Vehicular Communications 39 (2023) 100569

Contents lists available at ScienceDirect

Vehicular Communications

journal homepage: www.elsevier.com/locate/vehcom

Artificial Intelligence and Machine Learning as key enablers for V2X communications: A comprehensive survey

Maria Christopoulou [a,b,*], Sokratis Barmpounakis [c], Harilaos Koumaras [b], Alexandros Kaloxylos [a]

[a] Dept. of Informatics and Telecommunications, University of Peloponnese, Tripoli, Greece
[b] National Centre for Scientific Research "Demokritos", Agia Paraskevi, Greece
[c] Dept. of Informatics and Telecommunications, National and Kapodistrian University of Athens, Athens, Greece

ARTICLE INFO

Article history:
Received 5 September 2022
Received in revised form 18 November 2022
Accepted 20 December 2022
Available online 28 December 2022

Keywords:
Artificial Intelligence
Machine Learning
V2X
Deep Learning
Vehicular networks

ABSTRACT

The automotive industry is undergoing a profound digital transformation to create autonomous vehicles. Vehicle-to-Everything (V2X) communications enable the provisioning of transportation use cases for road traffic and safety management. At the same time, during the past decade, Artificial Intelligence (AI) and Machine Learning (ML) have been in the spotlight because of their outstanding performance in various domains, including natural language processing, and computer vision. Considering also current standardization efforts, towards incorporating AI and ML as integral sub-systems of beyond 5G and 6G networks, these technologies are considered very promising to optimize user, control, and management network functions, but also to support road safety and even entertainment applications. This survey systematically reviews existing research at the intersection of AI/ML and V2X communications, focusing on handover management, proactive caching, physical and computation resources allocation, beam selection optimization, packet routing, and QoS prediction in vehicular environments. We extract the underlying AI/ML techniques, the training features, their architecture and discuss several aspects regarding the intricacies of vehicular environments and ML. These aspects include time complexity of the algorithms, quality of real-world vehicle traces, suitability of AI/ML techniques in relevance to the designated network operation and the underlying automotive use case, as well as velocity and positioning accuracy requirements towards the creation of more realistic and representative synthetic data.

1. Introduction

During the past years, car manufacturers have introduced driver assistance systems to their models, coupled with onboard intelligence, leading to a higher perception of their surroundings. This enables the possibility to achieve different levels of autonomous driving. Autonomous driving is considered critical in improving car safety, eliminating accidents due to human error, reducing traffic congestion, and improving passenger comfort. The Society of Automotive Engineers (SAE) has defined six driving automation levels, ranging from no automation to full automation [1]. Communications among vehicles, infrastructure and road users, collectively defined as Vehicle-to-Everything (V2X), are essential in realizing safety and non-safety-related applications, such as autonomous driving, car platooning, information sharing among vehicles and high data-rate infotainment.

V2X may refer to Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N) or Vehicle-to-Pedestrian (V2P) communication. V2P refers to the communication among vehicles and pedestrians, cyclists or motorized two-wheeler operators, collectively called Vulnerable Road Users (VRUs) [2]. There are two leading technologies in V2X: *i*) the Cellular-V2X (C-V2X) based on cellular 4G/LTE [3] and 5G networks [4] and *ii*) the Dedicated Short-Range Communication (DSRC) [5] based on IEEE 802.11p [6]. These solutions will complement the sensor/camera/radar information and intelligently connect the car to its surroundings and the network.

From a network perspective, connected vehicles present challenges regarding network management, performance, and efficiency. Vehicular networks exhibit strong dynamics in traffic patterns, network topologies, and propagation channels. In addition, the rising popularity of mobile applications, such as in-vehicle in-

* Corresponding author.
E-mail addresses: m.christopoulou@go.uop.gr (M. Christopoulou), sokbar@di.uoa.gr (S. Barmpounakis), koumaras@iit.demokritos.gr (H. Koumaras), kaloxyl@uop.gr (A. Kaloxylos).

https://doi.org/10.1016/j.vehcom.2022.100569
2214-2096/

Summary of abbreviations

3GPP	3rd Generation Partnership Program
5G NR	5G New Radio
5GAA	5G Automotive Association
AI	Artificial Intelligence
ANN	Artificial Neural Network
AoA	Angle of Arrival
AoD	Angle of direction
AP	Access Point
ARIB	Association of Radio Industries and Businesses
BLER	Block Error Rate
BS	Base Station
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CQI	Channel Quality Indicator
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
C-V2X	Cellular V2X
DDPG	Deep Deterministic Policy Gradient
DL	Deep Learning
DNN	Deep Neural Network
DQN	Deep Q Network
DRL	Deep Reinforcement Learning
DSRC	Dedicated Short-Range Communication
E2E	End to End
ENI	Experiential Networked Intelligence
ETC	Electronic Toll Collection
ETSI	European Telecommunications Standards Institute
FES	Fixed Edge Servers
TCMA	Tiered Contention Multiple Access
TTT	Transport and Traffic Telematic
FL	Federated Learning
FR1	Frequency Range 1
FR2	Frequency Range 2
GHE	Geoffrey E. Havers
GHR	Gazis-Herman-Rothery
MAB	Multi-Armed Bandit
MAC	Medium Access Control
MBS	Master Base Station
MEC	Multi-access Edge Computing
ML	Machine Learning
MLP	Multi-Layer Perceptron
mmWave	millimeter Wave
MNO	Mobile Network Operator
NWDAF	Network Data Analytics Function
OBU	On-board Unit
OECD	Organization for Economic Co-operation and Development
O-RAN	Open Radio Access Network
PF	Prediction Function
PLMN	Public Land Mobile Network
PRR	Packet Reception Rate
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RRUs	Remote Radio Units
mmBS	Millimeter Base Station
ETC	Electronic Toll Collection
RSRP	Reference Signal Received Power
RSSI	Received Signal Strength Indicator
RSU	Roadside Unit
SAE	Society of Automotive Engineers
SAI	Securing Artificial Intelligence
SBS	Secondary Base Station
SDN	Software Defined networking
SDOs	Standardization Organizations
SINR	Signal-to-Interference-plus-Noise Ratio
SL	Sidelink
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular ad hoc networks
VES	Vehicular Edge Servers
VRU	Vulnerable Road Users
VUE	Vehicular User Equipment
WAVE	Wireless Access in Vehicular Environments
WLAN	Wireless Local Area Network
WMMSE	Weighted Mean Squared Error
UE	User Equipment
RL	Reinforcement Learning
VICS	Vehicle Information Communication Systems

fotainment, already introduces unprecedented demands on wireless networking infrastructure. As a result, Mobile Network Operators (MNOs) will need to collect an immense amount of heterogeneous data to monitor network performance and provide better services. Given the current network management techniques, maintaining the network in an operational state, while committing to diverse Service Level Agreements (SLAs), will become an increasingly complex task.

A solution to this issue is deploying more intelligence to the network [7]. For example, Self-Organizing-Networks (SONs) are adaptive and autonomous networks that interact with their environment, deciding on the necessary actions to maintain performance stability and improve their QoS level, based on previous resource management decisions. SON solutions are divided into three main categories: *i*) self-configurable, *ii*) self-optimized, and *iii*) self-healing. The first generation of SONs in 3G/4G networks faced limitations in their performance since they employed reactive techniques, where functions begin the self-organization phase only after a problem has occurred. Hence, these techniques cannot provide the desired level of adaptability to dynamic changes in the environment, preventing the network from complying with the requirements of many advanced use cases foreseen in future wireless networks. In this case, Artificial Intelligence (AI) and Machine Learning (ML) can be investigated as potential solutions for deploying intelligence in SONs, acting proactively in upcoming network issues, but also being able to face the complexity of the state space appropriately (e.g., number of network environment/parameters). This way, networks can reduce the excessive signaling overhead that results from network faults, minimizing recurrent delays in the provided services. AI and ML can play a similar role also for control and user plane functions handling the amount of heterogeneous data and improving overall network performance.

Today, there is no unified consensus on a single definition of AI. It is out of this publication's scope to discuss the reasons for the lack of a standard definition or its implications. For the remainder of this paper, we will adopt the definition of the Organization for Economic Co-operation and Development (OECD) [8] for the reasons outlined in [9]: "*An AI system is a machine-based system that* *can, for a given set of human-defined objectives, make predictions, recommendations, or decisions influencing real or virtual environments. AI systems are designed to operate with varying levels of autonomy*".

