]. In a 2024 study, Random Offset Attacks were shown to disrupt traffic flow in urban scenarios, leading to a 10% reduction in throughput for vehicle-to-vehicle (V2V) communication, as vehicles avoided non-existent congested zones [54]. Such disruptions not only waste resources but also exacerbate environmental impacts through increased emissions.

- **False Accidents**: By falsely reporting a crash, attackers can trigger unwarranted emergency responses. A simulated Eventual Stop Attack in the VeReMi dataset, where a vehicle halts position updates to mimic a stationary incident, prompted erroneous emergency alerts in 62% of test cases, diverting resources like ambulances to incorrect locations [53]. In a 2023 experiment, researchers demonstrated that a malicious vehicle broadcasting a fake collision on a highway segment caused a 20-minute delay in traffic management systems, as roadside units (RSUs) relayed false alerts to authorities [56]. These incidents highlight the potential for resource misallocation and public safety risks.

- **Safety Risks**: Collision avoidance systems depend on accurate location data, and false positions can induce hazardous reactions. In a 2020 study, a Random Position Attack in a VANET testbed caused a vehicle to misjudge an attacker's location, leading to unnecessary braking and a 30% increased likelihood of rear-end collisions. Similarly, a Constant Offset Attack (e.g., adding a fixed offset like $\Delta x = 300$, $\Delta y = -200$) in a simulated urban environment misled adaptive cruise control systems, resulting in abrupt speed changes in 45% of vehicles [54]. These scenarios underscore the critical safety threats posed by falsified data, potentially causing real accidents.

- **Economic and Operational Impacts**: Beyond immediate safety concerns, false position attacks can incur significant economic costs. A 2021 study noted that fake traffic jams induced by position falsification increased fleet operational costs by 5–7% due to longer routes and delayed deliveries [40]. Additionally, repeated false accident reports in simulated VANETs strained emergency response budgets, with one case estimating a $10,000 loss per incident due to misdeployed resources [56]. These impacts highlight the broader economic ramifications for municipalities and logistics sectors.

4.3.3 Long-Term Impacts and Challenges

False Position Attacks in Vehicular Ad Hoc Networks (VANETs) pose significant long-term challenges that extend beyond immediate disruptions. By broadcasting falsified location data, these attacks undermine the reliability of intelligent transportation systems (ITS), threatening their adoption and effectiveness [57]. The following outlines the key long-term impacts and challenges:

- **Trust Erosion in VANET Systems**: Persistent exposure to falsified position data erodes drivers' trust in VANET systems, leading them to disregard critical safety features such as collision warnings or traffic coordination. A 2023 study demonstrated that repeated false alerts reduced driver compliance with VANET-based warnings by up to 25% in simulated urban scenarios, significantly diminishing the benefits of ITS and hindering advancements in road safety and efficiency [56].

- **Economic and Environmental Impacts**: False position data misleads navigation systems, causing vehicles to take unnecessary detours, which increases fuel consumption and operational costs for drivers. A 2024 analysis using the VeReMi dataset showed that Constant Position Attacks resulted in an 8% increase in fuel usage and a 15% increase in travel time for affected vehicles. This higher fuel consumption also contributes to elevated carbon emissions, exacerbating environmental concerns in urban areas [53, 40].

- **Legal and Accountability Concerns**: False Position Attacks create significant ambiguity in determining liability for accidents caused by falsified data, raising the question of who is responsible—the attacker, vehicle manufacturer, or network operator. These issues remain unresolved in ongoing legal and policy discussions. A 2022 survey highlighted that unclear accountability frameworks deter investment in VANET infrastructure due to potential litigation risks, further complicating the deployment of secure ITS [58, 16].

- **Scalability and Detection Challenges**: The high mobility of vehicles in VANETs enables the rapid dissemination of false position data, amplifying the attack's impact across a large area. For instance, a 2024 study showed that Random Offset Attacks reduced vehicle-to-vehicle (V2V) communication throughput by 10%, as a single malicious vehicle on a highway affected hundreds of others within minutes. The decentralized nature of VANETs, lacking a central authority, complicates real-time attacker identification and

mitigation. Recent research emphasizes machine learning-based misbehavior detection systems (MDS), which achieve detection accuracies up to 99% using features like traffic flow and position differences, as a promising approach to address these challenges [59, 54, 60].

4.4 Machine Learning Algorithms for Misbehavior Detection

This section elaborates on the machine learning algorithms that form the core of the detection methodologies discussed and employed within this research for identifying misbehavior, such as position falsification, in vehicular ad-hoc networks (VANETs). The algorithms selected for their proven efficacy and relevance to this domain are Random Forest (RF), Support Vector Machine (SVM), and Extreme Gradient Boosting (XGBoost).

4.4.1 Random Forest (RF)

Random Forest (RF) stands as a versatile and widely adopted ensemble learning method, adept at handling tasks ranging from classification to regression [61]. The fundamental principle of RF involves the construction of a multitude of decision trees during the model's training phase. For classification tasks, the final output is determined by a majority vote among the individual trees, while for regression, it is the average of their predictions . A significant advantage of Random Forests is their inherent ability to mitigate the risk of overfitting, a common pitfall with single decision trees. This is achieved by injecting randomness into the tree-building process—specifically, by considering only a random subset of features when determining splits for each node, and by utilizing bagging (bootstrap aggregation) to create diverse training datasets for each tree from the original data . Consequently, even as more trees are added to the ensemble, the model tends not to overfit, a resilience attributed to the Law of Large Numbers [61].

RF have consistently demonstrated high accuracy in both classification and regression scenarios, often rivaling or even surpassing other ensemble techniques like boosting. Their robustness extends to handling noisy data and outliers effectively. Moreover, RF models are relatively faster to train compared to some other complex ensemble methods and offer valuable byproducts, such as internal estimates of generalization error, feature importance, data strength, and inter-feature correlations, providing deeper insights into the data structure [61].

4.4.2 Support Vector Machine (SVM)

The Support Vector Machine (SVM), an influential supervised learning algorithm introduced by Vapnik, is renowned for its proficiency in both classification and regression problems . SVMs are particularly lauded for their robust performance and effectiveness, especially when dealing with high-dimensional data spaces where the number of features can be substantial . The central tenet of SVM is the identification of an optimal hyperplane that creates the largest possible margin—or distance—between distinct classes of data points. The data points that lie closest to this separating hyperplane are termed "support vectors," and they are critical as they uniquely define the position and orientation of this optimal boundary [62].

In scenarios where data is linearly separable, SVM can delineate classes using a straightforward linear hyperplane. However, for more complex, non-linearly separable datasets, SVM employs a sophisticated technique known as the "kernel trick." By utilizing various kernel functions (such as the Radial Basis Function (RBF), polynomial, or sigmoid kernels), SVM implicitly maps the input data into a higher-dimensional feature space where linear separation becomes feasible, without the computational burden of explicit transformation . This capability makes SVM exceptionally well-suited for detecting misbehavior in VANETs, as these networks generate vast amounts of high-dimensional data encompassing vehicle kinematics (speed, location), communication patterns, and message authenticity, all of which SVMs can process effectively [62].

Compared to deep learning models that typically demand extensive datasets for training, SVMs are less prone to overfitting and can perform admirably even with moderately sized datasets . Their capacity to model non-linear relationships through kernel functions allows them to uncover intricate misbehavior patterns, such as sophisticated packet injection schemes or nuanced false location broadcasts, which might be missed by traditional rule-based detection mechanisms . A further practical advantage of SVMs is their efficiency during the prediction phase; once trained, an SVM model can classify new instances of behavior rapidly, making it a viable candidate for real-time security applications in dynamic VANET environments [62]. Despite these strengths, SVMs do present certain challenges, notably in terms of computational demands during training, especially with very large datasets where time complexity can escalate. Additionally, the performance of an SVM is highly contingent on the judicious selection of the kernel function and the meticulous tuning of its hyperparameters (like the regularization

parameter 'C' and kernel-specific parameters such as 'gamma'), which often requires careful experimentation to achieve an optimal balance between model accuracy and generalization [62].

4.4.3 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost) is a powerful and highly optimized machine learning algorithm, specifically engineered for speed, scalability, and performance on structured or tabular data Ḋeveloped by Tianqi Chen and Carlos Guestrin, XGBoost is an advanced implementation of gradient boosted decision trees (GBDTs), augmented with several significant algorithmic enhancements and system optimizations . Key among these are its capabilities for parallel processing (allowing for faster computation by building parts of trees or multiple trees concurrently where feasible), out-of-core computation (enabling it to handle datasets that exceed available system memory by processing data in disk-based blocks), and sparsity-aware learning (allowing it to intrinsically manage missing values or sparse features without requiring prior imputation) [63].

A distinguishing feature of XGBoost is its incorporation of a regularized objective function. This function penalizes model complexity (e.g., the number of leaves or the depth of the trees), thereby promoting simpler models that are less susceptible to overfitting the training data and thus generalize better to unseen data . For determining optimal split points in decision trees, especially for continuous features, XGBoost leverages sophisticated techniques like weighted quantile sketches, leading to more accurate splits . Furthermore, its architecture supports distributed computing, making it well-suited for tackling very large-scale datasets efficiently [63].

