This article has been accepted for publication in IEEE Communications Surveys & Tutorials. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/COMST.2023.3236448

A Survey on Machine Learning-based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks

Abdelwahab Boualouache, *Member*, *IEEE* and Thomas Engel, *Member*, *IEEE*

Abstract—Advances in Vehicle-to-Everything (V2X) technology and onboard sensors have significantly accelerated deploying Connected and Automated Vehicles (CAVs). Integrating V2X with 5G has enabled Ultra-Reliable Low Latency Communications (URLLC) to CAVs. However, while communication performance has been enhanced, security and privacy issues have increased. Attacks have become more aggressive, and attackers have become more strategic. Public Key Infrastructure (PKI) proposed by standardization bodies cannot solely defend against these attacks. Thus, in complementary of that, sophisticated systems should be designed to detect such attacks and attackers. Machine Learning (ML) has recently emerged as a key enabler to secure future roads. Various V2X Misbehavior Detection Systems (MDSs) have adopted this paradigm. However, analyzing these systems is a research gap, and developing effective ML-based MDSs is still an open issue. To this end, this paper comprehensively surveys and classifies ML-based MDSs as well as discusses and analyses them from security and ML perspectives. It also provides some learned lessons and recommendations for guiding the development, validation, and deployment of ML-based MDSs. Finally, this paper highlighted open research and standardization issues with some future directions.

Index Terms—5G and Beyond, Connected and Automated Vehicles, Machine Learning, Misbehavior Detection Systems, Security, Vehicle-to-Everything

1 Introduction

The emergence of the fifth-generation (5G) mobile communications networks has brought a technological revolution to the world, as it provides URLLC, high bandwidth, and scalable coverage [1]. As one of the transportation verticals, connected and automated vehicles are witnessing significant advances with the advent of 5G [2]. Equipped with sophisticated onboard sensors such as Radar, On-Board Unit (OBU), and Lidar, CAVs can collect and process sensitive and valuable whereabouts data. Sharing this data among V2X nodes helps to extend their perceptions for ensuring road safety, avoiding traffic congestion, and providing a better driving experience for users during their journey [3]. 5G-V2X communications come to support data sharing while addressing different application requirements. For time-sensitive V2X applications, 5G-V2X provides URLLC with sufficient bandwidth dedicated to infotainment applications. Combining 5G-V2X with other 5G enablers such as Software Defined Networking (SDN), Network Function Virtualization (NFV), and Network Slicing (NS) has opened up a new era to CAVs where several use cases such as teleoperated driving and autonomous driving have emerged [4]. However, 5G-V2X is facing a dangerous vector of attacks, leading to hazardous situations for drivers and passengers. For example, breaking the V2X link between two CAVs during overtaking or lane merging can lead to an accident. Moreover, broadcasting false information over the road network to create traffic congestion impacts convenience, and the business [5]. These security and privacy issues have been taken special attention by research communities since early research investigations on V2X [6]. Specifically, extensive research works have been carried out to protect V2X communications. Several cryptography solutions have been proposed for thwarting V2X attacks [7–9]. In addition, standardization bodies have designed a public key infrastructure to offer V2X security services, especially authentication, integrity, and confidentiality [10]. Standard specifications also defined message formats and all cryptography tools to sign and encrypt V2X messages. Cryptographic solutions allow for avoiding a significant vector of attacks, specifically external attacks launched by non-authenticated members. However, more aggressive attacks launched by internal attackers persist. More specifically, internal attacks such as Denial of Service (DoS), position falsification, and message droppings pose a real danger since attackers are already part of the network, making them resistant to cryptographic solutions [11]. In addition, attackers have become more intelligent and strategic in overcoming the defense lines [12]. In this context, misbehavior detection systems have been proposed as complementary to PKI to detect such attacks and exclude attackers from the 5G-V2X system. However, detecting these attackers is challenging and requires employing sophisticated and intelligent detection mechanisms.

Machine learning has recently emerged as a key intelligence enabler for future networks. It becomes obvious that ML algorithms will be one of the pillars of 5G and beyond and 6G mobile networks [13]. In addition, ML algorithms have already proven their success in network security [14]. Consequently, several ML-based MDSs have been proposed to detect attacks on 5G-V2X.

Figure 1 is a result of the quantitative study of the papers

A. Boualouache and T. Engel are with the Faculty of Science, Technology and Medicine (FSTM) at the University of Luxembourg, Esch-sur-Alzette, AVE, 4365, Luxembourg. E-mail: {abdelwahab.boualouache, thomas.engel}@uni.lu

TABLE 1: Abbreviations used throughout the paper.

Abbr	Description	Abbr	Description
3GPP	The 3rd Generation Partnership Project	MNO	Mobile Network Operator
5G	The 5th generation mobile network	NB	Naive Bayes
5GB	5G and Beyond	NEF	Network Exposure Function
AF	Application Function	NFV	Network Function Virtualization
AHC	Agglomerate Hierarchical Clustering	NFVI	NFV Infrastructure
AMF	Access and Mobility Management Function	NGAP	NG Application Protocol
ANN	Artificial Neural Network	NR	New Radio
AODV	Ad hoc On-Demand Distance Vector	NRF	Network Repository Function
ARP	Address Resolution Protocol	NS	Network Slicing
AUC	Area Under the Curve	NS2/3	Network Simulator Version 2/3
AUSF	Authentication Server Function	NSSF	Network Slice Selection Function
AoA	Angle of Arrival	OBU	On-Board Unit
BSM	Basic Safety Message	OMNeT++	Objective Modular Network Testbed in C++
BTP	Basic Transport Protocol	PDCP	Packet Data Convergence Protocol
C-V2X	Cellular Vehicle-to-Everything	PDR	Packet Delivery Ratio
CACC	Cooperative Adaptive Cruise Control	PDrR	Packet Drop Rate
CAM	Cooperative Awareness Message	PHY	PHYsical layer
CAV	Connected and Automated Vehicle	PKI	Public Key Infrastructure
CNN	Convolutional Neural Network	PMR	Packet Modification Ratio
CPF	Control Plane Function	R2L	Remote-to-Local
CPM	Collective Perception Message	RF	Random Forest
CPS	Collective Perception Service	RLC	Radio Link Control
DDoS	Distributed Denial of Service	RNN	Recurrent Neural Network
DENM	Decentralized Environmental Notification Message	ROC	Receiver Operator Characteristic
DL	Deep Learning	RSSI	Received Signal Strength and interference
DR	Detection Rate	RSU	Roadside Unit
DoS	Denial of Service	RTS	Request To Send
ET	Extra Tree	SCTP	Stream Control Transmission Protocol
ETSI	European Telecommunications Standards Institute	SDN	Software Defined Networking
FL	Federated Learning	SEPP	Security Edge Protection Proxy
FN	False Negative	SLR	Systematic Literature Review
FNR	False Negative Rate	SMF	Session Management Function
FP	False Positive	SQL	Structured Query Language
FPR	False Positive Rate	SST	Singular Spectrum Transformation
GAN	Generative Adversarial Network	SUMO	Simulation of Urban MObility
GPRS	General Packet Radio Service	SVM	Support-Vector Machine
GPS	Global Positioning System	TCP	Transmission Control Protocol
GRU	Gated Recurrent Unit	TN	True Negative
GTP	GPRS Tunneling Protocol	TNR	True Negative Rate
HTTP	HyperText Transfer Protocol	TP	True Positive
IBL	Instance-Based Learning	TPR	True Positive Rate
ICMP	Internet Control Message Protocol	U2R	Remote-to-Local
IEEE	Institute of Electrical and Electronics Engineers	UDM	Unified Data Management
IP	Internet Protocol	UDP	User Datagram Protocol
KNN	k-Nearest Neighbors	UPF	User Plan Function
LGBM	Light Gradient Boosting Machine	URLLC	Ultra-Reliable Low Latency Communications
LLC	Logical Link Control	V2I	Vehicle-to-Infrastructure
LR	Logistic Regression	V2N	Vehicle-to-Network
LSTM	Long Short-Term Memory	V2P	Vehicle-to-Pedestrian
LTE	Long-Term Evolution	V2V	Vehicle-to-Vehicle
MAC	Media Access Control	V2X	Vehicle-to-Everything
MANO	MANagement and Orchestration	VANET	Vehicular Ad-Hoc Network
MDS	Misbehavior Detection System	VNF	Virtual Network Function
MEC	Multi-access Edge Computing	VRU	Vulnerable Road Users
ML	Machine Learning	VUE	Vehicular User Equipment
MLOps	ML Operations	WiFi	Wireless Fidelity

surveyed in this paper. It shows the number of published papers on ML-based MDSs per year. As can be seen, recent years have witnessed a notable increase in the number of proposed ML-based MDSs. This is due to the trust of the research communities in ML for providing efficient and evolving MDSs for 5G-V2X [15]. However, analyzing these ML-based MDSs is still a research gap. Therefore, this survey comes to fill this gap and complements ongoing research and standardization activities on MDSs for V2X. [16, 17]. This survey analyzes existing ML-based MDSs not only from security but also from ML perspectives. Thus, it establishes analysis guidelines and presents learned lessons and recommendations for future ML-based MDSs. It also identifies open research and standardization gaps that need attention and priority to deploy ML-based MDSs successfully. Table 1 describes abbreviations used throughout the paper.

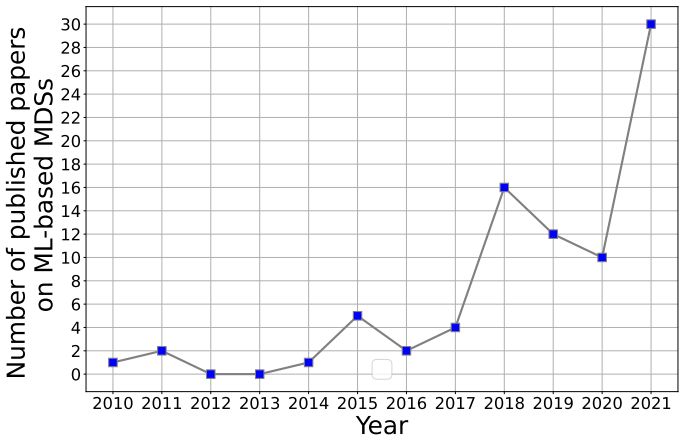


Fig. 1: Number of published papers on ML-based MDSs per year

Research method

The method used in this survey is a Systematic Literature Review (SLR) [18], which consists of five steps: research questions, literature retrieval, literature evaluation, data extraction, and results and discussion. Generally, three key questions lead our research: (i) "What are the ML-based MDSs exist for 5G and Beyond (5GB) vehicular networks?", (ii) "What are the features and limitations of the existing ML-based MDSs for 5GB vehicular networks?", and (iii) "What are the open research issues regarding ML-based MDSs for 5GB vehicular networks". Based on these research questions, we have used a combination of many keywords in the search engines. Specifically, we have combined three sets of keywords. The first set includes keywords regarding misbehavior detection, such as "misbehavior detection" or "intrusion detection", "attack detection", and "anomaly detection. In the second set, we have used key works to refer to machine learning, such as "machine learning", "artificial intelligence", or "deep learning". Finally, the third set includes keywords regarding vehicular networks such as "5G vehicular networks", "internet of vehicles", "connected vehicles", "autonomous vehicles", "connected and automated vehicles", "Vehicular Ad-Hoc Network (VANET)", and "VANET".

Around 200 research papers related to the research topic were found in the initial search. But after reading, papers were removed from the literature review if their contents were unrelated to our research questions. Specifically, Table 2 describes inclusion and exclusion criteria. This survey primarily includes a paper if several criteria meet: (i) the paper should be written in English and keywords appear in its title or abstract, (ii) the paper should be published in journals, books, or proceedings of conferences, symposiums, or workshops; and (iii) the paper should focus on detecting misbehaviors in vehicular communications. However, the same paper was excluded if an incomplete study or ML is used in other networking aspects but not in misbehavior detection. The paper was also excluded if ML was mentioned for detecting misbehaviors but not elaborated on or if methods other than ML were used. At the end of the SLR, 92 papers were left and were the focus of this survey.

TABLE 2: Inclusion and exclusion criteria

Inclusion Criteria	Exclusion Criteria
Keywords appeared in the title or abstract	Incomplete study
Written in English	ML is used in other networking aspects, but not in misbehavior detection
Published in journals, books, or proceedings of conferences, symposiums, or workshops	The use of ML is mentioned for detecting misbehaviors but not elaborated
Focus on detecting misbehavior in vehicular communications	The use of ML was mentioned for detecting misbehaviors but methods other than ML were used

Relevant surveys

Several surveys have been conducted on security and privacy in vehicular networks. The authors of [28] highlighted security challenges in the V2X environment. They also identified various cybersecurity risks and vulnerabilities and analyzed corresponding defense strategies for securing CAVs. The authors of [26] classified the available defense mechanisms into four categories: cryptography, network security, software vulnerability detection, and malware detection. The authors of [19] surveyed possible attacks and the corresponding detection mechanisms. The authors of [20, 27] reviewed detection schemes of data falsification attacks. The authors of [21] reviewed various MDSs and classified them into three different categories. The authors of [22] clearly defined V2X misbehavior. They also reviewed different MDSs and provided a comprehensive classification of the existing MDSs. However, all previous surveys have reviewed MDSs in general without focusing on ML aspects. The authors of [31, 32] discussed the role of ML in enabling efficient cybersecurity defense mechanisms for V2X. The authors of [29] classified the ML techniques according to their use in V2X applications and discussed approaches and working principles of these ML techniques in addressing various security challenges. The authors of [23] survey MDSs targeting only three communication attacks: false information, blackhole, greyhole and wormhole attacks, and DoS with explicit indication of MDSs that are based on ML. The authors of [24] surveyed ML-based MDSs detecting

TABLE 3: A comparison between this survey and relevant surveys on security for V2X

Year	Survey	Targeted attacks	Focus on MDS	ML Context	ML analyzes	Security analyzes	Recommendations and Open Issues
2017	[19]	Sybil, (D)DoS, Blackhole Wormhole, Position falsification, False information GPS spoofing, Replay, Eavesdropping	-	-	-	-	-
2018	[20]	False information	-	-	-	-	-
2018	[21]	(D)DoS, Position falsification, Tunneling Blackhole and Greyhole, False information Sybil, Replay, Impersonation	X	Limited	-	-	-
2018	[22]	Jamming, False information, Replay Blackhole and Greyhole, Sybil	X	-	-	-	-
2018	[23]	False information, (D)DoS Wormhole, greyhole/blackhole	X	Very Limited	-	-	-
2019	[24]	(D)DoS	X	X	X	-	-
2019	[25]	False information, (D)DoS, Blackhole and Greyhole Impersonation, Wormhole, Sybil	X	X	X	-	-
2020	[26]	(D)DoS, Blackhole, Replay Sybil, Impersonation, False information	-	-	-	-	-
2020	[27]	False information	-	-	-	-	-
2021	[28]	(D)DoS, Impersonation, Reply, Eavesdropping Blackhole and Greyhole, False information	-	-	-	-	-
2021	[29]	Impersonation, Position tracking, Eavesdropping (D)DoS, Jamming, Blackhole and Greyhole False information, GPS spoofing, Position falsification	-	X	X	-	-
2021	[30]	Eavesdropping, Jamming, Impersonation GPS spoofing, False information, Blackhole and Greyhole	X	X	X	-	-
2022	This survey	False information, Position falsification, Eavesdropping Impersonation, GPS spoofing, Blackhole and Greyhole (D)DoS, Wormhole, Jamming, Timing, Sybil, Position tracking, Tunneling, Reply	X	X	X	X	X

only DDoS attacks. The authors of [25] presented an SLR for some ML-based MDSs for V2X along with their ML algorithms, architectures, and datasets. The authors of [30] also reviewed some ML-based MDSs for V2X.

Table 3 compares this survey with relevant V2X security surveys considering different criteria: (i) the attacks targeted by the survey; (ii) whether the survey focuses on the MDS within the ML context; (iii) whether the survey analyzes ML-based MDSs from the ML and security perspective; (iv) whether the survey describes recommendations and open issues for building efficient ML-based MDSs. Only a few relevant surveys focus on MDSs while considering the ML context. In addition, although previous surveys cover some ML-based MDSs, they lack deep analysis from both ML and security perspectives. Furthermore, relevant surveys miss recommendations and open issues from building efficient ML-based MDSs. Thus, to complement these efforts and unlike previous surveys, this survey presents a wide-coverage and comprehensive review of existing ML-based MDSs for 5G-V2X. This survey includes in-depth technical analyzes from both ML and security perspectives. It also highlights recommendations and open issues to fill research and standardization gaps. To the best of our knowledge, we are the first to propose such a survey. We hope this survey will build guidelines to select the best ML-based MDSs to implement in the near deployment of 5G-V2X and shape future research directions on this topic.

Contributions

The main contributions of this paper can be summarized as follows:

- Survey and elaborate taxonomy of machine learning-based misbehavior detection systems.
- Perform in-depth technical analyzes from both ML and security perspectives of ML-based MDSs.
- Present lessons learned and recommendations for developing, evaluating, and deploying ML-based MDSs.

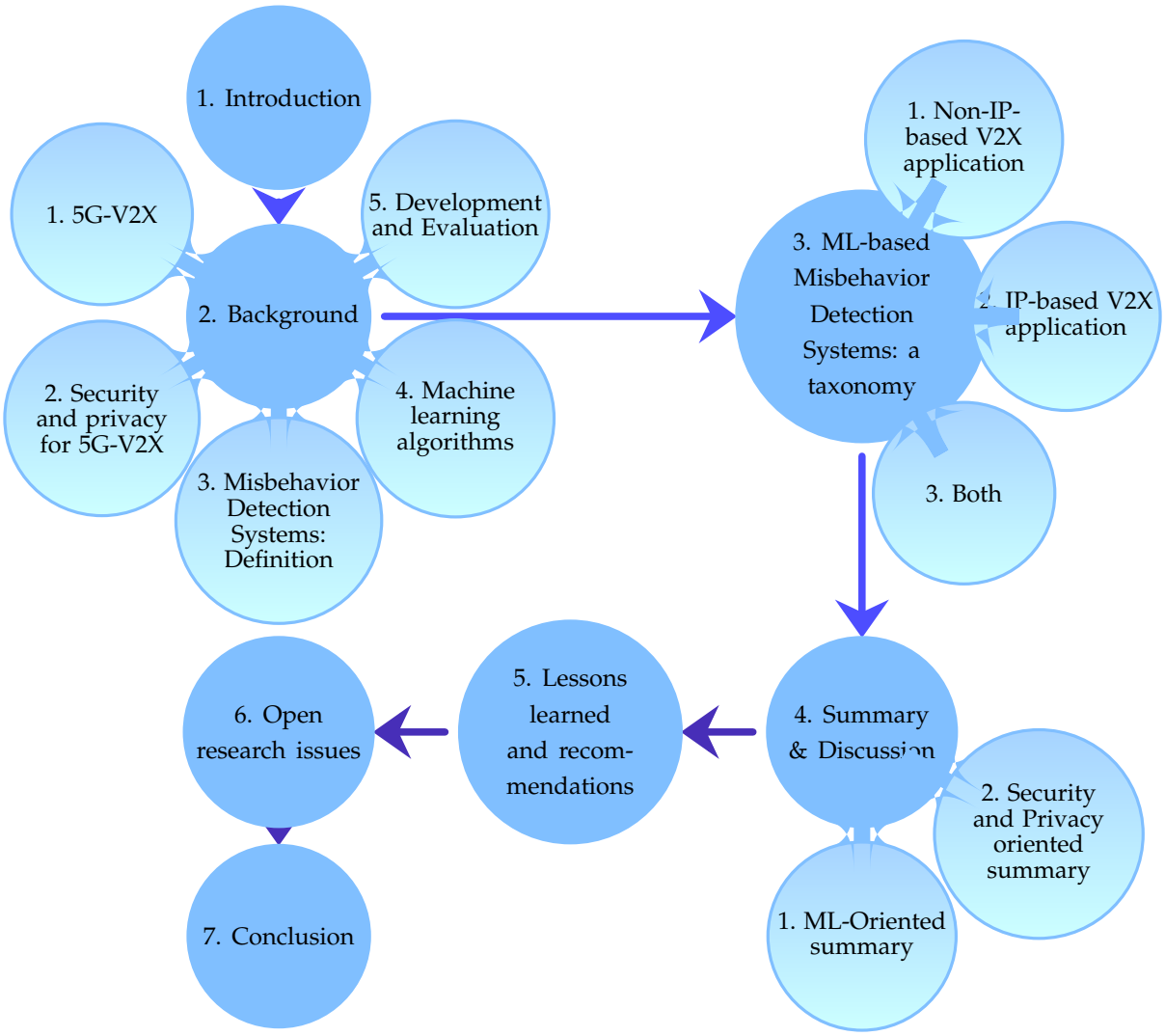


Fig. 2: Survey's roadmap

- Highlight open research and standardization issues on the topic.

The rest of the paper is organized as follows. Section 2 presents some necessary background information. A taxonomy of machine learning-based misbehavior detection systems for 5GB vehicular networks is presented in Section 3. Section 4 analyzes and discusses the presented ML-based MDSs. Lessons learned and recommendations are discussed in Section 5. Open research issues are given in Section 6. Finally, Section 7 concludes this survey. The roadmap of this survey is given in Figure 2.

2 Background

The purpose of this section is to give the reader the necessary background information to understand the research presented in this paper. This section is divided into five subsections. Firstly, it describes the architecture of 5G-V2X. Next, it overviews security requirements, attacker models, and attacks on 5G-V2X. Then, the definition of the MDS is given. After that, overviews various ML techniques and concepts. Finally, it describes the development and evaluation elements of ML-based MDSs.

2.1 5G-V2X

Figure 3 illustrates a CAV equipped with a V2X communication interface and several installed sensors such as Radar, OBU, Lidar, Camera, ultrasonic sensors, and Global Positioning System (GPS). Three principle essential services are enabled in CAVs: (i) the broadcast service allows broadcasting a message called Cooperative Awareness Message (CAM) in EU [33], and Basic Safety Message (BSM) in the US [34]. Both of these messages contain state information of vehicles such as their position, speed, acceleration, etc.; (ii) the decentralized environmental notification service allows triggering a Decentralized Environmental Notification Message (DENM), which includes information related to a road hazard or abnormal traffic conditions, such as its type and its position [35]; and (iii) the Collective Perception Service (CPS) which allow sharing sensor data between CAVs about non-connected objects such as non-V2X vehicles, obstacles, pedestrians, and animals. This service enables generating and consuming a message called Collective Perception Message (CPM), acting as a complement to messages generated in the two previous services [36, 37]. This section describes the main building blocks of 5G-V2X.

2.1.1 Communication technologies

Several communication technologies have been suggested for V2X communications. Two leading technologies have been designed and customized to support V2X communications and enable direct information sharing between CAVs. Currently seen as alternatives to each other, these technologies are IEEE 802.11p (ITS-G5 in Europe) and Cellular Vehicle-to-Everything (C-V2X) [38]. However, ITS-G5 has known a slow development on a wide scale in favor of C-V2X, which is witnessing significant growth led by the 3rd Generation Partnership Project (3GPP) [39].

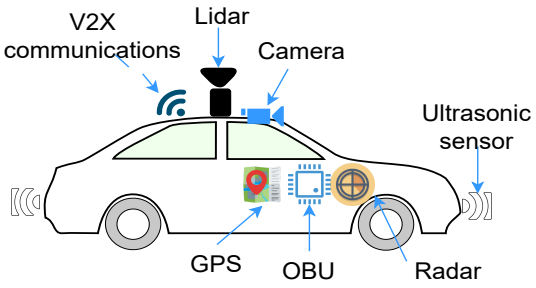


Fig. 3: An example of a connected and automated vehicle

C-V2X is already part of the completed 3GPP Long-Term Evolution (LTE) Releases 14 and 15 [40]. It is designed to support URLLC for V2X use case groups specified Release 16 [41]. This subsection describes the protocol stacks of ETSI ITS-G5 and C-V2X, respectively.

2.1.1.1 ETSI ITS-G5 protocol stack

ETSI ITS-G5 is based on physical and Media Access control (MAC) layers defined in the IEEE 802.11p protocol. The 802.11p modifies the physical and MAC layers of 802.11a to be adapted for V2X communications in a frequency band from 5.85 to 5.925 GHz, which is segmented into seven channels of 10 MHz each. ITS-G5 standard uses the decentralized congestion control protocol to minimize the probability of radio channel congestion. As shown in Figure 4 (a), for the Internet Protocol (IP)-based applications (non-safety applications), ITS-G5 uses the IP for the network layer and the User Datagram Protocol (UDP)/Transmission Control Protocol (TCP) for the transport layer. On the other hand, for non-IP-based applications (safety applications), ITS-G5 uses the Geonetworking protocol to enable packets' routing based on the geographic position of vehicles in the network layer [42], while the Basic Transport Protocol (BTP) is used to offer point-to-point connectionless network transport service in the transport layer [43]. ITS-G5 also introduces the facilities layer between the transport layer and the applications layer, where several messages were defined [44]. For example, the CAM was defined for the periodic messages, and the DENM was defined for the event messages.

2.1.1.2 C-V2X User Plan Stack

As shown in Figure 4 (b), the protocol stack of C-V2X via PC5 interface is mainly based on the 3GPP Releases for the low layers (PHY, MAC, RLC, and PDCP) and reuses the layer stacks from the Institute of Electrical and Electronics Engineers (IEEE) and European Telecommunications Standards Institute (ETSI) for the upper layers (network and transport layers) [45]. The PHYsical layer (PHY) transmits data on the sidelink, exploiting 10 MHz or 20 MHz bandwidths at the 5.9 GHz radio frequency band. The MAC layer implements the blind hybrid automatic repeat request without feedback. The Radio Link Control (RLC) layer is in charge of delivering service data units in sequence and segmenting and reassembling them. The Packet Data Convergence Protocol (PDCP) sub-layer separates 3GPP radio access protocol layers from those related to V2X applications [46]. As shown in Figure 4 (c), The protocol stack of C-V2X via Uu link for V2N is common for communications in 5G architecture.

2.1.2 Architecture

V2X communication types are classified as follows [47]: (i) Vehicle-to-Vehicle (V2V) for direct communications between Vehicular User Equipment (VUEs); (ii) Vehicle-to-Infrastructure (V2I) for communications between vehicles and the RSUs, which can be deployed as radio base stations in 5G networks (gNodeBs) or in standalone devices; (iii) Vehicle-to-Pedestrian (V2P) between VUEs and Vulnerable Road Users (VRUs) such as pedestrians and bikers; and (iv) Vehicle-to-Network (V2N) for communications with remote servers and cloud-based services reachable through the cellular infrastructure. The enhancement of 3GPP to support C-V2X communications concerns the radio access network (the New Radio) and the core network.

2.1.2.1 5G-NR V2X

Several enhancements are introduced in Release 16 within New Radio (NR) to support V2X applications' demands in terms of latency and reliability [41]. These enhancements range from introducing more disruptive radio technologies (e.g., flexible waveforms) to improving specific communication modes (e.g., multicast and groupcast). V2X NR covers both the PC5 and Uu radio interfaces. The PC5 radio interface (sidelink) is used for V2V, V2P, and V2I communications, bypassing the cellular infrastructure. In the absence of a cellular network, the 5.9 GHz radio frequency band is employed to ensure ultra-high availability in all regions, regardless of the Mobile Network Operator (MNO). 5G-V2X supports two resources allocation modes for PC5 for sidelink V2V communications [48]: Mode 3 (scheduled) – operates in coverage of a gNodeB, which is in charge of the allocation of radio resources, and Mode 4 (autonomous) – can operate both in- and out-of-coverage of an eNodeB, where the allocation of radio resources is agreed between vehicles without support for the infrastructure. V2N communications occur over the conventional cellular Uu interface operating in the licensed spectrum. This interface has been modified to handle unicast and multicast V2X communications with fewer changes that enable efficient V2X information sharing to meet the latency requirements of V2X applications.