1.1. Learning in V2X scenarios

Machine-based systems comprise agents that perceive their environment and adapt to changing circumstances through *learning*, in order to achieve a specific objective. ML is a central subfield of AI that studies how agents can improve their perception based on experience or data. ML algorithms extract underlying patterns among data by approximating a target function that predicts outcomes on previously unseen data. An important aspect of ML models is their ability to generalize their learning from seen training data to unseen examples [10]. In recent years, advances in Graphics Processing Units (GPUs) and the increased availability of data have given rise to Deep Learning (DL). DL is an important ML branch that outperforms conventional ML techniques in terms of accuracy when the amount of data is significant. DL relies on Artificial Neural Networks (ANNs), a popular ML technique inspired by the function of learning in the human nervous system.

Based on the above, machine-based systems have the potential to enable advanced transportation use cases, including collision avoidance, platooning, intersection movement, and emergency brakes. Such transportation use cases entail large amounts of data to train a learning agent according to specific constraints derived from the underlying use case. The agent will then make decisions and adapt them to changing network and road traffic conditions.

To enable such transportation use cases, it is imperative to preserve uninterrupted communication between vehicles, networks, and pedestrians, regardless of network and road traffic conditions. The vehicles transmit and receive awareness messages, meaning that the underlying network control and management procedures must be in place to allow uninterrupted V2X communications. Motivated by the capabilities of AI/ML, this survey examines applications on such procedures, including handovers, beam allocation, caching, radio network resource allocation, computation resources management, packet routing, and QoS prediction. Subsection 1.2 describes related work and the contributions of this survey.

1.2. Related work and survey contributions

In recent years, AI/ML applications in wireless networks and V2X communications have been extensively studied in the existing literature, as depicted in Table 1. Ye et al. [11] review advances in applying ML in vehicular networks for traffic flow prediction, local data storage, network congestion control, load balancing, vertical handover control and wireless resource management. The authors also discuss the challenges when applying ML techniques in vehicular networks, including the strong dynamics of vehicular networks, limited on-board computation resources for efficient model training and heterogeneous data generation across several network points. Liang et al. [12] describe potential applications of AI/ML in learning the dynamics of vehicular networks regarding mobility and traffic patterns, and elaborate on ML techniques that can contribute to optimizing network performance. The authors also present related works on network security, handover, resource management and congestion control. Noor-A-Rahim et al. [13] focus on resource allocation techniques on DSRC, C-V2X and heterogeneous vehicular networks. Although the main scope of the paper is not AI/ML on V2X, the authors provide a comprehensive overview of AI/ML techniques on several resource allocation tasks, such as user association, handover management and virtual resource management for V2V and V2I communications. Tong et al. [14] provide a comprehensive survey of AI/ML applications on transportation use cases, such as platooning, autonomous navigation and safety, and network-related use cases, including content delivery and offloading, edge computing and security in vehicular networks. They also review available software tools for AI and discuss research challenges on this joint research field. Tang et al. [15] discuss the challenges of resource allocation, network traffic control and security in vehicular networks and provide a survey on various ML techniques applied to each field. The authors extend their work in [16], by including cognitive radio, beamforming, routing, OFDM and NOMA tasks. Finally, Mchergui et al. [17] present AI/ML applications in Vehicular Ad Hoc Networks (VANETs) on network routing, resources allocation, security, mobility management and seamless integration of different technologies.

Driven by the recent advances in autonomous vehicles and V2X, our survey aims at providing a comprehensive review of AI/ML applications in V2X communications for a variety of different network operations/procedures, namely: handover/user-cell association, caching, physical and computation management, routing, beam management and QoS prediction. We investigate the AI/ML algorithms and the optimization objectives used in each category. Unlike earlier publications, however, we extend this survey by examining more aspects of AI/ML: *i*) the training architecture, i.e., centralized, distributed, or federated, *ii*) the features used for training the AI/ML models in each type of operation, and *iii*) the training and testing time complexity of popular AI/ML algorithms used in the surveyed papers, extracted from the literature, to provide context on the computational characteristics of AI/ML. We also extract the lessons learned in each task. We focus on network control and management procedures that depend on V2X communications. We do not examine higher-level applications, such as network security, mobility and trajectory prediction, or transportation use cases that do not target network-related optimizations. The survey's contributions are the following:

1) We systematically review existing research at the intersection between AI/ML and V2X communications and provide an extensive list of network-controlled functions optimized by data-driven approaches in vehicular environments. For each publication, we extract the underlying AI/ML technique(s), the training feature types and categories, the training architecture (i.e., centralized, distributed or federated), and we summarize the lessons learned in each category, identifying common trends and suitable solutions depending on the designated task.
2) We discuss the time complexity aspects of ML techniques regarding their suitability in enabling automotive use cases. These aspects include AI/ML model training, inference response time, and techniques for accelerating the model's execution to meet the highly dynamic changes of vehicular environments. We also provide training and testing time complexities of standard AI/ML algorithms (supervised, unsupervised, and reinforcement learning) extracted from related literature.
3) Finally, we discuss the quality aspects of real-world vehicle traces used in simulating vehicular networks, as well as velocity and positioning modeling aspects, concerning different automotive use cases requirements.

The remainder of this survey is organized as follows (Fig. 1). Section 2 briefly overviews V2X access technologies, namely DSRC, LTE, and 5G New Radio (NR). Section 3 provides a brief introduction to AI/ML techniques, time complexity according to the literature, and related standardization activities. Section 4 reviews existing works on different network procedures, such as handover/user-cell association, caching, physical and computation management, routing, beam management, and QoS prediction, and provides the lessons learned based on the surveyed solutions of each task. Sec-

Table 1
Existing surveys addressing applications of AI/ML on vehicular networks.

Ref.	Year	Summary	Handover	Beam Selection	Radio Access Network Allocation	Computation Resources	Routing	QoS Prediction	Network Security	Trajectory Prediction	Road Traffic Flow Prediction	Transporta-tion use cases	AI/ML Training Architecture	Time Complexity
[11]	2018	ML applications in vehicular networks (traffic flow prediction, local data storage, network congestion control, load balancing, vertical handoff control and wireless resource management)	✓		✓	✓					✓			
[12]	2019	ML methods in vehicular applications (learn dynamics of V2X networks and optimize network performance)	✓		✓	✓			✓	✓				
[13]	2019	A survey on resource allocation in DSRC and C-V2X networks that also presents AI/ML-based solutions on handover, radio network allocations and QoS	✓		✓			✓						
[14]	2019	Applications of AI/ML in vehicular use cases (content delivery, offloading, edge computing, security, AI/ML tools)				✓			✓		✓	✓		
[15]	2020	Application of AI/ML in radio network allocation, network traffic control and security in vehicular networks		✓	✓		✓		✓	✓				
[16]	2021	AI/ML on vehicular networks (OFDM, NOMA, cognitive radio, Beamforming, routing, security, mobility and trajectory prediction)		✓	✓	✓	✓		✓	✓	✓			
[17]	2021	AI in VANETs: safety applications, routing, security, mobility management			✓		✓		✓	✓		✓		
	2022	Our survey	✓	✓	✓	✓	✓	✓	Out of this survey's scope				✓	✓