Owing to its exceptional predictive accuracy, computational efficiency, and operational flexibility, XGBoost has become a dominant algorithm in machine learning competitions and has found widespread application in diverse domains, including fraud detection, sophisticated recommendation systems, and critical anomaly detection . Its remarkable ability to process massive datasets efficiently while consistently delivering high-accuracy results has solidified its reputation as one of the most effective and popular algorithms in contemporary applied machine learning [63]. XGBoost also implements system-level optimizations such as cache-aware access patterns and block structures for tree construction to further enhance its computational speed.

4.5 Detection and Preventing Techniques

4.5.1 Addressing False Position Attacks

Mitigating False Position Attacks in Vehicular Ad Hoc Networks (VANETs) requires robust detection and prevention strategies tailored to the dynamic and decentralized nature of these networks. By leveraging advanced verification techniques, machine learning, and cooperative mechanisms, the security of VANETs can be significantly enhanced [57]. The following strategies outline effective approaches to counter these attacks:

- **Position Verification Techniques**: Verifying the accuracy of reported vehicle positions is critical to detecting falsified data. Consistency checks analyze the plausibility of a vehicle's trajectory by cross-referencing reported positions with expected mobility patterns, such as speed and direction, derived from neighboring nodes. A 2023 study demonstrated that trajectory-based verification using Kalman filtering achieved a detection accuracy of 92% for Constant Position Attacks in simulated urban VANETs, significantly reducing the impact of false data [60].

- **Cooperative Detection Mechanisms**: Collaborative validation among vehicles enhances detection accuracy by leveraging shared observations to cross-check position data. Cooperative misbehavior detection systems (MDS) allow vehicles to share real-time information, reducing false positives and improving robustness. A 2022 study showed that cooperative MDS frameworks achieved a 90% reduction in false positives compared to standalone detection, making them effective against sophisticated attacks like Eventual Stop Attacks [16].

- **Certificate Revocation and Privacy Measures**: Mitigating the impact of detected attackers involves isolating them from the network. The Trusted Authority (TA) can revoke certificates of malicious nodes, preventing further participation. Additionally, implementing short-term pseudonyms and periodic key updates limits the attacker's ability to sustain deception. A 2024 analysis noted that pseudonym rotation every 5 minutes reduced the success rate of False Position Attacks by 70% in high-density VANETs, enhancing overall network security [64].

- **Machine Learning-Based Detection:** Machine learning algorithms offer promising

solutions for identifying position falsification by analyzing multiple features, such as signal strength, velocity, and movement patterns. For instance, XGBoost models have been shown to detect anomalies with up to 95% accuracy in real-time VANET scenarios by identifying deviations indicative of attacks like Random Offset Attacks. [54].

4.6 Conclusion

False Position Attacks pose a significant threat to the reliability and safety of VANETs by exploiting the critical dependency on accurate location data. This chapter has provided a comprehensive overview of the attack, detailing its characteristics and impacts on vehicular networks. The discussion of detection and mitigation techniques underscores the importance of integrating advanced methods, such as machine learning and cooperative verification, to address this threat effectively. By understanding the nature of False Position Attacks and implementing robust countermeasures, VANETs can better ensure the integrity of their communication systems, paving the way for safer and more efficient intelligent transportation systems.

Chapter 5

Contribution

5.1 Introduction

This chapter describes the experimental part of our work. It includes the dataset used, the types of attacks considered, the preprocessing steps applied to the data, the machine learning algorithms selected, the development environment, and the training procedure.
The experiments were conducted using **Python** version 3.11.11. The development environment utilized was a **Kaggle** kernel, equipped with an **Intel(R) Xeon(R) CPU @ 2.00GHz** (x86_64 architecture).
Key Python libraries employed for data manipulation, analysis, model implementation, and evaluation included **NumPy** for numerical operations, **Pandas** for data handling, **Matplotlib** and **Seaborn** for data visualization, and **scikit-learn** (sklearn) for its comprehensive suite of machine learning tools, including model selection, preprocessing utilities, and metrics.

5.2 Dataset and Attack Scenarios

This study uses the *VeReMi* dataset[1], a widely used benchmark for evaluating misbehavior detection approaches in vehicular networks. The dataset was generated using two simulation tools: *SUMO*, which simulates realistic vehicular mobility, and *OMNeT++*, which models communication between vehicles and its inclusion of diverse and relevant position falsification attack scenarios [43]. The messages exchanged follow the format of *Cooperative Awareness Messages* (CAMs), which are periodically broadcast by vehicles and include information such

[1] https://www.kaggle.com/datasets/nasiruddinthander/vanet-veremi-dataset

as position, speed, and direction.

The dataset contains a total of 512,434 timestamped events simulating various interactions under normal and attack scenarios. Initially, the data was provided in five separate CSV files, each corresponding to a specific attack type: **Constant Position** (AT1), **Constant Offset** (AT2), **Random Position** (AT4), **Random Offset** (AT8), and **Eventual Stop** (AT16). These files were merged into a single, unified dataset to facilitate consistent preprocessing and model training.

Upon merging, each data instance (or row) in this unified dataset contains 18 attributes. These attributes, derived from pairs of observed messages , are as follows:

- `sendtime_1`: Timestamp of the first message observation.
- `sender_1`: Identifier for the sender of the first message observation.
- `messageID`: Identifier for the message (typically associated with the first observation).
- `pos-x1`: X-coordinate from the first message observation.
- `pos-y1`: Y-coordinate from the first message observation.
- `pos-z1`: Z-coordinate from the first message observation.
- `spd-x1`: X-component of velocity from the first message observation.
- `spd-y1`: Y-component of velocity from the first message observation.
- `spd-z1`: Z-component of velocity from the first message observation.
- `AttackerType`: The label indicating if the behavior is normal (0) or one of the five malicious types (1, 2, 4, 8, or 16). This serves as the target variable for our classification task.
- `sendtime_2`: Timestamp of the second message observation.
- `sender_2`: Identifier for the sender of the second message observation.
- `pos-x2`: X-coordinate from the second message observation.
- `pos-y2`: Y-coordinate from the second message observation.
- `pos-z2`: Z-coordinate from the second message observation.

- **spd-x2**: X-component of velocity from the second message observation.
- **spd-y2**: Y-component of velocity from the second message observation.
- **spd-z2**: Z-component of velocity from the second message observation.

These initial attributes provide a rich set of information, including temporal markers (`sendtime_1`, `sendtime_2`), source and message identifiers (`sender_1`, `sender_2`, `messageID`), full 3D Cartesian coordinates (`pos-x1`, `pos-y1`, `pos-z1`, `pos-x2`, `pos-y2`, `pos-z2`), and 3D velocity vectors (`spd-x1`, `spd-y1`, `spd-z1`, `spd-x2`, `spd-y2`, `spd-z2`). The process of refining this full set of attributes into the specific features used for model training will be detailed in the Preprocessing section (Section 5.3.1).

5.3 Proposed approach

The misbehavior detection approach adopted in this work follows a structured process composed of four main stages: data preprocessing, model development, training, and testing. Each of these steps is essential to ensure a consistent and effective implementation of the classification system within the context of vehicular networks.

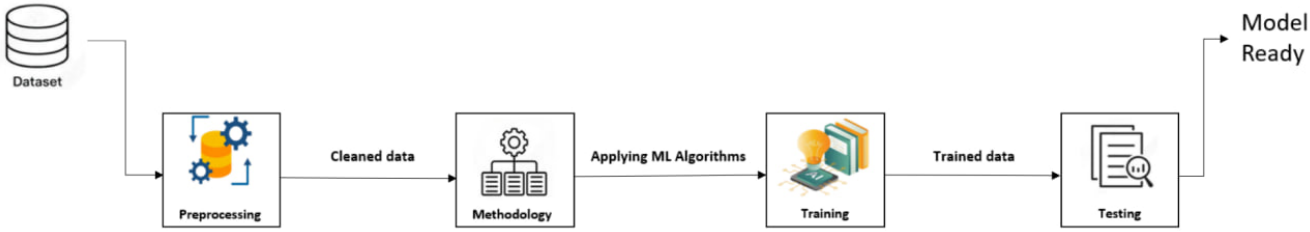

Figure 5.1: machine learning framework for misbehavior detection

5.3.1 Preprocessing

The primary goal of the preprocessing phase was to prepare the merged VeReMi dataset, initially comprising 18 attributes per instance (as detailed in Section 5.2), for effective use by the machine learning classifiers. This involved ensuring data integrity, engineering relevant features, subsetting the data based on a derived trust metric, and selecting the final set of predictive features.

5.3.1.1 Data Integrity Check and Initial Feature Reduction

First, an initial data quality assessment was performed on the unified dataset. This step involved checking for missing values and data type consistencies. The dataset was found to be clean in terms of the features considered for this study, with no missing values requiring imputation or the removal of instances.

Following this integrity check, a preliminary feature reduction step was undertaken. The four Z-axis related features, positions (`position z1`, , `position z2`) , and speeds (`speed z1`,`speed z2`) were removed from the dataset. This decision was based on the rationale that the scope of this research primarily addresses two-dimensional position falsification attacks, for which these 3D spatial and velocity components were deemed to offer limited direct relevance. After this step, 14 columns remained: 13 potential features and the **`AttackerType`** target variable.