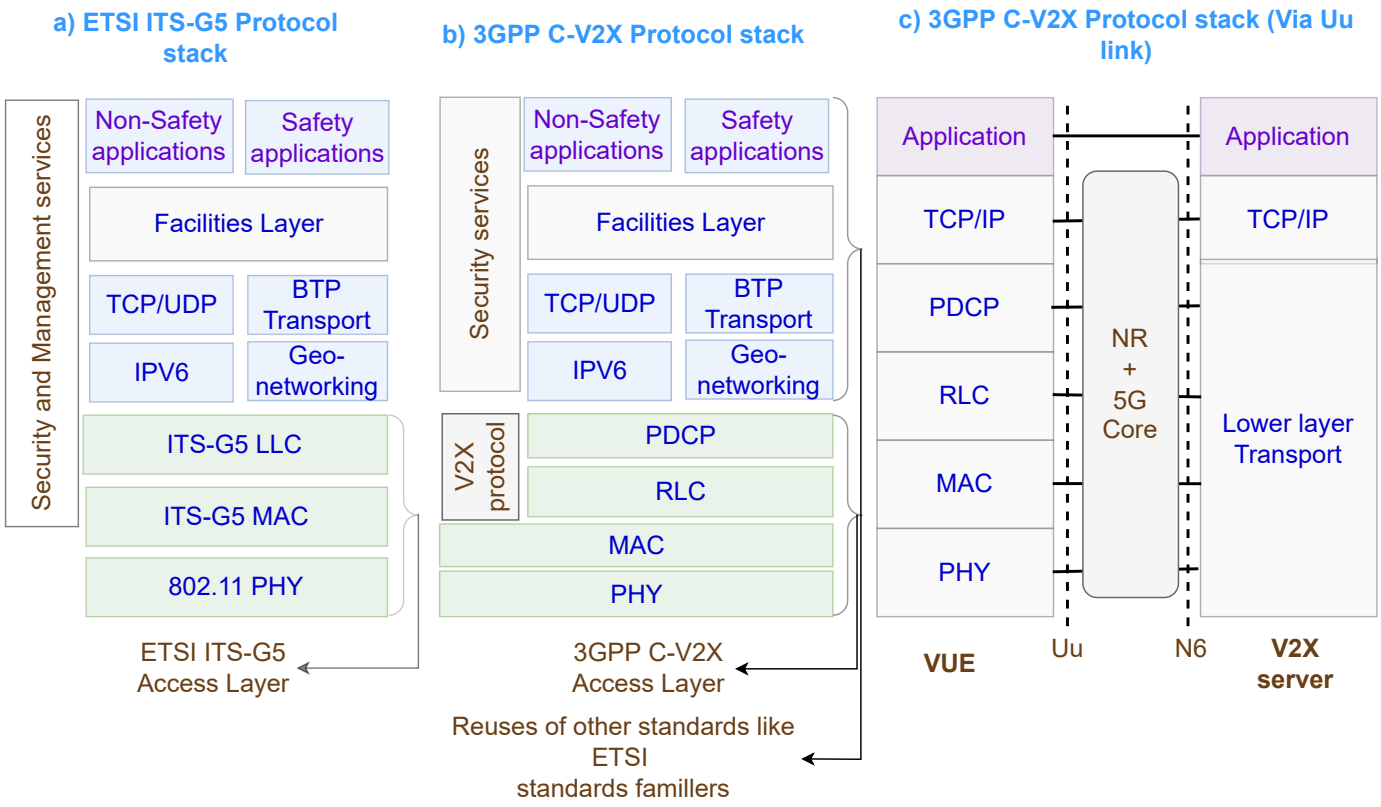


Fig. 4: V2X protocol stacks [45, 46]

2.1.2.2 5G Core Network

The 5G core network is designed to enable mobile data connectivity and support various verticals leveraging emerging technologies, such as SDN and NFV. By separating the User Plane Function (UPF) from the Control Plane Function (CPF), the 5G core becomes scalable and flexible. The building blocks of the 5G core are a set of Virtual Network Functions (VNFs), including the Authentication Server Function (AUSF), Access and Mobility Management Function (AMF), User Plan Function (UPF), Session Management Function (SMF), Network Slice Selection Function (NSSF), Unified Data Management (UDM), Application Function (AF), Network Repository Function (NRF), Network Exposure Function (NEF), and Security Edge Protection Proxy (SEPP).

Figure 5 shows a high-level view of the 5G-V2X architecture for V2X communication over PC5 and Uu reference points.

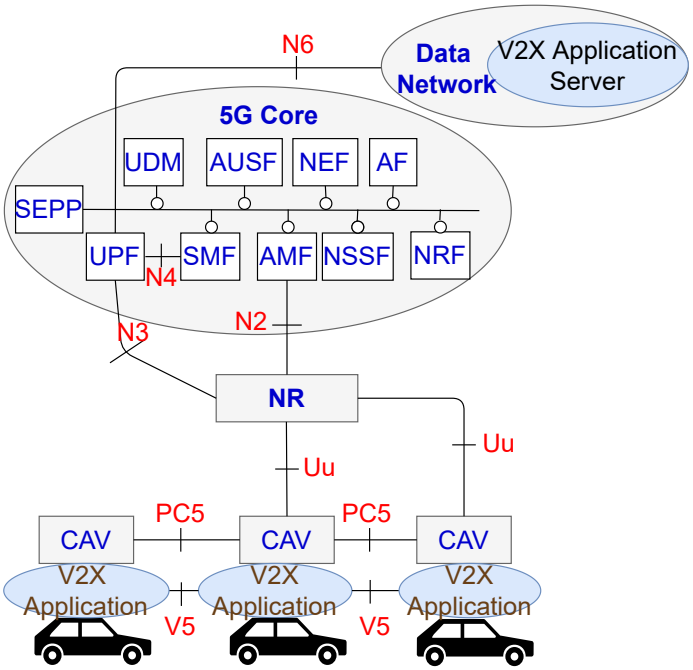


Fig. 5: The 5G-V2X architecture and its communication interfaces

2.1.3 CAV supporting technologies in 5G

In addition to 5G-V2X communication technology, CAVs are supported by other 5G enabling technologies. This subsection describes the key 5G enabling technologies supporting CAVs.

2.1.3.1 SDN

SDN is a paradigm that allows a network to be managed and controlled with a logically-centralized approach, separating the control and data planes. SDN brings programmability and flexibility to manage vehicular networks efficiently [49].

2.1.3.2 NFV

NFV allows virtualizing network services that are traditionally run on proprietary hardware, such as routers, firewalls, and load balancers. These services are packaged as virtual machines or containers on commodity hardware, which allows service providers to run their network on standard servers instead of proprietary ones [50]. NFV enables virtualizing 5GB vehicular network services, making them easy to manage and allocate [51].

2.1.3.3 Multi-access Edge Computing (MEC)

MEC moves services from a centralized cloud to the network's edge closer to the customer. MEC enables the collecting and processing of data close to the customer, reduces latency, and gives high-bandwidth applications real-time performance. The emergence of SDN and NFV has facilitated the deployment of MEC and accelerated its adoption. MEC brings several benefits to 5GB vehicular networks, including deploying highly virtualized computing services and offloading data processing tasks to MEC nodes near CAVs.

2.1.3.4 NS

NS aims to create multiple virtual networks on the same shared and programmable physical infrastructure. Network slicing then increases the exploitation degree of 5G physical infrastructure while boosting the performance of applications and services. For CAVs, NS can enable several 5G-V2X networks with different requirements to co-exist and operate together while enjoying isolation [39].

2.1.3.5 Mobility Management (Handover and Roaming)

Due to their high and dynamic mobility, CAVs are one of the most demanding 5GB verticals regarding mobility management. CAVs moving on roads switches their attachment to the network from one cell to another. Transferring data sessions from one cell to another cell is called handover. Handovers can be horizontal or vertical. Horizontal handovers are defined as handovers within the same access networks. In addition, vertical handovers are defined as handovers across heterogeneous access networks. CAVs frequently perform handovers, and 5GB procedures should provide seamless handovers to ensure V2X service continuously and URLLC requirements [52, 53].

On the other hand, in areas where the borders are open, like the European Schengen space, CAVs often cross country borders triggering inter-public land mobile network handover to switch from the home network to the visited network. In the roaming state, CAVs are served by the visited MNO, ensuring service and session continuity, 5GB core/MEC interconnection, and V2X application state transfer to meet the requirements [54].

2.1.4 Use case groups

3GPP technical report 22.886 [55] presents a comprehensive description of the envisaged 5G-V2X use case groups, which are given as follows [56]:

2.1.4.1 Vehicle Platooning

This group includes use cases that enable forming groups of vehicles in platoons while maintaining their functioning through the periodic exchange of messages. Vehicle platooning can be either centralized or decentralized. In the centralized case, the first vehicle in the platoon (the leader) controls the driving decisions of the rest (the followers). Moreover, in the decentralized case, each vehicle takes its driving decision according to information from other vehicles in the platoon.

2.1.4.2 Advanced Driving

This group comprises use cases allowing semi- or fully-automated driving while ensuring traffic efficiency and road safety. Specifically, advanced driving enables managing complicated maneuvers such as lane merging or overtaking, or cooperative collision avoidance, requiring sharing driving intentions with CAVs nearby. In such use cases, CAVs coordinate their trajectories or maneuvers by sharing driving plans and data obtained from their local sensors with CAV and/or MEC nearby.

2.1.4.3 Extended Sensors

This group aims to improve vehicle perception by exchanging data collected from different data sources, such as local sensors, RSUs, MECs, and VRUs. This group's use cases help CAVs to have a more holistic view of the local situation by providing an enhanced perception of the environment beyond what their sensors can detect. For example, a leading CAV can use a camera to capture the road in front of CAVs and continuously stream a video to the following CAVs with the help of MECs. Such use cases help to extend the perception of CAVs and greatly improve road safety in the case of complex manoeuvers on two-way roads.

2.1.4.4 Remote Driving

This group enables to drive CAVs remotely. In a basic remote driving scenario, remote human operators receive real-time data streams from CAV's sensors to create and send back commands over V2N links for controlling CAVs. Remote driving enables several use cases, including remote assistance to beginner drivers to overcome difficult road situations, facilitating autonomous public transportation services with predefined routes and stops, and supporting highly automated vehicles to perform complex maneuvers and overcome unfamiliar navigation environments.

2.2 Security and privacy for 5G-V2X

This section is divided into three subsections: security requirements, attacker models, and attack classification.

2.2.1 Requirements

5G-V2X communications are subject to an extensive range of attacks and cyber-threats, which can have significant negative consequences on the integrity and functionality of the 5G-V2X system, potentially putting drivers' lives at risk. Moreover, protecting privacy is crucial because the

lack of privacy may disturb the public acceptance and the successful deployment of 5GB vehicular networks [57]. This subsection discusses the security and privacy services for 5G-V2X networks.

2.2.1.1 Authentication

Authentication is a procedure or process for certifying the identity of users with the objective of authorizing access to resources and/or privileges. Entity authentication is required in 5GB-V2X networks to prevent unauthorized users from injecting fake messages. Apart from entity authentication, data authentication is also necessary to verify that the received data is not tampered with or replayed.

2.2.1.2 Integrity

Integrity allows for ensuring the accuracy, consistency, and immutability of data against malicious operations aimed at altering them. Data integrity is a must to protect drivers from malicious V2X participants, which can act to change data exchange between V2X nodes or can generate rogue messages similar to benign messages for affecting network operations.

2.2.1.3 Availability

Availability ensures that systems and services are available to users when they need them. Specifically, it comprises not only timely and reliable access to systems and services when required but also continuous availability for the time it is needed. In 5G-V2X networks, availability ensures that V2X messages are delivered to all of the intended recipients at the right moment. It also guarantees the continuity of V2X services.

2.2.1.4 Confidentiality

Confidentiality is the process of preventing unauthorized users from accessing sensitive information. In 5G-V2X networks, confidentiality ensures that only authorized V2X participants can access the exchanged data. However, applying confidentiality can add an overhead of processing data in time-sensitive 5G-V2X applications. Thus, using this security service highly depends on 5G-V2X use cases and the nature of the data to protect.

2.2.1.5 Non-repudiation

Non-repudiation protects against users who deny having performed specific actions such as creating, sending, and receiving messages using mechanisms checking if users took those actions or not. In 5G-V2X networks, non-repudiation is necessary to prevent legitimate V2X participants from denying the transmission or reception of the content of their messages.

2.2.1.6 Access control

Access control is the process of approving and rejecting specific requests to access and use information and services. Access control is necessary to ensure the system's reliability and security. The 5G-V2X system should be able to quickly revoke misbehaving V2X nodes from the system to protect the safety of legitimate V2X participants.

2.2.1.7 Privacy

Privacy is one of the fundamental human rights protected by laws. In 5G-V2X networks, users' personal and private data should be protected against such interference or attacks. However, many operations can violate users' privacy in 5GB-V2X. Confidential information regarding the driver and passengers must be protected, such as the number of passengers, their names, and their destinations can be indirectly obtained by monitoring CAVs' communications.

2.2.2 Attacker model

Because of the V2X system's intricacy, different adversaries can launch various attacks. The authors of [58] have thoroughly examined potential adversaries in V2X in [58] and identified the following types of attackers:

- **Global versus. Local:** A global attacker has a wider coverage of the V2X system than a local attacker. It can then eavesdrop on every message sent out by any vehicle.
- **Active versus. Passive:** An active attacker can alter or inject messages into the V2X system, while a passive attacker can only eavesdrop on messages.
- **Internal versus External:** An internal adversary is an authenticated participant of the V2X system, while an external adversary is an intruder.
- **Malicious versus rational:** A malicious attacker aims to damage the V2X system without caring about its interests, while a rational attacker aims to achieve its interests while performing attacks.

2.2.3 Attack classification

This subsection classifies and describes traditional attacks on 5GB vehicular networks. Figure 6 shows an overview of these attacks, while Table 4 specifies the applications targeted by these attacks and the types of attackers (internal or external) that can launch them. The next subsection focuses on attacks and threats on vehicular networks posed by 5G enabling technologies.

2.2.3.1 Attacks on authentication

- **Sybil**: This attack mainly concerns non-IP-based V2X (safety-related) applications where vehicles use multiple identifiers to protect their location privacy. However, these identifiers can also be exploited as Sybils, for example, to inject false information into the V2X system to alter the perception of vehicles or to create the illusion of traffic congestion.
- **Impersonation or masquerading**: The attacker exploits a valid identity to obtain V2X access for launching more advanced attacks and stealing private information. More specifically, this attack mainly exploits the vulnerabilities in IP-based applications to get remote access to the V2X node through a

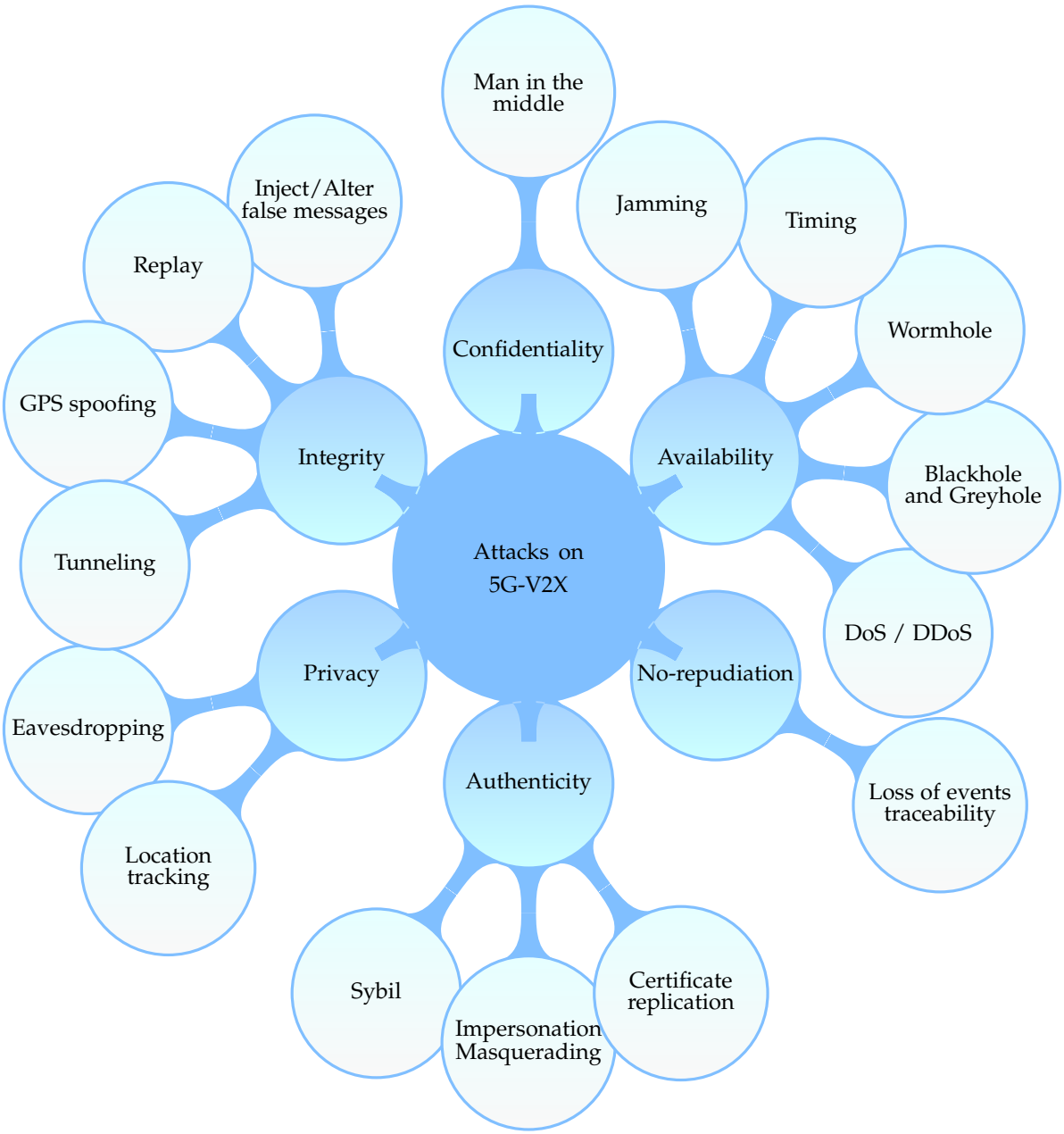


Fig. 6: 5G-V2X Attack classification

multi-stage process starting with probing and port scanning to network and application layers exploits such as Malware, Structured Query Language (SQL) injection, and DNS poisoning.

- **Certificate replication**: In this attack, malicious V2X nodes try to hide their identities by utilizing replicated certificates. Malicious nodes will no longer use certificates who blacklisted.

2.2.3.2 Attacks on integrity

- **Inject/Alter false messages**: In this attack, malicious nodes send wrong information (e.g., position, speed, etc.) to honest vehicles, which may put them in dangerous situations. This attack could be more likely in non-IP-based applications.
- **Replay**: In this attack, malicious V2X nodes replay messages captured at different times and show them as generated by original senders.
- **GPS spoofing**: Malicious nodes deceive GPS receivers of other V2X nodes by re-transmitting real GPS signals captured elsewhere at a different time or by transmitting inaccurate GPS signals.
- **Tunneling**: In this attack, the attacker controls at least two V2X nodes to establish a tunnel between them; hence, it can inject false data from one place to another. Tunneling can be considered a special case of a false message injection attack.

TABLE 4: Specification the type of attackers and the applications targeted by 5G-V2X attacks

Security Service	Attack	Non-IP-based V2X application	IP-based V2X applications	External	Internal
Authenticity	Impersonation or masquerading	X	X	X	X
	Sybil attack	X			X
	Certificate replication	X	X		X
Integrity	Inject/ Alter false messages	X	X		X
	Replay	X	X	X	X
	GPS spoofing	X	X	X	X
	Tunneling		X		X
Availability	Deny of service (DoS)	X	X	X	X
	Blackhole and Greyhole	X	X		X
	Jamming attack	X	X	X	X
	Wormhole		X		X
Confidentiality	Man in the middle	X			X
Non-repudiation	Loss of events traceability	X	X		X
Privacy	Eavesdropping	X	X	X	X
	Location tracking	X		X	X

2.2.3.3 Attacks on availability

- **DoS/Distributed Denial of Service (DDoS):** This attack prevents vehicles from having normal access to network services. A DDoS attack is a DoS attack variant involving a set of malicious V2X nodes. Both IP-based and non-IP-based V2X applications are vulnerable to DoS attacks. In non-IP-based applications, the DoS attacks can be achieved by increasing the frequency of periodic messages. In contrast, the DoS attack can be performed at different levels in IP-based applications, such as UDP flooding and Address Resolution Protocol (ARP) flooding.
- **Blackhole and Greyhole**: In these attacks, malicious V2X nodes stop disseminating received messages to the neighboring V2X entities. While in the greyhole only selected messages are dropped, in the blackhole attacker drops all received messages. These attacks mainly concern IP-based routing protocols (e.g., Ad-hoc On-Demand Distance Vector (AODV)) and position-based routing protocols (e.g., GeoNetworking).
- **Wormhole**: Similar to the tunneling attack, in the wormhole attack, attackers establish a tunnel between malicious V2X nodes to conduct a DoS attack disrupting IP-based routing protocols.
- **Timing**: In this attack, malicious V2X nodes intentionally delay forwarding the received messages to the following nodes in dissemination and routing protocols. This attack is hazardous, especially in time-sensitive safety-related applications.
- **Jamming**: In this attack, the attacker generates signals to corrupt the data or jam the radio channel. Both ETSI ITS-G5 and C-V2X standards are vulnerable to this attack.

2.2.3.4 Attacks on confidentiality

- **Man-in-the-middle**: The attacker establishes separate connections with the victims and passes messages between them to give the impression that they are in direct communication, but in fact, all conversations between the two victims are intercepted.

2.2.3.5 Attacks on Non-repudiation

- **Loss of events traceability**: In this attack, the attacker performs a set of actions to help in the denial of specified events. These actions mostly involve deleting its traces or causing confusion for the auditing entity.

2.2.3.6 Attacks on privacy

- **Eavesdropping**: Attackers gather data from the 5G-V2X networks to extract information from which they can benefit to identify drivers and passengers, track their trajectories, or use them for launching other attacks. The next attack will focus on trajectory tracking since attackers can easily eavesdrop to get position information from CAMs.
- **Location tracking**: CAMs aim to provide awareness for CAVs about their surrounding environment. They are sent in clear text and contain sensitive mobility information of CAVs, such as their identifier, position, speed, and acceleration. However, passive attackers can easily collect and exploit these messages for position and trajectory tracking [59–62]. Therefore, attackers will be able to know every location visited by drivers since this is a strong relationship between CAVs and their drivers. Trajectory tracking motivations vary and range from curiosity to criminal purposes.

2.2.4 Threats from 5G enabling technologies

While threats and attacks described in the previous section already pose serious security challenges to CAVs, 5G enabling technologies can worsen the situation by opening up new surfaces of threats and attacks [63]. This subsection describes the threats imposed by the main 5G enabling technologies to CAVs.

2.2.4.1 SDN-related threats

SDN provides 5GB vehicular networks with programmable and flexible network control through its application, control, and data layers. However, it also brought SDN threats to these networks. Specifically, each SDN layer has its security vulnerabilities [64]. The application layer is vulnerable to malicious application planting, data tampering, cross-application poisoning, and application eviction attacks. The control layer, the main component of SDN, is vulnerable to packet-in flooding attacks, unauthorized network control, and topology poisoning attack, among others. The data layer, including V2X nodes, is vulnerable to flow table overflow attacks, malicious control message injection, data leakage, and more [65].

2.2.4.2 NFV-related threats

By releasing network resources and functions from specialized physical equipment, NFV enables the physical network to support several logical networks. However, it makes the systems more vulnerable to data exfiltration, resource starvation, and side-channel attacks by widening the attack surface [66].

The NFV Infrastructure (NFVI), VNFs, and NFV MANagement and Orchestration (MANO) are the three key elements of the NFV architecture. NFVI is vulnerable to several attacks, such as compromised hypervisors and resource consumption attacks. VNFs are also vulnerable to attacks such as DoS and illegal VNF migration. In addition, the integrity and availability of VNFs and potentially the entire network may be affected by MANO threats such as sensitive data leakage and malicious manipulations [65].

2.2.4.3 MEC-related threats

Integrating MEC with 5GB vehicular networks increases the threat surface to these networks since MEC nodes are attractive targets for attacks. MEC nodes are vulnerable to DoS, making critical V2X safety available and putting CAVs in dangerous situations. Attackers can also gain access to MEC nodes, inject rogue hard-and software to run man-in-the-middle attacks, and derive private information regarding CAVs in proximity [67].

2.2.4.4 NS-related threats

5GB vehicular networks are vulnerable to several attack surfaces exposed by NS enabling technologies (SDN and NFV). As shown in the previous subsections, these enabling technologies also have vulnerabilities that attackers can exploit. In addition, attackers can use NS breaches to generate attacks against NS functioning, including [65] (i) Unauthorized Access, where attackers perform identity spoofing to gain unauthorized access to the V2X network slice. This attack is usually a prerequisite for achieving further attacks; (ii) DoS, where the attacker(s) overloads the target V2X network slice with high volumes of network traffic and requests, stopping the service provided by the slice. V2X Network slices, which share the same physical resources with the target network slice, could also be affected, leading to slice performance degradation and indirect DoS attack; and (iii) Cross-Slice Data Leakage: since CAVs can attach to multiple V2X network slices simultaneously, they can receive sensitive data on one V2X network slice and share it with other V2X network slices, occurring data leakage.

2.2.4.5 Mobility management-related threats

5G Mobility management procedures, including handover and roaming, can also bring attack surfaces to vehicular networks. Attackers can exploit inter-gNodeB handover authentication mechanisms vulnerabilities such as a false base-station attack, de-synchronization attack, or key compromise to break CAVs connectivity. Or, they can launch DoS during handovers to interrupt critical V2X safety services [68, 69]. Moreover, roaming crossing borders is sensitive due to the potential disparity of security levels between MNOs (home and visited), making interconnection points between MNOs easily exploitable by attackers.

2.3 Misbehavior detection systems: Definition

Several cryptography solutions have been proposed for thwarting V2X attacks. More specifically, standardization bodies have designed a PKI to offer V2X security services, especially authentication, integrity, and confidentiality [10]. Standard specifications define not only message formats but also all cryptography tools to sign and encrypt V2X messages [70]. However, although an important vector of attacks has been avoided using these solutions, attacks are still being performed, especially from internal attackers. In this context, misbehavior detection systems have been proposed complementary to PKI to detect attacks and exclude attackers from the V2X system.

In this survey, misbehavior refers to both faulty and malicious (intrusion) actions. Faulty nodes misbehave without malicious intent due to damage or other technical issues. For example, a malfunctioning vehicle's onboard GPS sensor can provide inaccurate position data. Moreover, malicious nodes or attackers act with malicious intent.

On the other hand, misbehavior detection systems can be classified into three groups described as follows [22]:

2.3.1 Node-centric

This category checks if the node's behaviors (e.g., message frequency and the ratio between received and forwarded packets) align with protocol specifications. They can be divided into two classes: (i) behavior-based: in which the attacker is detected in case of abnormal actions (e.g., message dropping), and (ii) Trust-based: in which trust values are assigned to V2X nodes. An attacker is detected if its trust value drops below a certain threshold. In this category,

ML models can be built based on statistical data on 5G-V2X nodes to detect misbehaviors.

2.3.2 Data-centric

This category focuses on the plausibility and consistency of data, which nodes can individually or collaboratively verify. These systems can also be divided into two classes: (i) Plausibility based: which use plausibility checks to decide on the correctness of data such as the received speed and position, and (ii) Consistency based: which inspect the relations between message to decide on the trustworthiness of newly received messages. For example, checking the difference between two received positions given a constant speed. In this category, ML models can be built based on the contents of messages exchanged between 5G-V2X nodes to detect misbehaviors.

2.3.3 Hybrid

This category adopts a combined approach that uses a node-centric system to evaluate nodes according to the correctness of the exchanged data, while the correctness of the information is verified using a data-centric mechanism. In this category, ML models can be built based on both statistical data on 5G-V2X nodes and the contents of messages exchanged between them to detect misbehaviors.

This survey reviews ML-based MDSs that detect misbehaviors (faults and intrusions). Data used by these ML-based MDSs can be from 5G-V2X node behaviors or/and data exchanged between 5G-V2X nodes.

2.4 Machine learning algorithms

"Machine learning is a branch of artificial intelligence (AI) and computer science which focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy" [71]. This section gives a brief introduction to various ML techniques and concepts that are used to build ML-based MDSs. Figure 7 summarizes these techniques and concepts. This section is divided into three subsections: traditional learning, Deep Learning (DL), and advanced ML concepts.

2.4.1 Traditional learning

Traditional learning refers to ML algorithms not based on DL, as explained in the following section.