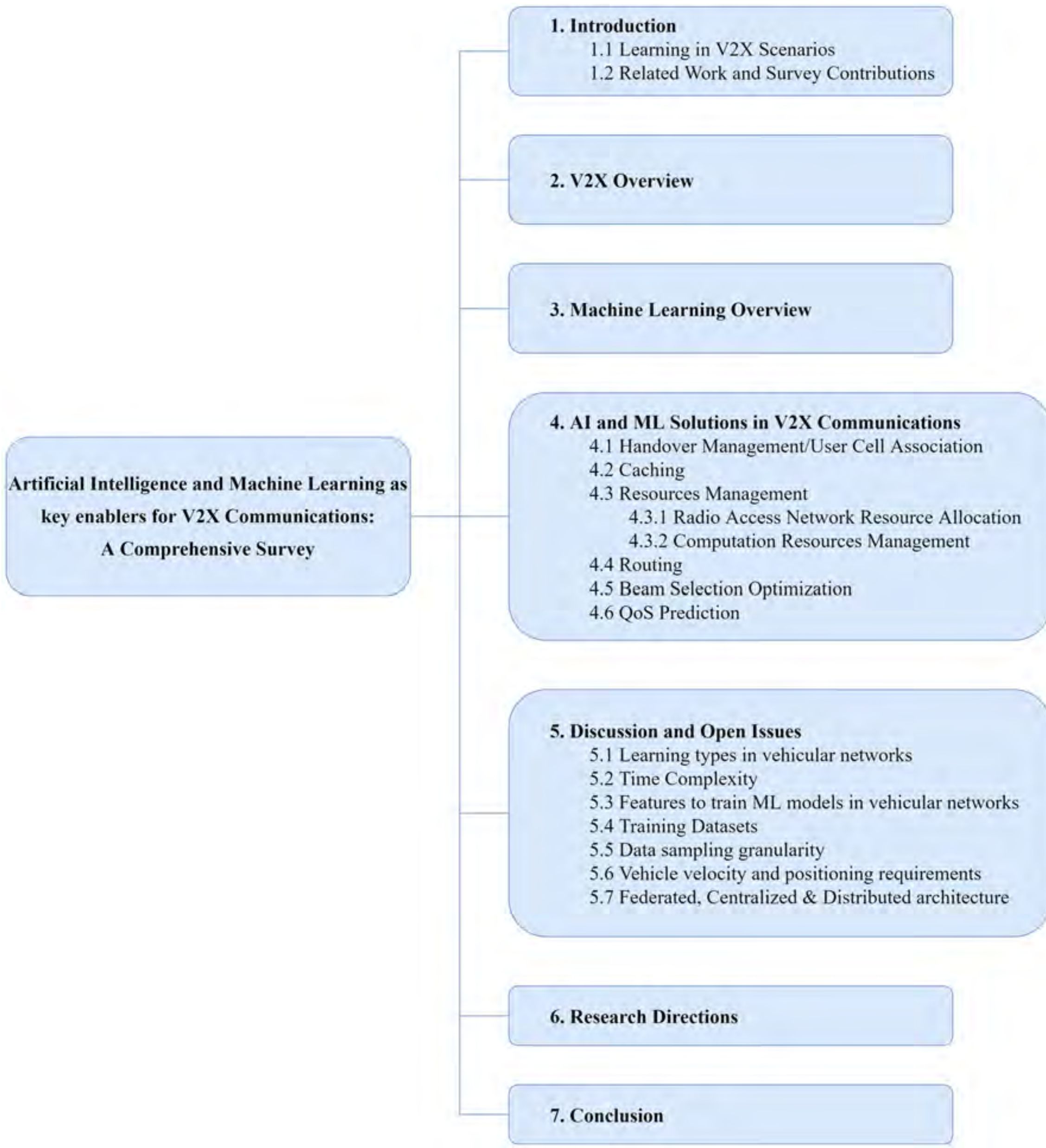

Fig. 1. Organization of the Survey.

tion 5 discusses significant aspects of AI/ML applications in V2X communications, including selecting appropriate AI/ML algorithms, training and response times, training features in vehicular environments, data collection methods, training architecture, dataset quality aspects, as well as vehicle velocity and positioning requirements, based on well-established automotive use cases. Section 6 presents research directions for AI/ML applications in V2X for next-generation networks. Section 7 concludes the survey.

2. V2X overview

This section briefly presents the two most prominent access technologies for V2X communications, namely *i*) IEEE 802.11p and *ii*) C-V2X, comprising LTE and 5G NR. The section serves as a preliminary introduction to V2X communications, so we encourage readers to refer to the provided sources for more details on these technologies.

V2X communications are significant for implementing Intelligent Transport Systems (ITSs), since they enable data exchange between vehicles and other entities in their surroundings. ITSs can be classified into Legacy and Advanced ITSs according to their technical characteristics. Legacy ITSs support only V2I communications and are already used in Electronic Toll Collection (ETC), Vehicle Information Communication Systems (VICS) and Transport and Traffic Telematic (TTT) services. Advanced ITSs provide enhanced capabilities such as improved data rates, larger packet sizes, and broader coverage than their legacy counterpart, enabling more advanced applications through V2I, V2V, V2N, and V2P connectivity.

The main components of ITSs are: *i*) the On-Board Units (OBUs), i.e., the radio units installed on the vehicles, *ii*) the Roadside Units (RSUs), the radio equipment on the roadside providing connectivity between the OBUs, the transportation infrastructure and the backhaul networks, and *iii*) the Roadside Equipment (RSE), referring to the ITS field equipment in general, such as Traffic Signal Controllers (TSC) and RSUs. The ITSs components regularly exchange messages to provide information on basic safety, VRU awareness, signal phase and timing, road/lane topology, traffic maneuver, platooning control, collective perception, and maneuver coordination.

Legacy ITSs rely on the Dedicated Short-Range Communication (DSRC) radio technology permitting data rates up to 4 MB/s [18]. Advanced ITSs use different radio technologies, namely the Wireless Access in Vehicular Environment (WAVE) [5] in the US, ETSI ITS-G5 in Europe, ITS Connect in Japan and Cellular [18]. The first three radiocommunication systems use modified versions of the IEEE 802.11p technology. The WAVE system is also called DSRC in the US, while in Europe, DSRC refers to Legacy ITSs [18].

The WAVE system comprises the IEEE 802.11p [6] and 1609 [19] family of standards. IEEE 802.11p is an amendment to the IEEE 802.11 Wi-Fi specification [20] to accommodate inter-vehicular communications and specifies the Physical (PHY) and Medium Access Control (MAC) layer procedures. IEEE 1609 define upper-layer procedures, such as networking, security, multi-channel operation and also describe the overall system architecture. IEEE 802.11p uses the Tiered Contention Multiple Access (TCMA) scheme [21], an extension of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) [22], which provides a shorter back-off time for higher priority messages when the channel is deemed busy. In addition, single-radio devices that listen to one channel at a time, are allowed to use both safety and service-related communication services, through a synchronization mechanism ensuring that all WAVE single and multi-channel devices monitor the control channel at specific time intervals [5]. As an evolution to IEEE 802.11p, the IEEE 802.11bd standard is currently under development to improve latency, reliability and throughput performance [23].

The regional standardization bodies designate the spectrum bands for both safety-specific and general V2X applications. To this end, ETSI and the Federal Communications Commission (FCC) have allocated 70 MHz and 75 MHz respectively on the 5.9 GHz ITS band, split into 10 MHz channels, whereas Japan's ARIB has allocated a single channel on 755,5-764,5 MHz [18].

The 3rd Generation Partnership Program (3GPP) has also entered the V2X communications field by introducing C-V2X, comprising LTE-V2X and 5G NR V2X. C-V2X includes two complementary communication modes: *i*) the Sidelink (SL) for V2V, V2I and V2P, without relaying data through the cellular network, and *ii*) the traditional cellular mode over a base station for V2N. 3GPP introduced the LTE-V2X standard in Release 14 [24] supporting two resource allocation modes: *i*) Mode 3, where the cellular network allocates resources used by the User Equipment (UEs) for their direct communications, and *ii*) Mode 4, where the UEs autonomously select and configure the radio resources, without requiring cellular coverage. In addition, LTE-V2X supports congestion control in Mode 4, by defining related metrics and possible mechanisms to reduce channel congestion. In Release 15 [25], 3GPP designed LTE-eV2X, which operates in Band 47 (5855-5925 MHz), corresponding to the European ITS spectrum, and included enhancements to address higher KPIs than Rel. 14. LTE-V2X [24].

While LTE-V2X addresses basic safety use cases, such as intersection collision warning and VRUs protection, 5G NR V2X was introduced in Release 16 [26] to complement LTE-V2X and accommodate more advanced use cases, such as autonomous driving [27]. 5G NR V2X SL supports flexible numerology that employs different Subcarrier Spacings (SCS), leading to different slot durations [28]. As a result, larger Subcarrier Spacings (SCS) are suitable for low latency applications because it results in shorter slot durations, reducing the overall user plane latency. Similarly to LTE-V2X, 5G NR V2X SL supports two resource allocation modes: *i*) Mode 1, where the network manages the resources for the UEs, and *ii*) Mode 2, where the UEs manage the resources autonomously, without requiring network coverage. 5G NR V2X operates both in the ITS band and the licensed mobile broadband spectrum, *i.e.*, Frequency Range 1 (FR1), that includes sub-6 GHz frequency bands and Frequency Range 2 (FR2), including 24.25-52.6 GHz.

3. Machine Learning overview

This section provides a brief overview of the AI/ML methods employed by the publications surveyed in Section 4. We do not provide a detailed presentation on these methods, considering that they have been already covered extensively in other publications. For the interested reader, Figs. 2–7 include key points, challenges, complexity aspects and references to related introductory publication for each method.

Typical AI/ML algorithms can be classified in four distinct categories based on the way learning is performed, namely supervised, unsupervised, semi-supervised and reinforcement learning.