5.3.1.2 Trust Feature Engineering

To further refine the dataset and concentrate the analysis on instances exhibiting potentially suspicious characteristics, a binary **trust** feature was engineered. This process involved several analytical steps:

First, a metric representing the Euclidean distance between the two observed positions in each data record was computed. This **estimated distance** was calculated using the standard formula:

$$\text{Estimated Distance} = \sqrt{(\texttt{position x2} - \texttt{position x1})^2 + (\texttt{position y2} - \texttt{position y1})^2}$$

where (**`position x1`**, **`position y1`**) and (**`position x2`**, **`position y2`**) are the coordinates from the paired message observations.

Subsequently, a **distance difference** metric was derived. This was defined as the absolute difference between this calculated Euclidean **estimated distance** and the x-component of the velocity reported in the first message observation (**`speed x1`**):

$$\text{Distance Difference} = |\text{Estimated Distance} - \texttt{speed x1}|$$

A rule-based classification logic was then applied to each data instance to assign a **trust**

label. This logic primarily utilized the **distance difference** metric. Alongside this, several other conceptual metrics were notionally part of this classification rule, including an `avg_distance_diff_percent`, `nearest_trusted_vehicle_distance`, `junction_distance_difference`, and an `AoA_match` (Angle of Arrival match) indicator. In the implemented experiment, these additional conceptual metrics were assigned fixed placeholder values (10, 15, 7, and True, respectively). The core of the trust decision was therefore driven by the **distance difference** relative to predefined thresholds (Th1=10, Th2=20, Th3=10). An instance was classified as 'trustable' (receiving a **trust** value of 1) if its **distance difference** was less than or equal to a specific threshold (Th3, i.e., $\leq$ 10) and the conditions related to the placeholder metrics were effectively met due to their fixed values. All other instances were classified as 'not trustable' (receiving a **trust** value of 0).

5.3.1.3 Dataset Subsetting for Model Training

Following the engineering of the **trust** feature, the dataset was partitioned. The primary analysis for detecting specific misbehavior types (as indicated by the `AttackerType` label) was subsequently focused exclusively on the subset of data that had been classified as 'not trustable' (i.e., where the **trust** feature was 0). This filtered subset, termed the `untrustable_vehicles` data, consisted of 247,378 instances and formed the complete basis for the training and evaluation of the machine learning models described in this study.

5.3.1.4 Final Feature Selection for Modeling

From this `untrustable_vehicles` subset, the `AttackerType` column was designated as the target variable (y) for classification. The set of input features (X) for the machine learning models was then constructed. This involved retaining relevant attributes from the `untrustable_vehicles` data while excluding the `AttackerType` column (as it is the target) and the intermediate **trust** column (as its role was in creating the data subset). This process resulted in the final set of **13 predictive features** used for modeling, categorized as follows:

- **Message Timestamps and Identifiers**: `sendtime_1`, `sender_1`, `messageID`, `sendtime_2`, `sender_2`.
- **Positional Coordinates (2D)**: `pos-x1`, `pos-y1`, `pos-x2`, `pos-y2`.
- **Velocity Components (2D)**: `spd-x1`, `spd-y1`, `spd-x2`, `spd-y2`.

These selected features collectively provide temporal context, source identification information, and the essential two-dimensional positional and kinematic data from the paired message observations, deemed crucial for detecting the targeted misbehaviors.

The final preprocessing step applied to these 13 features was feature scaling (standardization), which was performed after splitting the **`untrustable_vehicles`** data into training and testing sets, as will be detailed in the subsequent training methodology section (Section 5.3.3).

5.3.2 Model Development

For this study, a selection of well-established machine learning algorithms was implemented to detect and classify misbehavior in VANETs using the prepared VeReMi dataset. While much of the existing literature reports strong performance and widespread application of models such as SVM [65] and RF [66] for similar classification and intrusion detection tasks in vehicular networks, this research also aimed to explore and validate the efficacy of XGBoost.
XGBoost was selected due to its strong reputation for high performance, speed, and scalability, particularly with structured datasets, making it a dominant algorithm in many contemporary machine learning applications [67]. The primary goal was to conduct a comparative analysis to determine which of these models offers the most robust and accurate solution for the specific challenge of position falsification attack detection.

5.3.2.1 XGBOOST

XGBoost was a central focus of our model development phase due to its increasing prominence and demonstrated success in various machine learning competitions and complex classification problems. XGBoost is a highly efficient and scalable machine learning algorithm specifically engineered for structured or tabular data [67]. Developed by Tianqi Chen and Carlos Guestrin, it enhances traditional Gradient Boosted Decision Trees (GBDTs) through several key optimizations, including capabilities for parallel processing, out-of-core computation for large datasets, and sparsity-aware learning, which allows it to intrinsically handle missing values [67].

A distinguishing feature of XGBoost is its incorporation of a regularized objective function. This regularization helps to prevent overfitting by penalizing model complexity, leading to improved generalization on unseen data [67]. Furthermore, XGBoost employs sophisticated techniques like weighted quantile sketches for more accurate split selection in decision trees and

supports distributed computing, making it well-suited for tackling large-scale datasets efficiently [67]. Given its robust performance, flexibility, and the specific characteristics of the VeReMi dataset (structured, moderately large), XGBoost was hypothesized to offer a superior solution for accurately classifying nuanced misbehaviors in VANETs. The widespread adoption of XGBoost in fields requiring high accuracy, such as fraud detection and anomaly detection [67], further motivated its selection as a primary candidate for this research.

5.3.2.2 RF

RF was also selected for development and comparison in this study. As a well-regarded and versatile ensemble learning method, RF is known for its robustness and effectiveness in handling various classification tasks, often demonstrating high accuracy. Many existing studies on misbehavior detection in VANETs have reported good accuracy and reliable performance with RF models [66], making it an important benchmark for our comparative analysis.

5.3.2.3 SVM

SVM was the third algorithm developed for this comparative study. SVMs are powerful classifiers, particularly effective in high-dimensional spaces and renowned for their proficiency in both classification and regression problems. They have been successfully applied to numerous security-related classification problems. Similar to RF, SVMs are frequently cited in the literature for achieving good performance in network intrusion and misbehavior detection within vehicular contexts [65], thus providing another critical benchmark against which XGBoost's performance could be evaluated.

5.3.3 Training

This section details the training process for the selected machine learning models: XGBoost, RF, and SVM. The primary objective of this phase was to develop robust classifiers capable of accurately identifying different types of position falsification attacks based on the features derived from the `untrustable_vehicles` subset of the VeReMi dataset.

Data Preparation for Training: The `untrustable_vehicles` DataFrame, identified through the trust feature engineering process, served as the basis for model training. This subset consisted of 247,378 instances. From this data, the input features (X) were defined as the 13

selected attributes (excluding **AttackerType** and the intermediate **trust** column). The target variable (y) was the **AttackerType** column.

This data was then partitioned into training and testing sets. An 80/20 split was utilized, allocating 80% of the **untrustable_vehicles** instances for training the models and the remaining 20% for subsequent testing and performance evaluation. The parameter **random_state = 42** was used during this split to ensure the reproducibility of the results across different experimental runs.

Feature Scaling Prior to model fitting, the training features (**X_train**) were standardized using the **StandardScaler** from the scikit-learn library. This process transformed the numerical features to have zero mean and unit variance. The scaler was fit exclusively on the **X_train** data, and the same fitted scaler was then used to transform the **X_test** features. This standardization is beneficial for optimizing the performance of algorithms sensitive to feature magnitudes, such as SVM, and can aid the convergence of gradient-based methods like XGBoost. This globally scaled **X_train** was used for training all three developed models.

Label Encoding To ensure a consistent numerical format suitable for all classifiers, the categorical target variable **y_train** was converted into a 0-indexed numerical representation using "LabelEncoder" from scikit-learn. This resulted in **y_train_encoded**, which was used as the target variable for training all three machine learning models. The same encoding transformation was subsequently applied to **y_test** to create **y_test_encoded** for the evaluation phase.

5.3.3.1 Model Configurations and Training

The selected models were configured and trained as follows, using parameters identified from the experimental code:

- **XGBoost**: The **XGBClassifier** was initialized with the following hyperparameters:
  - **n_estimators**: 100 (The model builds and combines 100 individual decision trees).
  - **learning_rate**: 0.1 (Scales the contribution of each tree by a factor of 0.1 to make the learning process more robust).

- **max_depth**: 3 (Limits each individual decision tree to a maximum of 3 levels of splits to control model complexity).
- **random_state**: 42 (Ensures that the results are reproducible by fixing the seed for any random processes within the algorithm).

The XGBoost model was then trained using the scaled training features (**X_train**) and the encoded target labels (**y_train_encoded**).

- **Support Vector Machine (SVM)**: For SVM, the **LinearSVC** (Linear Support Vector Classification) model was employed, configured with:
  - **C**: 1.0 (Regularization parameter).
  - **dual**: `False` (Set to `False`, often more efficient when the number of samples exceeds the number of features).
  - **max_iter**: 1000 (Maximum number of iterations for the solver to converge).
  - **random_state**: 42 (Ensures reproducible results by fixing the seed for any random processes within the algorithm).

  The **LinearSVC** model was trained on the scaled training features (**X_train**) and the encoded target labels (**y_train_encoded**).