2.4.1.1 Supervised learning

It is an ML approach that leverages labeled datasets to train or "supervise" algorithms in classifying data or predicting outcomes. Supervised learning can be separated into two types of problems: classification and regression:

- **Classification** problems use an algorithm to classify data into specific categories. For example, the problem of classifying safety messages into two groups: malicious or normal. Common classification algorithms are Naive Bayes (NB), Logistic Regression (LR), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), Artificial Neural Network (ANN), Extra Tree (ET), AdaBoost, Decision Stump, Ensemble learning (bagging, boosting, stacking), XGBoost, Light Gradient Boosting Machine (LGBM), Instance-Based Learning (IBL), and Ensemble voting.
- **Regression** problems use an algorithm to predict real or discrete input variables. For example, predicting a trust value of a V2X node or the number of attackers in the 5GB-V2X network. Common regression algorithms are linear and polynomial.

2.4.1.2 Unsupervised learning

In contrast to supervised learning, unsupervised learning uses unlabeled datasets for finding patterns that help understand data structure. Unsupervised learning can be classified into three types of problems: anomaly detection, clustering, and dimensionality reduction.

- **Clustering** is commonly used to organize data into groups that are easier to comprehend and manage. Common clustering algorithms are k-means, hierarchical, and Gaussian mixture models.
- **Anomaly detection** consists in identifying unexpected items or events in the dataset without any prior knowledge. Common anomaly detection algorithms are Elliptic Envelope Algorithm, Isolation Forest Algorithm, One-class SVM Algorithm, Singular Spectrum Transformation (SST), and Local Outlier Factor (LOF) Algorithm.
- **Dimensionality reduction** consists of transforming data from a high-dimensional space into a low-dimensional space while preserving some necessary information quantity from the original data. Common dimensionality reduction algorithms are Principal Component Analysis and Missing Value Ratio.

2.4.2 Deep learning

Deep learning is a subset of ML that is based on ANN. "Deep" refers to the number of hidden layers required to train ML models. The DL algorithm outperforms ML algorithms, especially in large data sets with a huge number of features and rows. DL algorithms have enabled advances in several applications such as computer vision, natural language processing, and machine translation. DL is offering efficient learning algorithms for both supervised and unsupervised tasks.

2.4.2.1 Supervised learning

Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are the more common DL learning algorithms:

- **CNN:** are specialized DL algorithms designed for computer vision applications. CNN architectures take images represented as a matrix of pixels. CNN combines traditional layers in ANN with more sophisticated operators such as convolution and polling operators to learn fine-tuned features from the figures.

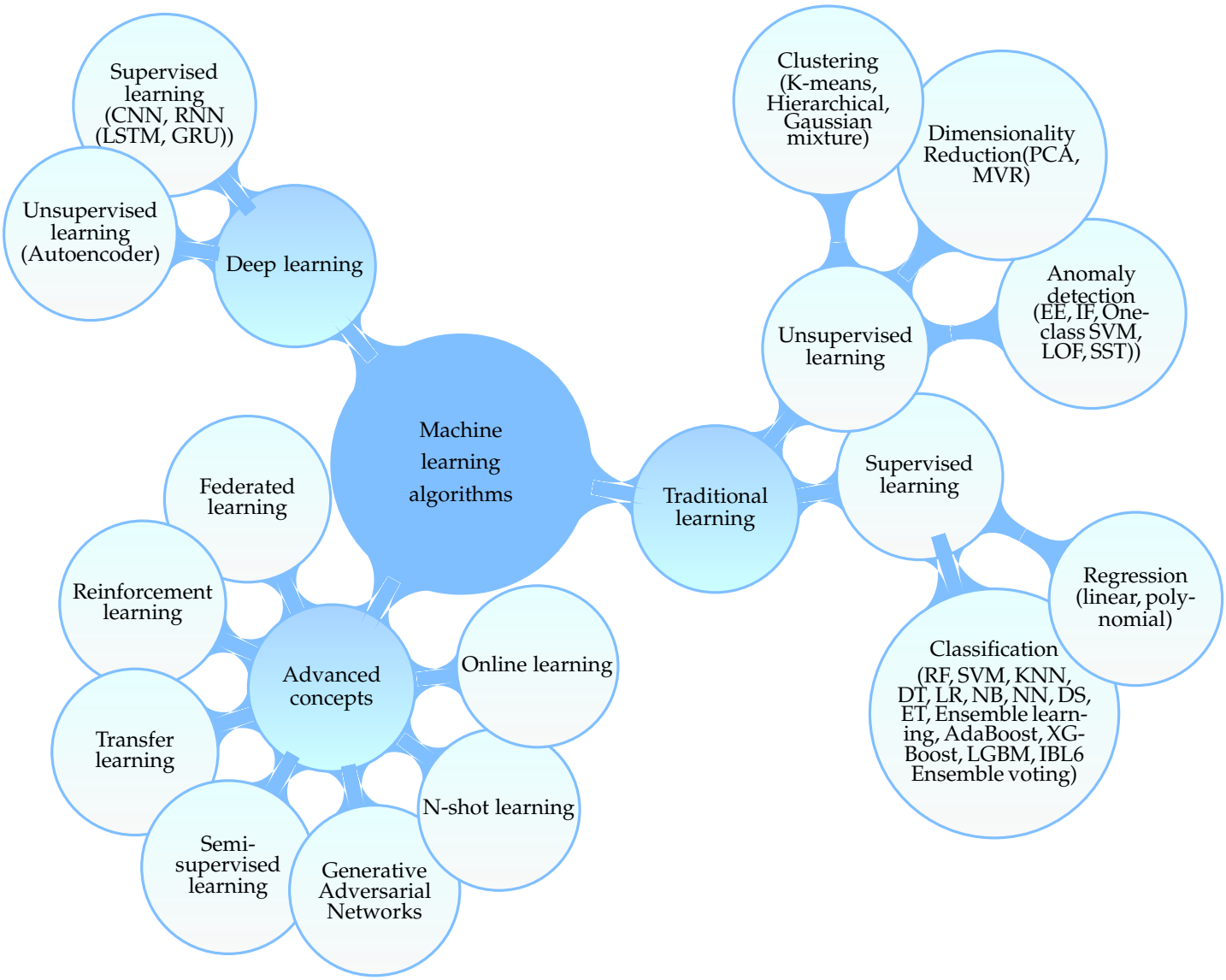


Fig. 7: Taxonomy of the machine learning algorithms

- **RNN:** RNNs are DL algorithms addressing problems involving data sequences or time series such as speech recognition, natural language processing, and language translation. RNNs are inter-connecting learning nodes enabling a kind of memory that takes knowledge from previous data sequences to influence the current data sequence and the output. Several advanced RNN architectures are proposed, such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs).

2.4.2.2 Unsupervised learning algorithms

- **Autoencoder** is an unsupervised deep learning algorithm that uses a neural network architecture with a tiny bottleneck layer in the middle that contains the input data's encoding representation to reconstruct the input data in the output. Specifically, the autoencoder consists of (i) the encoder that compresses the data inputs to encoding presentation with a smaller size than the input and (ii) the decoder that takes the encoding representation and tries to reconstruct the input data. In unsupervised anomaly detection, Autoencoders aim to minimize the reconstruction error as part of their training. The reconstruction loss is used to detect the anomalies.

2.4.3 Advanced ML concepts

2.4.3.1 Federated learning (FL)

FL is a distributed ML technique enabling collaboration between multiple nodes to build a global model without sharing their data sets collaboratively. The training of the global model is performed within several rounds until the FL server achieves a satisfactory global model. The FL server sends the global model to a set of selected nodes in each round. Each learning node uses its local labeled dataset to calculate its local updates of the global model. At the end of the round, all the selected learning nodes send their local updates to the FL server. Once all the updates are received,

the FL server aggregates local updates for calculating the new global model.

2.4.3.2 Reinforcement learning

It is a type of goal-oriented learning that trains models on how to achieve specified goals while maximizing outcomes over time. It is based on rewarding positive behaviors while penalizing those that are undesirable. Reinforcement learning agents can perceive and interpret their surroundings, taking actions and learning through trial and error. Deep reinforcement learning combines reinforcement learning with DL.

2.4.3.3 Transfer learning

It is an ML technique that exploits the knowledge gained by solving a given problem to apply it to another related problem. For example, the knowledge acquired from learning to detect DoS attacks could be used to detect DDoS attacks.

2.4.3.4 Semi-supervised learning

It is similar to supervised learning, but the training process combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised is usually used where unlabeled data is accessible but labeled data is hard to obtain.

2.4.3.5 Generative Adversarial Networks (GAN)

GAN is a deep learning network that can create data similar to the input data. It consists of two networks that train together: (i) Generator: which generates data with similar characteristics as the training data, and (ii) Discriminator: which attempts to categorize observations as "real" or "created" given data comprising observations from both the training data and producing data from the generator.

2.4.3.6 N-shot learning

N-shot learning is an ML sub-area that classifies new data based on zero or only a few supervised training samples. N-shot learning is the general concept where N is the number of used training samples. As a result, three particular cases could be identified: (i) zero-shot learning, (ii) one-shot learning, and (iii) few-shot learning. Zero-shot learning aims to classify unseen classes without any training examples, while one-shot learning needs only one sample of each class, and two to five samples per class are required for few-shot learning [72].

2.4.3.7 Online learning

As part of ML, online learning enables learning from progressively arriving data. Unlike offline learning, online learning methods allow updating ML models progressively with one data point at a time. In other words, each online learning stage is quick and allows the ML model to adapt to new knowledge in real time. Moreover, online learning is helpful, especially when computing resources are limited since storing training samples is not required after learning them, saving considerable storage space.

2.5 Development and Evaluation

This subsection describes different elements used to develop and evaluate ML-based MDSs. As depicted in Figure 8, these elements include public datasets, network simulators and emulators, and evaluation metrics.

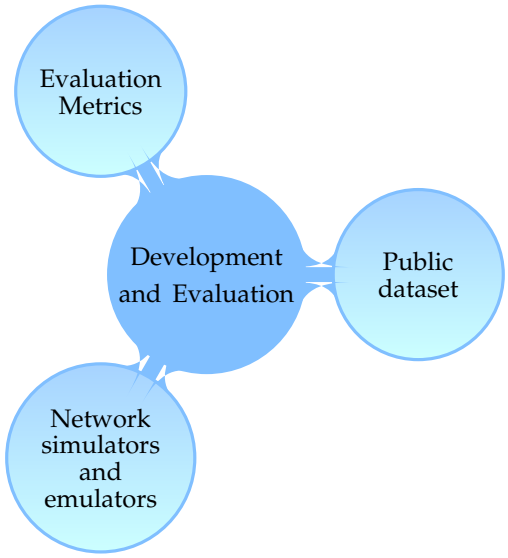


Fig. 8: Cornerstones of the development and evaluation of ML-based MDSs

2.5.1 Public datasets

Several public security datasets have been used to build ML-based MDSs. These datasets are briefly described in the following.

2.5.1.1 VeReMi

VeReMi [73, 74] is a simulated dataset generated using simulation tools such as OmNeT++ and Veins. Five types of position falsification were implemented. i) Constant: which consists in broadcasting fixed positions; (ii) Constant offset: which consists in broadcasting a fixed offset added to the real positions; (iii) Random: which consists of broadcasting random positions belonging to the simulated area; (iv) Random offset: which consists in broadcasting random positions that belong to a rectangle around the vehicle; and (v) Eventual stop: in which the attacker behaves normally for some time and then attacks by broadcasting a fixed position for a period. This dataset is generated for different traffic densities and different attacker radios.

2.5.1.2 VeReMi extension

VeReMi extension [75, 76] is also a simulated dataset generated using the Framework For Misbehavior Detection [77, 78], which is based on OmNeT++ and Veins. This dataset represents an extension of VeReMi implementing nine types of attacks: (1) Position falsification (constant, random, constant offset, and random offset); (2) Speed Malfunctions

(constant, random, constant offset, and random offset); (3) Delayed Messages; (4) DoS attacks; (5) DoS Random; (6) Data Replay; (7) Disruptive; (8) Eventual Stop; (9) Traffic congestion Sybil.

2.5.1.3 DARPA

DARPA [79, 80] are popular intrusion detection datasets created using an emulated network environment at the MIT Lincoln Lab. They implement attacks on authentication such as scanning attacks, User-to-Root (U2R), and Remote-to-Local (R2L) attacks. They also implement attacks on availability like DoS attacks.

2.5.1.4 CAIDA DDos2007

CAIDA DDos2007 [81] includes approximately one hour of traffic traces from a DDoS attack (UDP flooding) attempting to block access to a server by consuming computing resources on the server and all of the bandwidth of the network connecting the server to the Internet. The traces only includes attack traffic to the victim and responses to the attack from the victim.

2.5.1.5 AWID 2

AWID 2 [82, 83] dataset implements popular attacks on 802.11 generated based on a small testbed. It includes various attacks on authentication (e.g., ARP injection), availability (e.g., Request To Send (RTS) beacon), and confidentiality (e.g., rogue access point). The AWID dataset comprises packets of different sizes captured at different times, with various equipment,, and in different environments. Each trace includes 154 features and a label indicating whether the trace is benign or an attack.

2.5.1.6 KDD CUP 99

KDD CUP 99 [84] is a popular intrusion detection dataset proposed in an international competition aimed at building a predictive model that can distinguish between normal and malicious network traffic. KDD CUP 99 includes 23 attacks on authentication and availability, such as R2L, probing attacks, DoS, and U2R simulated in a military network environment.

2.5.1.7 NSL-KDD

NSL-KDD [85, 86] dataset includes the same attacks as the KDD CUP 99. It mainly came to enhance the KDD CUP 99 by removing duplication and creating more sophisticated sub-datasets. Specifically, unlike KDD CUP 99, NSL-KDD does not include redundant traces in the train and test datasets, which helps eliminate bias toward more frequent records.

2.5.1.8 Kyoto

Kyoto [87, 88] is a honeypot dataset that contains real packet-based traffic converted into a new format called sessions. Each session comprises 24 features, 14 of which are inspired by the KDD CUP 99 dataset. The remaining 10 are flow-based features such as IP addresses, ports, or duration.

2.5.1.9 UNSW-NB15

UNSW-NB15 [89, 90] has been created in a small emulated environment over 31 hours. The dataset includes raw network packets. The number of records in the training dataset is 175,341 records, and the test dataset is 82,332 records from the different types (normal and attack). Specifically, It includes nine attacks, such as backdoors, DoS, exploits, fuzzes, or worms.

2.5.1.10 ISCX 2012 IDS

ISCX 2012 IDS [91] has been generated using a small testbed within seven days of network activity (normal and malicious). It consists of labeled network traces, including full packet payloads. Specifically, it includes attacks such as infiltration, HyperText Transfer Protocol (HTTP), DoS, DDoS, and brute force SSH.

2.5.1.11 CICIDS2017

CICIDS2017 [92, 93] has been created within an emulated environment for five days. It is generated with realistic traffic background abstracting behavior of human interactions and generating naturalistic benign background traffic. It contains a wide range of attack types like SSH brute force, heartbleed, botnet, DoS, DDoS, web, and infiltration attacks.

2.5.1.12 CRAWDAD (mobiclique)

CRAWDAD (mobiclique) [94] includes the traces of Bluetooth encounters, opportunistic messaging, and social profiles of 76 users of the MobiClique application at SIGCOMM 2009. Specifically, each device performs a periodic Bluetooth discovery to discover nearby devices and records the results of the discovery and all data communications. Moreover, the devices record details of the user's social profile, its evolution, and application-level messaging.

2.5.1.13 NGSIM trajectory datasets

The NGSIM program [95] collected high-quality traffic datasets at four different locations in the United States, including two freeway segments and two arterial segments, between 2005 and 2006. The datasets collected and generated for each location include longitudinal and lateral positioning information for all vehicles in certain regions.

Table 5 shows the types of security attacks provided by each of the security datasets concerning security services described in the subsection 2.2.1. It is worth mentioning that any dataset includes attacks on non-repudiation. In addition, as can be seen, the table does not include CRAWDAD (Mobiclique) and NGSIM trajectory datasets since they do not initially include attacks. Still, the authors use them after pre-processing, like injecting noise.

TABLE 5: The security services targeted by the attacks included in the datasets used to build ML-based MDSs

Dataset	Integrity	Authentication	Availability	Confidentiality
VeReMi	X			
VeReMi extension	X		X	
DARPA		X	X	
CAIDA DDos 2007			X	
AWID2		X	X	X
KDD CUP 99		X	X	
NSL-KDD		X	X	
Kyoto		X	X	
UNSW-NB15		X	X	
CICIDS2017	X	X	X	

2.5.2 Network simulators and emulators

Several network simulators have been used to generate customized security datasets. These network simulators are described in the following.

2.5.2.1 Objective Modular Network Testbed in C++ (OMNeT++)

OMNeT++[1] is a modular, component-based C++ simulation library and framework, primarily for building network simulators. OMNeT++ itself is a simulation framework without models for network protocols like IP or HTTP. The main computer network simulation models are available in several external frameworks.

2.5.2.2 Simulation of Urban MObility (SUMO)

SUMO[2] is a microscopic mobility simulator. It allows for building realistic traffic and mobility models for various application areas. It supports modeling pedestrians, bicycles, passenger cars, trucks, buses, trains, and even ships. SUMO includes many tools for the creation, execution, and evaluation of traffic simulations, such as network import, route calculations, and visualization. SUMO also provides various modules to remotely control the simulation.

2.5.2.3 Veins

Veins[3] is an open-source framework for running vehicular network simulations. It is based on two well-established simulators: OMNeT++ and SUMO. It extends these to offer a comprehensive suite of models serving as a modular framework for simulating V2X applications. Each model includes one or more OMNeT++ models, which can be instantiated to provide the required simulation functionalities.

2.5.2.4 PLEXE

PLEXE[4] is a cooperative driving framework extending SUMO and Veins, allowing realistic simulations of vehicle platooning systems. It offers realistic vehicle dynamics and several cruise control models, enabling control systems analysis, large-scale and mixed scenarios, networking protocols, and cooperative maneuvers.

2.5.2.5 Network Simulator Version 2/3 (NS2/NS3)

NS2/NS3[5] is an open-source event-driven simulator explicitly designed for research in computer communication networks. It simulates wired and wireless networks, considering protocols such as TCP, FTP, UDP, and HTTP. NS3 is an extension of NS2, focusing on improving the core architecture, software integration, and model components.

2.5.2.6 VanetMobiSim

VanetMobiSim[6] is an extension of the CANU Mobility Simulation Environment (CanuMobiSim) supporting vehicular mobility and features new realistic automotive motion models at macroscopic and microscopic levels. According to these models, vehicles regulate their speed depending on nearby vehicles, overtake each other and act according to traffic signs in the presence of intersections.

2.5.2.7 CTUns-5.0

CTUns-5.0 [96] is a high-fidelity and extensible network simulator that can test various protocols and topologies used in wired and wireless IP networks. It also provides a framework to develop simulations for evaluating advanced V2V and V2I applications. It also supports simulating multi-interface mobile nodes equipped with multiple heterogeneous wireless interfaces.

2.5.2.8 GloMoSim (Global Mobile Information System Simulator)

GloMoSim [97] is a network simulation software that can simulate large-scale wired and wireless in different configurations enabling several use cases like mobile ad-hoc networks. GloMoSim supports protocols for purely wireless networks and works using parallel discrete event simulation capabilities.

2.5.2.9 Mininet

Mininet[7] is a network emulator which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow [98] for highly flexible custom routing and SDN. Mininet was used to design under-attack SDN V2X scenarios by creating OpenFlow devices, including flow tables, and defining SDN rules. Mininet-Wireless Fidelity (WiFi) is an extension of Mininet, which allows the using

1. https://omnetpp.org/
2. https://www.eclipse.org/sumo/
3. https://veins.car2x.org/
4. https://plexe.car2x.org/
5. https://www.nsnam.org/
6. http://vanet.eurecom.fr/
7. http://mininet.org/

WiFi stations and access points. Two surveyed papers have exploited Mininet-WiFi to generate datasets by extracting information from SDN network flows.

2.5.2.10 RDS1000

RDS1000 is a simulation platform that may serve for research, training, or automotive product development. It has real-vehicle equipment with a customizable virtual dashboard. RDS-1000 offers an actual steering wheel with control, loaded steering, real accelerator, and brake pedals [99].

2.5.3 Evaluation metrics

Various evaluation metrics have been used to evaluate ML-based MDSs. The following gives the calculating formula for each metric with a short description.

- **Accuracy** is the ratio of the correctly detected attackers to the total of vehicles.

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

  - **True Positive (TP)** is the number of cases correctly identified the attacks.
  - **False Positive (FP)** is the number of cases incorrectly identified the attacks.
  - **True Negative (TN)** the number of cases correctly identified the benign events.
  - **False Negative (FN)** is the number of cases incorrectly identified the benign events.

- **Precision** calculates the ratio of correctly detected attacks to the total detected attacks.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

- **Recall** calculates the ratio of correctly detected attacks to the total actual attacks.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

- **F1-score** can be interpreted as a weighted average of precision and recall.

$$F1 - score = 2X\frac{Precision * Recall}{Precision + Recall} \tag{4}$$

- **True Positive Rate (TPR)**, so-called also Sensitivity, is the proportion of attacks which has a positive detection result.

$$TPR = \frac{TP}{TP + FN} \tag{5}$$

- **True Negative Rate (TNR)**, so-called also Specificity is the proportion of benign events which has a negative detection result.

$$TNR = \frac{TN}{FP + TN} \tag{6}$$

- **False Positive Rate (FPR)** is the proportion of benign events, which has a positive detection result.

$$FPR = \frac{FP}{FP + TN} \tag{7}$$

- **False Negative Rate (FNR)**: is the proportion of attacks which has a negative detection result.

$$FNR = \frac{FN}{TP + FN} \tag{8}$$

- **The Receiver Operator Characteristic (ROC) curve** shows the trade-off between sensitivity and specificity. Classifiers with curves that are closer to the top-left corner perform better.
- **The Area Under the Curve (AUC)** is used as a summary of the ROC curve. It measures the ability of a classifier to distinguish between classes.

3 ML-BASED MISBEHAVIOR DETECTION SYSTEMS: A TAXONOMY

This section reviews different ML-based MDSs proposed in the literature. It classifies the proposed ML-based MDSs into three categories: (i) ML-based MDSs for Non-IP-based (safety) applications: that mainly detect attacks on the facilities layer; (ii) ML-based MDSs for IP-based (non-safety) applications: that mainly detect attacks on the transport and networking layers; and (iii) ML-based MDSs that can be used for both: that mainly detect attacks on the physical layer. Each category is divided into subcategories according to the attack detected by the ML-based MDS as shown in Figure 9.

3.1 Non-IP-based V2X applications

This category includes ML-based MDSs detecting position falsification, false information, Sybil, position tracking, and multi-attacks. The false information subcategory can comprise the position falsification subcategory. However, this subsection categorized them separately due to the important works on position falsification. In addition, the multi-attack category includes ML-based MDSs that can detect two or more attacks.

3.1.1 Position falsification

So et al. [100] proposed an ML-based MDS that can detect position falsification attacks. The proposed system built a supervised learning model based on the VeReMi dataset. The authors considered six features for the training: (1) location plausibility check; (2) movement plausibility check; (3) average distance between the first received beacon and the final received beacons; (4) average velocity between the first received beacon and the final received beacon; the feature (5) is the magnitude of features 3 and 4; finally, the feature (6) is the total displacement between two received messages. Moreover, they used two ML algorithms (SVM and KNN) to train their model, which was evaluated using precision and recall metrics. Le et al. [101] also proposed a supervised learning-based MDS based to detect position falsification attacks. The paper leverages comparing the trajectory of vehicles with the trajectories of legitimated vehicles. The authors proposed three features to compare the trajectories: (i) movement plausibility check: which checks if their positions changed between two consecutive messages received from the same vehicle; (ii) minimum distance to trajectories: which measures the similarity between observed trajectories

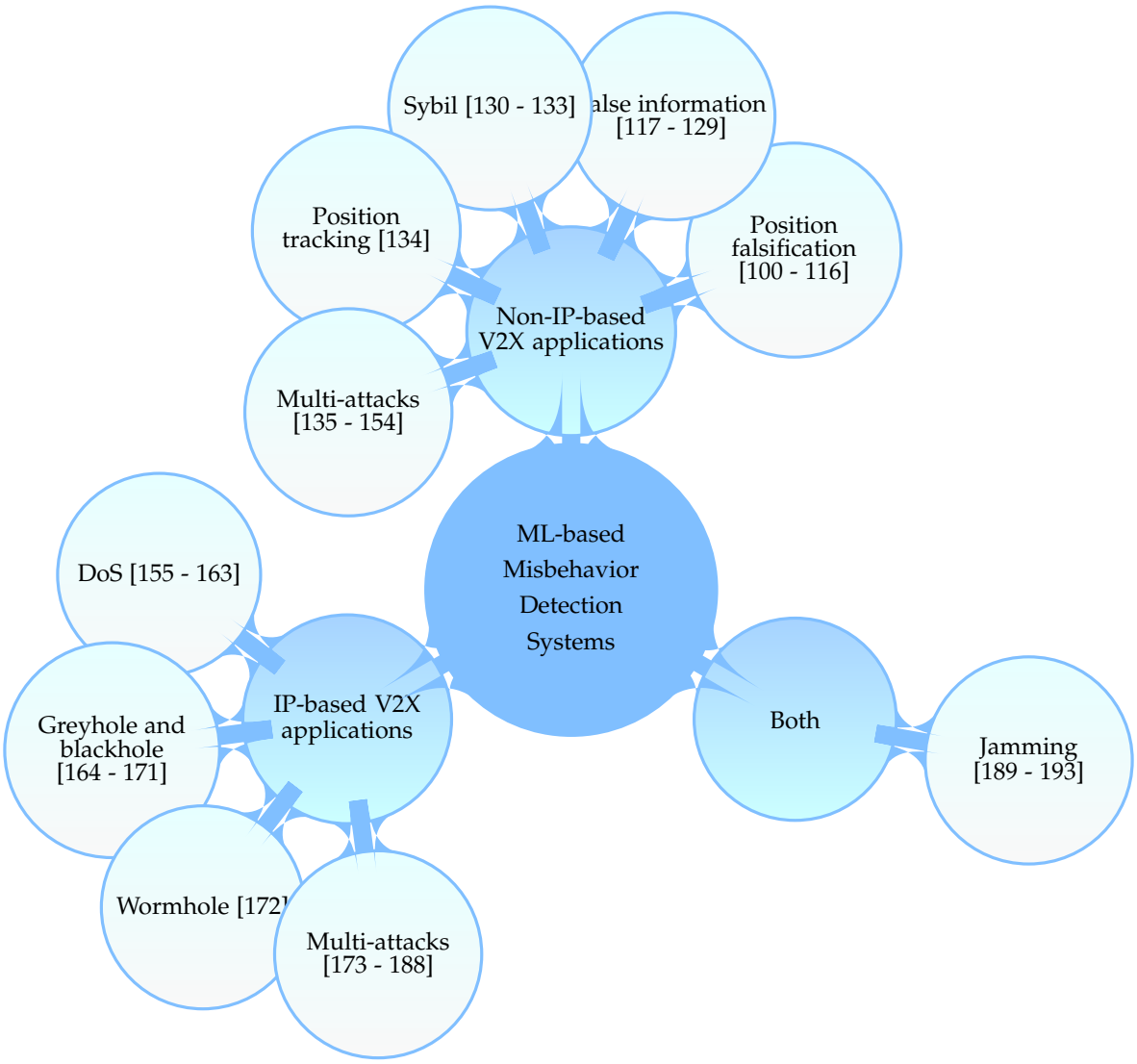


Fig. 9: Taxonomy of ML-based Misbehavior Detection Systems

and legitimate trajectories; and (iii) minimum translation distance to trajectories: which checks if any offset was added to received positions. Based on the proposed features, the paper trains a multi-class classifier on the VeReMi dataset to detect five false position attacks. The authors tested two classification algorithms (SVM and KNN) based on MATLAB implementation and evaluated the model using precision and recall. Singh et al. (1) [102] proposed a supervised-based MDS to detect position falsification attacks. They trained the system based on the VeReMi dataset. Three combinations of features were tested (i) (position, speed) (ii) (position + Δ position (between the sender and the receiver), and (iii) (position, speed, Δ position, Δ speed). The authors tested two ML algorithms (SVM and LR) for binary classification and evaluated the system using the F1-score. Sharma et al. (1) [103] proposed an ML-based MDS to detect position falsification attacks. The proposed system combines plausibility checks with ML models. They trained the system on the VeReMi dataset based on a supervised learning approach. The authors selected four features for the training: position coordinates (x,y) and speed coordinates (v_x, v_y). They compared six ML algorithms (SVM, KNN, NB, RF, ensemble boosting, and ensemble voting) and evaluated the system using accuracy, precision, recall, and F1-score. Kosmanos et al. [104] proposed a supervised approach to detect position falsification attacks. The ML model was based on a binary classification and was trained based on a dataset generated using the Veins network simulator. The authors selected four features for the training (i) the signal strength indicator; (ii) the signal quantity indicator; (iii) the packet Delivery Ratio (PDR); and (iv) the position verification using relative speed, which is a position verification check based on the speed and the GPS position. Moreover, they tested two classification algorithms (KNN and RF) and evaluated the system using FPR, TPR, and ROC. Montenegro et al. [105] proposed an ML-based MDS for detecting position falsification attacks based on a supervised learning approach. The authors implemented four types of position falsification attacks using the Veins simulator. The trust value is the only feature considered for the training. The trust value is calculated based on the weighted sum of the normalized speed and the normalized received power. In addition, the