In supervised learning, model training takes place using both input and output data, called *features* and *labels* of the dataset, respectively. The label largely determines the task we are assigned to solve, i.e., whether it is a regression (label is a continuous variable) or classification problem (label is a discrete variable). The features are associated with the labels through a target function. During the training phase, the model is learning an approximation of this target function by adjusting its parameters until it reaches an optimal solution. The learning process is controlled by a set of pre-defined hyperparameters, such as the learning rate or the batch size, which the user defines after fine-tuning prior to the training process. In effect, this is an optimization problem of finding the minimum or maximum of an objective function depending on the algorithm of choice. We evaluate the solution on a separate testing dataset, not used during the training phase to avoid overfitting.

Fig. 2 summarizes the key points, challenges and time complexities of supervised learning algorithms surveyed in this paper, namely k-Nearest Neighbors (kNN) [29], [30], Random Forests [31], Support Vector Machines (SVMs) [32], and Artificial Neural Networks (ANNs), namely Multi-layer Perceptron (MLP) [33], Convolutional Neural Networks (CNNs) [34], and Recurrent Neural Networks (RNNs) [35]. The first three methods belong to the ML field. At the same time, ANNs are part of the DL field and have been in the spotlight in recent years due to their ability to handle massive volumes of data more effectively than traditional ML techniques. ANNs are able to model non-linear functions and consist of interconnected nodes called neurons, dispersed on three layers: the input layer, the hidden layers and the output layer. The features enter the input layer and are propagated through the hidden layers to the output layer. Neurons in the hidden layers transform the information via specific computations and finally, the output layer extracts the predicted values.

Fig. 3 depicts the time complexities for training various supervised learning algorithms used in the papers of our survey. We have made some assumptions about the values of parameters included in the Big-O notation mentioned in Fig. 2, based on values supplied by the papers included in our survey or commonly used in the literature, whenever this information was not available. We have not included CNNs in this plot because their reported complexity does not depend on the number of training samples. The authors in [34] provide extensive comparisons between well-known CNNs architectures.

Fig. 4 summarizes the key points, challenges and time complexity of unsupervised learning algorithms used in the surveyed publications, namely k-Medoids [36], k-Means [37] and Affinity Propagation [38]. In unsupervised learning, the learning model is provided only with unlabeled datasets and extrapolates patterns without knowing any labels. Unsupervised learning tasks include clustering, *i.e.*, grouping of similar data into groups, dimensionality reduction, *i.e.*, decreasing the number of input features while preserving as much of the original information as possible, and density estimation, *i.e.*, estimating an underlying probability distribution function.

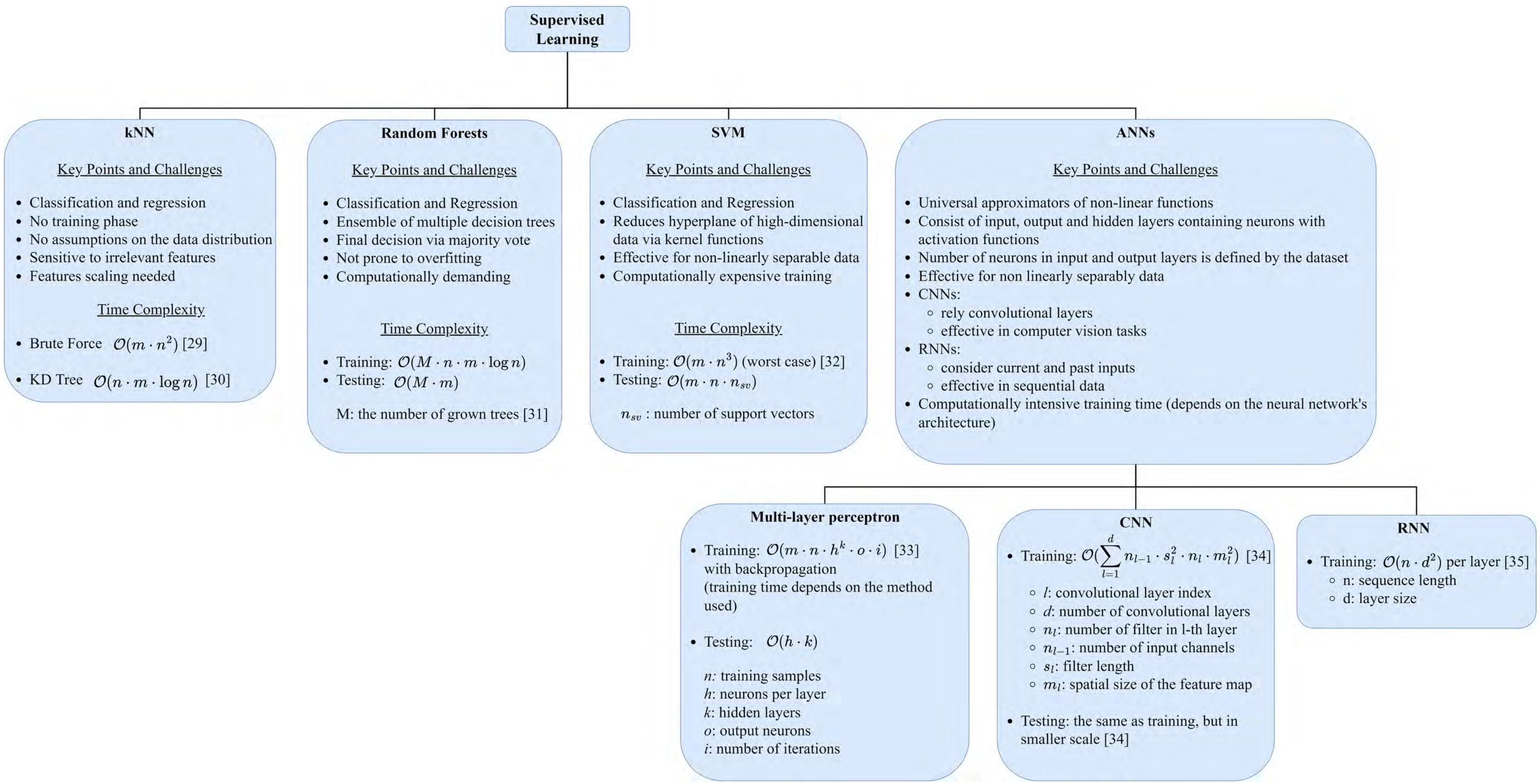

Fig. 2. Supervised Learning Algorithms Overview.

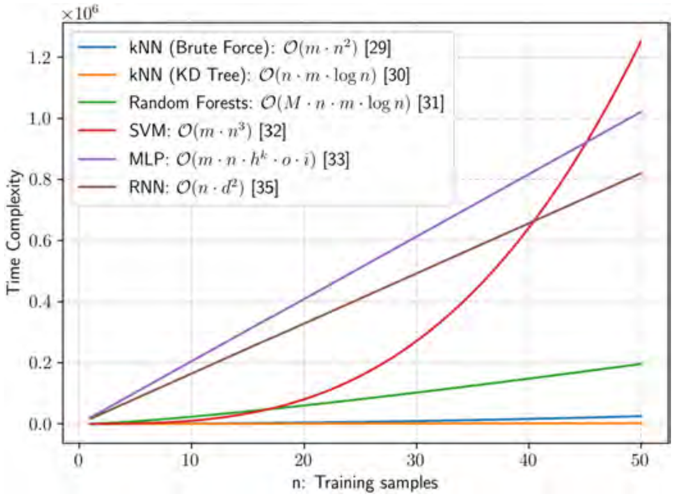

Fig. 3. Time Complexities for Training of common Supervised Learning Algorithms. We have assumed the following values of the parameters in the Big-O notations of Fig. 2: M = 100 trees, m = 10 features, h = 32 neurons per layer, k = 2 hidden layers, o = 6 output neurons, d = 128 neurons in a single RNN layer, and i = 10 epochs.

Fig. 5 depicts the time complexities of k-Means, k-Medoids and Affinity Propagation for $k = 5$ clusters.

The third learning approach is semi-supervised learning, standing in the middle between supervised and unsupervised learning. In semi-supervised learning, a model is trained with both labeled and unlabeled data. However, the portion of labeled data is relatively small compared to the unlabeled portion. Semi-supervised learning aims to achieve higher performance by leveraging the unlabeled data compared to using only labeled data. There is a vast amount of semi-supervised learning techniques. None of the surveyed papers presented in Section 4 use semi-supervised learning methods, so we omit an extensive presentation of related methods. Engelen et al. [39] provide an extensive survey on semi-supervised learning.

Reinforcement Learning (RL) is a branch of ML used for decision-making problems. The agent and the environment are the essential components of RL. Like humans, agents learn to select the optimal actions by interacting with their environment. The agent executes actions and receives feedback from the environment, termed “reward”. At each time step, the agent receives a representation of the environment’s state and selects an action based on this representation. The agent then receives a reward and finds itself in a new state [40]. Through this closed-loop interaction, the agent gains experience and selects actions that maximize rewards from the environment.