- **Random Forest (RF)**: The **RandomForestClassifier** was configured with the following parameters:
  - **n_estimators**: 100 (Number of trees in the forest).
  - **random_state**: 42 (Ensures reproducible results by fixing the seed for random processes like bootstrap sampling).
  - **max_depth**: 5 (Maximum depth of the individual trees).
  - **min_samples_split**: 10 (Minimum number of samples required to split an internal node).
  - **min_samples_leaf**: 5 (Minimum number of samples required to be at a leaf node).
  - **max_features**: 0.5 (The number of features to consider when looking for the best split, set to 50% of total features).

The Random Forest model was also trained using the scaled training features (**X_train**) and the encoded target labels (**y_train_encoded**).

Each model was trained by fitting it to the prepared training data. The resulting trained models were then subsequently evaluated on the unseen test set to assess their generalization performance, as detailed in the following section.

5.3.4 Testing

The **test set**, comprising 20% of the **untrustable_vehicles** data previously held out during the splitting process, was used to evaluate the **generalization performance** of the trained **XGBoost**, **Random Forest**, and **SVM** models on unseen instances. A critical step in this phase was the consistent application of **preprocessing**: the **X_test** features were transformed using the **StandardScaler** instance that had been previously fitted on the **X_train** data. This ensured that the test data was **scaled** in exactly the same manner as the training data, which is vital for a fair and accurate assessment of the models' performance.

Each of the three **trained models** was then employed to make **predictions** on this prepared **X_test** feature set. Given that the models were trained to predict the **label-encoded** version of **AttackerType**, their initial outputs (raw predictions) were also in this encoded numerical format. To facilitate meaningful comparison with the **original class labels** and for clear interpretation in the subsequent performance evaluation, these encoded **predictions** were subsequently **inverse-transformed** back to their original **AttackerType** categorical labels. This was achieved using the **LabelEncoder** instance that had been fitted on the training labels during the training preparation phase.

These final **predictions**, along with the **y_test** labels, formed the basis for the subsequent comprehensive **performance assessment**. This **rigorous testing** on **unseen data** plays a crucial role in assessing each model's ability to generalize and accurately predict new instances of misbehavior under realistic conditions.

5.4 Performance Evaluation

5.4.1 Evaluation Metrics

To evaluate the effectiveness of the proposed machine learning approach for detecting position falsification attacks in Vehicular Ad Hoc Networks (VANETs), we employed two key performance indicators: Accuracy and F1-score [60, 54].

Accuracy represents the overall correct classification ratio, measuring the proportion of true predictions for both malicious and legitimate cases relative to all instances. But , in imbalanced datasets like VeReMi, where malicious nodes are less frequent, accuracy can be misleading. To address this, the F1-score, which balances precision and recall, is used as a complementary metric.

Precision is the ratio of correctly identified malicious nodes to all nodes predicted as malicious, while recall is the ratio of correctly identified malicious nodes to all actual malicious nodes. As improving precision tends to lower recall and vice versa, the F1-score serves as a balanced metric by taking their harmonic mean, the F1-score, is essential. The F1-score calculates the harmonic mean of precision and recall, providing a robust measure for imbalanced datasets [54].

These metrics are computed using the following formulas:

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

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

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

The variables are defined as:

TP (True Positive) Correctly identified false position attacks.

TN (True Negative) Correctly identified legitimate vehicle positions.

FP (False Positive) Legitimate positions incorrectly classified as attacks.

FN (False Negative) False position attacks incorrectly classified as legitimate.

5.4.2 Results and discussion

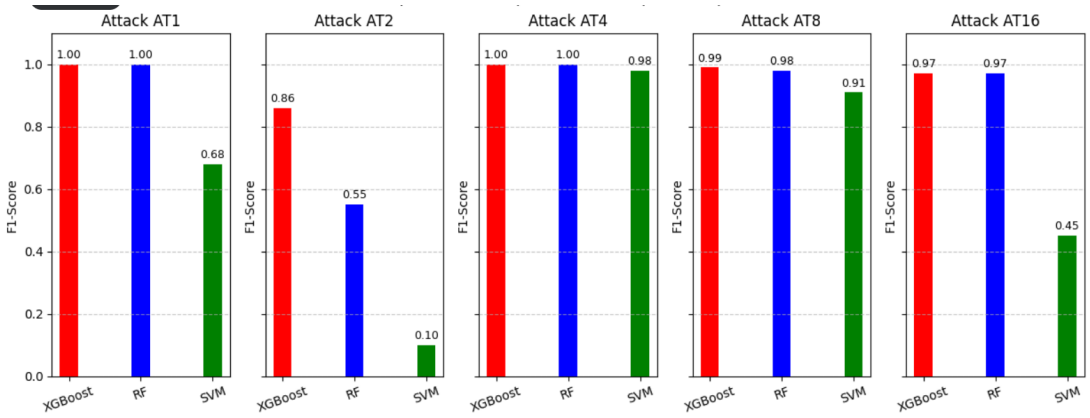

Figure 5.2: Performance Comparison by Attack

The experimental results, depicted in Figure 5.2 demonstrate the superior performance of the XGBoost model in detecting various types of attacks. Specifically, it achieves a perfect F1-score of 1.00 for attacks AT1 and AT4, while maintaining high performance on others, notably 0.99 for AT8 and 0.97 for AT16. In contrast, the SVM model demonstrates significant instability, with a very low score of 0.10 for AT2 and only 0.45 for AT16, despite showing good performance on AT4 (0.98) and AT8 (0.91). The Random Forest model delivers generally satisfactory results, with scores close to those of XGBoost, although it shows a noticeable drop for AT2 (0.55). These findings confirm the robustness and stability of the XGBoost model when facing various types of attacks, making it a strong candidate for machine learning-based intrusion detection systems.

Another way to evaluate the models used is by referring to Table 5.1. XGBoost stands out as the most effective model, achieving the highest accuracy (0.9828), a perfect recall (1.00), and a high F1-score (0.99), demonstrating a strong balance between precision and recall. Random Forest follows closely with a solid accuracy (0.9259) and an F1-score of 0.94, although its recall (0.90) is slightly lower, which may lead to some missed cases. SVM, despite having a high recall (0.98), suffers from lower precision (0.85) and an F1-score of 0.92, indicating a higher rate of false positives. Overall, XGBoost proves to be the most reliable model, followed by Random Forest, while SVM appears less suitable without further optimization.

Table 5.1: Models Evaluations

ML Model	Random Forest (RF)	XGBoost	SVM
Accuracy	0.92	0.98	0.87
F1-score	0.94	0.99	0.92

5.5 Conclusion

Preventing and defending against false position attacks on vehicles in CITS has become a significant chal- lenge especially in safety and emergency applications, where precise location data is crucial. Correct positioning reduces the risk of accidents and improves route planning and traffic man- agement. Therefore, developing strong and efficient machine learning models for misbehavior detection holds great potential to enhance CITS applications. In this paper, we explored only machine-learning-based approaches for misbehavior detection in VANETs. XGBoost is a powerful and efficient tool for Cooperative Intelligent Transport Systems (CITS), particularly in detecting misbehavior and potential attacks. Its scalability and strong performance on structured data make it well-suited for secur- ing intelligent transportation environments. Results revealed that this algorithm consistently outperformed the other models across all experiments. Beyond enhancing security, such advancements in CITS also support broader sustainability goals. By enabling more reliable, efficient, and cooperative traffic systems, CITS serve as a pivotal solution for building greener roads and fostering environmentally responsible engineering practices.

Evaluation Matrix Formulas	
Metric	**Equation**
Accuracy:	$\frac{TP + TN}{TP + FP + TN + FN}$
F1-score:	$2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$
Precision:	$\frac{TP}{TP + FP}$
Recall:	$\frac{TP}{TP + FN}$

Table 5.2: Formulas for key evaluation metrics

General Conclusion

Preventing and defending against false position attacks on vehicles in CITS has become a significant chal lenge—especially in safety and emergency applications, where precise location data is crucial. Correct positioning reduces the risk of accidents and improves route planning and traffic man agement. Therefore, developing strong and efficient machine learning models for misbehavior detection holds great potential to enhance CITS applications. This work focused on machine-learning-based approaches for misbehavior detection in Vehicular Ad Hoc Networks (VANETs). The Extreme Gradient Boosting (XGBoost) algorithm emerged as a highly effective and efficient tool for C-ITS, particularly in its ability to identify anomalies and potential attacks. Its inherent characteristics, such as excellent scalability and strong performance when dealing with structured datasets, make it exceptionally well-suited for securing intelligent transportation environments. The experimental findings clearly indicated that this algorithm consistently surpassed the performance of the other evaluated models across all conducted tests. Future work could explore several avenues. This includes investigating the performance of the proposed models with real-time VANET data streams, which may present additional challenges related to data latency and volume. Another direction involves exploring the deployment of these machine learning models on edge computing devices within Road Side Units (RSUs) or On-Board Units (OBUs) to enable more immediate, localized misbehavior detection, thereby reducing reliance on centralized authorities. Given privacy concerns in vehicular networks, research into federated learning approaches is also promising, where models are trained collaboratively on local vehicle data without centralizing raw information, enhancing privacy while maintaining detection capabilities.