paper built a binary classifier based on KNN and evaluated it using accuracy, recall, TPR, and FPR metrics. Ecran et al. (1) [106] proposed a supervised learning approach to detect position falsification attacks. They trained the system based on the VeReMi dataset and selected two combinations of features: (i) Received Signal Strength and Interference (RSSI), position, the distance between sender and receiver, Δ position of the sender, the estimated Angle of Arrival (AoA), the estimated distance between the sender and the receiver), and (ii) (Position, the distance between sender and receiver, estimated AoA, estimated distance between sender and receiver). Moreover, they used ML algorithms (KNN and RF) to build a binary classifier and evaluated it using precision, recall, accuracy, and F1-Score. The authors proposed an extension of this work in Ecran et al. (2) [107]. Unlike their previous work, they trained multi-class classifiers instead of binary classifiers. In addition, they tested ensemble learning by combining KNN and RF. Hawlader et al. [108] proposed an ML-based MDS based on a supervised learning approach. They trained binary and multi-class classifiers on the VeReMi dataset and selected twenty features for training based on the difference in positions sent by vehicles. They also used six ML algorithms (SVM, DT, RF, KNN, NB, and LR) for building the models. Moreover, they evaluated the models using accuracy, precision, recall, and F1-score metrics and validated them using simulations. Okamura et al. [109] proposed unsupervised anomaly-based MDS that detects position falsification attacks. They implemented four types of position falsification attacks similar to the ones presented in VeReMi using the Scenargie network simulator. The authors proposed to deploy their MDS on the cloud. The MDS leverages SST-based transformed time series of positions to detect attacks. Moreover, the authors evaluated their solution using precision, recall, and F1-score. Grover et al. (1) [110] proposed an unsupervised anomaly detection MDS to detect position falsification attacks. They used the VeRemi dataset for training and comparing several ML models: GRU (1 layer), LSTM (1 layer with changing the number of neurons), and stacked LSTM by changing the number of the layers from 2 to 5. The authors evaluated their ML models using accuracy and recall and suggested deploying them on the edge nodes. Sedar et al. [111] proposed an ML-based MDS based on reinforcement learning to detect sudden-stop (eventual stop) attacks, which is a specific type of position falsification attack. The authors used the VeReMi extension dataset to train their model based on one feature (position/speed) and evaluate it using precision, recall, and F1-score metrics. Uprety et al. [112] leveraged FL to propose privacy preservation collaborative ML-based MDS for position falsification attacks. The authors used the VeReMi dataset and selected four features for the training. The first two features are the difference between calculated average velocity and predicted ones in x and y directions, respectively. Feature 3 is the magnitude of features 1 and 2. The last feature is the difference between the calculated and predicted total displacement. The evaluation metrics were precision and recall. Sharma et al. (2) [113, 114] proposed a supervised-based MDS to detect position falsification attacks. The authors used the VeReMi dataset and selected features for training, including position and speed features from two consecutive beacons. They trained binary and multi-class classifiers to detect the attacks based on several ML algorithms (KNN, RF, NB, and DT). Finally, they evaluated the ML classifiers using several metrics, including precision, recall, and F1-score. Mankodiya et al. [115] proposed a supervised-based MDS to detect position falsification attacks. They built a binary classifier based on the VeReMi dataset selecting six features for training consisting of (X, Y, Z) coordinates of position and speed. They also used three ML algorithms (RF, DT, and AdaBoost) for the training and evaluated the classifier using precision, recall, and F1-score metrics. Recently, Aliev et al. [116] proposed an ML-based MDS for detecting position falsification attacks. The authors used VeReMi to train a multi-class classifier based on a multi-head DL architecture consisting of multiple CNN networks stacked with an LSTM network. They first trained every CNN network based on speed and position data generated by only one vehicle. After this stage, they concatenated all output layers of CNN networks to serve as an input for the LSTM network. Finally, they used the accuracy to evaluate the system.

3.1.2 False Information

Ghaleb et al. [117] proposed an ML-based MDS to detect data injection. The mobility traces were extracted for the NGSIM dataset and replayed in MATLAB. The authors implemented the data injection attack by injecting dynamic noise where 20% of vehicles are considered malicious. They also trained an ML model using seven features: (1) overlaying check; (2) consistency of reported uncertainties; (3) mobility message prediction error; (4) communication-based feature; (5) appearance position-based features; (6) average mobility messages prediction error, (7) the time to last received mobility message. The paper adopted a supervised approach based on ANN and evaluated the ML model using accuracy, F1-score, recall, and precision metrics. Monteuuis et al. [118] proposed supervised learning-based MDS to detect the absence of correlation between the type V2X entity and the dimension of the vehicles. Open datasets about cars, motorcycles, and pedestrians were collected from the internet and processed (data cleaning and feature reduction) for training and evaluation. Three features were selected: the width, length, and type of V2X entity. The authors generated misbehaviors by injecting noise into the dataset. They also used three ML algorithms (ANN, AdaBoost, and RF) and several metrics for the evaluation, including TPR, TNR, FPR, FNR, accuracy, and F1-score. Singh et al. (2) [119] proposed an ML-based MDS for detecting malicious infrastructure nodes reporting false information to the traffic management center. They generated their dataset using SUMO. Their system consists of ML models built using ANN and LSTM to predict traffic congestion (the halt time in traffic segment) based on loop detectors' data. The information reported by the infrastructure nodes was compared with the output of the proposed model to detect the attack. Gyawali et al. [120, 121] proposed an ML-based MDS for detecting false alert and position falsification attacks. The authors trained a binary classifier to detect false alert attacks based on the difference between the average flow value and the received flow value from the vehicles. The flow value is calculated based on the density of vehicles and the average speed of vehicles. The dataset for this attack was generated

using the Veins simulation platform. On the other hand, the authors trained a multi-class classifier to detect position falsification attacks based on the VeReMi dataset. The considered features were the change in speed and position between two consecutive beacons, the receiving distance, the RSSI, and its speed and position. The paper used several classification algorithms (LR, KNN, DT, Bagging, and RF) and evaluated the models using precision, recall, and F1-score metrics. Negi et al. [122] proposed an anomaly detection-based MDS. The proposed system leverages an unsupervised learning approach based on LSTM. The authors generated datasets from experiments performed on a treadmill-based autonomous car simulator at the University of Waterloo. This system focus on detecting anomalies in big data generated by CAVs rather than focusing on V2X attacks. To speed up the training process, a cluster of servers is used instead of one server. After the training, the anomaly detection model is distributed to vehicles for the real-time detection of anomalies. The model was retrained over time based on newly collected data, and the new model's parameters were distributed to vehicles to keep the model updated. AUC was used as an evaluation metric. Almalki et al. [123] proposed a supervised-based MDS to detect false data inject attacks. The authors proposed to take several contextual data in addition to data collected in real-time for attack detection. The authors used the NGSIM dataset, which contains data acquired from the environment using a set of sensors. In this MDS, data undergo several pre-processing steps, including missing values imputation based on the local and global fuzzy-clustering correlation approach. The ML models were trained using LR, SVM, and CNN. Accuracy, F1-score, Detection Rate (DR), and FPR were used as evaluation metrics. Ko et al. [124] proposed a supervised-based MDS to detect false information attacks in Cooperative Adaptive Cruise Control (CACC) for CAVs platooning. More specifically, these attacks falsify speed and acceleration to destabilize the platoon. The authors proposed to deploy an ML-based MDS on each platooning vehicle to detect attacks. They used a dataset generated from PLEXE platooning simulator. They trained their model based on the information received by the ego vehicle from three predecessor vehicles in the platoon. This information consists of speed, acceleration of the predecessor vehicle, the time difference between the generation of the message at the preceding vehicle, and the local time of the ego vehicle. The built ML model was a binary classifier trained based on LSTM and evaluated using accuracy and F1-score. Boddupalli et al. (1) [125] proposed an unsupervised-based MDS to detect false acceleration values in CACC. They trained their model based on a dataset generated using the RDS1000 simulator. Their dataset considers various simulation environments: (1) three road typologies (highway, suburban, and city), (2) four weather parameters (rain, snow, clear, and windy), and (3) two diurnal parameters (day and night). They trained their ML model to detect anomalies only based on normal data. The model was built using the ANN algorithm and evaluated using FP and FN metrics. Wang et al. [126] proposed an unsupervised-based MDS to detect false information in CACC. Their ML model considers three types of stealthy attacks. The first attack adds a constant to the velocity of the leading vehicle, while the second attack multiplies it by a constant. The third attack adds an offset to position, velocity, and acceleration. The authors trained two anomaly detection models on LSTM and autoencoders, respectively. Their models took a dataset generated using SUMO, consisting of temporal trajectory windows. Each window consists of w vectors of trajectories. Each trajectory vector consists of positions, velocities, and accelerations of both ego and leading vehicles. The F1-score was used as an evaluation metric while the system was proposed to be deployed on the Roadside Unit (RSU). Boddupalli et al. (2) [127] proposed an ML-based MDS to detect false information (position and velocity) attacks in CAVs platooning. Their system only detects attacks generated from followers. They used a dataset generated based on the RDS1000 simulator. The system first predicts the ego vehicle's acceleration and compares it with the actual acceleration value. The attack was detected if the difference between the predicted and the actual acceleration values was greater than a certain threshold. The RF regressor was used for the prediction based on five features: the velocity and the position of the ego vehicle, the velocity and the preceding vehicle, and the leader's velocity. Moreover, the authors proposed deploying the system on vehicles. Sarker et al. [128] proposed an ML-based MDS to detect false information (velocity (v), acceleration (a), brake status (b), and steering angle (θ)) attacks. Their MDS predicts the current

TABLE 6: Specification of the type of false information detected by ML-based MDSs

	False information							
	Not specified	Dimension and type	Position	Velocity	Acceleration	Brake status & Steering angle	Alert	Road traffic
Ghaleb et al. [117]	X							
Monteuuis et al. [118]		X						
Singh et al. (2) [119]								X
Gyawali et al. [120, 121]			X				X	
Negi et al. [122]	X							
Almalki et al. [123]	X							
Ko et al. [124]				X	X			
Boddupalli et al. (1) [125]						X		
Wang et al. [126]			X	X	X			
Boddupalli et al. (2) [127]			X	X				
Sarker et al. [128]				X	X	X		
Ayoob et al. [129]			X					X

driving state r(t) = (v(t), a(t), b(t), θ (t)) of the vehicle based on its previous driving states using a Gaussian mixture model-based Mixture Density Network incorporating RNN. The attack is detected by comparing the predicted driving state with the actual driving state. The used dataset combined a real driving dataset collected from 29 participants and a simulated dataset generated using SUMO. In addition, the authors proposed deploying the system on vehicles. Precision and recall were used as evaluation metrics. Ayoob et al. [129] proposed an ML-based MDS to detect false information about road traffic and position. They used a dataset generated based on NS2 collecting features about the traffic flow characteristics and the deviation between the position coordinate of the neighboring vehicle and the measurement position. Then, they trained a binary classifier based on ANN to detect attacks. Although their model was not validated, they conducted experiments to compare situations before and after the system's deployment.

Table 6 summarizes the type of false information detected by previously described ML-based MDSs. There are seven types of false information: dimension and type of vehicles, position, velocity, acceleration, brake status & steering angle, alert, and road traffic. It can be seen that three of the described ML-based MDSs do not explicitly specify the type of detected false information.

3.1.3 Sybil

Gu et al. [130] proposed a supervised learning-based MDS to detect Sybil attacks. They used a dataset generated using SUMO. The driving patterns of vehicles were modeled as matrices. Each matrix line contains five fields: time, location, velocity, acceleration, and acceleration variation at time t. The two max eigenvalues of the matrix were used as training features. The authors then built a binary classifier based on SVM and ANN algorithms and used TPR, FPR, and FNR as evaluation metrics. The same authors in [131] also proposed a similar approach but used the KNN algorithm for binary classification and the accuracy as an evaluation metric instead. Kamel et al. (1) [132] proposed an ML-based MDS for detecting Sybil attacks. They defined four types of Sybil attacks: (i) Traffic Congestion Sybil, (ii) Data reply Sybil, (iii) Dos Random Sybil, and (iv) Dos Disruptive Sybil. Their solution consists of two systems: Local and global. The authors proposed to deploy the local system at the vehicle level, where a set of plausibility and consistency checks to detect misbehavior and report it to the global system. The latter was equipped with an ML-based MDS and placed in the cloud. In addition, the authors trained their system based on the VeReMi extension dataset. They used thirty features to build the multi-class attack classifier based using LSTM. Their classifier was evaluated using various metrics such as F1-score, recall, and precision. Quevedo et al. [133] proposed an ML-based MDS for detecting Sybil attacks. They adopted a supervised learning approach to build their ML models, which are trained and deployed on edge nodes for attack detection. Their collected data consists of a set of matrices describing vehicles' driving patterns. The columns of matrices are the features considered for learning. Each row of a matrix contains vehicle driving information at time t. The authors used an unsupervised learning technique to reduce the dimensionality of matrices. They also used extreme ML for classification based on the dataset generated using SUMO containing between 1% and 20% Sybil attackers. They also used accuracy as a metric to evaluate their system.

3.1.4 Position tracking

Boualouache et al. [134] proposed an ML-based MDS for detecting position-tracking attacks. The authors first identified strategies that attackers can use to track vehicles without being visually detected efficiently. The authors then generated a syntactic dataset to train the ML models based on these strategies. Their MDS enables FL at the edge to ensure collaborative learning while preserving the privacy of vehicles. In addition, FL clients (vehicles) use a semi-supervised learning approach for self-labeling. Furthermore, their FL architecture used a DL model for binary attack classification. Their results were compared with six traditional ML algorithms: LR, KNN, SVM, NB, DT, and RF in terms of precision, recall, F1-score, and accuracy.

3.1.5 Multi-attacks

Grover et al. (2) [135] proposed a supervised ML-based MDS to detect six attacks: (i) Impersonation, (ii) False position, (iii) Combination of impersonation and false position, (iv) greyhole, (v) reply, and (vi) timing. The authors generated the dataset using the NCTUns-5.0 network simulator. After that, they selected eleven features for training and building the ML models: (i) position, (ii) acceptance range, (iii) speed deviation, (iv) RSS, (v) packet transmitted, (vi) PDR, (vii) packet drop ratio, (vii) packet capture ratio, (ix) packet capture ratio, (x) packet collision ratio, and (xi) packet retransmission error ratio. Thus, they built and compared binary and multi-class classifiers using five ML algorithms NB, IBK, RF, DT, and Adaboost. The metrics are TPR, FPR, TNR, and FNR. Grover et al. (3) [136] also proposed a similar ML-based MDS in [135]. Their results demonstrated that ensemble-based learning gives better results than in [135] in the case of binary classification. Li et al. [137] (1) proposed a supervised learning-based MDS to detect packet dropping, packet modification, and RTS flooding attacks. They generated the dataset using the GloMoSim network simulator and selected three features to train the ML model: Packet Drop Rate (PDrR), Packet Modification Rate (PMR), and RTS flooding rate. The authors also considered using other contextual information such as velocity, channel status, temperature and wind speed, and GPS coordinates and altitude. But this information is not used in the evaluation. In addition, they built their binary classifier using the SVM algorithm and evaluated it using precision and recall. Zhang et al. (1) [138] proposed an ML-based MDS for detecting false messages and message suppression. The dataset was generated from simulations using the Veins simulation framework. The authors selected five features to detect false message attacks: (i) VehicleType, (ii) MessageType, (iii) Reputation, (iv) The distance between the sender of the message and the location of the event, and (v) Message forwarding status. They also used four features to detect suppression message attacks: (i) PDrR, (ii) Packet Delay Forward Rate, (iii) PMR, and (iv) Packet Misroute rate. Thus, they trained two binary classifiers for each attack based on SVM, evaluated them using TPR, FPR, and accuracy,

and proposed deploying them on CAVs. Eziama et al. [139] proposed an unsupervised-based MDS based on Bayesian ANN that combines DL with probabilistic modeling. The paper also described three attacks: timing, Sybil, and position falsification. However, the authors did not evaluate the models. Mahmoudi et al. [140] proposed a supervised-based MDS approach to detect attacks and multiple attacks defined in the VeReMi extension dataset. Several training features were used, including local detection, Kinematic data, and generic features. The authors developed a multi-class classifier based on five ML algorithms: RF, XGboost, LightGBM, ANN, and LSTM, and evaluated their classifier using various metrics: precision, recall, and F1-score. Kamel et al. (2) [141] also proposed a supervised-based MDS to detect multiple attacks defined in the VeReMi extension dataset. The features considered for learning are almost similar to the previous work [132]. The authors developed ML models based on three ML algorithms: SVM, ANN, and LSTM. The evaluation metrics are recall, precision, F1-score, and accuracy. Alladi et al. (1) [142] proposed a supervised-based to detect multiple attacks considered in VeReMi extension dataset. The authors used the messages of each vehicle to generate sequences of 20 messages with 7 data fields: position (X, Y), velocity (V_X,V_Y), timestamp, pseudo-id, and label. In addition, they built two multi-class classifiers. The first one considers two classes: normal and attacks. The second one considered position falsification attacks as faults; thereby, three classes: normal, faults, and attacks. In addition, the authors used two DL architectures: stacked LSTM and CNN-LSTM. Based on these architectures, they built different models and evaluated them using accuracy, precision, recall, and F1-score metrics. Alladi et al. (2) extended the previous work in [143]. The authors proposed a similar supervised-based MDS and suggested deploying it as a detection engine in the MEC. The authors considered two detection methods: (1) Sequence Classification and (2) Sequence-image classification. They developed two models for sequence classification: (i) four stacked layers LSTM and (ii) CNN-LSTM. They also developed two models, CNN and MLP, for sequence-image classification. Alladi et al. (3) [144] proposed an ML-based MDS to detect the same attacks considered in their previous works but based on an unsupervised learning approach. They trained the ML models only on normal data using the VeReMi extension dataset. The considered DL architectures are similar to auto-encoders that take sequences of 20 messages as input and encode&decode them to reconstruct the normal traces. The anomaly detection threshold was adjusted according to the accuracy of reconstructing the normal trace. The authors considered two models: Model 1 (CNN-LSTM) and Model 2 (stacked 4-layer LSTM). They used precision, recall, F1-score, and accuracy metrics. Alladi et al. (4) also extended this work by considering further models for enhancing detection performance. Kushardianto et al. [145] proposed a supervised-based MDS to detect multiple attacks. They used the position and velocity features to train two ML models based on the VeReMi extension dataset. The first model is a binary classifier that detects attacks without specifying the type. The second model is a multi-class classifier that determines the attack type. The authors used four ML algorithms to build their classifiers: RF, LSM, GRU, and Deep Belief Network, and evaluated them using accuracy as a metric. Gonçalves et al. [146, 147] proposed an ML-based MDS to detect DoS and false information (speed, acceleration, and heading) attacks. They trained several multi-class classifiers based on a dataset generated in their previous work [148]. Several features were used for the training, including position, speed, and heading. The proposed system has a hierarchical architecture with four different levels. The first level (vehicle) deploys a Decision Stump classifier, while the second level is to forward messages from vehicles to RSUs. The third level (RSU) deploys an RF classifier. Finally, the fourth level (cloud) deployed an ensemble learning classifier that combines ANN, DT, and RF. The authors evaluated their system using accuracy, TPR, and TPR metrics.

Hsu et al. [150] proposed a supervised-based MDS to detect attacks described in the VeReMi extension dataset. The proposed ML model is a binary classifier based on SVM and trained using eleven features. The authors divided the features into four classes: (1) behavioral deviation, (2) location plausibility, (3) velocity information, and (4) comprehensive information. The main feature considered in behavior deviation is the mean absolute error extracted from the position reconstruction procedure based on a CNN-LSTM model, very similar to the proposed in [110, 144, 149]. The authors evaluated the model using accuracy, precision, recall, and F1-score and suggested deploying the system on RSUs. Boddupalli et al. (3) [151] proposed an ML-based MDS, deployable on vehicles, to detect multiple attacks in CACC. These attacks are impersonation, greyhole, DoS, jamming, and false information. The authors generated their dataset based on the RDS1000 simulator. The authors used the ANN to predict the acceleration based on the velocity and acceleration of the preceding vehicle, the acceleration of the ego vehicle, and the gap between the two vehicles. Their system detects the attack if the difference between the actual acceleration and the predicted one is greater than a certain threshold. The authors evaluated the system using recall, precision, F1-score, and FPR metrics. Liu et al. [152] proposed an ML-based MDS to detect DoS/DDoS, Sybil, and replay attacks. The authors first simulated these attacks using Veins. Then, they used two-stage multi-class classification to build the ML model. In the first stage, they employed four primary classification algorithms (KNN, DT, AdaBoost, and RF), while in the second stage, they used LR as a secondary classifier. More specifically, they trained four primary classifiers based on 5-fold cross-validation. After that, the results were aggregated and passed to the second classifier. They also evaluated their system using accuracy, precision, recall, and F1-score metrics. Sedjelmaci et al. [153] proposed a hybrid MDS combining ML and rules-based methods. The authors generated a dataset using the NS3 simulator, consisting of information about PDrR, Packets Sent Ratio, Message Duplication Ratio, and Signal Strength Intensity. The proposed system trains a binary classifier based on SVM to detect DoS, Sybil, greyhole/Blackhole, and wormhole attacks. The authors evaluated their system using the DR and FPR as metrics. Zeng et al. (2) [154] proposed an ML-based MDS to detect Sybil and DDoS attacks. The authors built their model using a dataset generated using the Ns3 simulator. They also used stacked DL architecture combining CNN and LSTM for building their multi-class

TABLE 7: Specification of attacks detected by multi-attack ML-based MDSs for non-IP-based applications

	Position falsification	False information	DoS/DDoS	Sybil	Replay	Timing	Greyhole/blackhole	Impersonation	Wormhole
Grover et al. (2) [135]	X				X	X	X	X	
Grover et al. (3)~[136]	X				X	X	X	X	
Li et al. (1) [137]		X	X			X	X		
Zhang et al. (1) [138]	X						X		
Eziama et al. [139]	X			X		X			
Mahmoudi et al. [140]	X	X	X	X	X	X			
Kamel et al. (2) [141]	X	X	X	X	X	X			
Alladi et al. (1) [142]	X	X	X	X	X	X			
Alladi et al. (2) [144]	X	X	X	X	X	X			
Alladi et al. (3) [143]	X	X	X	X	X	X			
Alladi et al. (4) [149]	X	X	X	X	X	X			
Kushardianto et al. [145]	X	X	X	X	X	X			
Gonçalves et al. [146, 147]		X	X						
Hsu et al. [150]	X	X	X	X	X	X			
Boddupalli et al. (3) [151]		X	X				X	X	
Liu et al. [152]			X	X	X				
Sedjelmaci et al. [153]			X	X			X		X
Zeng et al. (2) [154]			X	X			X		X

classifier models. The authors evaluated their system using precision, F1-score, and recall metrics and proposed a mechanism to update ML models.

Table 7 summarizes the detected attacks of each previously described ML-based MDSs. These attacks include position falsification, false information, DoS/DDoS, Sybil, replay, timing, greyhole/blackhole, wormhole, and impersonation.

3.2 IP-based V2X applications

This category includes ML-based MDSs detecting DoS, greyhole/blackhole, Sybil, Wormhole, and multi-attacks. The multi-attack category includes ML-based MDSs that can detect two or more attacks.

3.2.1 DoS

Tan et al. [155] proposed an unsupervised learning-based MDS for detecting DoS attacks. In their paper, RSUs collect vehicles' traffic flow, defined as a sequence of packets from the source to the destination. Each flow contains σ network packets with the corresponding time series. The Agglomerate Hierarchical Clustering (AHC) was applied to create clusters of similar traffic flows. In each step of the clustering algorithm, the dynamic time wrapping distance [156] is used to calculate the distance between the time series of different traffic follow. The authors built their model based on a Python-generated dataset and evaluated it using the DR as a metric. Singh et al. (3) [157] proposed a supervised learning-based MDS to detect DDoS attacks in SDN-based vehicular networks. The authors built a binary classifier based on a dataset generated using the Mininet-WiFi emulation tool. They also tested serval ML algorithms, including LR, DT, NB, SVM, KNN, ANN, and Gradient boosting. Finally, they evaluated their system using TP, TN, FP, and FN metrics. Yu et al. [158] also proposed an ML-based MDS for detecting DDoS attacks in SDN-based vehicular networks. The paper suggested the integration of OpenFlow [98] with vehicular networks with a focus on DDoS. A set of features are extracted from an open flow table to use for training a set of binary classifiers based on TCP, UDP, and Internet Control Message Protocol (ICMP). The authors built their models using SVM based on well-known datasets, including DARPA, CAIDA, and DDos2007. They also evaluated their system using the DR metric. Sharshembiev et al. [159] proposed an unsupervised learning-based MDS to detect DoS attacks. The authors generated their dataset using the Veins simulation platform and then used the entropy-based anomaly detection technique to detect attacks based on the generated network flows. Finally, they evaluated their system using precision, recall, and F1-score as metrics. Narayanadoss et al. [160] proposed an ML-based MDS to detect a type of DDoS called Crossfire in SDN-based vehicular networks. Crossfire attacks aim at isolating a specific region from the network by launching coordinated flooding on the principal network links. The authors generated a dataset implementing normal and under-attack scenarios based on Mininet-WiFi. In addition, they trained a binary classifier using ANN, CNN, and LSTM and evaluated it using accuracy, precision, recall, and F1-score metrics. The authors also suggested deploying

their system on top of the SDN controller to detect attacks efficiently. Maglaras et al. [161] proposed an ML-based MDS to detect DoS attack. They trained their system based on a dataset generated by a customized network simulator. The system leverages an unsupervised anomaly detection method combining KNN with one-class SVM called K-OCSVM. While accuracy was the evaluation metric, the authors suggested installing this system on CAVs or RSUs. Tian et al. [162] proposed an ML-based MDS to detect DoS attacks in BUSNet, a virtual mobile backbone infrastructure constructed using public buses. BUSNet consists of three layers: (i) vehicles, (ii) buses, and (iii) RSUs. The proposed ML-based MDS was installed on the backbone and used an unsupervised anomaly detection method to detect the attacks. The authors built their model using ANN based on a dataset generated using NS2 and validated it using FP and FN metrics. Nie et al. [163] proposed an ML-based MDS to detect DoS/DDoS attacks. They employed several spatial and temporal features to detect the attack. In addition, they trained their model based on data generated using a private testbed. Specifically, the authors used a CNN-based anomaly detection method to train the model and TPR as an evaluation metric.

3.2.2 Greyhole and blackhole

Gruebler et al. [164] proposed a supervised learning-based MDS to detect blackhole attacks. The authors generated a dataset based on NS2 and SUMO and selected fifteen features for learning, such as payload size, type, IP source and destination, and sequence number. They also trained a binary classifier based on the ANN algorithm and evaluated it using TP, TN, FP, and FN metrics. Alheeti et al. (1) [165, 166] proposed a supervised learning-based MDS to detect greyhole attacks. The authors generated the datasets by simulating greyhole with an adapted version of the AODV protocol on the NS2 simulator and SUMO. They also selected fifteen features and performed feature fuzzification to train two binary classifiers. In addition, they used two classification algorithms (SVM and ANN) for training and four metrics (accuracy, TP, TN, FN, and FP) for evaluation. Zeng et al. (1) [167] proposed a multi-level ML-based MDS for detecting greyhole/blackhole attacks. The proposed system consists of two binary classification models. The first binary classifier was an ANN deployed on the RSU, while the second was an SVM classifier deployed on the cluster head. The authors generated their dataset using GlobMoSim, employed the techniques described in [164] to extract the features, and adopted accuracy and DR as evaluation metrics. Siddiqui et al. [168] proposed a hybrid ML-based MDS that combines unsupervised and supervised learning for detecting greyhole attacks. The authors used CRAWDAD (mobiclique) dataset to extract three features: similarity, familiarity, and PDR. After data prepossessing, they applied an unsupervised technique for labeling. Then, they trained a binary classifier using two classification algorithms (KNN and SVM) and evaluated it using the accuracy metric. Acharya and Oluoch [169] proposed a supervised learning-based MDS to detect blockhole attacks. They generated their dataset based on a modified version of AODV implemented on the NS3 simulator. Then, they selected seven features for the training: (i) source IP address, (ii & iii) source and destination ports, (iv) timeFirstRxpaket, (v) timeLastRxPacket, (vi) lost packets, and (vii) throughput. Finally, the authors trained a binary classifier using five ML algorithms: NB, LR, KNN, SVM, and gradient boosting and evaluated its performance using recall, precision, F1-score, accuracy, FPR, FNR, and ROC_AUC score metrics.