RL agents select actions based on policies, *i.e.*, functions that map states to actions and dictate the agent’s behavior. Solving RL problems essentially decomposes into finding an optimal policy for the given problem. An RL agent also includes a value function that calculates the expected reward if the agent began in a specific state-action pair and acted according to a policy. There are different forms of value functions in RL, which obey the Bellman equations [41]. Thus, the agent gains experience by trial and error; it calculates the expected reward for each state-action pair and stores the result in a lookup table. This tabular representation, however, is not feasible in large state spaces. In Deep Reinforcement Learning (DRL), a neural network is used (that may include multiple hidden layers to capture the intrinsic representations of the problem, that is why it is termed “deep”) and approximates the expected value of each state-action pair, instead of just storing it in a lookup table.

RL algorithms specify how the agent changes its policy due to its experience. There are three approaches to solving RL: i) value-based RL, where one estimates the optimal value function, which is the optimal value achievable under any policy, *ii*) policy-based RL, where one searches directly for the optimal policy that achieves the maximum future reward, and *iii*) model-based RL, where one builds a model of the environment and makes decisions based on this model. Value- and policy-based algorithms are called *model-free* and are currently more popular than model-based approaches. It is more flexible to define the value functions or policies than the model for the environment.

There is a vast amount of RL and DRL algorithms. Following the taxonomy in [42], Fig. 6 summarizes key points, challenges, and sample complexities for common RL algorithms used in the surveyed papers presented in Section 4. We supply the sample complexities for RL algorithms, which define the number of steps required to obtain a satisfactory policy for a given problem, *i.e.*, an ε-optimal policy with probability at least 1-δ, where $\delta \in (0,1]$ is the failure probability. The sample complexities are critical in RL, since they dictate the exploration vs. exploitation trade-off, i.e., exploring an uncertain environment while maximizing a reward.

Fig. 7 depicts sample complexities of selected algorithms used in the papers in Section 4, including Q-learning [43], Actor Critic [44] and Multi-armed Bandit [45] for values of $\varepsilon \in (0,1]$. For increased accuracy, *i.e.*, smaller values of ε, Q-learning requires a larger number of samples compared to Multi-armed Bandit and Actor Critic. We plotted the sample complexity of Q-learning in a separate plot due to the very high number of samples required. To generate the plots, we have made the following assumptions, based on representative values used in the surveyed papers of Section 4: i) we consider the number of arms n = 15, in Multi-armed Bandit, *ii*) we consider the number of states S = 36 and number of *Actions* = 4, and *iii*) we consider the discount factor γ = 0.99, which determines the importance of future rewards.

We presented the three learning types for training an AI/ML model. There are paradigms that accelerate and improve these learning types that have emerged recently, namely i) Federated Learning, *ii*) Meta-Learning, and *iii*) Transfer Learning. Federated Learning (FL) is an emerging solution where model training takes place across local edge devices (e.g., mobile devices, Mobile Edge Computing servers on base stations) on their own generated datasets. The edge devices use these datasets to build local models, which in turn are aggregated by a global server, generating one global model. Finally, the global server broadcasts the updated global model across the edge devices and then the training process continues.

In contrast to centralized approaches, where edge devices transmit their generated data directly to a centralized server, where model training takes place, FL provides lower latency, higher bandwidth, and privacy preservation since the user data stays on the edge device. Meta-Learning is described as “learning to learn”. It refers to gaining experience from observing the performance of different ML models on specific tasks and then use this experience to enhance one’s model performance and training speed when learning new tasks. The experience is gained by “meta-data” such as results from model evaluation, task properties or even performance of prior models. Transfer Learning is another ML paradigm where a model trained on a specific task, is used as a starting point for creating a model on a related task. In effect, pre-trained models are used to improve a model’s generalization ability and can be used in both ML and DL.

Each solution comes with its advantages and disadvantages. Different learning types are suited for different kinds of problems, as it will be discussed in Section 4. It is imperative to consider these aspects when selecting AI/ML-based solutions and formulate the problem accordingly, based on the designated requirements of the automotive use case at hand.

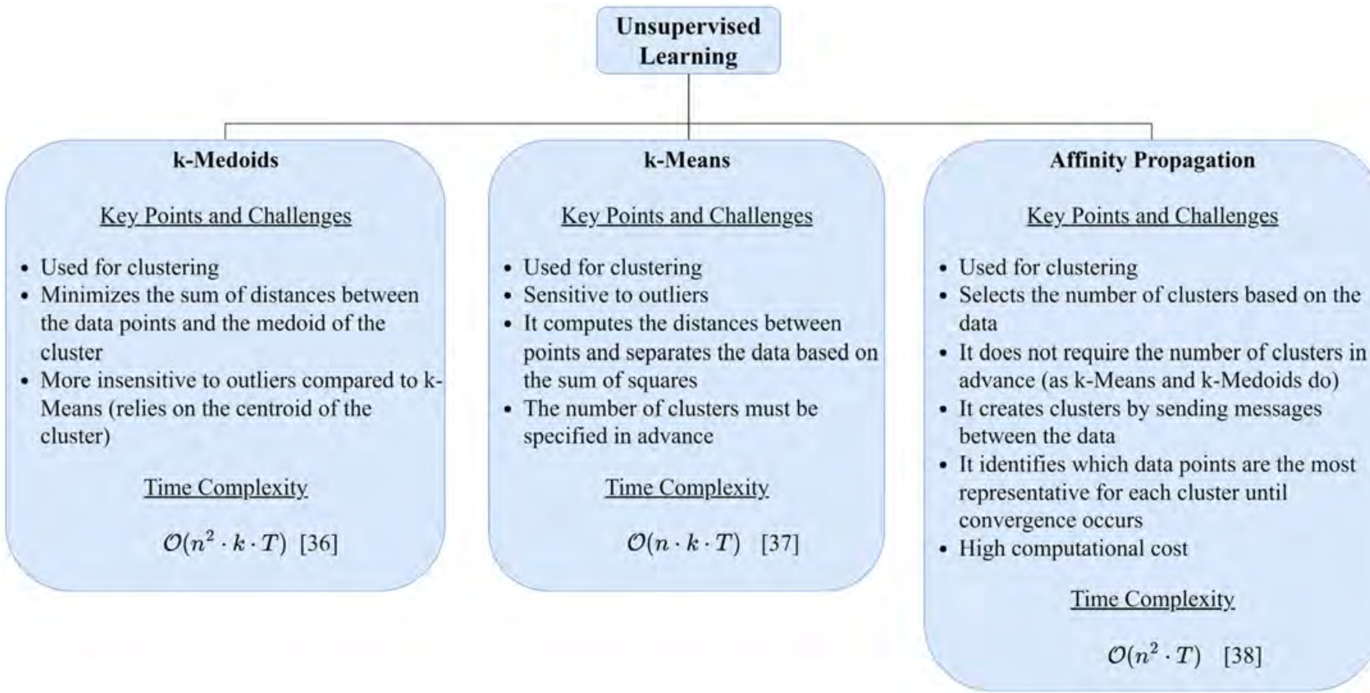

Fig. 4. Unsupervised Learning Algorithms Overview.

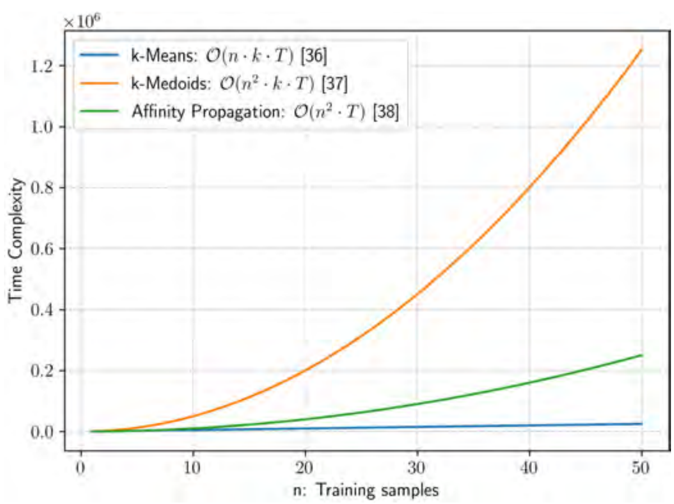

Fig. 5. Time Complexities of Unsupervised Learning Algorithms for $k = 5$ clusters.