Bibliography

[1] A. Manderna, S. Kumar, U. Dohare, M. Aljaidi, O. Kaiwartya, and J. Lloret, "Vehicular network intrusion detection using a cascaded deep learning approach with multi-variant metaheuristic," *Sensors*, vol. 23, no. 21, p. 8772, 2023, published 2023 Oct 27. [Online]. Available: https://www.mdpi.com/1424-8220/23/21/8772

[2] W. contributors, "Intelligent transportation system — Wikipedia, The Free Encyclopedia," https://en.wikipedia.org/w/index.php?title=Intelligent_transportation_system&oldid=1270473349, 2025, [Online; accessed 29-April-2025].

[3] J. Scholliers, "Improving safety and mobility of vulnerable road users through intelligent transport systems," http://blogs.biomedcentral.com/on-society/2019/05/07/improving-safety-and-mobility-of-vulnerable-road-users-through-intelligent-transport-systems/, 2019, accessed: 2025-04-29.

[4] M. Almutiq, L. Sellami, and B. Alaya, "Dynamic vehicular clustering enhancing video on demand services over vehicular ad-hoc networks," *Computers, Materials and Continua*, vol. 72, no. 2, pp. 3493–3510, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1546221822014801

[5] V. Hassija, V. Chamola, V. R. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on security issues in vehicular ad hoc networks," *Applied Sciences*, vol. 12, no. 15, p. 7922, 2020, adapted and reused in Encyclopedia.pub. [Online]. Available: https://encyclopedia.pub/entry/27657

[6] K. Moghraoui, "Gestion de l'anonymat des communications dans les réseaux véhiculaires ad hoc sans fil (vanets)," Ph.D. dissertation, Université du Québec à Trois-Rivières, 2015.

[7] Wikipédia, "Infrastructure à clés publiques — wikipédia, l'encyclopédie libre," 2024, [En

ligne; Page disponible le 13-février-2024]. [Online]. Available: http://fr.wikipedia.org/w/index.php?title=Infrastructure_%C3%A0_cl%C3%A9s_publiques&oldid=212437729

[8] S. Diop, "Une infrastructure à clés publiques (pki) pour sécuriser les messages dans un réseau v2g," Ph.D. dissertation, Université du Québec à Trois-Rivières, 2018.

[9] DigiCert, "What is pki | public key infrastructure," https://www.digicert.com/what-is-pki, 2021, accessed: 19 May 2025.

[10] Entrust, "What is pki (public key infrastructure)? pki meaning," https://www.entrust.com/knowledgebase/pki/what-is-pki, 2025, accessed: 19 May 2025.

[11] IDManagement.gov, "Public key infrastructure 101," https://www.idmanagement.gov/pki/, 2025, accessed: 19 May 2025.

[12] Wikipedia, "Public key infrastructure," https://en.wikipedia.org/wiki/Public_key_infrastructure, 2023, accessed: 19 May 2025.

[13] M. Li, "Security in vanets," Washington University in St. Louis, Tech. Rep., December 2014, a paper written under the guidance of Prof. Raj Jain. Available at http://www.cse.wustl.edu/ jain/cse571-14/ftp/vanet_security/index.html (Accessed: [Your Access Date]).

[14] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "Vanet security surveys," *Computer Communications*, vol. 44, pp. 1–13, 2014.

[15] N. GUERFI, "La navigation des véhicules autonomes par un algorithme génétique."

[16] P. Mundhe, S. Verma, and S. Venkatesan, "A comprehensive survey on authentication and privacy-preserving schemes in vanets," *Computer Science Review*, vol. 41, p. 100411, 2021.

[17] S. Russell and P. Norvig, "Artificial intelligence: a modern approach, 4th us ed," *aima: website. URL: https://aima. cs. berkeley. edu/(date of access : 26.02. 2023)*, 2021.

[18] K. Kahn and N. Winters, "Constructionism and ai: A history and possible futures," *British Journal of Educational Technology*, vol. 52, no. 3, pp. 1130–1142, 2021.

[19] GeeksforGeeks. (2024) Top 10 branches of artificial intelligence. Last updated: 28 Jun, 2024. Accessed: 28 May, 2025. [Online]. Available: https://www.geeksforgeeks.org/top-10-branches-of-artificial-intelligence/

[20] S. Y. Liu, "Harnessing artificial intelligence," *Case Studies: Insights on Agriculture Innovation 2020 (IAAS Series)*, p. 13, 2021.

[21] A. Biswal. (2024) Top 24 artificial intelligence applications and uses. Simplilearn, Lesson 2 of 21. Accessed: 28 May, 2025. [Online]. Available: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/artificial-intelligence-applications

[22] E. Dilek and M. Dener, "Computer vision applications in intelligent transportation systems: A survey," *Sensors*, vol. 23, no. 6, p. 2938, 2023.

[23] Ultralytics, "Computer vision in 2025: Trends and applications," *Ultralytics Blog*, 2025.

[24] ImageVision.ai, "Key trends in computer vision for 2025," *ImageVision.ai Blog*, 2024.

[25] Picsellia, "2025 trends in computer vision: What to expect," *Picsellia Blog*, 2024.

[26] J. P. Bharadiya, "Artificial intelligence in transportation systems a critical review," *American Journal of Computing and Engineering*, vol. 6, no. 1, pp. 35–45, 2023.

[27] GeeksforGeeks, "Machine learning tutorial," https://www.geeksforgeeks.org/machine-learning/, 2025, last updated: 03 May 2025. [Online]. Available: https://www.geeksforgeeks.org/machine-learning/

[28] GeeksforGeeks, "Supervised machine learning," https://www.geeksforgeeks.org/supervised-machine-learning/, 2025, accessed: 2025-05-26.

[29] 'GeeksforGeeks', "What is unsupervised learning?" https://www.geeksforgeeks.org/unsupervised-learning/, 2025, accessed: 2025-05-30.

[30] GeeksforGeeks, "What is reinforcement learning?" https://www.geeksforgeeks.org/what-is-reinforcement-learning/, 2025, accessed: 2025-05-30.

[31] Coursera. (2023) 10 examples of deep learning applications. Coursera. Accessed: 2025-05-28. [Online]. Available: https://www.coursera.org/articles/deep-learning-applications

[32] Coursera. (2023) What is deep learning? definition, examples, and careers. Accessed: 2025-05-29. [Online]. Available: https://www.coursera.org/articles/what-is-deep-learning

[33] A. Mosavi, S. Ardabili, and A. R. Varkonyi-Koczy, "List of deep learning models," in *International conference on global research and education*. Springer, 2019, pp. 202–214.

[34] T. Chaymae, H. Elkhatir, and A. Otman, "Recent advances in machine learning and deep learning in vehicular ad-hoc networks: A comparative study," in *International Conference on Electrical Systems & Automation*. Springer, 2021, pp. 1–14.

[35] Y. Jernite and Collaborators, "Dziribert: A pre-trained language model for algerian arabic," https://huggingface.co/DziriBERT, 2022.

[36] U. d'Alger and Collaborators, "Darjabert: A transformer model for algerian dialect nlp," 2023, internal project or unpublished prototype (to be confirmed).

[37] R. Rabah, "Les surcoûts des mécanismes de sécurité dans les vanets," Ph.D. dissertation, UNIVERSITE BADJI MOKHTAR ANNABA, 2021.

[38] M. S. Sheikh, J. Liang, and W. Wang, "A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets)," *Sensors*, vol. 19, no. 16, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/16/3589

[39] J. Ren, Y. Cheng, and S. Xu, "Edppa: An efficient distance-based privacy preserving authentication protocol in vanet," *Peer-to-Peer Networking and Applications*, vol. 15, pp. 1–13, 05 2022.

[40] A. Mahmood, B. Butler, W. E. Zhang, Q. Z. Sheng, and S. Siddiqui, "Security in vehicular ad hoc networks: Challenges and countermeasures," *Security and Communication Networks*, vol. 2021, pp. 1–24, 2021.

[41] F. Ahmad, A. Adnane, and V. N. Franqueira, "A systematic approach for cyber security in vehicular networks," *Journal of Computer and Communications*, vol. 4, no. 16, pp. 38–62, 2016.

[42] F. Azam, S. Kumar, N. Priyadarshi, S. Padmanaban, and R. Bansal, "A comprehensive review of authentication schemes in vehicular ad-hoc network," *IEEE Access*, vol. PP, 02 2021.

[43] F. Hawlader, A. Boualouache, S. Faye, and T. Engel, "Intelligent misbehavior detection system for detecting false position attacks in vehicular networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[44] A. Boualouache, B. O. Soufiene, P. Muhlethaler, and T. Engel, "A survey on machine learning-based misbehavior detection systems for 5g and beyond vehicular networks," *IEEE Access*, vol. 11, pp. 2519–2551, 2023.

[45] J. Grover, M. S. Gaur, and V. Laxmi, "A novel defense mechanism against sybil attacks in vanet," *2011 Third International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–9, 2011.

[46] J. H. Jafarian and T. Alpcan, "Software-defined radio based gnss spoofing and detection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2004–2016, 2018.

[47] P. Papadimitratos, L. Buttyan, T. Holczer, and E. Schoch, "Secure vehicular communication systems: Design and architecture," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, 2008.

[48] T. Engel and N. Lagraa, "Security challenges in vehicular networks: A holistic view," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3456–3470, 2019.