Abdel Wahab et al. [170] proposed an ML-based MDS to detect greyhole/blackhole attacks. They generated the dataset using VanetMobisim and selected several features to train the ML model, including the number of packets to be forwarded and the number of packets forwarded. In addition, they built a binary classifier using SVM and validated it using accuracy, DR, and FPR metrics. Shams et al. [171] proposed an ML-based MDS to detect greyhole/blackhole attacks. The authors trained a binary classifier based on the dataset generated using the NS2 simulator. They also selected several features for the training of the ML model, including packet drop count, packet transfer delay, and packet forward interval. Their model was built using SVM and evaluated precision, recall, and F1-score metrics. The authors also proposed deploying the system on CAVs.

3.2.3 Wormhole

Singh et al. (4) [172] proposed a supervised learning-based to detect wormhole attacks. They generated their dataset using the NS3 simulator and selected several features to train their ML models, including source and destination IP addresses, transmitted and received Bytes, dropped Bytes, FirstRxBytesTime, FirstTxBytesTime, and throughput. In addition, they trained a binary classifier using two ML algorithms (KNN and SVM) and evaluated it using TP, TN, FP, and FN metrics.

3.2.4 Multi-Attacks

Alheet et al. (2) [173, 174] proposed a supervised learning-based to detect multiple attacks. The training was done using the Kyoto dataset, which includes different attacks: SQL, TCP, Malware, shellcodes, and exploit codes. The authors also proposed a technique to reduce the number of features. Using the ANN algorithm, they trained a multi-class classifier to detect three classes: normal, known, and unknown attacks. They also evaluated it using TP, TN, FN, and FP metrics. Kim et al. [175] proposed an ML-based MDS for SDN-based vehicular networks where CAVs analyze the incoming traffic and forward selected data flows to the SDN controller. Based on these data flows, the SDN controller trains a multi-class classifier on the KDD dataset based on SVM. The authors selected six features for the training: PDrR, PMR, RTS flooding rate, channel status, packet interval, average packet interval in the flow, and packet size. They also evaluated their model using accuracy, precision, and recall metrics. The trained model is forwarded to vehicles, which use it to detect misbehavers. Zhang et al. (2) [176, 177] proposed an ML-based MDS based on a distributed ML approach. The authors assumed that each vehicle has its own labeled data. Vehicles collaboratively build the model without exchanging data sets between them. Instead of sharing the data sets, the vehicles share only updates of the loss functions. The authors proposed a dual variable perturbation to provide dynamic differential privacy to prevent privacy leakage. They trained a binary

classifier using LR based p, NSL-KDD dataset and used the loss function output as an evaluation metric. Ghaleb et al. [178] proposed a collaborative ML-based MDS for multiple attacks. The system consists of four phases. In the first phase, each vehicle builds its local model based on its collected data. In the second phase, vehicles share models according to the requests received from the neighbors. In the third phase, vehicles evaluate the received models to detect malicious models (nodes). Finally, in the last phase, vehicles build a collaborative model based on the valid models checked in the third phase. The authors used the NSL-KDD dataset to train a binary classifier, leveraging three algorithms (RF, XGBoost, and SVM). In addition, they evaluated their model using accuracy, precision, recall, F1-score, FPR, and FNR metrics. Ashraf et al. [179] proposed unsupervised learning-based MDS for detecting multiple network attacks. They trained their model based on the UNSW-NB15 dataset, including exploits attacks, generic attacks, DoS attacks, Fuzzer attacks, and Recon attacks. They also used a statistical method to extract the features. Their model was based on LSTM autoencoder architecture and evaluated using several metrics, including precision, recall, accuracy, F1-score, and TPR. Shu et al. [180] proposed a collaborative ML-based MDS based on SDN. The proposed MDS used DL with GAN to jointly enable multiple distributed SDN controllers to train the ML model for the entire network. The authors trained their model based on the KDD99 dataset and evaluated it using various metrics, including accuracy, precision, recall, and F1-score. Li et al. (2) [181] proposed an ML-based MDS to detect multiple attacks, including false information, DoS, and impersonation. They used a transfer learning approach to transfer the knowledge acquired by building ML models built based on a large amount of labeled data regarding well-known attacks to detect new ones with a small amount of labeled data. The authors proposed two transfer learning approaches for ML model updating. The first is cloud-assisted, in which the cloud serves to label the data, update the model and send it back to vehicles. The second is the local update, in which vehicles use a pre-trained model to label data and transfer learning to update the model locally. In addition, the authors trained a multi-class classifier on the AWID public data. Their experiments showed how to exploit knowledge built from detecting injection and impersonation attacks for detecting flooding attacks. They also evaluated their MDS using accuracy, and FN. Bangui et al. [182] proposed a hybrid approach to detect multiple attacks. The MDS combines a binary multi-classifier model to detect know attacks and unsupervised learning to detect unknown attacks. The model was trained based on the CICIDS2017 dataset using RF and a variation of Kmeans and evaluated using F1-score and accuracy metrics. Yang et al. (1) [183] proposed a multi-tiered hybrid intrusion detection system. The MDS uses multi-class classification models to detect known attacks based on the CICIDS2017 dataset and unsupervised anomaly detection models to detect unknown attacks. The authors exploited several algorithms to train the model, including DT, RF, ET, XGBoost, and a stacking ensemble model. They also used a Bayesian optimization with a tree Parzen estimator to optimize the classification. Their anomaly-detection system consists of cluster labeling, two biased classifiers, and a Bayesian optimization with a Gaussian process method for unsupervised learner optimization. Moreover, they evaluated their system using various metrics, including DR, FPR, and F1-score. Khan et al. [184] proposed an unsupervised anomaly detection system to detect multiple attacks, such as DoS, reconnaissance, exploits, fuzzes, and generic attacks, consisting of two stages. The authors proposed two models based on the standard state-based method for the first stage and a Bidirectional LSTM-Based model for the second stage. They trained their models based on UNSWNB15 datasets and suggested deploying them on the CAVs' gateways. Finally, they evaluated their models using accuracy, recall, precision, and F1-score metrics. Liu et al. [185] combined blockchain and FL to build collaborative FL-based MDS. RSUs select FL workers from vehicles under their coverage and exploit them to build local models. Then, each RSU uses local models collected after each learning round to train global models. In addition, the blockchain system, which consists of RSUs, stores global models obtained after running consensus processes to select the block's miner. The authors built a binary classifier based on the KDD-Cup99 dataset to evaluate their system. Then they used accuracy, precision, and recall metrics for evaluation. Rahal et al. [186] proposed a supervised learning-based MDS to detect DoS and eavesdropping attacks. The authors trained a multi-class classifier based on a dataset generated using NS3. They also used several ML algorithms to build this classifier, including KNN, ANN, SVM, RF, DT, and NB. In addition, they evaluated their system using precision, recall, F1-score, accuracy, FPR, and FNR metrics. Liu et al. (1) [187] proposed an ML-based detection system to detect DoS/DDoS and impersonation attacks. The authors assumed a city partitioned into cells mapped to virtual machines equipped with ML-based MDSs. They built their ML-based MDS based on a dataset generated using their private testbed. In addition, they trained a binary classifier using traditional algorithms such as NB and LR and evaluated it using precision, recall, and F1-score metrics. Zeng et al. (2) [154] proposed an ML-based MDS to detect DDoS and impersonation attacks. They built their model using ISCX 2012 IDS public dataset and trained several multi-class classifier models based on stacked DL architecture combining CNN and LSTM. The authors also proposed a mechanism to update the ML model. In addition, they evaluated their models using precision, F1-score, and recall metrics. Yang et al. (2) [188] proposed an ML-based MDS to detect multiple attacks, including DDoS/DoS and impersonation. The ML model is a multi-class classifier trained based on the CIC-IDS2017 dataset. The authors used transfer learning to build the model. They transferred learning parameters from well-known CCN architectures, including VGG16, VGG19, Xception, Inception, and Inception Resnet. Then, they employed a hyper-parameter optimality method and ensemble learning to obtain the best results. The system was validated using accuracy, precision, recall, and F1-score.

Table 8 summarizes the detected attacks of each previously described ML-based MDSs. These attacks include impersonation, DoS/DDoS, false information, and eavesdropping. As can be seen, almost all works can detect impersonation and DoS/DDoS attacks.

TABLE 8: Specification of attacks detected by multi-attack ML-based MDSs for IP-based applications

	Impersonation	DoS /DDoS	False information	Eavesdropping
Alheet et al. (2) [173, 174]	X	X		
Kim et al. [175]	X	X		
Zhang et al. (2) [176, 177]	X	X		
Ghaleb et al. [178]	X	X		
Ashraf et al. [179]	X	X		
Shu et al. [180]	X	X		
Li et al. (2) [181]	X	X	X	
Bangui et al. [182]	X	X		
Yang et al. (1) [183]	X	X		
Khan et al. [184]	X	X		
Liu et al. (2) [185]	X	X		
Rahal et al. [186]		X		X
Liu et al. (1) [187]	X	X		
Zeng et al. (2) [154]	X	X		
Yang et al. (2) [188]	X	X		

3.3 Both

This category only includes jamming attacks.

3.4 Jamming

Karagiannis et al. [189] proposed an unsupervised learning-based MDS to distinguish between intentional interference (jamming) and unintentional interference. The authors selected several features for learning, including RSSI, PDR, signal-to-noise and interference ratio, and relative speed variation. In addition, they generated a dataset using the R programming language, considering a scenario with interference and different types of radio jammers. In addition, they used the k-means clustering algorithm for anomaly detection and evaluated their system based on a specific metric for identifying differences between interference and jamming cases. Lyamin et al. [190] proposed an unsupervised learning-based MDS for jamming attacks. The authors considered two jamming attacks: (i) Random jamming: each transmitted CAM is jammed independently with a probability p, and (ii) ON-OFF jamming: in which K subsequent CAMs are destroyed with probability one only in the ON state. They also generated their dataset using MATLAB simulations, considering radio inference on CAVs. In addition, they combined their solution with their previous work [191] to propose a hybrid method for enhancing the results. Finally, they evaluated their solution using the F1-score, TPR, and TNR metrics. Abhishek and Gurusamy [192] proposed an unsupervised learning-based MDS to detect jamming attacks. They generated a dataset using the NS3 simulator and selected two features for the training: PDrR and inverse PDR. In addition, they used the one-class SVM algorithm to build their model and evaluated it using the detection probability as a metric. Kosmanos et al. [193] proposed an ML-based MDS to detect jamming and GPS spoofing attacks in CACC. The authors generated their dataset using Veins, considering information from both the application and the physical layers to detect these attacks. They selected the RSSI, the SINR, and the PDR as features from the physical layer, the relative Speed (δu), and the GPS coordinates from the application layer. They also used both supervised and unsupervised learning to detect the attack. Specifically, they exploited ensemble learning, combining SVM and RF for supervised learning and OCSVM for unsupervised learning. Finally, they evaluated their solution using the ROC curve as a metric.

4 Summary & Discussion

This section provides summaries and discussions to get an overview of ML-based MDSs for 5GB vehicular networks. This section is divided into two parts: security and privacy-oriented summary and ML-oriented summary. The security and privacy-oriented summary mainly focuses on detected attacks, the general design, and other different security-related aspects. The ML-oriented summary mainly focuses on the ML model used to detect the attack. It also analyzes ML model characteristics such as the data sets, used ML algorithms, and metrics.

4.1 Security and Privacy oriented summary

Table 9 lists the publication year and the attacks targeted by ML-based MDSs for 5GB vehicular networks. As shown in Figure 1, since 2014, these ML-based MDSs have witnessed an increasing interest from the research community. This is not only due to the topic's importance and the considerable ML advances done in recent years but also to the emergence of interesting datasets such as VeReMi and VeReMi extension. As depicted in Figure 10, the majority of proposed ML-based MDSs are targeting Dos/DDoS attacks, position falsification, and false information attacks. This might be mainly due to the availability of the datasets. However, the major issue here is that existing ML-based MDSs do not consider attacks enabled by the integration of vehicular networks with 5G described in subsection 2.2.4, such as attacks on NS, handover, and roaming. This issue is discussed further in section 6.

Tables 10, 12, and 11 show the security and privacy-oriented summary for ML-based MDSs for Non-IP-based V2X applications, for IP-based V2X applications, and for both of them respectively. The following defines different criteria used in this summary.

- **Platooning-dedicated**: This column indicates whether the ML-based MDS was dedicated to vehicle platooning or not.
- **Type of the misbehavior**: This column mentions which type of misbehavior is detected by the ML-based MDS. It can be an attack performed intentionally by a malicious node or an anomaly caused by a malfunctioning node.
- **Unseen attacks (Yes/Maybe/No)**: This column indicates whether the ML-based MDS can detect unseen attacks or not.
- **Learning model**: This column indicates whether the model was built by a single node (Single) or multiple nodes (Collaborative).

TABLE 9: Specification of attack(s) detected by each surveyed ML-based MDS

Year	Work	Position falsification	False information	Sybil	Position tracking	Dos/DDoS	Reply	Timing	Greyhole /black-hole	Jamming	Impersonation	Wormhole	Eavesdropping	GPS Spoofing
2010	Tian et al. [162]					X								
2011	Grover et al. (2) [135]	X					X	X	X		X			
2011	Grover et al. (3)[136]	X					X	X	X		X			
2014	Liu et al. (1) [187]					X					X			
2015	Maglaras et al. [161]					X								
2015	Sedjelmaci et al. [153]			X		X			X			X		
2015	Gruebler et al. [164]								X					
2015	Li et al. (1) [137]		X			X		X	X					
2015	Alheeti et al. (1) [165, 166]								X					
2016	Abdel Wahab et al. [170]								X					
2016	Alheet et al. (2) [173, 174]					X					X			
2017	Ghaleb et al. (2) [117]		X											
2017	Kim et al. [175]					X					X			
2017	Gu et al. (1) [130]			X										
2017	Gu et al. (2) [131]			X										
2018	Sarker et al. [128]		X											
2018	Zhang et al. (1) [138]	X							X					
2018	Nie et al. [163]					X								
2018	Ayoob et al. [129]	X	X											
2018	Zhang et al. (2) [176, 177]					X					X			
2018	So et al. [100]	X												
2018	Shams et al. [171]								X					
2018	Zeng et al. (1) [167]								X					
2018	Eziama et al. [139]		X	X				X						
2018	Karagiannis et al. [189]									X				
2018	Tan et al. [155]					X								
2018	Monteuuis et al. [118]		X											
2018	Lyamin et al. [190]									X				
2018	Singh et al. (3) [157]					X								
2018	Singh et al. (2) [119]		X											
2018	Yu et al. [158]					X								
2019	Boddupalli et al. (1) [125]		X											
2019	Kosmanos et al. [193]									X				X
2019	Narayanadoss et al. [160]					X								
2019	Siddiqui et al. [168]								X					
2019	Mahmoudi et al. [140]	X	X	X		X	X	X						
2019	Zeng et al. (2) [154]			X		X			X		X			
2019	Kamel et al. (1) [132]			X										
2019	Kamel et al. (2) [141]	X	X	X		X	X	X						
2019	Le et al. [101]	X												
2019	Singh et al. (1) [102]	X												
2019	Gyawali et al. [120, 121]	X	X											
2019	Singh et al. (4) [172]											X		
2020	Ghaleb et al. [178]					X					X			
2020	Quevedo et al. [133]			X										
2020	Ashraf et al. [179]					X					X			
2020	Sharma et al. (1) [103]	X												
2020	Shu et al. [180]					X					X			
2020	Negi et al. [122]		X											
2020	Li et al. (2) [181]		X			X					X			
2020	Kosmanos et al. [104]	X												
2020	Bangui et al. [182]					X					X			
2020	Montenegro et al. [105]	X												

TABLE 9: Specification of attack(s) detected by each surveyed ML-based MDS – continued from previous page

Year	Work	Position falsification	False information	Sybil	Position tracking	Dos/DDoS	Reply	Timing	greyhole /black-hole	Jamming	Impersonation	Wormhole	Eavesdropping	GPS Spoofing
2021	Almalki et al. [123]		X											
2021	Yang et al. (1) [183]					X					X			
2021	Ecran et al. (1) [106]	X												
2021	Ecran et al. (2) [107]	X												
2021	Hawlader et al. [108]	X												
2021	Okamura et al. [109]	X												
2021	Grover et al. (1) [110]	X												
2021	Sedar et al. [111]	X												
2021	Uprety et al. [112]	X												
2021	Boualouache et al. [134]				X									
2021	Alladi et al. (1) [142]	X	X	X		X	X	X						
2021	Khan et al. [184]					X					X			
2021	Alladi et al. (2) [143]	X	X	X		X	X	X						
2021	Liu et al. (2) [185]					X					X			
2021	Alladi et al. (3) [144]	X	X	X		X	X	X						
2021	Alladi et al. (4) [149]	X	X	X		X	X	X						
2021	Kushardianto et al. [145]	X	X	X		X	X	X						
2021	Gonçalves et al. [146, 147]		X			X								
2021	Sharshembiev et all [159]					X								
2021	Acharya and Oluoch [169]								X					
2021	Abhishek et Gurusamy [192]									X				
2021	Sharma et al. (2) [113, 114]	X												
2021	Mankodiya et al. [115]	X												
2021	Hsu et al.[150]	X	X	X		X	X	X						
2021	Mankodiya et al. [115]	X												
2021	Ko et al. [124]		X											
2021	Wang et al. [126]		X											
2021	Boddupalli et al. (2) [127]		X											
2021	Boddupalli et al. (3) [151]		X			X			X		X			
2021	Aliev et al. [116]	X												
2022	Rahal et al. [186]					X							X	
2022	Liu et al. [152]			X		X	X							
2022	Yang et al. (2) [188]					X					X			

- **Learning mode**: This column indicates the learning mode depending on the learning model. If the model was built by a single node, it can then be on a single server or data center. But, if the model was constructed collaboratively by multiple nodes, it can then be done in federated or peer-to-peer learning.
- **Privacy preservation (Yes/No)**: This column indicates if privacy preservation was ensured by the ML-based MDS or not.
- **Context-aware (Yes/No)**: This column indicates if the ML-based MDS takes into account the context parameters to change the security parameters.
- **SDN-oriented**: This column indicates whether the ML-based MDS was built on top of an SDN architecture or not.
- **Secure (Yes/No)**: This column indicates if the ML-based MDS is secured or not.
- **Communication Overhead (Large/Low)**: This column indicates the ML-based MDS's communication overhead, which depends on the learning mode. According to [194], centralized architectures generate significant communication overhead since they collect all data in a single place. Similarly, distributed data center architectures need row data exchanges conducted among servers, causing considerable communication overhead. In FL, communication overhead is smaller than in other learning approaches since communications are only required between the central server and each client. In peer-to-peer architectures, overhead is more significant than FL because more signaling overheads are needed to achieve synchronization among multiple clients
- **Validation (dataset/simulation)**: This column indicates how the ML model was validated whether

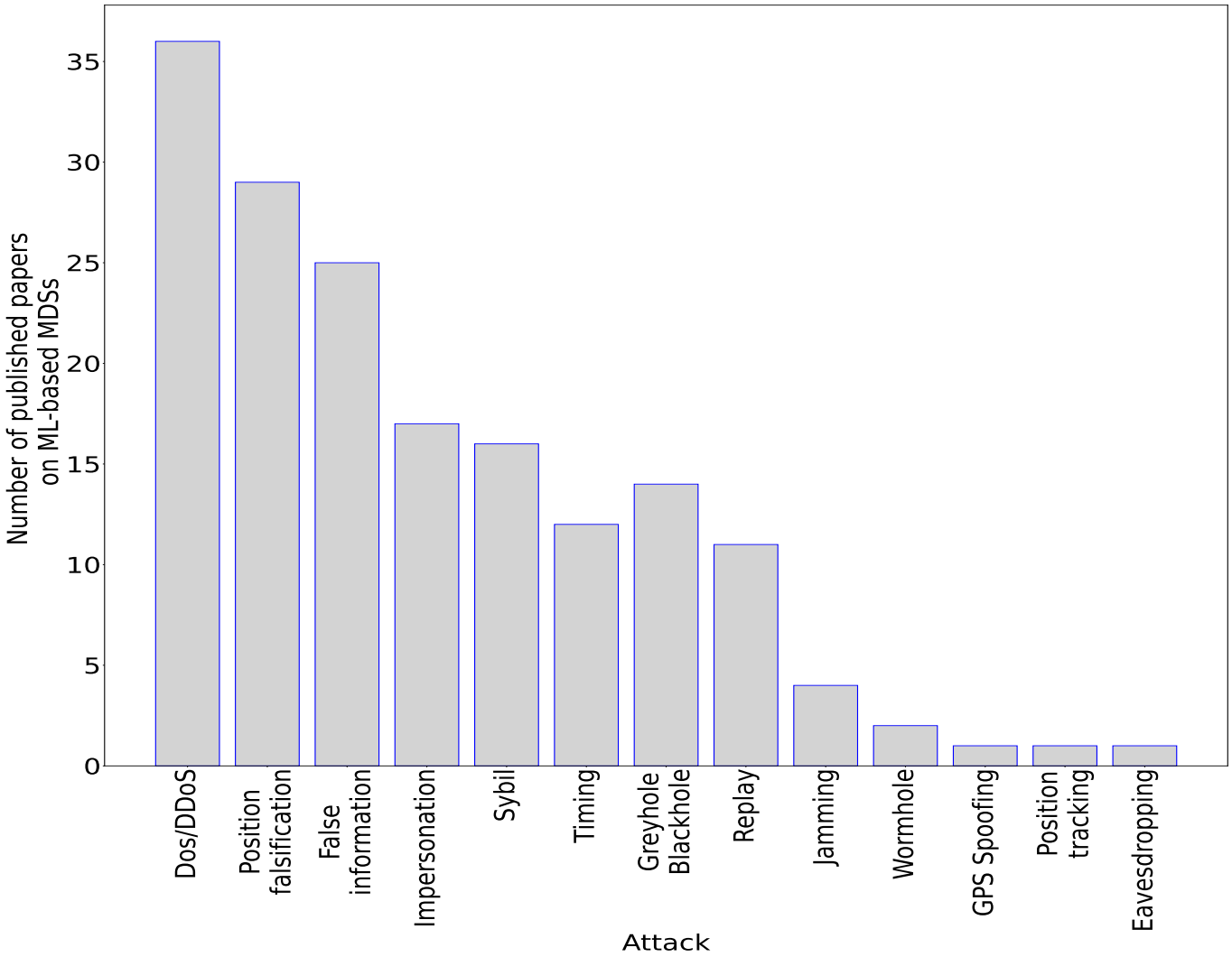


Fig. 10: The number of papers published on ML-based MDSs per attack

using datasets or through simulations.

From the analyzes of Tables 10, 12, and 11, it can be seen most of ML-based MDSs for 5GB vehicular are general and only a few ML-based MDSs are specific to vehicle platooning. It can also be seen that almost all the proposed ML-based MDSs focus on detecting attacks instead of anomalies. In addition, since most of the proposed ML-based MDSs are supervised, most MDS focus on detecting specific attacks instead of previously unseen attacks. However, unsupervised-based MDSs try to detect attacks by detecting deviations from normal behavior. Thus, it cannot be confirmed that these ML-based MDSs can detect unseen attacks since the authors only evaluated them on well-known attacks and do not identify new attacks. For this reason, the column "Unseen attacks" includes the value "maybe" as a value. Moreover, in most works, the ML model was built based on a single node. Only a few are based on collaborative learning, and most of them leverage FL. This particular set of MDSs is privacy-preserving, unlike the others. Context awareness was also not considered in most works; only two include this property. Furthermore, some ML-based MDSs incorporate SDN in their design alongside SDN-enabled vehicular networks. However, these ML-based MDSs do not consider detecting attacks and threats from this SDN integration. Finally, it is worth noting that almost all the existing works were validated using only datasets; only one uses simulations and datasets to validate the ML model.

In summary, this survey points out that most ML-based MDSs are generic rather than specific to a given application (e.g., vehicle platooning) to secure, as many as possible, 5GB-V2X applications. In addition, ML-based MDSs aim not only to detect attacks but also to identify them. However, existing ML-based MDSs for 5GB vehicular networks have targeted detecting traditional attacks and have ignored attacks resulting in 5G enabling technologies. In addition, research communities focused less on detecting and identifying unseen attacks. Besides, recent ML-based MDSs have adopted an FL approach instead of a centralized approach due to the benefits of FL on privacy-preserving. However, securing ML-based MDSs and context awareness are less addressed. Moreover, a few ML-based MDSs have considered support from 5GB enabling technologies like SDN but ignored threats from these technologies, as previously mentioned. Most of them also marginalized the deployment of ML-based MDSs.

TABLE 10: Security and privacy-oriented summary of ML-based MDSs proposed for non-IP-based applications

Collab: Collaborative | DS: DataSet | SIM: Simulation

Work	Platooning-dedicated	Misbehavior Type	Unseen attacks	Learning model	Learning mode	Privacy preservation	Context aware	SDN-based	Secure	Communication Overhead	Validation
So et al. [100]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Le et al. [101]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Singh et al. (1) [102]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Sharma et al. (1) [103]	No	Attack	No	Single	Centralized	No	No	No	No	Large	Sim
Montenegro et al. [105]	No	Attack	No	Single	Centralized	No	No	No	No	Large	Sim
Ecran et al. (1) [106]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Ecran et al. (2) [107]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Hawlader et al. [108]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS Sim
Okamura et al. [109]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Grover et al. (1) [110]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Sedar et al. [111]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Uprety et al. [112]	No	Attack	No	Collab	Federated	Yes	No	No	No	Small	DS
Sharma et al. (2) [113, 114]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Mankodiya et al. [115]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Ghaleb et al. [117]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Monteuuis et al. [118]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Singh et al. (2) [119]	No	Attack	No	Single	Centralized	No	No	No	No	Large	-
Gyawali et al. [120, 121]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Negi et al. [122]	No	Anomaly	Maybe	Single	Data center distributed	No	No	No	No	Large	DS
Almalki et al. [123]	No	Attack	No	Single	Centralized	No	Yes	No	No	Large	DS
Gu et al. (1) [130]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Gu et al. (2) [131]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Kamel et al. (1) [132]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Quevedo et al. [133]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Boualouache et al. [134]	No	Attack	No	Collab	Federated	Yes	No	No	No	Small	DS Sim
Grover et al. (2) [135]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Grover et al. (3) [136]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Li et al. (1) [137]	No	Attack	No	Single	Centralized	No	Yes	No	No	Large	DS
Zhang et al. (1) [138]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Eziama et al. [139]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	-
Mahmoudi et al. [140]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Kamel et al. (2) [141]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Alladi et al. (1) [142]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Alladi et al. (2) [143]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Alladi et al. (3) [144]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Alladi et al. (4) [149]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Kushardianto et al. [145]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Gonçalves et al. [146, 147]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Hsu et al.[150]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Mankodiya et al. [115]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Ko et al. [124]	Yes	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Boddupalli et al. (1) [125]	Yes	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Wang et al. [126]	Yes	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Boddupalli et al.(2) [127]	Yes	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Boddupalli et al. (3) [151]	Yes	Attack	No	Single	Centralized	No	No	No	No	Large	DS

TABLE 10: Security and privacy-oriented summary of ML-based MDSs proposed for non-IP-based applications – continued from previous page

Collab: Collaborative | DS: DataSet | SIM: Simulation

Work	Platooning-dedicated	Misbehavior Type	Unseen attacks	Learning model	Learning mode	Privacy preservation	Context aware	SDN-based	Secure	Communication Overhead	Validation
Sarker et al. [128]	Yes	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Liu et al. [152]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Sedjelmaci et al. [153]	No	Attack	No	Single	Centralized	No	No	No	No	Large	Sim
Ayoob et al. [129]	No	Attack	No	Single	Centralized	No	No	No	No	Large	Sim
Aliev et al. [116]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS

TABLE 11: Security and privacy-oriented summary of ML-based MDSs proposed for both IP-based and non-IP-based applications

Collab: Collaborative | DS: DataSet | SIM: Simulation

Work	Platooning-dedicated	Misbehavior Type	Unseen attacks	Learning model	Learning mode	Privacy preservation	Context aware	SDN-based	Secure	Communication Overhead	Validation
Karagiannis et al. [189]	No	Attack	Yes	Single	Centralized	No	No	No	No	Large	DS
Lyamin et al. [190]	No	Attack	Yes	Single	Centralized	No	No	No	No	Large	DS
Abhishek et Gurusamy [192]	No	Attack	Yes	Single	Centralized	No	No	No	No	Large	DS
Kosmanos et al. [193]	Yes	Attack	No	Single	Centralized	No	No	No	No	Large	DS

4.2 ML-Oriented summary

Tables 13, 14, and 15 show the ML-oriented summary for MDS for Non-IP-based V2X applications, IP-based V2X applications, and both of them respectively. The following defines different criteria used in this summary.