Considering the emerging trend of AI/ML in V2X, we highlight some of the on-going standardization activities by Standards Development Organizations (SDOs) and Industry initiatives, as a final note in this introductory section on AI and ML. 3GPP integrated into its Rel.15 5G Core Architecture [46] the Network Data Analytics Function (NWDAF) [47] to provide network data analytics services to other network functions. ETSI has established two Industry Specification Groups (ISGs), the ETSI Experiential Networked Intelligence (ENI) [48] and ETSI Securing Artificial Intelligence (SAI) [49], whose main activities concern AI. ETSI ENI ISG has defined a high-level architecture based on AI techniques and context-aware policies to adjust services offered by MNOs, according to the changing user needs and business goals. ETSI SAI ISG focuses on enhancing the security of AI entities from an ever-growing list of potential attacks. ITU has also investigated the use of AI and ML in future networks through its "ITU-T Focus Group on Machine Learning for Future Networks including 5G" (FG-ML5G) [50].

Many industry initiatives have examined the use of AI in networking. O-RAN Alliance [51] is an industry group founded in 2018 to develop an Open Radio Access Network (O-RAN) with embedded intelligence based on general-purpose hardware. O-RAN has integrated data analytics services into its architecture, by introducing Radio Intelligent Controllers (RIC) for interfacing the RAN with third-party applications, which is impossible with current proprietary RAN solutions. In addition, the 5G Automotive Association (5GAA) has introduced the mechanism of Predictive QoS [52], allowing 5G mobile networks to notify the QoS prediction service consumers of predicted QoS changes to adjust the V2X application behavior in advance. In automotive use cases, such as tele-operated driving and high-density platooning, the network must provide a certain level of QoS. The Predictive QoS mechanism includes a Prediction Function (PF) that collects heterogeneous data from the vehicles, the network or third-party applications and provides predictions to interested consumers. Such predictions can be carried out using ML or DL techniques.

4. AI and ML solutions in V2X communications

This section presents applications of ML on vehicular networks based on recently surveyed publications. We examine several network control and management operations, including *a*) handovers and user association, *b*) caching, *c*) physical resources allocation (*i.e.*, power/frequency and channel access management), *d*) computation resources management (*i.e.*, service offloading in Mobile Edge computing environments), *e*) routing, *f*) beam selection optimization and *g*) QoS prediction. For each operation, we present relevant papers, identifying their designated scenarios, the underlying ML method in use, the overall objective of the method, the architecture (*i.e.*, centralized, distributed or federated), the list of features used during training/exploration stages and finally, the lessons learned with regards to the results. All scenarios involve only vehicular environments.

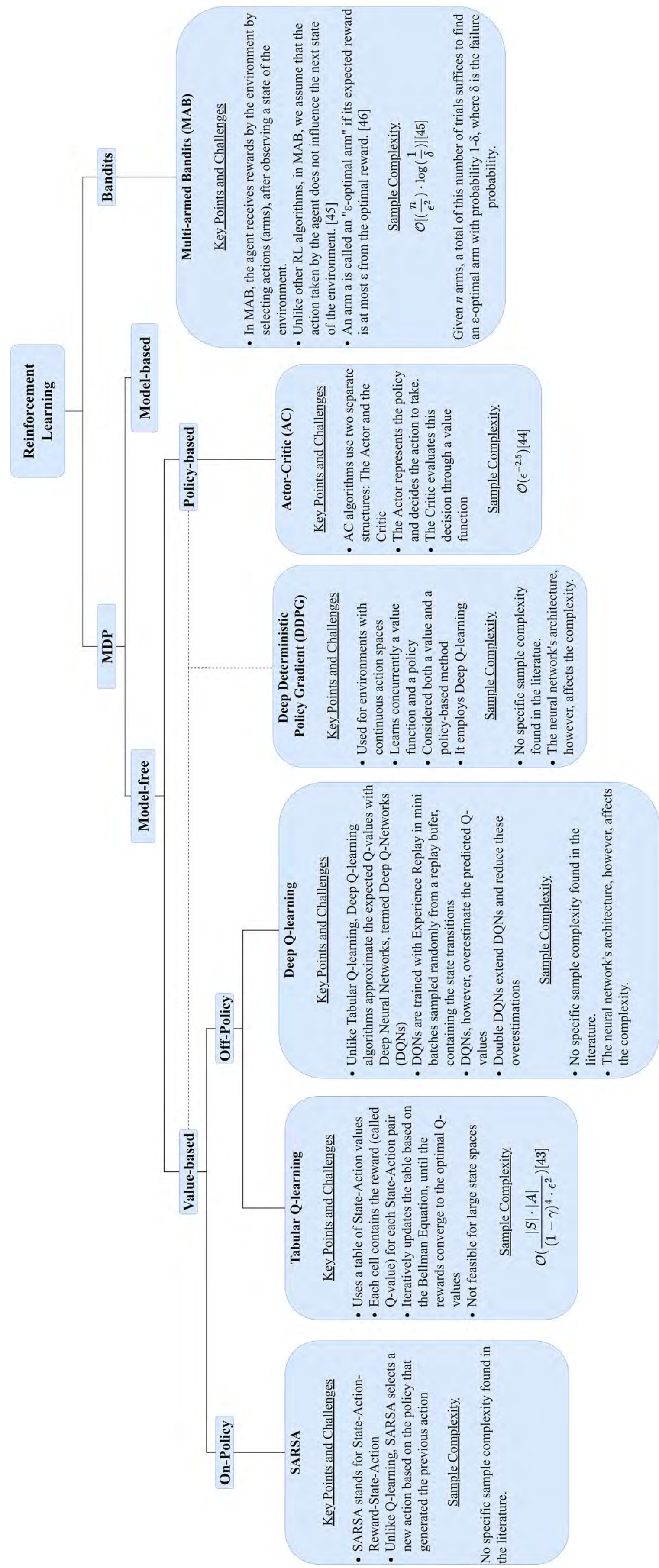

Fig. 6. Reinforcement Learning Algorithms Overview following the taxonomy in [42].

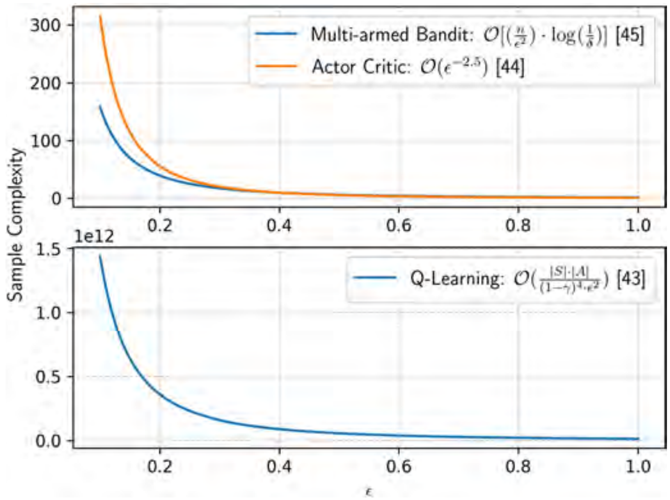

Fig. 7. Sample complexities of RL algorithms. We assume number of arms $n = 15$ (Multi-armed Bandit), number of States $S = 36$ and number of Actions = 4.

4.1. Handover management/user cell association

In wireless networks, handover management is essential in maintaining continuous service delivery during the UE's movement between neighboring cells. Handover is the process where the primary serving base station is substituted from a new one, according to some criteria (e.g., pre-defined radio signal thresholds). Depending on the radio conditions, a handover may impact the quality of the service to the end-users. Especially in vehicular networks, handovers are too frequent due to the velocity of the UEs and can result in service disconnections until the connection to a subsequent base station is established. ML is used in a predictive approach to improve the handover decision process, the base station discovery rate, and reduce the number of unnecessary handovers. Table 2 provides recent papers that have applied ML for handover management in vehicular networks.

In [53], Aljeri et al. employ a two-tier handover scheme implemented in the vehicles, where the first tier uses a Long Short-Term Memory (LSTM) model to forecast Received Signal Strength Indicator (RSSI) values to reduce the number of lost packets and the number of unnecessary handovers. The second tier utilizes a stochastic Markov model to select the next access point based on vehicle mobility projections. Vehicle speeds vary from 0 to 30 m/s. The LSTM model shows higher accuracy compared to traditional algorithms, such as simple moving average (SMA), simple moving median (SMM), and exponential smoothing that perform poorly in dynamic environments, like vehicular networks. The two-tier framework reduced the number of dropped packets by 18%, while yielding a higher packet delivery ratio because of the appropriate handover trigger time. However, the proposed solution did not consider the access point load or required QoS, left as improvements in future work.