[49] M. S. Sheikh, J. Liang, and W. Wang, "A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets)," *Sensors*, vol. 19, no. 16, p. 3589, 2019.

[50] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2006.

[51] A. Studer, E. Shi, and F. Bai, "Tacking together efficient authentication, revocation, and privacy in vanets," *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pp. 1–9, 2009.

[52] M. A. H. Al Junaid, A. A. Syed, M. N. M. Warip, K. N. F. K. Azir, and N. H. Romli, "Classification of security attacks in vanet: A review of requirements and perspectives," in *MATEC Web of Conferences*, vol. 150. EDP Sciences, 2018, p. 06038.

[53] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2018, pp. 318–337.

[54] T. Alladi, A. Agrawal, V. Chamola, and B. Sikdar, "Feature engineering impact on position falsification attacks detection in vehicular ad-hoc network," *IEEE Internet of Things Journal*, vol. 11, no. 5, pp. 9058–9070, 2024.

[55] R. Kolandaisamy, M. Y. I. Idris, and S. Bhattacharyya, "A review of security attacks and intrusion detection in the vehicular ad hoc networks," *Journal of Computer Virology and Hacking Techniques*, vol. 19, pp. 637–655, 2022.

[56] P. Kolodziej, J. Smith, and R. Patel, "Intelligent defense strategies: Comprehensive attack detection in vanet with deep reinforcement learning," *Journal of Network Security*, vol. 12, no. 3, pp. 45–60, 2023.

[57] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[58] M. S. Al-Kahtani, "Survey on security attacks in vehicular ad hoc networks (vanets)," *International Journal of Computer Science and Network Security*, vol. 12, no. 6, pp. 67–74, 2012.

[59] T. Leinmuller, E. Schoch, and F. Kargl, "Position verification approaches for vehicular ad hoc networks," *IEEE Wireless Communications*, vol. 13, no. 5, pp. 16–21, 2006.

[60] S. Ercan, M. Ayaida, and N. Messai, "Misbehavior detection for position falsification attacks in vanets using machine learning," *IEEE Access*, vol. 9, pp. 105 277—-105 289, 2021.

[61] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[62] D. M. Abdullah and A. M. Abdulazeez, "Machine learning applications based on SVM classification a review," *Qubahan Academic Journal*, vol. 1, no. 2, pp. 81–90, 2021.

[63] T. Chen and C. 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.

[64] J. Liang, J. Chen, Y. Zhu, and R. Yu, "A novel intrusion detection system for vehicular ad hoc networks (vanets) based on differences of traffic flow and position," *Applied Soft Computing*, vol. 75, pp. 712–727, 2019.

[65] J. Gu and S. Lu, "An effective intrusion detection approach using svm with naïve bayes feature embedding," *Computers and Security*, vol. 103, p. 102158, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404820304314

[66] M. Zang and Y. Yan, "Machine learning-based intrusion detection system for big data analytics in vanet," pp. 1–5, 2021.

[67] T. Chen and C. 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.

Machine Learning for Misbehavior Detection in Next-Generation Vehicular Networks: Towards Safer and Greener Mobility

1st Boubakeur Moussaoui
Computer Science Department
University of Mohamed el Bachir el Ibrahimi
Bordj Bou Arreridj, Algeria
Laboratory of Informatics and its Applications of M'sila (LIAM)
b.moussaoui@univ-bba.dz

2nd Baya Mekhfi
Computer Science Department
University of Mohamed el Bachir el Ibrahimi
Bordj Bou Arreridj, Algeria
baya.mekhfi@univ-bba.dz

3rd Ahmed Salah Eddine Madi
Computer Science Department
University of Mohamed el Bachir el Ibrahimi
Bordj Bou Arreridj, Algeria
ahmedsalaheddine.madi@univ-bba.dz

***Abstract*—Connected vehicles have great potential to enhance road safety, reduce traffic congestion, and play a vital role in green engineering by reducing pollution and fuel consumption. By enabling more efficient traffic flow, eco-routing, and optimized driving behaviors, connected vehicles contribute to a cleaner environment. However, when a vehicle is compromised, it can pose a serious threat to the entire network due to the potential harm it can cause. One of the major challenges in vehicular networks is the detection of misbehaving vehicles, which should then be blacklisted or their certificates revoked. In this paper, we propose a novel scheme that leverages machine learning to accurately detect and classify vehicle behavior, enabling effective identification and management of misbehaving vehicles. To assess the effectiveness of our approach, a comprehensive comparative analysis was performed. The results demonstrate that our model outperforms existing methods in accurately classifying vehicle behaviors, highlighting its potential for real-world deployment in securing vehicular networks.**

***Index Terms*—Misbehavior detection, Security, Machine learning, Classification, Eco-Friendly Mobility.**

I. Introduction

Cooperative Intelligent Transport Systems (C-ITS) promise, on the one hand, to enhance road safety by reducing accidents. On the other hand, they play a crucial role in creating a greener, more sustainable transportation ecosystem by reducing pollution and promoting environmentally conscious travel [1]. These goals can be achieved through real-time communication between vehicles (V2V), infrastructure (V2I), and traffic management systems. C-ITS improve traffic flow and fuel efficiency, significantly reducing emissions. These systems offer eco-driving advice, such as optimal acceleration and braking patterns, and suggest routes that minimize congestion and fuel consumption [2]. C-ITSs also support green engineering objectives by facilitating the integration of electric vehicles, prioritizing public transport and non-motorized modes like bicycles, and promoting smart mobility planning. In addition, they contribute to long-term environmental goals by reducing traffic-related energy use and enabling infrastructure design that aligns with sustainable development principles. By combining safety improvements with pollution reduction and energy efficiency, C-ITS are a cornerstone of future green transportation systems.

In order to meet the objectives mentioned above, C-ITS enable smooth data exchange between mobile nodes through wireless communication technologies such as Dedicated Short-Range Communications (DSRC) and 5G. C-ITS applications share sensitive data (such as identity, location, speed, direction, etc.) for informed decision-making. The accuracy of a node's location information is especially critical, as it directly affects the reliability of the system. Unfortunately, vehicles are exposed to various cyber-security threats, including Sybil attacks, where an attacker generates multiple fake identities to manipulate the network, denial of service (DoS) attacks that disrupt communication by overloading the network, as well as identity spoofing and eavesdropping, which compromise data integrity and confidentiality.

Intrusion detection in CITS has been extensively studied using centralized, decentralized, and distributed approaches to combat various attacks such as Sybil, black-hole, DoS, and position falsification. Centralized systems rely on trusted authorities to collect and analyze data, while decentralized approaches use local nodes like Road Side Units (RSU) or cluster heads to monitor behavior using game theory, neural networks, or filtering techniques. Distributed systems allow vehicles to detect threats collaboratively through consistency checks, signal analysis, or trajectory comparisons. Recently, machine learning-based methods have gained traction, particularly for detecting position falsification attacks. These models

use various features such as signal strength, velocity, trajectory plausibility, and movement patterns [3], [4], [5].

In this study, we focus exclusively on machine learning-based approaches to address position falsification attacks. We propose a novel detection model that uses the XGBOOST framework, which is designed to improve accuracy, robustness, and real-time performance. Our approach demonstrates the potential of advanced machine learning techniques in securing location-based services against malicious manipulation.

II. RELATED WORK

In this section, we are specifically interested in studies that focus on the use of machine learning (ML) algorithms for the detection of position falsification attacks in Vehicular Ad-hoc Networks (VANETs), as part of enabling Cooperative Intelligent Transport Systems (C-ITS).

Authors in [6] introduce a lightweight ML-based misbehavior detection system focused on identifying position falsification attacks in VANETs. Unlike prior works relying on extensive feature sets, their approach extracts features solely from received position data, enabling real-time, low-latency detection. Evaluated on the VeReMi dataset and simulation-generated data, their multi-class classifiers—particularly Decision Tree and Logistic Regression—achieved near-perfect accuracy with minimal computational overhead.

Another work published by [7] proposes a distributed IDS for detecting position falsification attacks in VANETs using machine learning and novel location-based features. Their system combines common mobility indicators with new features such as angle of arrival and estimated distance based on RSSI. After centralised training, vehicles perform real-time detection locally using ensemble learning (Stacking of kNN and Random Forest). Evaluation with the VeReMi dataset shows improved accuracy and significantly reduced computation time compared to prior work.

III. SYSTEM MODEL

This section provides a concise overview of the three main components of this study: the CITS system, the machine learning algorithms employed (XGBoost, SVM, and Random Forest), and the attacker model.

1) Overview of CITS: As shown in Figure 1, the architecture consists of three main entities: the vehicle, which is the principal component, along with the Trusted Authority (TA) and the Road-Side Unit (RSU), each playing a key role in the system.