- **ML method**: this column mentions the ML method used to train the ML model including supervised, unsupervised, hybrid (combines supervised and unsupervised methods), and reinforcement learning.
- **Dataset**: This column specifies the dataset used to train the ML model. The name of the dataset is mentioned if it is publicly available. Otherwise, the network simulator used to generate the dataset is mentioned.
- **ML Task**: According to the used ML method, this column indicates the type of ML task. For supervised learning: the task can be regression, binary classification, or multi-class classification. For unsupervised learning: the task can be anomaly detection or clustering. For reinforcement learning, the task can be Markov Decision Process or Q-learning.
- **Update model (Yes/No)**: This column indicates whether the ML model will be updated over time or not.
- **ML Algorithm**: this column mentions the ML algorithms used to train the model and whether these algorithms are traditional or based on DL.
- **Metrics**: this column indicates which are the metrics used to evaluate the ML model.
- **Inference loc**: this column mentions the location where the ML model is deployed after the validation tests. This could be vehicles, edge nodes, RSUs, or the Cloud.

From the analyzes of Tables 13, 14, and 15, it can be seen that most of ML-based MDSs use a supervised ML method. Also, only three ML-based MDSs use a hybrid ML method combining unsupervised and supervised methods. Thus, most ML tasks are classification tasks that use either binary classifiers or binary classifiers. It can also be noticed that unsupervised tasks are mostly anomaly detection tasks.

Regarding the used datasets, this survey concludes that datasets used to build ML-based MDSs for non-IP-based based V2X applications (Table 13) have been generated using network simulators. In addition, the majority of authors prefer to use public datasets (VeReMi and VeReMi extension) than generating their own datasets. But, ML-based MDSs for IP-based V2X applications have mostly used public datasets generated using computer network testbeds. On the other hand, most of the works built their ML models using traditional ML algorithms. However, the latest works have started to focus on more DL algorithms and advanced concepts. It can also be seen that the used evaluation metrics are different from one ML-based MDS to another and mainly depend on the authors' perspectives, which are dependent on what to demonstrate. In addition,

TABLE 12: Security and privacy-oriented summary of ML-based MDSs proposed for IP-based applications

Collab: Collaborative | DS: DataSet | SIM: Simulation

Work	Platooning-dedicated	Misbehavior Type	Unseen attacks	Learning model	Learning mode	Privacy preservation	Context aware	SDN-based	Secure	Communication Overhead	Validation
Tan et al. [155]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Singh et al. (3) [157]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Yu et al. [158]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Sharshembiev et al. [159] [158]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Gruebler et al. [164]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Alheeti et al. (1) [165, 166]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Zeng et al. (1) [167]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Siddiqui et al. [168]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Acharya and Oluoch [169]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Singh et al. (4) [172]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Alheet et al. (2) [173, 174]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Kim et al. [175]	No	Attack	No	Single	Centralized	No	No	Yes	No	Large	DS
Zhang et al. (2) [176, 177]	No	Attack	No	Collab	Peer-to-peer distributed	(Yes)	No	No	No	Large	DS
Ghaleb et al. [178]	No	Attack	No	Collab	Peer-to-peer distributed	No	No	No	No	Large	DS
Ashraf et al. [179]	No	Attack	Maybe	Single	Centralized	No	No	No	No	Large	DS
Shu et al. [180]	No	Attack	No	Collab	Centralized	No	No	Yes	No	Large	DS
Li et al. (2) [181]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Bangui et al. [182]	No	Attack	Yes	Single	Centralized	No	No	No	No	Large	DS
Yang et al. (1) [183]	No	Attack	Yes	Single	Centralized	No	No	No	No	Large	DS
Khan et al. [184]	No	Attack	Yes	Single	Centralized	No	No	No	No	Large	DS
Liu et al. (2) [185]	No	Attack	No	Collab	Federated	Yes	No	No	Yes	Small	DS
Rahal et al. [186]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Abdel Wahab et al. [170]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Narayanadoss et al. [160]	No	Attack	No	Single	Centralized	No	No	Yes	No	Large	DS
Maglaras et al. [161]	No	Attack	No	Single	Centralized	No	No	No	No	Large	Sim
Tian et al. [162]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Liu et al. (1) [187]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Nie et al. [163]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Zeng et al. (2) [154]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS
Shams et al. [171]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS Sim
Yang et al. (2) [188]	No	Attack	No	Single	Centralized	No	No	No	No	Large	DS

in most ML-based MDSs the models were not updated after their deployment. Only a few works explicitly mentioned the update process, especially where the update of ML models was done by design like in FL-based MDS. Finally, the majority of works do not mention the inference location, but those who mention were mostly CAVs.

In summary, traditional ML techniques are dominant in the current ML-based MDSs for 5GB vehicular networks. However, ML-based MDS's development has followed the advancement in the ML fields. As a result, recent ML-based MDSs are based on advanced ML techniques such as sophisticated DL architecture and advanced ML concepts. In addition, the development of ML-based MDSs has also followed the availability of datasets. Simulation tools have generated new datasets to support the development of new ML-based MDSs for non-IP V2X applications. However, ML-based MDSs for IP-based applications still leverage datasets generated from computer network testbeds. Finally, we can see that the attention to updating mechanisms to ML models is also weak, even though updating security ML models is crucial to ensure the continuous detection of threats.

TABLE 13: ML-oriented summary of ML-based MDSs proposed for non-IP-based applications)

S: Supervised | U: Unsupervised | H: Hybrid | M-class: Multi-class | B-class: Binary-class | Reg: Regression | AD: Anomaly Detection | Clust: Clustering | Pre: Precision | Recall: Rec | Acc: Accuracy | F1: F1-score | Tra: Traditional | Deep: Deep learning

Work	ML method	Dataset	ML task	Model update	ML Algorithms	Metrics	Inference loc
So et al. [100]	S	VeRiMi	M-class	No	Tra (SVM, KNN)	Pre, Rec	-
Le et al. [101]	S	VeRiMi	M-class	No	Tra (SVM, KNN)	Pre, Rec	-
Singh et al. (1) [102]	S	VeRiMi	B-class	No	Tra (LR, SVM)	F1	-
Sharma et al. (1) [103]	S	VeRiMi	M-class	No	Tra (SVM, KNN, NB, RF, Ensemble, boosting voting)	Acc, Pre, Rec, F1	-
Kosmanos et al. [104]	S	Generated (Veins)	B-class	No	Tra (KNN and RF)	FPR, TPR, ROC curve	-
Montenegro et al. [105]	S	Generated (Veins)	B-class	No	Tra (KNN)	Acc, Rec, FPR, TPR	-
Ecran et al. (1) [106]	S	VeReMi	B-class	No	Tra (KNN, RF)	Pre, Rec, Acc, F1.	-
Ecran et al. (2) [107]	S	VeReMi	M-class	No	Tra(KNN, RF , ensemble Learning	Pre, Rec, Acc, F1	-
Hawlader et al. [108]	S	VeReMi	B-class + M-class	No	Tra (SVM, DT, RF, KNN, NB, and LR)	Acc, Prec, Rec, F1	-
Okamura et al. [109]	U	Generated (Scenargy)	-	No	Tra (SST)	Pre, Rec, F1-score	Cloud
Grover et al. (1) [110]	U	VeReMi	-	No	Deep (GRU, LSTM)	Acc, Rec	Edge
Sedar et al. [111]	S	VeReMi extension	-	No	Reinforcement learning	Pre, Rec, F1-score.	-
Uprety et al. [112]	S	VeReMi	B-class	Yes	Deep (FL)	Pre, Rec.	-
Sharma et al. (2) [113, 114]	S	VeReMi	B-class M-class	Yes	Tra (KNN, RF, NB, DT)	Pre, Rec, F1-score.	Cloud
Mankodiya et al. [115]	S	VeReMi	B-class	No	Tra (RF, KNN, AdaBoost)	Pre, Rec, F1-score.	-
Ghaleb et al. [117]	S	Generated (NGSIM + Matlab)	B-class	No	Tra (ANN)	Acc, F1, Rec, Pre	-
Monteuuis et al. [118]	S	Collected (internet)	B-class	No	Tra (ANN, AdaBoost and RF)	TPR, TNR, FPR, FNR, Acc, F1	-
Singh et al. (2) [119]	S	Generated (SUMO)	Reg	No	Tra (ANN) + Deep (LSTM)	-	-
Gyawali et al. [120, 121]	S	VeReMi + Generated (Veins)	B-class + M-class		Tra (LR, KNN, DT, Bagging, RF)	Prec, Rec , F1	CAV
Negi et al. [122]	U	Generated + Real dataset	AD	Yes	Deep (LSTM)	AUC	CAV
Almalki et al. [123]	S	NGSIM dataset	B-class	No	Tra (LR, SVM) Deep (LSTM)	Acc, F1 DR, FPR	-
Gu et al. (1) [130]	S	Generated (SUMO)	B-class	No	Tra (SVM, ANN)	TPR, FPR, FNR	-
Gu et al. (2) [131]	S	Generated (SUMO)	B-class	No	Tra (KNN)	Acc	-

TABLE 13: ML-oriented summary of ML-based MDSs proposed for non-IP-based applications – continued from previous page

S: Supervised | U: Unsupervised | H: Hybrid | M-class: Multi-class | B-class: Binary-class | Reg: Regression | AD: Anomaly Detection | Clust: Clustering | Pre: Precision | Recall: Rec | Acc: Accuracy | F1: F1-score | Tra: Traditional | Deep: Deep learning

Work	ML method	Dataset	ML task	Model update	ML Algorithms	Metrics	Inference loc
Kamel et al. (1) [132]	S	VeReMi extension	M-class	No	Deep (LSTM)	Acc, F1, Rec, Pre	CAV, Cloud
Quevedo et al. [133]	S	Generated (SUMO)	B-class	No	Tra (ANN)	Acc	Edge
Boualouache et al. [134]	S	Syntactic dataset	B-class + M-class	Yes	Tra (LR, KNN, SVM, NB, DT, RF) Deep (FL)	Pre, Rec , F1, Acc	CAV
Grover et al. (2) [135]	S	Generated (NCTUns-5.0)	B-class +M-class	No	Tra (NB,RF, DT, Adaboost, IBL)	TPR, FPR, TNR, FNR	-
Grover et al. (3) [136]	S	Generated (NCTUns-5.0)	B-class	No	Tra (RF, DT, Adaboost, Ensemble based learning,IBL)	TPR, FPR, TNR, FNR	-
Li et al. (1) [137]	S	Generated (GloMoSim)	B-class	No	Tra (SVM)	Pre, Rec	-
Zhang et al. (1) [138]	S	Generated (Veins)	B-class	No	Tra (SVM)	TP, FP, Acc	CAV
Eziama et al. [139]	U	-	AD	No	Tra (Bayesian ANN)	-	-
Mahmoudi et al. [140]	S	VeReMi extension	M-class	No	Tra (RF, XGboost, LGBM, ANN) + Deep (LSTM)	Pre, Rec, F1	-
Kamel et al. (2) [141]	S	VeReMi extension	B-class	No	Tra (SVM, ANN) + Deep (LSTM)	Rec, Prec, F1, Acc	-
Alladi et al. (1) [142]	S	VeReMi extension	B-class M-class	No	Deep (LSTM, CNN)	Acc, Pre, Rec, F1	-
Alladi et al. (2) [143]	S	VeReMi	M-class	No	Deep(RNN/LSTM , CNN)	Acc, Pre Rec, F1	MEC
Alladi et al. (3) [144]	U	VeReMi extension	AD	No	Deep(CNN-LSTM)	Pre, Rec, F1, Acc	RSU
Alladi et al. (4) [149]	U	VeReMi extension	AD	No	Deep(CNN-LSTM)	Pre, Rec, F1, Acc.	RSU
Kushardianto et al. [145]	S	VeReMi extension	B-class M-class	No	Tra (RF) Deep(DL, GRU,LSTM)	Acc	-
Gonçalves et al. [146, 147]	S	Generated (NS3) [148]	M-class	No	Decision Stump RF MLP, J48, RF	Accuracy TPR, FPR	(CAV, RSU, Cloud)
Hsu et al.[150]	S	VeReMi Extension	B-class	No	Tra (SVM) Deep (CNN-LSTM)	Acc, Pre, Rec, F1	RSU
Mankodiya et al. [115]	S	VeReMi	B-class	No	Tra (RF, DT, AdaBoost)	Pre, Rec, F1	-
Ko et al. [124]	S	Generated (PLEXE)	B-class	No	Deep (LSTM)	Acc, F1, DR	CAV
Boddupalli et al. (1) [125]	U	Generated (RDS1000)	AD	No	Tra (ANN)	FR, FN	-
Wang et al. [126]	U	Generated (SUMO)	AD	No	Deep (LSTM, Autoencoder)	F1	RSU
Boddupalli et al. (2) [127]	S	Generated (RDS1000)	Reg	No	Tra (RF Regressor)	-	CAV
Boddupalli et al. (3) [151]	S	Generated (RDS1000)	Reg	No	Tra (ANN)	Rec, Pre, F1, FPR	CAV
Sarker et al. [128]	S	Public dataset + Generated (SUMO)	Reg	No	Deep (ANN, RNN)	Pre, Rec	CAV

TABLE 13: ML-oriented summary of ML-based MDSs proposed for non-IP-based applications – continued from previous page

S: Supervised | U: Unsupervised | H: Hybrid | M-class: Multi-class | B-class: Binary-class | Reg: Regression | AD: Anomaly Detection | Clust: Clustering | Pre: Precision | Recall: Rec | Acc: Accuracy | F1: F1-score | Tra: Traditional | Deep: Deep learning

Work	ML method	Dataset	ML task	Model update	ML Algorithms	Metrics	Inference loc
Liu et al. [152]	S	Generated (Veins)	B-class	No	Tra (KNN, DT, Adaboost, RF, LR)	Acc, Pre, Rec, F1	-
Sedjelmaci et al. [153]	S	Generated (Ns3)	B-class	No	Tra (SVM)	DR, FPR	CAV
Ayoob et al. [129]	S	Generated (Ns2)	B-class	No	Tra (ANN)	-	CAV
Aliev et al. [116]	S	VeReMi	M-class	No	Deep (multi-head CNN-LSTM)	Acc	-

5 Lessons learned and recommendations

Interesting lessons and recommendations could be concluded from the results and analyses presented in the previous section. First, existing ML-based MDSs for 5GB vehicular networks only focus on detecting traditional attacks and ignore threats for 5GB enabling technologies. This survey recommends focusing on the detection of attacks and threats related to 5GB enabling technologies and non-5G domains. It also recommends further investigating the detection of traditional attacks in the 5GB settings since the scenarios can be more complicated. Second, the datasets used to build ML-based MDSs for 5GB vehicular networks are either generated using network simulators in a non-5G setting for non-IP-based V2X applications or obtained from classical network testbeds for IP-based V2X applications. Even though these data are still for 5GB vehicular networks, efforts should be made to generate security datasets considering both the unique characteristics of vehicular networks (e.g., mobility) and 5GB settings, as discussed in subsection **??**. Third, the proposed ML-based MDSs strongly depend on the training dataset. Specifically, the attacks detected by these MDS only cover the attacks included in datasets, unlike what was described in the titles and the abstracts of most research papers. These latter give the impression that they are proposing solutions that address all the attacks instead of specific attacks. This survey strongly recommends authors be more specific when presenting their ML-based MDSs. It also recommends proposing holistic security frameworks integrating different ML-based MDSs for 5GB vehicular networks to cover various existing attacks, including unseen attacks. Fourth, this survey concludes that the existing ML-based MDSs for 5GB vehicular networks are generally incomparable due to the absence of benchmark security datasets and the unuse of unified evaluation metrics. Indeed, an essential part of research papers generates their datasets, making it difficult to reproduce and compare their results with other solutions. The rest of the research papers use public data sets such as VeReMi and VeRemi extension. However, these datasets are difficult to be appointed as benchmark datasets due to several data described in the open issues section (6.2). To this end, this survey recommends gathering efforts for defining unified benchmark datasets. It also recommends specifying a common evaluation framework consisting of all metrics used to evaluate and compare ML-based MDSs. This survey encourages researchers to reproduce the results of different ML-based MDSs and compare the results. This task cannot be easy since the majority of authors did not well explain the methodology and the parameters of the ML models. Thus, this survey recommends researchers working in this field to: (i) deepen their knowledge in ML, (ii) include the required parameters to reproduce their results, and (iii) make the implementation publically available. Fifth, as stated in the ML-oriented summary, most ML-based MDSs leverage conventional ML approaches to detect attacks. However, while conventional ML approaches demonstrate good performance in detecting some attacks, they have also shown several limitations, such as (i) low performance in detecting specific attacks, (ii) detecting attacks with small datasets, and (iii) privacy preservation while detecting attacks. These limitations demonstrate the need for more sophisticated ML approaches to address them. For example, DL approaches and architecture have been proposed to enhance the performance results. FL comes to address privacy preservation while building attack detection models. Transfer learning addresses the problem of dataset scarcity regarding specific types of attacks and enables knowledge built on detecting attacks with large datasets to detect attacks with small data. This survey recommends investigating more sophisticated ML algorithms and advanced concepts for building ML-based MDSs for 5GB vehicular networks to address open issues such as detecting zero-day attacks and autonomous attack detection. Here, advanced concepts mean the concepts that have emerged in the ML field which have not yet been applied in ML-based MDSs for 5G vehicular networks. Sixth, most of the proposed ML-based MDSs for 5GB vehicular networks have not considered the inference location and the ML model's update, ignoring thus the deployment phase of ML-based MDSs. Therefore, this survey recommends paying

TABLE 14: ML-oriented summary of ML-based MDSs proposed for IP-based applications

S: Supervised | U: Unsupervised | H: Hybrid | M-class: Multi-class | B-class: Binary-class | Reg: Regression | AD: Anomaly Detection | Clust: Clustering | Pre: Precision | Recall: Rec | Acc: Accuracy | F1: F1-score | Tra: Traditional | Deep: Deep learning

Work	ML method	Dataset	ML task	Model update	ML Algorithms	Metrics	Inference loc
Tan et al. [155]	U	Generated (Python)	Clust	No	Tra (AHC)	DR	CAV RSU
Singh et al. (3) [157]	S	Generated (Mininet-WiFi)	B-class	No	Tra (LR, DT, NB, SVM, KNN, ANN Gradient boosting)	TP, TN FP, FN	SDN
Yu et al. [158]	S	DARPA CAIDA DDos2007	B-class	No	Tra (SVM)	DR	-
Sharshembiev et al. [159]	U	Generated (Veins)	AD	No	Tra (Entropy-based)	Pre Rec F1	-
Gruebler et al. [164]	S	Generated (NS2 and SUMO)	B-class	No	Tra (ANN)	TP, TN FP, FN	-
Alheeti et al. (1) [165, 166]	S	Generated (NS2 and SUMO)	B-class	No	Tra (SVM, ANN)	Acc, TP, TN, FN, FP	-
Zeng et al. (1) [167]	S	Generated (GloMoSim)	B-class	No	Tra (ANN, SVM)	Acc	CAV RSU
Siddiqui et al. [168]	H	CRAWDAD	B-class	No	Tra (KNN, SVM)	Acc	-
Acharya and Oluoch [169]	S	Generated (NS3)	B-class	No	Tra (NB, LR, KNN, SVM, Gradient boosting)	Rec, Prec, F1, Acc, FP, FN, ROC_AUC	-
Singh et al. (4) [172]	S	Generated (SUMO)	B-class	No	Tra (KNN, SVM)	TP, TN FP, FN	-
Alheet et al. (2) [173, 174]	S	Kyoto	M-class	No	Tra (ANN)	TP, TN FP, FN	-
Kim et al. [175]	S	KDD CUP 1999	M-class	No	Tra (SVM)	Acc, Pre Rec.	CAV
Zhang et al. (2) [176, 177]	S	NSL-KDD	B-class	Yes	Tra (LR)	Loss	CAV
Ghaleb et al. [178]	S	NSL-KDD	B-class	Yes	Tra (RF, XGBoost, SVM)	Acc, Pre, FP F1-score, FN DR	CAV
Ashraf et al. [179]	U	UNSW-NB15	AD	No	Deep (LSTM)	Pre, Rec, TPR Acc, F1	CAV
Shu et al. [180]	S	KDD99	B-class	No	Deep (GAN)	Acc, Pre, Rec, F1	Cloud
Li et al. (2) [181]	S	AWID	M-class	Yes	Tra (SVM +RF) Transfer learning	Acc, FPR FNR	CAV
Bangui et al. [182]	H	CICIDS2017	M-class AD	No	Tra (RF based)	Acc, F1	-
Yang et al. (1) [183]	H	CICIDS2017	M-class AD	No	Tra (DT, RF,ET, XGBoost, stacking)	Acc, F1, DR, False alarm	CAV

TABLE 14: ML-oriented summary of ML-based MDSs proposed for IP-based applications – continued from previous page

S: Supervised | U: Unsupervised | H: Hybrid | M-class: Multi-class | B-class: Binary-class | Reg: Regression | AD: Anomaly Detection | Clust: Clustering | Pre: Precision | Recall: Rec | Acc: Accuracy | F1: F1-score | Tra: Traditional | Deep: Deep learning

Work	ML method	Dataset	ML task	Model update	ML Algorithms	Metrics	Inference loc
Khan et al. [184]	U	UNSWNB15	AD	No	Deep (LSTM)	Acc, Rec, Pre, F1	CAV
Liu et al. (2) [185]	S	KDDCup99	B-class	Yes	DL FL	Acc, Pre Rec	-
Rahal et al. [186]	S	Generated (NS3)	M-class	No	Tra (KNN, ANN, SVM RF, DT, NB)	Acc, Pre Rec, F1	Vehicle
Abdel Wahab et al. [170]	S	Generated (VanetMobiSim)	B-class	No	Tra (SVM)	Acc, DR FPR	CAV
Narayanadoss et al. [160]	S	Generated (MiniNet-WiFi)	B-class	No	Tra (ANN) Deep (CNN, LSTM)	Acc, Pre, Rec, F1	SDN Controller
Maglaras et al. [161]	U	Generated (private Simulator)	AD	No	Tra (K-OCSVM (KNN+OCSVM))	Acc	CAV RSU
Tian et al. [162]	U	Generated (NS2)	AD	No	Tra (ANN)	FP, FN	Backbone
Liu et al. (1) [187]	S	Generated (Testbed)	B-class	No	Tra (NB, LR)	Pre, Rec, F1	Backbone
Nie et al. [163]	U	Generated (Testbed)	AD	No	Deep (CNN)	TPR	-
Zeng et al. (2) [154]	S	ISCX 2012 IDS + Generated (Ns3)	M-class	Yes	Deep (CNN+LSTM)	Pre, Rec, F1-score	-
Shams et al. [171]	S	Generated (Ns2)	B-class	No	Tra (SVM)	Pre, Rec, F1	CAV
Yang et al. (2) [188]	S	CIC-IDS 2017	M-class	No	Deep (ensemble learning, CCN), Transfer learning	Acc, Pre, Rec, F1	-

TABLE 15: ML-oriented summary of ML-based MDSs proposed for both IP-based and non-IP-based applications

S: Supervised | U: Unsupervised | H: Hybrid | M-class: Multi-class | B-class: Binary-class | Reg: Regression | AD: Anomaly Detection | Clust: Clustering | Pre: Precision | Recall: Rec | Acc: Accuracy | F1: F1-score | Tra: Traditional | Deep: Deep learning

Work	ML method	Dataset	ML task	Model update	ML Algorithms	Metrics	Inference loc
Karagiannis et al. [189]	U	Generated (R)	AD	No	Tra (K-means)	-	-
Lyamin et al. [190]	U	Generated (MATLAB))	AD	No	Tra (Statistical Data mining)	F1	CAV
Abhishek and Gurusamy [192]	U	Generated (NS3)	AD	No	Tra (One-class SVM)	DR	CAV
Kosmanos et al. [193]	S U	Generated (Veins)	B-class AD	No	Tra (SVM, RF Ensemble learning, OCSVM)	ROC curve	-

careful attention to the deployment phase, which has an important impact on the detection rate and the feasibility of the ML-based MDS. In addition, evaluation should include new indirect metrics to study ML-based MDSs deployments, such as the size of the ML model and the inference time (time to detect the attack). It is also important to carefully examine inference location from the detection and security and privacy perspectives. On the other hand, model update mechanisms should be defined to prevent loss of accuracy with time. In this direction, collaborative MDS offers interesting opportunities to update the ML model and provide privacy preservation smoothly. For this reason, this survey recommends promoting research in this direction by exploring more advanced ML concepts such as online learning and reinforcement learning. Seventh, ML-based MDSs, including collaborative ones, still face various security threats, such as adversarial attacks and poisoning. The analysis in this survey identifies only one work that considers the security of the ML-based MDS. To this end, the authors of this survey believe that the security of ML-based MDSs is an urgent issue that requires concerted efforts.

Key highlights from the lessons learned

1) Current ML-based MDSs for 5GB vehicular networks focus on detecting traditional attacks and ignore attacks from 5GB enabling technologies.
2) There is a lack of security datasets for 5GB vehicular networks, which consider unique characteristics of vehicular networks (e.g., mobility).
3) Titles and abstracts of most papers on ML-based MDSs for 5GB vehicular networks do not precisely mention detected attacks, which depend on the used datasets.
4) Current ML-based MDSs for 5GB vehicular networks are generally incomparable due to the absence of benchmark datasets and the unuse of unified evaluation metrics.
5) Most solutions leverage traditional ML approaches to detect attacks. Some solutions have exploited advanced ML techniques to overcome the limitations of conventional ML approaches.
6) Most ML-based MDSs for 5GB vehicular networks do not consider the inference location and the ML model's update, ignoring thus solutions' deployment.
7) Most ML-based MDSs for 5GB vehicular networks face various security threats such as adversarial attacks and poisoning.

6 Open Research Issues

Several parameters involve in building effective ML-based MDSs for 5GB vehicular networks, such as the quality of datasets and the used ML algorithms. However, although considerable efforts have been made, several open issues still need more attention to achieve the aimed ML-based MDS. These issues, illustrated in Figure 11, are discussed in this section. This section also discusses some potential solutions to address these issues.