Following a distributed approach, the authors in [54] and [55], employ DRL in urban environments. In [54], Lin et al. develop a DRL-based approach using DDPG for reducing the handover overhead during User-Centric clustering migration of high speed vehicular users (VUEs) relying on RSU cooperation and V2V communication. All VUEs act as agents that recommend a clustering policy based on their locations, observable RSU, vehicular Access Points (APs), and associated transmitters at previous timeslots. The DRL-based solution reduces the frequency of HOs at least by 50% compared to traditional baselines, especially when the number of RSUs is equal or larger than 14. The clustering design achieves at least 30% higher per-user average trade-off utility function and 25% higher per-user average data rate than the benchmarkers.

In [55], Khan et al. present a DRL-based scheme that uses the Actor-Critic Algorithm for maximizing the average rate per vehicular user, while maintaining a target minimum. Each RSU includes a local agent that determines the optimal RSU-vehicle association based on the past observed channels, vehicles' experienced rate, and a threshold violation rate indicator defined for each vehicle. The vehicles move with an average speed of 25 km/h. The proposed solution achieves up to 15% gains in terms of sum rate and 20% reduction in VUE outages compared to several baseline designs.

Tan et al. [56] proposed a DRL-based handover framework with double deep Q-learning to improve the handover decision time and maximize the vehicles' throughput. Double deep Q-learning stabilizes the DRL training and improves the chances of convergence, overcoming the potential problem of non-convergence due to training instabilities of the DQN algorithm. A centralized agent observes the Reference Signal Received Power (RSRP) measured by each vehicle from all surrounding base stations and decides in which base station it will connect to, if needed. The authors use metrics from the E-UTRAN measurement report to enable flexible integration of the proposed framework into a live cellular network. Results show that the DRL solution improves the handover triggering instant by 10.04 sec and 42.62% cumulative packet loss for all trajectories compared to the traditional A3 RSRP handover algorithm.

In [57], Souza et al. propose an RL-based framework using Tabular Q-learning to minimize the number of handovers in vehicular, fog computing environments. An agent determines whether the vehicle should connect to an Access Point (AP), remain connected to an AP, disconnect or remain disconnected. The agent interacts with the environment via a state space comprising the position of the moving device at a given time, the position immediately before (east, west, north, south) and the device connectivity status at that moment (disconnected or connected to one of the APs). The RL-based solution presented an average reduction of 16.2% on the number of handovers and an improvement of 16.9% on the average uninterrupted connection time among the different scenarios.

A promising field of AI/ML applications is on the mmWave spectrum, as shown by the following two papers. In [58], Yan et al. use a control/user plane decoupled network architecture with mmWave Remote Radio Units (RRUs) assisted by lower sub-6 GHz LF-RRUs, which pre-activate appropriate beams according to the vehicle's position. The authors employ a Kernel-based method for predicting the vehicle's positions based on the Channel State Information (CSI) signal. Although its training has a high computational complexity, the Kernel-based method achieves the same and better accuracy than a Kalman-filtering scheme. Then, the vehicle's position along with the serving mmWave RRUs index and the serving beam index of handover vehicles are used as input features to a kNN classifier located on the mmWave RRUs to predict the selected target mmWave RRU index and the target beam index of the handover vehicle. Regarding the handover decision process, the kNN scheme reduces time consumption down to approximately 2.40 ms compared to conventional beam training schemes (40.96 ms) and provides low estimation errors.

In [59], the authors propose a FL framework for proactive handover in a heterogeneous network with a Master Base Station (MBS) and mmWave Secondary Base Stations (SBSs). The MBS broadcasts a global model across all users of a specific region. Each moving user receives the global model and initiates training of their local model, an MLP, in order to select the next associated SBS based on the observed Signal-to-Noise Ratio (SNR). All users participate in the training and upload their local models to the MBS asynchronously, where the update of the global model takes place. This scheme addresses the limited storage capacity of the users, includes more participant users in the global training

Table 2
Publications on AI-enabled handover management in vehicular networks.

Ref.	Scenario	ML method	Objective	Architecture	Conclusion
[53]	Random Access Points	(Supervised) • LSTM	Reduction of number of unnecessary handovers	Distributed	The ML-based handover trigger scheme results in less dropped packets compared to a hybrid solution (80000 vs. 140000 packets). Higher accuracy of the LSTM model compared to various forecasting models.
[54]	Urban multi-lane freeway	(DRL) • DDPG	Reduction of number of handovers considering the average data rate per user	Distributed	The proposed technique reduces handover frequency in different scenarios (approximately 20 handovers vs. 100 of the traditional RSS-based handover schemes.
[55]	Urban micro environment, with LOS connectivity	(DRL) • Actor Critic algorithm	Maximization of the average data rate of the VUEs	Distributed	15% gains in terms of sum data rate and 20% reduction in VUE outages compared to several baseline designs.
[56]	Real-world scenario	(DRL) • Double Deep Q-Learning	Improvement of handover decisiontime and throughput maximization	Centralized	The proposed technique reduces the handover delay by 11.56 seconds and the packet loss by 25.73% per handover compared to the A3 RSRP baseline.
[57]	Grid-shaped Road network	(RL) • Tabular Q-learning	Reduction of number of handovers	Centralized	The scheme achieves 16.2% reduction in the number of handovers and improves by 16.9% the average uninterrupted connected time compared to a greedy benchmark.
[58]	Intersection	**Localization:** (Supervised) • Kernel-based **HO Decision:** (Supervised) • kNN	**Localization** Improvement of target discovery rate **HO Decision** Improvement of handover decision time	Distributed	**Localization** Achieves the same or better positioning accuracy with lower over compared to Kalman filtering. **HO Decision** kNN reduces time consumption and provides low estimation errors.
[59]	HetNet	(Supervised) • MLP	Reduction of handover rate	Federated	Reduces frequent handovers and improves QoS of users simultaneously.
[60]	50 gNBs with 10 UEs	(RL) • SARSA (Q-learning variation)	Select optimal cell during handovers	Centralized	For varying UE speeds, the solution outperforms other baselines consistently, achieving higher throughput (∼80 Mbps against ∼70 Mbps at 150 km/h), lower packet loss rate (∼13% against ∼10% and ∼6% at 150 km/h), packet delay reduction (38.98%) and handover latency reduction (24.87%) compared to Multi-armed Bandits.
[61]	Madrid Grid with 5G mmWave stations	(DRL) • DDQN	Avoid sudden Radio Link Failures while providing highest throughput	Centralized (The UE decides the action)	The DDQN-based solution presents consistently higher average throughput gains compared to traditional DAPS and CHO procedures for various UE velocities (∼600 MBps at 12 m/s UE velocity).

process and effectively reduces handovers. Simulations show that the federated learning scheme decreases the number of handovers compared to reactive handover schemes (from approximately 30 to 14 handovers for user velocities around 15-20 m/s compared to a baseline scheme), while it increases the average SNR of the users, reflecting an improved QoS across the region.

Karmakar et al. [60] present an RL-based solution in 5G NR based on SARSA algorithm -a variation of Q-learning- to select the target cell during handovers. The first step includes a Kalman filter that predicts the RSRPs of the serving and neighbor cells. The second step employs SARSA; this algorithm depends on the RSRP of the serving and neighbor cells, the last handover Q-value and the RSRQ of the neighbor cell. The authors employ RSRQ as the reward, as it: i) includes both RSRP and RSSI, and ii) captures the overall handover performance. For UE speeds varying from 50 km/h to 350 km/h, results indicate that the proposed solution achieves consistently higher throughput (∼80 Mbps against ∼70 Mbps of the baselines at 150 km/h) and lower packet loss rate (∼13% against ∼10% and ∼6% at 150 km/h) compared to selected baseline mechanisms, i.e., a Multi-Armed Bandit algorithm and a doppler-based mobility solution. SARSA also provides consistently lower packet delay and handover latency -the difference between the reception of the last packet through the old connection and the first packet in the new connection- for increasing cell crossing rates (24.87% latency reduction and 21.12% packet loss reduction compared to MAB).

Lee et al. [61] present a DRL-based solution with DDQN in 5G mmWave networks, applied in Dual Active Protocol Stack (DAPS) handover. The DAPS handover is triggered following the usual handover events (e.g., A3, A4, etc), and maintains the connection to both the source and target cells. However, event-based Handovers are not suitable for mmWave links, which are highly volatile, while handover events assume more gradual changes in channel conditions. The solution employs DDQN where a learning agent is deployed on the UEs and learns when it is best to conduct a handover, avoiding sudden radio link failures. The DDQN-based solution presents consistently higher average throughput gains compared to traditional DAPS and CHO procedures for various UE velocities (∼600MBps at 12 m/s UE velocity).