- **Vehicle:** A vehicle in a VANET is equipped with an 802.11p network interface, enabling communication with RSUs (V2I) and with other vehicles (V2V). It periodically broadcasts safety messages containing sensitive information such as location, timestamp, velocity, identity, and more. Our misbehavior detection module, based on machine learning, will be installed on each vehicle to ensure a rapid response to any suspicious or abnormal situations. The module continuously monitors incoming messages and classifies them in real time, reporting any detected anomalies to the Trusted Authority (TA) immediately.
- **Trusted Authority (TA):** The TA serves as the unique trusted entity in the system and acts as the Public Key Infrastructure (PKI). It manages all critical operations such as vehicle registration, key management, verification, and more. Due to its ample storage and computational resources, the TA securely handles these essential transactions. Additionally, if a vehicle exhibits malicious behavior, the TA has the authority to revoke its certificate to maintain system integrity.
- **RSU:** An RSU is the third component of our system, typically fixed at locations such as intersections or traffic light panels. These units communicate with vehicles in their range through a wireless connection, with other RSUs using a wired connection, and with the Trusted Authority (TA) via a 4G/5G interface. The RSU acts as an intermediary in the backbone network, facilitating communication between the TA and the vehicles.

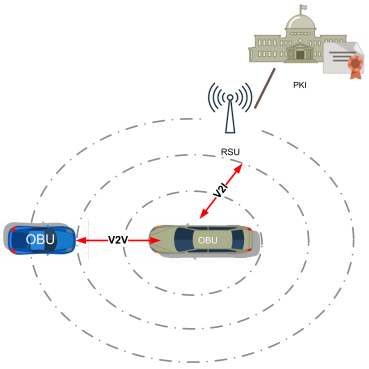

Fig. 1: System Architecture

2) Employed ML Algorithms:

- **RF Method**

Random forests (RF) are an ensemble learning method used for classification, regression, and other tasks. They work by constructing multiple decision trees during the training phase and outputting either the mode of the classes (for classification) or the mean prediction (for regression) of the individual trees.

RF help reduce of overfitting in decision trees by introducing randomness into the tree-building process and combining multiple trees. Key features of random forests include:

a) They use a random subset of features to split nodes when building trees.
b) They employ bagging (bootstrap aggregation) to create diverse training sets for each tree.
c) They do not overfit as more trees are added, due to the Law of Large Numbers.

RF have been shown to be accurate classifiers and regressors, often performing comparably or better than other

ensemble methods like boosting. They are also robust to noise and outliers, faster to train than some other ensemble methods, and provide useful internal estimates of error, strength, correlation, and variable importance. [8]

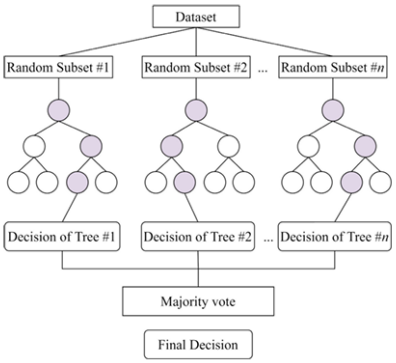

Fig. 2: Random Forest

- **SVM Method**
  Support Vector Machine (SVM), developed by Vapnik, is a powerful supervised learning algorithm commonly used for both classification and regression tasks. It is particularly valued for its robustness and effectiveness in high-dimensional data spaces . The core idea behind SVM is to find the optimal hyperplane that maximally separates two classes of data points. The data points closest to this hyperplane, known as support vectors, play a crucial role in defining the decision boundary.
  When the data is linearly separable, SVM identifies a straight hyperplane. However, for non-linearly separable data, SVM employs kernel functions (such as the Radial Basis Function (RBF) or polynomial kernels) to map the data into a higher-dimensional space, making separation easier .
  SVM is particularly useful for detecting misbehavior in VANETs (Vehicular Ad Hoc Networks), as it can efficiently process the large volumes of data generated by vehicle movement, network communication, and potential security threats. VANET datasets often include multiple high-dimensional features (such as speed, location, message authenticity, and signal strength) which SVM can handle effectively.
  Unlike deep learning models that require extensive training data, SVM is less prone to overfitting and performs well even with smaller datasets. This makes it a practical choice for real-world applications. Furthermore, its ability to model non-linear relationships using kernel tricks enables it to detect complex misbehavior patterns, such as malicious packet injection or false location broadcasting-scenarios that may not be easily identified through traditional rule-based approaches.
  Another key advantage of SVM is its short prediction time. Once trained, it can quickly classify instances of misbehavior, making it highly suitable for real-time VANET security applications.
  However, SVM also presents certain challenges, primarily related to computational complexity and parameter tuning. Training SVM on large datasets can be resource-intensive, as time complexity increases with the number of samples. In addition, selecting the appropriate kernel function and optimizing hyperparameters (such as C and gamma) are crucial to achieving high accuracy, requiring careful fine-tuning to balance performance and efficiency. Despite these challenges, SVM remains a powerful tool for detecting and mitigating security threats in VANETs, thanks to its precision, reliability, and ability to process diverse types of network data. [9]

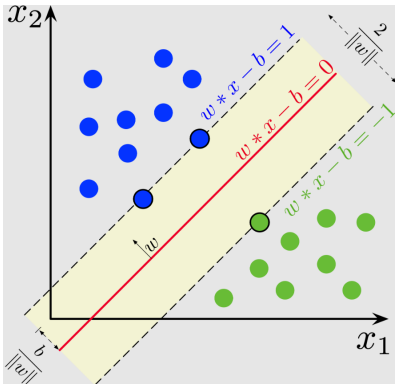

Fig. 3: SVM Decision Boundary and Support Vectors

- **XGBOOST Method** XGBoost (Extreme Gradient Boosting) is a highly efficient and scalable machine learning algorithm designed for structured data. Developed by Tianqi Chen and Carlos Guestrin, XGBoost enhances traditional Gradient Boosted Decision Trees (GBDTs) with several optimizations, including parallel processing, out-of-core computation, and sparsity-aware learning. The algorithm introduces a regularized objective function, improving model generalization and preventing overfitting. Additionally, XGBoost leverages weighted quantile sketches for better split selection in decision trees and supports distributed computing, making it suitable for large-scale datasets. Due to its exceptional performance and flexibility, XGBoost has been widely adopted in machine learning competitions, fraud detection, recommendation systems, and anomaly detection. Its ability to process massive datasets efficiently while maintaining high accuracy has made it one of the most popular algorithms in applied machine learning. [10]

3) Attacker Model: In this paper, we consider several types of position falsification attacks as defined in [6]:

- **Constant position attack**: the attacker transmits a fixed, pre-configured constant position.
- **Constant offset Position attack**: in this case, the attacker generates a fixed offset and adds it to the real

positions, which helps the attacker pretend to change the line on the road trip and hide the actual position.

- **Random position attack**: the attacker transmits a newly generated random position from the vehicular area without considering the real position.
- **Random offset attack**: this case of attack gen erates a random offset from a pre-configured rect angular area around the vehicle, which could be considered a close variation of the random attacks. However, in this case, the vehicle chooses a random value that ranged over the rectangle region around the vehicle.
- **Eventual stop attack**: an attacker behaves normally for a certain time period and then eventually attacks by broadcasting the last position repeatedly (i.e., like it had stopped). This attack can be very harmful as the attacker can pretend not to be on board.

IV. Proposed approach

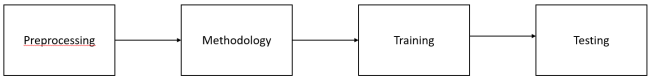

Fig. 4: machine learning framework for misbehavior detection

Figure 4 illustrates the core components of our proposed misbehavior detection framework, structured into four key stages:

TABLE I: Data Types and Corresponding Variables

Type	Variable Names
`float64`	`sendtime`, `sender`, `messageID`, `position`, `speed`, `estimated_distance`, `distance_diff`
`int64`	`AttackerType`, `avg_distance_diff_percent`, `nearest_trusted_vehicle_distance`, `junction_distance_difference`, `trust`

1) **Preprocessing**: This phase involved addressing missing values, removing non-informative features such as pos-z1, spd-z1, pos-z2, and spd-z2, and standardizing data formats to ensure consistency. These preprocessing steps were essential to enhance data quality and prepare the dataset for effective learning by machine learning classifiers.
2) **Model Development**: This stage involves the implementation and comparative evaluation of multiple machine learning algorithms, including RF, SVM, and XGBoost. Among these, XGBoost is emphasized due to its efficiency in handling structured data and its superior predictive performance.
3) **Training**: In this phase, we train the models using historical data, fine-tuning them through hyperparameter optimization. For XGBoost, we particularly focus on essential parameters such as learning rate, maximum depth, number of estimators, and regularization techniques, ensuring an optimal balance between accuracy and generalization.
4) **Testing** : Upon completion of the training phase, the model performance is assessed on a separate test set to evaluate its generalization capability on unseen data. The model's predictions are compared with actual labels, and performance is assessed using accuracy, precision, recall, and F1-score. A confusion matrix helps visualize correct and incorrect classifications. This evaluation ensures the model is reliable and may indicate areas for improvement, such as hyperparameter tuning or better data preprocessing.