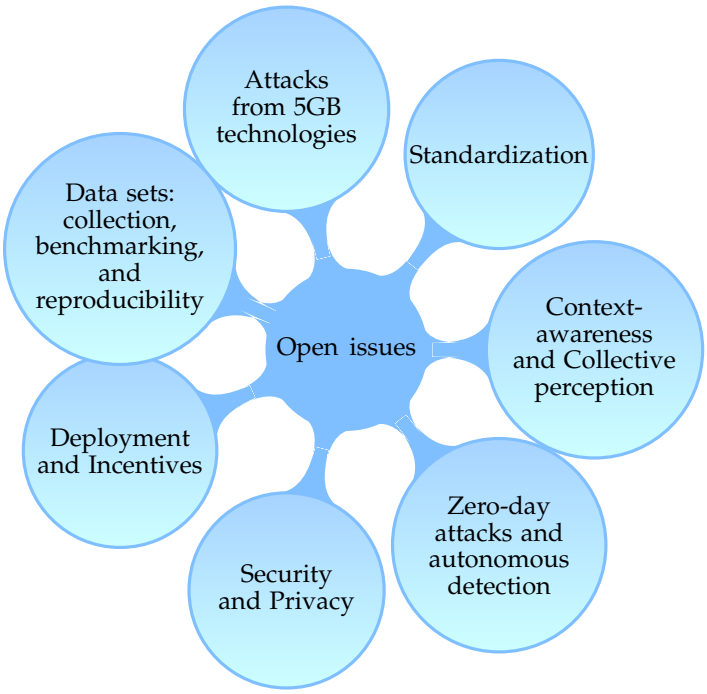


Fig. 11: Open issues of ML-based MDSs for 5GB vehicular networks

6.1 Attacks from 5GB technologies

Empowering CAVs with 5GB technologies has brought many opportunities but, at the same time, opened up new vulnerabilities and surfaces of attacks to CAVs. While traditional attacks have been widely covered, attacks posed by 5G enabling technologies have been less considered [65]. The survey authors believe ML could also be a key enabler to effectively detect attacks from both 5GB and non-5GB domains. For example:

- ML-based MDSs could early and efficiently detect attacks on V2X network slices. Attacks on NS could be: (i) intra-slice: in which the attacker(s) and the target(s) belong to the same V2X network slice, or (ii) inter-slice attacks in which the attacker(s) and the target(s) belong to different V2X network slices. ML-based MDSs could be efficient in detecting both of these attack types. In addition, the FL concept could help detect attacks while preserving the isolation of V2X network slices.
- Mobility management for CAVs in 5G is challenging. CAVs are characterized by high mobility, performing frequent horizontal and vertical handovers [195]. Handovers may create a performance constraint for CAVs, especially road safety applications. Specifically, if the loss of the network connection impedes real-time decision-making, the consequences could be disastrous. Context-aware mobility management solutions have been proposed to deal with this. These solutions leverage predictive ML models taking context information of CAVs as input to proactively prepare handovers so that the network connection can be seamlessly transitioned while meeting the delay requirements [196]. Moreover, CAVs in 5G need fre-

quent mutual authentications along with handovers to prevent external attacks such as impersonation and man-in-the-middle [69]. However, while current authentication schemes are efficient in preventing these attacks, they generally come at the expense of high computation costs and long handover delays [197, 198]. Using lightweight cryptographic-based [199] and secure context information-based authentication [200] schemes seems promising to reduce delays, but they will also reduce the security and privacy levels.

To this end, ML-based MDSs come to complement authentication schemes to secure mobility management by proactively detecting attacks conducted during handover and roaming processes and protecting context information shared during these processes [201]. Specifically, the use of ML-based MDSs is crucial, especially in the case existing malicious actors can launch cyberattacks by taking advantage of handover and roaming characteristics, such as different security configurations and needs at different MNOs. Here collaborative ML architecture could be more appropriate to support the mobility of CAVs while detecting attacks. ML-based MDSs can be deployed at access points (gNodeB) and MEC stations, usually used to manage context information for predicting handovers. These ML-based MDSs can collaborate and coordinate to detect attacks on handovers. However, detecting attacks on roaming scenarios could be more complicated since home and visited MNOs are non-cooperative in sharing information regarding their networks. Thus, the challenge is how MNOs can collaborate to detect attacks in roaming while preserving their private data. FL could be a good candidate to use here to enable privacy-preserving collaboration in building global attack detection models. Specifically, each MNO can train local models based on their data and contribute to building a global model for detecting roaming attacks without sharing data.

- ML-based MDSs could detect attacks coming from non-5G domains. On the one hand, ML-based MDSs built for 5GB vehicular networks are still relevant for non-5G vehicular domains such as ITS-G5 since both 5G and non-5G vehicular domains share the same vectors of traditional attacks. However, the focus should be put on detecting attacks that might happen to CAVs during the vertical handovers between the two vehicular domains. On the other hand, for other non-5G domains, cellular (i.e., LTE) or not cellular such as satellite or WiFi, detecting these attacks should be ensured by the 5GB network core, which acts as a broker between these domains and 5GB vehicular networks. Thus, ML-based MDSs could be built and deployed on the 5GB network core to protect CAVs from attacks coming from non-5G domains [202].

Besides, traditional security and privacy attacks still need further attention in the 5GB domain. For example, trajectory tracking can be done by collecting and linking CAMs messages broadcast by CAVs. Using pseudonyms can help to some extent to protect against these attacks [203]. However, identifiers used by CAVs in the 5GB environment can provide assets for attackers to link the CAMs. A first work demonstrated that ML-MDSs could help in detecting passive adversaries [134].

6.2 Data sets for 5GB vehicular networks: collection, benchmarking, and reproducibility

As already discussed in the learned lessons section, the absence of benchmark security datasets for 5GB vehicular networks is a major open issue in reproducing results of existing ML-based MDSs and comparing them. All datasets used by current ML-based MDSs for 5GB vehicular networks have not been generated using cellular testbeds or cellular simulations. It is hard to find real-world cellular security datasets because telco companies prefer to keep their data (especially data on security) confidential to avoid hurting their business and reputation. However, all datasets listed in Table 5 are still relevant to the 5GB vehicular context. This is because: (i) 5GB vehicular networks share the same protocols with non-cellular networks in several protocol stack layers such as IP, TCP/UDP, and HTTP. Thus, attacks on these protocols are not only valid in non-cellular networks but also in cellular network domains like 5GB vehicular networks, and (ii) Attacks on V2X applications are independent of the underlying protocol stack, whether non-cellular (e.g., ITS-G5) or cellular (5G-V2X). In other words, attack datasets on V2X applications generated in a non-cellular setting are still relevant for 5GB vehicular networks. However, existing datasets still miss attacks on telco-specific protocols such as GPRS Tunneling Protocol (GTP), NG Application Protocol (NGAP), and Stream Control Transmission Protocol (SCTP), which can only be generated using a cellular testbed. Some lab testbeds have recently been set up in this direction to generate such data [204, 205]. However, these testbeds concern only the 5G network core part, and their produced datasets are not publically available. To this end, generating security datasets for vehicular networks in a 5GB setting is still an issue. 5GB vehicular network security datasets should consider the core network, but most importantly, the mobility of CAVs and 5GB enabling technologies, including mobility management procedures such as handover and roaming. Moreover, the authors of this survey believe that it is difficult to appoint benchmark datasets without standardized procedures that clearly define V2X attack scenarios. They believe efforts should be gathered to define clear and standardized V2X attack scenarios to generate unified benchmark datasets using realistic 5GB vehicular testbeds.

Besides, a common evaluation framework consisting of all metrics used to evaluate and compare ML-based MDSs should also be specified. In addition, the reproducibility of results is another issue since authors tend to neither mention ML parameters nor make their source codes public as discussed in section 5. For this reason, this survey encourages authors to adopt a result reproducibility methodology. It also encourages researchers to reproduce the results of different ML-based MDSs and compare the results.

6.3 Zero-day attacks and autonomous detection

Zero-day attacks are vectors of unseen attacks that appear over time due to the evolution of technologies (i.e., network slicing [206]) and attacker strategies [207]. 5GB vehicular networks are highly vulnerable to zero days attacks since CAVs are under the control of their users, which can then modify software and hardware to generate attacks. Most existing ML-based MDSs were built on a supervised approach, allowing the detection of known attacks listed in the used datasets. In addition, the rest of the ML-based MDSs use unsupervised models based only on normal data. Thus, they can detect anomalies but cannot identify them. We believe that investigating recent ML advances could help detect zero-day attacks. For example, N-shot learning has shown promising results for identifying unseen classes of images in vision applications [208]. This could be a motivation to use them for detecting zero-day attacks. In addition, new emerging hybrid semi-automatic frameworks, including humans in the detection loop, are promising [209]. The system leverage ML to detect anomalies and the security operations center for identifying zero-day attacks and updating the ML system. However, the ultimate goal is to detect attacks on 5GB vehicular networks automatically since there is currently no such approach that enables autonomous detection in different contexts for CAVs under the umbrella of the zero-touch paradigm [210].

6.4 Context-awareness and Collective perception

Due to their high mobility, 5GB vehicular networks are exposed more than other applications to the context-changing. Almost all the existing ML-based MDSs for 5GB vehicular networks leverage the direct parameters to detect attacks while ignoring indirect parameters, which also influence attack detection accuracy. For example, detecting message greyhole/blackhole attacks requires monitoring message exchanges between CAVs. However, the messages could be suppressed intentionally due to environmental characteristics such as obstacles and interference. Thus, other indirect parameters such as channel status, temperature, and speed should also be considered to detect such behavior. Although two ML-based MDSs [123, 137] consider the context, their contributions are still limited since indirect parameters are not included in the ML training. In this vein, Besides, SDN approaches to change security parameters according to the context are interesting to investigate [180, 211].

On the other hand, for V2V communications, existing ML-MDSs have mainly focused on detecting misbehavior in periodic messages, i.e., CAM, and even-trigger messages, i.e., DEMN. However, less attention has been paid to secure CPM messages generated by collection perception services. As aforementioned, this service is very important for CAVs to detect non-V2X objects such as legacy vehicles, pedestrians, and animals and to extend vehicle perception through a fusion process with CAM and DENM messages. Thus, attacking CPMs can have catastrophic results on road safety. To this end, developing MDSs to detect misbehaviors has become a must. Ansari et al. have recently conducted a threat assessment on CPS use cases defined by ETSI [36, 37]. In addition, a few non-ML-based MDSs have been proposed to detect CPM misbehaviors [212–216]. However, more efforts still have to be performed in this direction, and ML is one promising technique to improve detection capabilities. Indeed, ML algorithms have proven this efficiency in sensor data carried out by CPM messages. For example, CNNs are mainly designed for visual and imagery applications.

6.5 Security and Privacy

As discussed in the analysis section, most existing ML-based MDSs of 5GB vehicular networks are centralized, where the datasets are collected, and the ML model is trained in one location. This exposes them to serious security and privacy issues for the following reasons: (i) these solutions are suffering from single-point-of-failure attacks since the model is trained on a single location; and (ii) collecting datasets by one entity could have several privacy violations since datasets may contain sensitive information about the behavior and movement patterns of V2X nodes. Collaborative FL-based MDSs [112, 134, 185] came to partially address privacy issues since datasets in these systems are shared among learning nodes instead of centrally stored. However, security issues are multiplied by the number of components in collaborative systems. In addition, FL-based MDSs are still suffering from single-point-of-failure attacks since the global model is aggregated and calculated in one FL server. To this end, the authors of this survey believe that blockchain could be an efficient technology to secure ML-based MDSs [185]. However, some problems still need to be addressed in the blockchain design to achieve this aim, such as the consensus algorithms and optimizing smart contracts.

Besides, like other ML-based systems, ML-based MDSs are also suffering from adversarial ML [217, 218]. Adversarial ML is a set of techniques that try to exploit models by using the information obtained from models to launch advanced attacks. For example, learning nodes still send small updates to the FL server even if the datasets are not shared in FL. An attacker can use this information to infer sensitive information about the model and thereby launch attacks to poison the model [219]. Although recent work has addressed the adversarial ML attacks issue [220], the survey authors believe this issue still needs careful attention.

6.6 Deployment and Incentives

Building a successful ML-based MDS for 5GB vehicular networks depends on not only the development and validation phases but also the deployment phase. In addition to the datasets and the model accuracy, ML-based MDSs should be developed to fit the V2X environment (software and hardware) in which these systems will be deployed. The deployment of ML-MDSs in 5GB vehicular networks differs from other applications due to many factors such as mobility, equipment heterogeneity, and hardware performance. Therefore, ML-based MDSs should be developed with the end in mind considering metrics such as the model's size and processing resources required to run the models in the evaluation part. Besides, the placement of ML-based MDS components on 5GB vehicular networks should be studied to provide early attack detection and a rapid reaction while protecting them from vulnerabilities. The authors of this

survey believe that deploying ML-based MDSs in 5GB vehicular networks is an open issue that requires efforts from both research and industry. Besides, since the deployment of ML-based MDSs in V2X nodes will consume storage and processing resources, the manager of these nodes might be against deploying ML-based MDSs. Thus, the incentive issue should also be addressed to ensure the continuity of ML-based MDS services. Some works have started interesting in incentive modeling using game theory frameworks [221]. However, the authors of this survey believe more efforts can still be made in this direction.

6.7 Standardization

The first efforts on standardization of MDSs for vehicular networks are ongoing. ETSI has recently published a technical report on the pre-standardization study of V2X misbehavior detection [16] and is currently working on the technical specification [17]. Although several detection techniques are mentioned in this report, the role of ML is not well emphasized. The authors of this survey believe standardization bodies should focus more on defining a toolbox for developing and validating ML-based MDSs for 5GB vehicular networks. Consequently, this survey can identify several standardization opportunities: 1) defining attack scenarios in complementary with the ETSI technical report 102 893 [222]; (2) defining benchmark datasets; (4) defining validation KPIs; (5) specifying evaluation metrics; and (6) establish clear validation procedures. On the other hand, standardization bodies should organize plug-tests events (e.g. [223]) that gather several stakeholders for testing and validating ML-based MDSs with reporting relevant results, as is the case [224–226].

7 Conclusion

Misbehavior Detection Systems are key building blocks for securing 5GB vehicular networks. Machine Learning is an indispensable tool of the design of these systems. An increasing effort is ongoing to provide effective ML-based MDSs. This paper surveyed and classified relevant ML-based MDSs for 5GB vehicular networks. It also analyzed and discussed them from security and ML perspectives. Finally, It gave some learned lessons and shed light on open research and standardization issues for building effective ML-based MDSs.

This survey showed that ML-based MDSs for 5GB vehicular networks are still in their first stage of development. Much effort is still being made in this research area. The results of this survey can be used to build a roadmap for the research community to accelerate the development of ML-based MDSs for 5GB vehicular networks. The starting step could be to enlarge knowledge of attacks against CAVs in the 5GB environment and their potential scenarios. This step is essential for building realistic testbeds to generate reliable attack datasets, serving as benchmarks to validate and compare results.

Following trends in ML fields and employing state-of-art ML algorithms and concepts is also essential to enhance attack detection results, detect unseen attacks, update ML models, and provide early detection. For example, N-shot learning could detect and identify unseen classes of zero-day attacks. Online learning and deep reinforcement learning could help to build context-aware ML-based MDSs for 5GB vehicular networks that can be self-adapted to different contexts. Moreover, multiple ML models can be combined inside the same ML-based MDS, which switches between them according to triggering contexts. Besides, it is also promising to investigate semi-automatic ML systems that involve humans in the loop to detect unseen attacks. These systems use ML to detect anomalies and security analysts to identify attacks and update the ML models.

The next step is to study the deployment of ML-based MDSs carefully, considering different performance metrics, the security of ML models, and incentive aspects to ensure sustainability. ML Operations (MLOps) is a key enable in this stage. ML Operations (MLOps) approach is a key enable in this stage. MLOps is a developing field with principles and tools to help with the ML project lifecycle, especially data processing, model building, and deployment. MLOps can address both ML and software engineering issues in deploying ML-based MDSs. ML issues mainly include data and concept drifts. "Data drift" happens when data distribution changes after the deployment. "Concept drift" happens when the mapping between the input and output of the ML models changes. For example, high network traffic generated by CAVs should have been detected as anomalies using ITS-G5. After 5G-V2X, the same network traffic should not cause an anomaly with the high bandwidth offered by the 5G technologies. Thus, MLOps can detect and manage changes and adapt the ML-based MDS to avoid these issues.

On the other hand, MLOps can help tackle software engineering issues. According to detected attacks, MLOps can help decide whether ML-based MDSs deployed to make real-time or batch detection and what is the best place to deploy ML-based MDS (e.g., CAVs, MEC, and/or core). In addition, MLOps tools allow monitoring of how much ML-based MDSs consume processing and memory resources. Other real-time software engineering metrics, such as latency and throughput, can also be monitored. Moreover, MLOps provides services to log data for analysis and review and to provide more data for retraining ML-based MDSs. Finally, regarding security and privacy, MLOps can help to customize an appropriate level of security and privacy on ML-based MDSs based on data sensitivity and regulatory requirements.

Finally, condensing standardization activities will help to accelerate the adoption of ML-based MDSs by the industry. More specifically, the definition of an ML methodology for detecting misbehaviors on top of the plausibility checks specified by current standards is required. In addition, standardization bodies should further focus on detecting misbehaviors in cooperative perception services. Defining standard specifications will enable the creation of new business opportunities in ML-based MDSs for 5GB vehicular networks.

Acknowledgment

This work was supported by the 5G-INSIGHT bilateral project (ID: 14891397) / (ANR-20-CE25-0015-16), funded by

the Luxembourg National Research Fund (FNR), and by the French National Research Agency (ANR).

Special thanks go to Hadjer Benseghir for supporting this work.

REFERENCES

[1] H. Yin, L. Zhang, and S. Roy, "Multiplexing URLLC Traffic within eMBB Services in 5G NR: Fair Scheduling," *IEEE Transactions on Communications*, vol. 69, no. 2, pp. 1080–1093, 2020.

[2] J. Wang, J. Liu, and N. Kato, "Networking and Communications in Autonomous Driving: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1243–1274, 2018.

[3] G. Volk, Q. Delooz, F. A. Schiegg, A. Von Bernuth, A. Festag, and O. Bringmann, "Towards Realistic Evaluation of Collective Perception for Connected and Automated Driving," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1049–1056.

[4] D. Hetzer, M. Muehleisen, A. Kousaridas, and J. Alonso-Zarate, "5g connected and automated driving: Use cases and technologies in cross-border environments," in *2019 European Conference on Networks and Communications (EuCNC)*. IEEE, 2019, pp. 78–82.

[5] V. Sharma, I. You, and N. Guizani, "Security of 5G-V2X: Technologies, standardization, and research directions," *IEEE Network*, vol. 34, no. 5, pp. 306–314, 2020.

[6] R. Lu, L. Zhang, J. Ni, and Y. Fang, "5G vehicle-to-everything services: Gearing up for security and privacy," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 373–389, 2019.

[7] J. Wang, Y. Shao, Y. Ge, and R. Yu, "Physical-layer authentication based on adaptive Kalman filter for V2X communication," *Vehicular Communications*, vol. 26, p. 100281, 2020.

[8] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban, and R. Bansal, "A Comprehensive Review of Authentication Schemes in Vehicular Ad-Hoc Network," *IEEE Access*, vol. 9, pp. 31 309–31 321, 2021.

[9] Y. Yang, L. Wei, J. Wu, C. Long, and B. Li, "A Blockchain-based Multi-domain Authentication Scheme for Conditional Privacy Preserving in Vehicular Ad-hoc Network," *IEEE Internet of Things Journal*, 2021.

[10] ETSI TS 102 941, "Intelligent Transport Systems (ITS); Security; Trust and Privacy Management; Release 2 ," Oct 2021.

[11] S. Gyawali, Y. Qian, and R. Q. Hu, "Deep Reinforcement Learning based Dynamic Reputation Policy in 5G based Vehicular Communication Networks," *IEEE Transactions on Vehicular Technology*, 2021.

[12] E. Bout, V. Loscri, and A. Gallais, "How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021.

[13] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "AI Models for Green Communications Towards 6G," *IEEE Communications Surveys & Tutorials*, 2021.

[14] A. L. Buczak and E. Guven, "A survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.

[15] F. Tang, B. Mao, N. Kato, and G. Gui, "Comprehensive Survey on Machine Learning in Vehicular Network: Technology, Applications and Challenges," *IEEE Communications Surveys & Tutorials*, 2021.

[16] ETSI TR 103 460 , "Intelligent Transport Systems (ITS);Security; Pre-standardization study on Misbehaviour Detection; Release 2," Oct 2020.

[17] ETSI TS 103759, "Intelligent transport systems (its); security; misbehaviour reporting service," Dec 2021.

[18] S. Keele *et al.*, "Guidelines for performing systematic literature reviews in software engineering," Technical report, ver. 2.3 ebse technical report. ebse, Tech. Rep., 2007.

[19] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Networks*, vol. 61, pp. 33–50, 2017.

[20] M. Arshad, Z. Ullah, N. Ahmad, M. Khalid, H. Criuckshank, and Y. Cao, "A survey of local/cooperative-based malicious information detection techniques in VANETs," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, pp. 1–17, 2018.

[21] S. Sharma and A. Kaul, "A survey on Intrusion Detection Systems and Honeypot based proactive security mechanisms in VANETs and VANET Cloud," *Vehicular communications*, vol. 12, pp. 138–164, 2018.

[22] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 779–811, 2018.

[23] G. K. Rajbahadur, A. J. Malton, A. Walenstein, and A. E. Hassan, "A Survey of Anomaly Detection for Connected Vehicle Cybersecurity and Safety," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 421–426.

[24] A. M. Alrehan and F. A. Alhaidari, "Machine learning techniques to detect DDoS attacks on VANET system: a survey," in *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2019, pp. 1–6.

[25] F. Gonçalves, B. Ribeiro, O. Gama, A. Santos, A. Costa, B. Dias, J. Macedo, and M. J. Nicolau, "A Systematic Review on Intelligent Intrusion Detection Systems for VANETs," in *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2019, pp. 1–10.

[26] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, Y. Zhang, Y. Xiang, and S. Yu, "Attacks and defences on intelligent connected vehicles: a survey," *Digital Communications and Networks*, vol. 6, no. 4, pp. 399–421, 2020.

[27] S. A. Almalki and J. Song, "A Review on Data Falsification-Based attacks In Cooperative Intelligent Transportation Systems," *International Journal of Computer Science and Security (IJCSS)*, vol. 14, no. 2, p. 22, 2020.

[28] X. Sun, F. R. Yu, and P. Zhang, "A Survey on Cyber-Security of Connected and Autonomous Vehicles (CAVs)," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[29] A. Talpur and M. Gurusamy, "Machine Learning for Security in Vehicular Networks: A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021.

[30] H. Bangui and B. Buhnova, "Recent Advances in Machine-Learning Driven Intrusion Detection in Transportation: Survey," *Procedia Computer Science*, vol. 184, pp. 877–886, 2021.

[31] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, 2019.

[32] M. Dibaei, X. Zheng, Y. Xia, X. Xu, A. Jolfaei, A. K. Bashir, U. Tariq, D. Yu, and A. V. Vasilakos, "Investigating the Prospect of Leveraging Blockchain and Machine Learning to Secure Vehicular Networks: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[33] ETSI EN 302 637-2, "Intelligent Transport Systems (ITS); Vehicular Communications;Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," Nov 2014.

[34] SAE J2735_202007, "V2X communications message set dictionary," Jul 2020.

[35] ETSI EN 302 637-2, "Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," Sep 2014.

[36] ETSI TR 103 562, "Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Analysis of the Collective Perception Service (CPS); Release 2 ," Dec 2019.

[37] ETSI TS 103 324, "Intelligent transport system (its); vehicular communications;basic set of applications; specification of the collective perception service," May 2021.

[38] A. Bazzi, A. Zanella, I. Sarris, and V. Martinez, "Co-channel Coexistence: Let ITS-G5 and Sidelink C-V2X Make Peace," in *2020 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. IEEE, 2020, pp. 1–4.

[39] C. Campolo, A. Molinaro, and V. Sciancalepore, "5G Network Slicing for V2X Communications: Technologies and Enablers," *Radio Access Network Slicing and Virtualization for 5G Vertical Industries*, pp. 239–257, 2021.

[40] 3GPP TS 22.186, "Service requirements for enhanced V2X scenarios," Jul 2018.

[41] 3GPP TS 23.287, "Architecture enhancements for 5G System (5GS) to support Vehicle-to-Everything (V2X) services," Jul 2020.

[42] ETSI EN 302 636-1, "Intelligent Transport Systems (ITS);Vehicular Communications; GeoNetworking; Part 1: Requirements," Feb 2014.

[43] ETSI TS 102 636-5-1, "Intelligent Transport Systems (ITS);Vehicular Communications; GeoNetworking; Part 5: Transport Protocols; Sub-part 1: Basic Transport Protocol," Feb 2011.

[44] A. Abunei, C.-R. Comşa, and I. Bogdan, "Implementation of ETSI ITS-G5 based inter-vehicle communication embedded system," in *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*. IEEE, 2017, pp. 1–4.

[45] Qualcomm 80-PE732-63 Rev B, "Cellular-V2X Technology Overview," 2019.

[46] Qualcomm 80-PE732-64 Rev A, "Its stack," 2019.

[47] 3GPP TS 22.185, "Service requirements for V2X services," Mar 2017.

[48] D. Sempere-García, M. Sepulcre, and J. Gozalvez, "LTE-V2X Mode 3 scheduling based on adaptive spatial reuse of radio resources," *Ad Hoc Networks*, vol. 113, p. 102351, 2021.

[49] X. Ge, Z. Li, and S. Li, "5G Software Defined Vehicular Networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 87–93, 2017.

[50] "What is NFV?" https://www.ibm.com/cloud/learn/machine-learning, accessed: 2022-07-25.

[51] H. Cao, H. Zhao, D. X. Luo, N. Kumar, and L. Yang, "Dynamic Virtual Resource Allocation Mechanism for Survivable Services in Emerging NFV-Enabled Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[52] M. M. Elsayed, K. M. Hosny, M. M. Fouda, and M. M. Khashaba, "Vehicles communications handover in 5G: A survey," *ICT Express*, 2022.

[53] V. O. Nyangaresi and A. J. Rodrigues, "Efficient handover protocol for 5G and beyond networks," *Computers & Security*, vol. 113, p. 102546, 2022.

[54] K. Trichias, P. Demestichas, and N. Mitrou, "Inter-PLMN Mobility Management Challenges for Supporting Cross-Border Connected and Automated Mobility (CAM) Over 5G," *Journal of ICT Standardization*, pp. 113–146, 2021.

[55] 3GPP TR 22.886, "Study on enhancement of 3GPP Support for 5G V2X Services," Dec 2018.

[56] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A Tutorial on 5G NR V2X Communications," *IEEE Communications Surveys Tutorials*, vol. 23, no. 3, pp. 1972–2026, 2021.

[57] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "A survey on pseudonym changing strategies for vehicular ad-hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 770–790, 2017.

[58] M. Raya and J. Hubaux, "Securing Vehicular Ad Hoc Networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, jan 2007.

[59] M. Terrovitis, G. Poulis, N. Mamoulis, and S. Skiadopoulos, "Local Suppression and Splitting Techniques for Privacy Preserving Publication of Trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 7, pp. 1466–1479, 2017.

[60] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, "A Dual Privacy Preserving Scheme in Continuous Location-Based Services," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.

[61] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Protecting Trajectory From Semantic Attack Consider-

ing {k} $-Anonymity$, {l}-Diversity, and {t}-Closeness," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 264–278, 2018.

[62] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, and J. Li, "Privacy preservation in location-based services: A novel metric and attack model," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 3006–3019, 2020.

[63] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2019.

[64] A. Akhunzada and M. K. Khan, "Toward Secure Software Defined Vehicular Networks: Taxonomy, Requirements, and Open Issues," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 110–118, 2017.

[65] J. Wang and J. Liu, "Secure and Reliable Slicing in 5G and Beyond Vehicular Networks," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 126–133, 2022.

[66] A. Dutta and E. Hammad, "5G Security Challenges and Opportunities: A System Approach," in *2020 IEEE 3rd 5G World Forum (5GWF)*. IEEE, 2020, pp. 109–114.

[67] R. Soua, I. Turcanu, F. Adamsky, D. Führer, and T. Engel, "Multi-Access Edge Computing for Vehicular Networks: A Position Paper," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–6.

[68] S. Gupta, B. L. Parne, and N. S. Chaudhari, "Security Vulnerabilities in Handover Authentication Mechanism of 5G Network," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, 2018, pp. 369–374.

[69] C. Lai, R. Lu, D. Zheng, and X. Shen, "Security and Privacy Challenges in 5G-enabled Vehicular Networks," *IEEE Network*, vol. 34, no. 2, pp. 37–45, 2020.

[70] ETSI TS 103 601, "Intelligent Transport Systems (ITS); Security; Security management messages communication requirements and distribution protocols ," Oct 2020.

[71] "IBM Machine Learning Definition," https://www.ibm.com/cloud/learn/machine-learning, accessed: 2021-09-30.

[72] D. LI, C. GU, and Y. ZHU, "Gene Fingerprinting: Cracking Encrypted Tunnel with Zero-Shot Learning," *IEICE TRANSACTIONS on Information and Systems*, vol. 105, no. 6, pp. 1172–1184, 2022.

[73] 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.

[74] "VeReMi Data Set," https://github.com/VeReMi-dataset/VeReMi/releases, accessed: 2021-10-06.

[75] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "VeReMi extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[76] "VeReMi Extended Data Set," https://github.com/josephkamel/VeReMi-Dataset, accessed: 2021-10-11.

[77] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, and P. Urien, "Simulation Framework for Misbehavior Detection in Vehicular Networks," *IEEE transactions on vehicular technology*, vol. 69, no. 6, pp. 6631–6643, 2020.

[78] "Framework For Misbehavior Detection (F2MD)," https://github.com/josephkamel/F2MD, accessed: 2021-10-11.

[79] "DARPA Intrusion Detection Evaluation Dataset 1999," https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset, accessed: 2021-10-11.

[80] "DARPA Intrusion Detection Evaluation Dataset 2000," https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets, accessed: 2021-10-11.

[81] "The CAIDA UCSD "DDoS Attack 2007" Dataset," https://www.caida.org/catalog/datasets/ddos-20070804_dataset/, accessed: 2021-10-11.

[82] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184–208, 2016.

[83] "The AWID2 Dataset," https://icsdweb.aegean.gr/awid/awid2, accessed: 2022-01-11.

[84] "KDD Cup 1999," https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, accessed: 2021-10-11.

[85] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE, 2009, pp. 1–6.

[86] "NSL-KDD dataset," https://www.unb.ca/cic/datasets/nsl.html, accessed: 2021-10-11.

[87] J. Song, H. Takakura, and Y. Okabe, "Description of kyoto university benchmark data," *Available at link: http://www. takakura. com/Kyoto_data/BenchmarkData-Description-v5. pdf [Accessed on 15 March 2016]*, 2006.

[88] "Traffic Data from Kyoto University's Honeypots," http://www.takakura.com/Kyoto_data/, accessed: 2021-10-11.

[89] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[90] "The UNSW-NB15 Dataset," http://www.takakura.com/Kyoto_data/, accessed: 2021-10-11.

[91] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[92] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, 2018.

[93] "Intrusion Detection Evaluation Dataset (CIC-IDS2017)," https://www.unb.ca/cic/datasets/ids-2017.html, accessed: 2021-10-11.

[94] A.-K. Pietilainen and C. Diot, "CRAWDAD dataset thlab/sigcomm2009 (v. 2012-07-15)," Downloaded from https://crawdad.org/thlab/sigcomm2009/

20120715, Jul. 2012.

[95] "NGSIMtrajectory datasets," https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm, accessed: 2022-01-19.

[96] "CTUns network simulator," http://nsl.cs.nctu.edu.tw/NSL/nctuns.html, accessed: 2022-01-19.

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

[98] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM computer communication review*, vol. 38, no. 2, pp. 69–74, 2008.

[99] "RDS1000 simulator," https://www.faac.com/realtime-technologies/products/rds-1000-single-seat-simulator/, accessed: 2022-02-16.

[100] S. So, P. Sharma, and J. Petit, "Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 564–571.

[101] A. Le and C. Maple, "Shadows Don't Lie: n-Sequence Trajectory Inspection for Misbehaviour Detection and Classification in VANETs," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–6.

[102] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, and S. Nandi, "Machine Learning Based Approach to Detect Position Falsification Attack in VANET," in *International Conference on Security & Privacy*. Springer, 2019, pp. 166–178.

[103] P. Sharma and H. Liu, "A Machine-Learning-Based Data-Centric Misbehavior Detection Model for Internet of Vehicles," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4991–4999, 2020.

[104] D. Kosmanos, A. Pappas, L. Maglaras, S. Moschoyiannis, F. J. Aparicio-Navarro, A. Argyriou, and H. Janicke, "A novel intrusion detection system against spoofing attacks in connected electric vehicles," *Array*, vol. 5, p. 100013, 2020.

[105] J. Montenegro, C. Iza, and M. Aguilar Igartua, "Detection of Position Falsification Attacks in VANETs Applying Trust Model and Machine Learning," in *Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, 2020, pp. 9–16.

[106] S. Ercan, M. Ayaida, and N. Messai, "New Features for Position Falsification Detection in VANETs using Machine Learning," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[107] ——, "Misbehavior Detection for Position Falsification Attacks in VANETs using Machine Learning," *IEEE Access*, 2021.

[108] 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.

[109] T. Okamura, K. Sato *et al.*, "Misbehavior Detection Method by Time Series Change of Vehicle Position in Vehicle-to-Everything Communication," *Journal of Transportation Technologies*, vol. 11, no. 02, p. 284, 2021.

[110] H. Grover, T. Alladi, V. Chamola, D. Singh, and K.-K. R. Choo, "Edge Computing and Deep Learning Enabled Secure Multi-Tier Network for Internet of Vehicles," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[111] R. Sedar, C. Kalalas, F. Vázquez-Gallego, and J. Alonso-Zarate, "Reinforcement Learning-based Misbehaviour Detection in V2X Scenarios," Sep 2021.

[112] A. Uprety, D. B. Rawat, and J. Li, "Privacy Preserving Misbehavior Detection in IoV using Federated Machine Learning," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021, pp. 1–6.

[113] A. Sharma and A. Jaekel, "Machine Learning Approach for Detecting Location Spoofing in VANET," in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–6.

[114] ——, "Machine Learning Based Misbehaviour Detection in VANET Using Consecutive BSM Approach," *IEEE Open Journal of Vehicular Technology*, 2021.

[115] H. Mankodiya, M. S. Obaidat, R. Gupta, and S. Tanwar, "XAI-AV: Explainable Artificial Intelligence for Trust Management in Autonomous Vehicles," in *2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*. IEEE, 2021, pp. 1–5.

[116] H. Aliev and H. Kim, "Misbehavior Detection based on Multi-Head Deep Learning for V2X Network Security," in *2021 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2021, pp. 1–2.

[117] F. A. Ghaleb, A. Zainal, M. A. Rassam, and F. Mohammed, "An effective misbehavior detection model using artificial neural network for vehicular ad hoc network applications," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*. IEEE, 2017, pp. 13–18.

[118] J.-P. Monteuuis, J. Petit, J. Zhang, H. Labiod, S. Mafrica, and A. Servel, ""my autonomous car is an elephant": A Machine Learning based Detector for Implausible Dimension," in *2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC)*. IEEE, 2018, pp. 1–8.

[119] P. K. Singh, M. K. Dash, P. Mittal, S. K. Nandi, and S. Nandi, "Misbehavior detection in c-its using deep learning approach," in *International Conference on Intelligent Systems Design and Applications*. Springer, 2018, pp. 641–652.

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

[121] S. Gyawali, Y. Qian, and R. Q. Hu, "Machine Learning and Reputation based Misbehavior Detection in Vehicular Communication Networks," *IEEE Transactions on Vehicular Technology*, 2020.

[122] N. Negi, O. Jelassi, H. Chaouchi, and S. Clemençon,

"Distributed online Data Anomaly Detection for connected vehicles," in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2020, pp. 494–500.

[123] S. A. Almalki and F. T. Sheldon, "Deep Learning to Improve False Data Injection Attack Detection in Cooperative Intelligent Transportation Systems," in *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE, 2021, pp. 1016–1021.

[124] B. Ko and S. H. Son, "An Approach to Detecting Malicious Information Attacks for Platoon Safety," *IEEE Access*, vol. 9, pp. 101 289–101 299, 2021.

[125] S. Boddupalli and S. Ray, "REDEM: Real-Time Detection and Mitigation of Communication Attacks in Connected Autonomous Vehicle Applications," *IFIP advances in information and communication technology*, vol. 574, 2019.

[126] S.-L. Wang, S.-Y. Wu, C.-C. Lin, S. Boddupalli, P.-J. Chang, C.-W. Lin, C.-S. Shih, and S. Ray, "Deep-Learning-Based Intrusion Detection for Autonomous Vehicle-Following Systems," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 865–872.

[127] S. Boddupalli, A. Hegde, and S. Ray, "Replace: Real-time Security Assurance in Vehicular Platoons Against V2V Attacks," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1179–1185.

[128] A. Sarker and H. Shen, "A Data-Driven Misbehavior Detection System for Connected Autonomous Vehicles," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, pp. 1–21, 2018.

[129] A. A. Ayoob, G. Su, and G. Al, "Hierarchical growing neural gas network (HGNG)-based semicooperative feature classifier for IDS in vehicular Ad Hoc network (VANET)," *Journal of Sensor and Actuator Networks*, vol. 7, no. 3, p. 41, 2018.

[130] P. Gu, R. Khatoun, Y. Begriche, and A. Serhrouchni, "Support Vector Machine (SVM) Based Sybil Attack Detection in Vehicular Networks," in *2017 IEEE Wireless communications and networking conference (WCNC)*. IEEE, 2017, pp. 1–6.

[131] ——, "k-Nearest Neighbours classification based Sybil attack detection in Vehicular networks," in *2017 Third International Conference on Mobile and Secure Services (MobiSecServ)*. IEEE, 2017, pp. 1–6.

[132] J. Kamel, F. Haidar, I. B. Jemaa, A. Kaiser, B. Lonc, and P. Urien, "A Misbehavior Authority System for Sybil Attack Detection in C-ITS," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 1117–1123.

[133] C. H. Quevedo, A. M. Quevedo, G. A. Campos, R. L. Gomes, J. Celestino, and A. Serhrouchni, "An Intelligent Mechanism for Sybil Attacks Detection in VANETs," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[134] A. Boualouache and T. Engel, "Federated Learning-based Scheme for Detecting Passive Mobile Attackers in 5G Vehicular Edge Computing," *Annals of Telecommunications*, pp. 1–20, 2021.

[135] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, "Machine Learning Approach for Multiple Misbehavior Detection in VANET," in *International conference on advances in computing and communications*. Springer, 2011, pp. 644–653.

[136] J. Grover, V. Laxmi, and M. S. Gaur, "Misbehavior detection based on ensemble learning in vanet," in *International Conference on Advanced Computing, Networking and Security*. Springer, 2011, pp. 602–611.

[137] W. Li, A. Joshi, and T. Finin, "SVM-CASE: An SVM-Based Context Aware Security Framework for Vehicular Ad-Hoc Networks," in *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. IEEE, 2015, pp. 1–5.

[138] C. Zhang, K. Chen, X. Zeng, and X. Xue, "Misbehavior detection based on support vector machine and Dempster-Shafer theory of evidence in VANETs," *IEEE Access*, vol. 6, pp. 59 860–59 870, 2018.

[139] E. Eziama, K. Tepe, A. Balador, K. S. Nwizege, and L. M. Jaimes, "Malicious Node Detection in Vehicular Ad-Hoc Network Using Machine Learning and Deep learning," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–6.

[140] I. Mahmoudi, J. Kamel, I. Ben-Jemaa, A. Kaiser, and P. Urien, "Towards a Reliable Machine Learning Based Global Misbehavior Detection in C-ITS: Model Evaluation Approach," in *International Workshop on Vehicular Adhoc Networks for Smart Cities (IWVSC'2019)*, 2019.

[141] J. Kamel, I. B. Jemaa, A. Kaiser, L. Cantat, and P. Urien, "Misbehavior Detection in C-ITS: A comparative approach of local detection mechanisms," in *2019 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2019, pp. 1–8.

[142] T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "Securing the Internet of Vehicles: A Deep Learning-Based Classification Framework," *IEEE Networking Letters*, vol. 3, no. 2, pp. 94–97, 2021.

[143] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "Artificial Intelligence (AI)-Empowered Intrusion Detection Architecture for the Internet of Vehicles," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 144–149, 2021.

[144] T. Alladi, A. Agrawal, B. Gera, V. Chamola, B. Sikdar, and M. Guizani, "Deep Neural Networks for Securing IoT Enabled Vehicular Ad-Hoc Networks," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[145] N. C. Kushardianto, Y. El Hillali, and C. Tatkeu, "2-Step Prediction for Detecting Attacker in Vehicle to Vehicle Communication," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–5.

[146] F. Gonçalves, J. Macedo, and A. Santos, "Intelligent Hierarchical Intrusion Detection System for VANETs," in *2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2021, pp. 50–59.

[147] ——, "An Intelligent Hierarchical Security Framework for VANETs," *Information*, vol. 12, no. 11, p. 455,

2021.

[148] F. Gonçalves, B. Ribeiro, O. Gama, J. Santos, A. Costa, B. Dias, M. J. Nicolau, J. Macedo, and A. Santos, "Synthesizing Datasets with Security Threats for Vehicular Ad-Hoc Networks," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[149] T. Alladi, B. Gera, A. Agrawal, V. Chamola, and F. R. Yu, "DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12 013–12 023, 2021.

[150] H.-Y. Hsu, N.-H. Cheng, and C.-W. Tsai, "A Deep Learning-Based Integrated Algorithm for Misbehavior Detection System in VANETs," in *Proceedings of the 2021 ACM International Conference on Intelligent Computing and its Emerging Applications*, 2021, pp. 53–58.

[151] S. Boddupalli, A. S. Rao, and S. Ray, "Resilient Cooperative Adaptive Cruise Control for Autonomous Vehicles Using Machine Learning," *arXiv preprint arXiv:2103.10533*, 2021.

[152] Y. Liu, H. Xue, W. Zhuang, F. Wang, L. Xu, and G. Yin, "CT2-MDS: Cooperative trust-aware tolerant misbehaviour detection system for connected and automated vehicles," *IET Intelligent Transport Systems*, vol. 16, no. 2, pp. 218–231, 2022.

[153] H. Sedjelmaci and S. M. Senouci, "An accurate and efficient collaborative intrusion detection framework to secure vehicular networks," *Computers & Electrical Engineering*, vol. 43, pp. 33–47, 2015.

[154] Y. Zeng, M. Qiu, D. Zhu, Z. Xue, J. Xiong, and M. Liu, "Deepvcm: A deep learning based intrusion detection method in vanet," in *2019 IEEE 5th intl conference on big data security on cloud (BigDataSecurity), IEEE intl conference on high performance and smart computing,(HPSC) and IEEE intl conference on intelligent data and security (IDS)*. IEEE, 2019, pp. 288–293.

[155] H. Tan, Z. Gui, and I. Chung, "A Secure and Efficient Certificateless Authentication Scheme with Unsupervised Anomaly Detection in VANETs," *IEEE Access*, vol. 6, pp. 74 260–74 276, 2018.

[156] H. Izakian, W. Pedrycz, and I. Jamal, "Fuzzy clustering of time series data using dynamic time warping distance," *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 235–244, 2015.

[157] P. K. Singh, S. K. Jha, S. K. Nandi, and S. Nandi, "ML-Based Approach to Detect DDoS Attack in V2I Communication Under SDN Architecture," in *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018, pp. 0144–0149.

[158] Y. Yu, L. Guo, Y. Liu, J. Zheng, and Y. Zong, "An Efficient SDN-Based DDoS Attack Detection and Rapid Response Platform in Vehicular Networks," *IEEE access*, vol. 6, pp. 44 570–44 579, 2018.

[159] K. Sharshembiev, S.-M. Yoo, and E. Elmahdi, "Protocol misbehavior detection framework using machine learning classification in vehicular Ad Hoc networks," *Wireless Networks*, vol. 27, no. 3, pp. 2103–2118, 2021.

[160] A. R. Narayanadoss, T. Truong-Huu, P. M. Mohan, and M. Gurusamy, "Crossfire Attack Detection Using Deep Learning in Software Defined ITS Networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–6.

[161] L. A. Maglaras, "A Novel Distributed Intrusion Detection System for Vehicular Ad Hoc Networks," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 4, pp. 101–106, 2015.

[162] D. Tian, Y. Wang, G. Lu, and G. Yu, "A vehicular ad hoc networks intrusion detection system based on BUSNet," in *2010 2nd International Conference on Future Computer and Communication*, vol. 1. IEEE, 2010, pp. V1–225.

[163] L. Nie, Y. Li, and X. Kong, "Spatio-Temporal Network Traffic Estimation and Anomaly Detection Based on Convolutional Neural Network in Vehicular Ad-hoc Networks," *IEEE Access*, vol. 6, pp. 40 168–40 176, 2018.

[164] A. Gruebler, K. D. McDonald-Maier, and K. M. A. Alheeti, "An Intrusion Detection System against Black Hole Attacks on the Communication Network of Self-Driving cars," in *2015 sixth international conference on emerging security technologies (EST)*. IEEE, 2015, pp. 86–91.

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

[166] K. M. Ali Alheeti, A. Gruebler, and K. McDonald-Maier, "Intelligent Intrusion Detection of Grey Hole and Rushing Attacks in Self-Driving Vehicular Networks," *Computers*, vol. 5, no. 3, p. 16, 2016.

[167] Y. Zeng, M. Qiu, Z. Ming, and M. Liu, "Senior2local: A Machine Learning Based Intrusion Detection Method for VANETs," in *International conference on smart computing and communication*. Springer, 2018, pp. 417–426.

[168] S. A. Siddiqui, A. Mahmood, W. E. Zhang, and Q. Z. Sheng, "Machine Learning Based Trust Model for Misbehaviour Detection in Internet-of-Vehicles," in *International Conference on Neural Information Processing*. Springer, 2019, pp. 512–520.

[169] A. Acharya and J. Oluoch, "A Dual Approach for Preventing Blackhole Attacks in Vehicular Ad Hoc Networks Using Statistical Techniques and Supervised Machine Learning," in *2021 IEEE International Conference on Electro Information Technology (EIT)*. IEEE, 2021, pp. 230–235.

[170] O. A. Wahab, A. Mourad, H. Otrok, and J. Bentahar, "CEAP: SVM-based intelligent detection model for clustered vehicular ad hoc networks," *Expert Systems with Applications*, vol. 50, pp. 40–54, 2016.

[171] E. A. Shams, A. Rizaner, and A. H. Ulusoy, "Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks," *Computers & Security*, vol. 78, pp. 245–254, 2018.

[172] P. K. Singh, R. R. Gupta, S. K. Nandi, and S. Nandi, "Machine Learning Based Approach to Detect Wormhole Attack in VANETs," in *Workshops of the international conference on advanced information networking and applications*. Springer, 2019, pp. 651–661.

[173] K. M. A. Alheeti and K. McDonald-Maier, "Hybrid in-

trusion detection in connected self-driving vehicles," in *2016 22nd International Conference on Automation and Computing (ICAC)*. IEEE, 2016, pp. 456–461.

[174] K. M. Ali Alheeti and K. McDonald-Maier, "Intelligent intrusion detection in external communication systems for autonomous vehicles," *Systems Science & Control Engineering*, vol. 6, no. 1, pp. 48–56, 2018.

[175] M. Kim, I. Jang, S. Choo, J. Koo, and S. Pack, "Collaborative security attack detection in software-defined vehicular networks," in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2017, pp. 19–24.

[176] T. Zhang and Q. Zhu, "Distributed Privacy-preserving Collaborative Intrusion Detection Systems for VANETs," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 148–161, 2018.

[177] ——, "Differentially Private Collaborative Intrusion Detection Systems For VANETs," *arXiv preprint arXiv:2005.00703*, 2020.

[178] F. A Ghaleb, F. Saeed, M. Al-Sarem, B. Ali Saleh Al-rimy, W. Boulila, A. Eljialy, K. Aloufi, and M. Alazab, "Misbehavior-Aware on-Demand Collaborative Intrusion Detection System Using Distributed Ensemble Learning for VANET," *Electronics*, vol. 9, no. 9, p. 1411, 2020.

[179] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled lstm autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[180] J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative Intrusion Detection for VANETs: a Deep Learning-based Distributed SDN Approach," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[181] X. Li, Z. Hu, M. Xu, Y. Wang, and J. Ma, "Transfer Learning based Intrusion Detection Scheme for Internet of Vehicles," *Information Sciences*, 2020.

[182] H. Bangui, M. Ge, and B. Buhnova, "A Hybrid Data-driven Model for Intrusion Detection in VANET," *Procedia Computer Science*, vol. 184, pp. 516–523, 2021.

[183] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A Multi-Tiered Hybrid Intrusion Detection System for Internet of Vehicles," *IEEE Internet of Things Journal*, 2021.

[184] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An Enhanced Multi-Stage Deep Learning Framework for Detecting Malicious Activities From Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[185] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing," *IEEE Transactions on Vehicular Technology*, 2021.

[186] R. Rahal, A. Amara Korba, N. Ghoualmi-Zine, Y. Challal, and M. Y. Ghamri-Doudane, "AntibotV: A Multi-level Behaviour-Based Framework for Botnets Detection in Vehicular Networks," *Journal of Network and Systems Management*, vol. 30, no. 1, pp. 1–40, 2022.

[187] X. Liu, G. Yan, D. B. Rawat, and S. Deng, "Data Mining Intrusion Detection in Vehicular Ad Hoc Network," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 7, pp. 1719–1726, 2014.

[188] L. Yang and A. Shami, "A Transfer Learning and Optimized CNN Based Intrusion Detection System for Internet of Vehicles," *arXiv preprint arXiv:2201.11812*, 2022.

[189] D. Karagiannis and A. Argyriou, "Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning," *Vehicular Communications*, vol. 13, pp. 56–63, 2018.

[190] N. Lyamin, D. Kleyko, Q. Delooz, and A. Vinel, "AI-Based Malicious Network Traffic Detection in VANETs," *IEEE Network*, vol. 32, no. 6, pp. 15–21, 2018.

[191] N. Lyamin, A. Vinel, M. Jonsson, and J. Loo, "Real-Time Detection of Denial-of-Service Attacks in IEEE 802.11p Vehicular Networks," *IEEE Communications letters*, vol. 18, no. 1, pp. 110–113, 2013.

[192] N. V. Abhishek and M. Gurusamy, "JaDe: Low Power Jamming Detection using Machine Learning in Vehicular Networks," *IEEE Wireless Communications Letters*, 2021.

[193] D. Kosmanos, A. Pappas, F. J. Aparicio-Navarro, L. Maglaras, H. Janicke, E. Boiten, and A. Argyriou, "Intrusion Detection System for Platooning Connected Autonomous Vehicles," in *2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*. IEEE, 2019, pp. 1–9.

[194] Z. Du, C. Wu, T. Yoshinaga, K.-L. A. Yau, Y. Ji, and J. Li, "Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 45–61, 2020.

[195] N. Aljeri and A. Boukerche, "Mobility management in 5g-enabled vehicular networks: models, protocols, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–35, 2020.

[196] M. R. Palattella, R. Soua, A. Khelil, and T. Engel, "Fog computing as the key for seamless connectivity handover in future vehicular networks," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 1996–2000.

[197] E. Gures, I. Shayea, A. Alhammadi, M. Ergen, and H. Mohamad, "A comprehensive survey on mobility management in 5g heterogeneous networks: Architectures, challenges and solutions," *IEEE Access*, vol. 8, pp. 195 883–195 913, 2020.

[198] D. Zhao, Z. Yan, M. Wang, P. Zhang, and B. Song, "Is 5g handover secure and private? a survey," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 855–12 879, 2021.

[199] M. J. Alam and M. Ma, "Dc and comp authentication in lte-advanced 5g hetnet," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[200] X. Duan and X. Wang, "Authentication handover and privacy protection in 5g hetnets using software-defined networking," *IEEE Communications Magazine*,

vol. 53, no. 4, pp. 28–35, 2015.

[201] I. H. Abdulqadder and S. Zhou, "Sliceblock: Context-aware authentication handover and secure network slicing using dag-blockchain in edge-assisted sdn/nfv-6g environment," *IEEE Internet of Things Journal*, 2022.

[202] L. Zhang, C. An, Q. Zhang, and C. Tang, "Misbehavior Detection Algorithm in CCSDS Space Telecommand System," *IEEE communications letters*, vol. 14, no. 8, pp. 746–748, 2010.

[203] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "Privanet: An efficient pseudonym changing and management framework for vehicular ad-hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3209–3218, 2019.

[204] O. A. Fernando, H. Xiao, and J. Spring, "Developing a Testbed with P4 to Generate Datasets for the Analysis of 5G-MEC Security," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2022, pp. 2256–2261.

[205] Y.-E. Kim, Y.-S. Kim, and H. Kim, "Effective Feature Selection Methods to Detect IoT DDoS Attack in 5G Core Network," *Sensors*, vol. 22, no. 10, p. 3819, 2022.

[206] H. Mun, M. Seo, and D. H. Lee, "Secure Privacy-Preserving V2V Communication in 5G-V2X Supporting Network Slicing," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[207] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[208] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[209] H. Sedjelmaci, S. M. Senouci, N. Ansari, and A. Boualouache, "A Trusted Hybrid Learning Approach to Secure Edge Computing," *IEEE Consumer Electronics Magazine*, 2021.

[210] C. Benzaid and T. Taleb, "AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions," *IEEE Network*, vol. 34, no. 2, pp. 186–194, 2020.

[211] A. Boualouache, R. Soua, and T. Engel, "SDN-based Misbehavior Detection System for Vehicular Networks," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[212] M. Ambrosin, L. L. Yang, X. Liu, M. R. Sastry, and I. J. Alvarez, "Design of a Misbehavior Detection System for Objects Based Shared Perception V2X Applications," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1165–1172.

[213] M. Tsukada, S. Arii, H. Ochiai, and H. Esaki, "Misbehavior Detection Using Collective Perception under Privacy Considerations," *arXiv preprint arXiv:2111.03461*, 2021.

[214] C. Buerkle, F. Geissler, M. Paulitsch, and K.-U. Scholl, "Fault-Tolerant Perception for Automated Driving A Lightweight Monitoring Approach," *arXiv preprint arXiv:2111.12360*, 2021.

[215] X. Liu, L. Yang, I. Alvarez, K. Sivanesan, A. Merwaday, F. Oboril, C. Buerkle, M. Sastry, and L. G. Baltar, "MISO-V: Misbehavior Detection for Collective Perception Services in Vehicular Communications," in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2021, pp. 369–376.

[216] F. Geissler, A. Unnervik, and M. Paulitsch, "A Plausibility-based Fault Detection Method for High-level Fusion Perception Systems," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 1, pp. 176–186, 2020.

[217] P. Sharma, D. Austin, and H. Liu, "Attacks on Machine Learning: Adversarial Examples in Connected and Autonomous Vehicles," in *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*. IEEE, 2019, pp. 1–7.

[218] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and the Way Forward," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 998–1026, 2020.

[219] L. Lyu, H. Yu, J. Zhao, and Q. Yang, "Threats to Federated Learning," in *Federated Learning*. Springer, 2020, pp. 3–16.

[220] A. Talpur and M. Gurusamy, "Adversarial Attacks Against Deep Reinforcement Learning Framework in Internet of Vehicles," *arXiv preprint arXiv:2108.00833*, 2021.

[221] R. Xing, Z. Su, N. Zhang, Y. Peng, H. Pu, and J. Luo, "Trust-Evaluation-Based Intrusion Detection and Reinforcement Learning in Autonomous Driving," *IEEE Network*, vol. 33, no. 5, pp. 54–60, 2019.

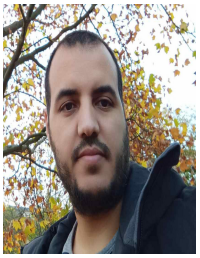
[222] ETSI TR 102 893, "Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)," Mar 2017.

[223] "3rd C-V2X PLUGTESTS," https://www.etsi.org/events/1759-cv2x-plugtests-3, accessed: 2022-01-17.

[224] ETSI TS 103 525-1, "Intelligent Transport Systems (ITS); Testing; Conformance test specifications for ITS PKI management; Part 1: Protocol Implementation Conformance Statement (PICS)," Jan 2022.

[225] ETSI TS 103 525-2, "Intelligent Transport Systems (ITS); Testing; Conformance test specifications for ITS PKI management; Part 2: Test Suite Structure and Test Purposes (TSS & TP) ," Jan 2022.

[226] ETSI TS 103 525-3, "Intelligent Transport Systems (ITS); Testing; Conformance test specifications for ITS PKI management; Part 3: Abstract Test Suite (ATS) and Protocol Implementation eXtra Information for Testing (PIXIT)," Jan 2022.



Abdelwahab Boualouache (M'15) is a research associate at the FSTM - Faculty of Science, Technology, and Medicine, University of Luxembourg. He received a doctorate of science in Mobile Computing from the University of Sciences and Technology Houari Boumediene (USTHB), Algeria, in 2016. His current research interests include applying advanced Machine Learning and Blockchain for security and privacy in 5G and beyond networks, focusing on the Connected and Automated Vehicles vertical.

Thomas Engel is a Professor of Computer Networks at the University of Luxembourg. He received the title Dr. rer. nat from the University of Saarbruecken, Germany, in 1996. Since 2002, he has taught and conducted research as a professor at the IST/University of Luxembourg. He is the head of the SECAN-Lab research group conducting internationally competitive fundamental and applied research in computer networking, privacy, and security, namely in the areas of privacy-by-design network and system security, SCADA and cyber security, IoT, vehicular communication and multimodal traffic management, and wireless networks and mobile security.