- step 1:Preprocessing: To ensure reliable and accurate model evaluation, this study leverages the VeReMi dataset, a widely used benchmark for misbehavior detection in vehicular networks [6]. The dataset contains 512,434 timestamped events representing simulated network interactions under various attack scenarios. Originally distributed across five separate CSV files—each corresponding to a specific attack type (AT1, AT2, AT4, AT8, and AT16)—the data was consolidated into a single unified dataset for comprehensive analysis. Table II shows the attacker parameters used in the VeReMi dataset. Prior to model training, a detailed preprocessing phase was conducted to enhance data quality and consistency. This included handling missing values, removing non-informative features (pos-z1, spd-z1, pos-z2, and spd-z2), and preparing the data for effective learning.
  Each instance in the final dataset is labeled either as normal behavior (class 0) or as one of five distinct misbehavior types (classes 1, 2, 4, 8, and 16). Fourteen key features describe interactions between pairs of nodes based on three principal dimensions: temporal markers indicating precise message transmission times, Cartesian coordinates (pos-x, pos-y) mapping the spatial location of nodes, and velocity vectors (spd-x, spd-y) capturing their movement dynamics. These attributes enable fine-grained behavioral analysis under different adversarial conditions. The attacks are modeled with varying levels of complexity and deception. AT1 represents a static transmitter fixed at coordinates (5560, 5820), while AT2 applies a constant offset to positions, simulating linear drift. AT4 introduces random jumps in position to mimic unpredictable behavior. AT8 adds erratic changes in both speed and direction, increasing the challenge of detection. The most subtle, AT16, simulates an eventual stop: vehicles behave normally for a period before gradually freezing, with a 2.5× increased probability per second of halting completely, imitating sudden and deceptive inactivity.
- step 2:Model Development
  The experimental framework implemented three machine

learning algorithms with carefully selected hyperparameters to ensure optimal performance for network attack detection. XGBoost was configured with 100 gradient-boosted trees (n-estimators=100), a conservative learning rate of 0.1 to prevent overshooting, and a maximum tree depth of 3 to balance model complexity with generalization capability. The SVM implementation utilized a linear kernel with primal optimization (dual=False) for efficient high-dimensional processing, standardized regularization strength (C=1.0), and 1000 maximum iterations to ensure convergence, while automatically adjusting for class imbalance. Random Forest employed 100 fully-grown decision trees with balanced class weights to handle the uneven distribution of attack types. All models shared a fixed random seed (random-state=42) for reproducibility and were evaluated using consistent metrics including accuracy and detailed classification reports.

This systematic configuration enabled direct comparison of algorithmic approaches while respecting each method's inherent characteristics. The XGBoost parameters emphasized computational efficiency and generalization, SVM settings prioritized linear separability with class balance, and Random Forest preserved complete feature interactions through unrestricted tree growth. The hyperparameter configurations are formally compared below:

- step 3 : Training : To train the models effectively, the dataset was first divided into input features (X) and output labels (y). The input features include behavioral metrics of vehicles, while the labels represent the corresponding attack types. The dataset was then split into training and testing subsets using an 80/20 ratio, with 80% of the samples reserved for training.

  Prior to model fitting, the training features (X_train) were standardized using a standard scaler, which transforms the data to have zero mean and unit variance. This step is essential for many machine learning algorithms, particularly Support Vector Machine, as it ensures that all features contribute equally to the learning process. The scaler was fit exclusively on the training data to prevent data leakage and ensure a fair evaluation during testing.
- step 4 : testing The remaining 20% of the data was used as a test set (X_test, y_test) to evaluate the model's performance on unseen data. The standardization parameters (mean and standard deviation) computed from the training set were applied to the test set using the transform() method. This ensures consistent scaling across both sets while preserving the integrity of the evaluation process. The test set plays a crucial role in assessing the model's ability to generalize and accurately predict new instances of misbehavior under realistic conditions.

TABLE II: Different attack types and their parameters.

Attack ID	Attack Name	Parameters
1	Constant	$x = 5560$, $y = 5820$
2	Constant Offset	$\Delta x = 250$, $\Delta y = -150$
4	Random	Uniformly random in playground
8	Random Offset	Δx, Δy uniformly random from $[-300, 300]$
16	Eventual Stop	Stop probability increases by 0.025 each position update (10 Hz)

TABLE III: Hyperparameter Comparison Across Models

Parameter	XGBoost	SVM	Random Forest
Complexity Control	max_depth=3	C=1.0	Full depth
Class Handling	Implicit	class_weight	class_weight
Iteration Control	100 trees	1000 iterations	100 trees

V. Performance Evaluation

A. Evaluation metrics

To evaluate the effectiveness of the proposed approach, we utilize two key performance indicators: Accuracy (ACC) and F1-score. These metrics are computed using the following formulas:

Accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

F1-score:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

Precision:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

Recall:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

Where TP (True Positive) denotes the number of correctly detected misbehavior attacks; FP (False Positive) refers to the number of normal behaviors incorrectly identified as attacks; TN (True Negative) is the number of normal behaviors correctly identified as normal; and FN (False Negative) denotes the number of attacks incorrectly identified as normal behaviors.

B. Results and discussion

The experimental results, depicted in Figure 6, demonstrate the superior performance of the XGBoost model in detecting various types of attacks. Specifically, it achieves a perfect F1-score of 1.00 for attacks AT1 and AT4, while maintaining high

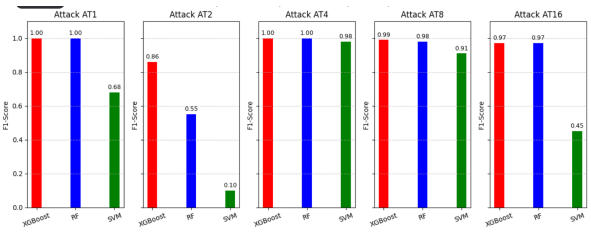

Fig. 5: Performance Comparison by Attack

performance on others, notably 0.99 for AT8 and 0.97 for AT16. In contrast, the SVM model demonstrates significant instability, with a very low score of 0.10 for AT2 and only 0.45 for AT16, despite showing good performance on AT4 (0.98) and AT8 (0.91). The Random Forest model delivers generally satisfactory results, with scores close to those of XGBoost, although it shows a noticeable drop for AT2 (0.55). These findings confirm the robustness and stability of the XGBoost model when facing various types of attacks, making it a strong candidate for machine learning-based intrusion detection systems.

Another way to evaluate the models used is by referring to Table IV. XGBoost stands out as the most effective model, achieving the highest accuracy (0.9828), a perfect recall (1.00), and a high F1-score (0.99), demonstrating a strong balance between precision and recall. Random Forest follows closely with a solid accuracy (0.9259) and an F1-score of 0.94, although its recall (0.90) is slightly lower, which may lead to some missed cases. SVM, despite having a high recall (0.98), suffers from lower precision (0.85) and an F1-score of 0.92, indicating a higher rate of false positives. Overall, XGBoost proves to be the most reliable model, followed by Random Forest, while SVM appears less suitable without further optimization.

TABLE IV: Models Evaluations

ML Model	**Random Forest (RF)**	**XGBoost**	**SVM**
Accuracy	0.92	0.98	0.87
F1-score	0.94	0.99	0.92

VI. Conclusion

Preventing and defending against false position attacks on vehicles in CITS has become a significant challenge—especially in safety and emergency applications, where precise location data is crucial. Correct positioning reduces the risk of accidents and improves route planning and traffic management. Therefore, developing strong and efficient machine learning models for misbehavior detection holds great potential to enhance CITS applications. In this paper, we explored only machine-learning-based approaches for misbehavior detection in VANETs.

XGBoost is a powerful and efficient tool for Cooperative Intelligent Transport Systems (CITS), particularly in detecting misbehavior and potential attacks. Its scalability and strong performance on structured data make it well-suited for securing intelligent transportation environments. Results revealed that this algorithm consistently outperformed the other models across all experiments.

Beyond enhancing security, such advancements in CITS also support broader sustainability goals. By enabling more reliable, efficient, and cooperative traffic systems, CITS serve as a pivotal solution for building greener roads and fostering environmentally responsible engineering practices.

References

[1] S. Thangam, M. Jahnavi, K. Chandana, K. Dheeraj, et al., Smart traffic management in green vanets: Ant-inspired routing strategies with local search optimization, in: 2024 International Conference on Recent Innovation in Smart and Sustainable Technology (ICRISST), IEEE, 2024, pp. 1–6.

[2] H. Y. Adarbah, M. Sookhak, M. Atiquzzaman, A digital twin-based traffic light management system using birch algorithm, Ad Hoc Networks 164 (2024) 103613.

[3] K. Soares, A. A. Shinde, Intrusion detection systems in vanet: A review on implementation techniques and datasets, in: 2024 5th International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), IEEE, 2024, pp. 897–905.

[4] H. E. Idris, I. Hosni, Machine learning-based systems for intrusion detection in vanets, in: Intelligent Systems Conference, Springer, 2024, pp. 603–614.

[5] S. Amaouche, AzidineGuezzaz, S. Benkirane, MouradeAzrour, Ids-xgbfs: a smart intrusion detection system using xgboostwith recent feature selection for vanet safety, Cluster Computing 27 (3) (2024) 3521–3535.

[6] F. Hawlader, A. Boualouache, S. Faye, T. Engel, Intelligent misbehavior detection system for detecting false position attacks in vehicular networks, in: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2021, pp. 1–6.

[7] S. Ercan, M. Ayaida, N. Messai, Misbehavior detection for position falsification attacks in vanets using machine learning, IEEE Access 10 (2021) 1893–1904.

[8] L. Breiman, Random forests, Machine learning 45 (2001) 5–32.

[9] D. M. Abdullah, A. M. Abdulazeez, Machine learning applications based on svm classification a review, Qubahan Academic Journal 1 (2) (2021) 81–90.

[10] T. Chen, C. 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.