Lessons learned: We can extract a few key takeaways, based on the previous analysis. Handover management in vehicular environments is usually formulated as a problem for reducing the number of handovers. In addition, centralized architectures are common, considering the nature of handovers. In this case, training is conducted on a central entity, such as a base station, which collects data from the vehicular users, trains a central model and performs actions according to this model. Distributed approaches are also present, where training takes place independently on vehicles or base stations. Based on the surveyed papers, FL has not been used extensively in handovers. However, considering privacy concerns, storage limitations and network overhead during data exchanges in V2X environments, it is anticipated that FL approaches will begin surging.

Regarding the selection of ML algorithms, RL, DRL and supervised learning techniques have been used. Supervised learning is either used in regression, where one predicts time series, e.g., RSSI, or in classification, where one formulates handover management as a classification problem, e.g., selecting the optimal access point. RL/DRL are also popular because they are appropriate for decision making problems. In addition, RL does not need a labeled dataset, as supervised learning does since the agent learns directly from its environment. A disadvantage of RL/DRL is considered the exploration time, i.e., the time it takes the agent to learn an optimal policy and is characterized by the sample complexity, which in [55] is addressed using offline training. We present such typical sample complexities extracted from the literature in Fig. 7.

All approaches that have been used in handover management seem promising. From [58], it is learned that kernel-based methods can predict effectively the position of vehicles based only on the CSI, considering that such techniques yield high computational complexity in training ($\mathcal{O}(n^3)$), while kNN can reduce handovers at low cost compared to traditional beam training techniques since it does not require separated training and testing phases. As indicated in [59], FL can reduce the handovers and increase the SNR of the moving users, while addressing the limited storage capacity of the vehicles. In [53], it is shown that neural networks (LSTM and CNN) are more effective in predicting RSSI values in dynamic environments than traditional algorithms like moving average. Following [54] and [55], distributed DRL can help reducing the number of handovers and increase the data rate of vehicles. Finally, it is learned from [57] that tabular Q-learning can reduce the handovers and improve the uninterrupted connection time of vehicles.

4.2. Caching

In content caching, popular content requested frequently by users is pre-cached on edge cloud infrastructures or even on the vehicles themselves. This content is retrieved via V2V or V2I links, reducing the transmission latency and improving the overall backhaul capacity. Table 3 highlights the parameters of the surveyed publications regarding AI/ML for caching in vehicular networks.

Considering a DDPG-based edge cooperative cache algorithm over a high-speed free-flow road segment, the authors in [62] develop a cloud cache placement method for reducing the transmission delay of requested contents. A centralized agent decides on the amount of V2I and V2V link bandwidth allocation and the edge cloud cache ratio by exploring the vehicle's instantaneous speed, position and remaining data that need to be obtained by the vehicle and the edge cloud. Compared to two baseline cache placement strategies (non-cooperative and random), the RL-based cooperative cache scheme reduces the minimum content transmission delay and minimizes system overhead.

Except for traditional RL, the authors in [63] employ DRL on a multi-timescale model to jointly optimize the cost of communication, storage, and computation costs according to hard deadline constraints. The set of possible connected RSUs and neighboring vehicles, the decision on whether to offload the computation task, and the number of packets that should be cached are determined by both a large and a small timescale model. These models operate over each epoch (i.e., the duration of communication for a requested content) and over each timeslot, comprising each epoch, respectively, based on the state of the available RSU/MEC servers', vehicles' and caches' availability. The multi-timescale framework allows an estimation of the reward, when the agent takes an action each epoch, while in the small timescale model, the authors calculate the exact immediate reward achieved every time the agent takes an action. The suggested scheme achieves higher success probability (meaning the tagged vehicle completes downloading the requested content and offloading its corresponding tasks for computing within the hard deadline) and significant performance gain compared to alternatives, outperforming the single time scale framework in terms of cost gain vs vehicle's mobility intensity, backhaul capacities and cloud computing resources.

Similarly in [64] and [65], the authors develop centralized frameworks based on DRL, to maximize the profits of an MNO. In [64], a BS along with RSUs deliver contents and execute tasks for 20 vehicles with average velocity of 30 km/h, based on real traffic data in Hangzhou, China. An agent decides the number of computing tasks and requested contents to be executed at different MEC servers, and how many resources are allocated to vehicles, based on the vehicle's location and velocity, the total size of computing tasks and requested contents, the popularity of requested contents, the size of residual contents, the remaining computing tasks and required computation resources, the required CPU cycles for the task computing and the available resources of each MEC server. The proposed scheme achieves higher MNO's profits considering different scenarios of increasing number of vehicles, computing charging time, caching charging price, vehicular velocity and convergence speed. In [65], a central agent decides which BS is assigned to the vehicle, whether the requested content should be cached on the BS and whether the computation task should be offloaded to the MEC server. The state space comprises the State of available BS, MEC Server and cache for a vehicle in specific timeslots. The proposed DRL solution is compared to four alternative schemes and achieves higher utility, considering increasing content size, charging price for accessing the virtual networks, charging price for MEC offloading and unit charging price for connecting to the cache servers.

In [66], the authors develop a distributed DL framework for enhancing infotainment services on self-driving cars by caching requested content on the RSUs, based on passenger's features such as age, emotion and gender. The objective is to minimize the total delay for retrieving infotainment contents. To this end, the authors employ a CNN to obtain passenger's features from the self-driving vehicle and an MLP to extract the probabilities of contents to be requested in specific areas and cache them at the RSUs and other self-driving cars. They also use k-Means to cluster the MLP outputs and apply Block Successive Majorization-Minimization to solve the optimization problem. The proposed scheme achieves higher MNO's profits considering different scenarios of increasing number of vehicles, computing charging time, caching charging price, vehicular velocity and convergence speed.

Hou et al. [67] develop a centralized RL-based approach with Q-learning to determine the optimal caching policy using a heuristic greedy process, minimizing the latency of caching services. The system space includes the number of cache units that has been occupied in a specific RSU. They also employ an LSTM model for predicting the vehicle's mobility at an intersection with RSUs. The prediction accuracy of LSTM reaches up to 87.6%. The proposed Q-learning scheme is compared to other baselines: i) maximum, where the predictor knows exactly the vehicle's behavior, ii) minimum, where there is no mobility predictor, iii) no-caching scheme, where the content is not cached on the RSUs, and iv) greedy strategy scheme, where the system caches the content based on the local optimum in each state. The proposed Q-learning scheme achieves highest reward compared to schemes ii), iii) and iv) with increasing number of available caches and prediction accuracy.

In [68], the authors propose a mobility-aware proactive edge caching scheme with FL (MPCF), based on adversarial autoencoders, in order to maximize the cache-hit ratio. The cache hit ratio measures the effectiveness of a cache in fulfilling content requests. The input features to the autoencoder are retrieved from the MovieLens 1 m Dataset [69] and include movie IDs, rating, user IDs, user contextual information (gender, age, occupation, zip code), while the output label is the predicted cache content. The proposed scheme presents the best cache-hit ratio compared with other solutions. Moreover, cache replacement in MPCF results in higher cache hit ratio when used (24,25% vs 22,2%), while MPCF achieves higher cache-hit ratio in a shorter time than a typical FL training process (MPCF achieves the target cache hit ratio of 16% in less than 60 seconds, while the typical FL process needs more than 600 seconds).

In [70], the authors propose FlexiCache, a centralized framework for adaptive edge caching based on Kernel Ridge Regression (KRR), in order to optimize the content retrieval time by allocating different proportions of the cache to safety-related and infotainment data. The input features to KRR include the average rate of arrival of interests for safety-critical data, the average rate of arrival of interests for infotainment data, the average frequency of interests for the same infotainment data object in a given period and the average content retrieval in seconds. The algorithm's output is a value that splits the RSU cache in chunks holding safety-critical and infotainment data. The proposed solution outperforms undifferentiated caching and the feedback mechanism of FlexiCache results in very low difference between the target and actual retrieval times, even for highly mobile urban road topologies.

Finally, Song et al. [71] consider a DRL based approach (DDPG) to maximize the QoE of the requested files by moving vehicles. The authors propose a class-based user interest model to predict the access probability of files by users, considering that users will request files of the same class in consecutive requests. The agent learns a policy that specifies whether to cache requested files in any RSUs, based on a reward that follows the proposed class-based user interest model. For a single vehicle, with the proposed method, the QoE increases as cache size increases, while vehicle speed affects the QoE (0.3 against 0.2 for the largest cache size). For multiple vehicles, the proposed method is compared to traditional cache algorithms for different number of file classes, where results show that as the number of file classes increases, (i.e., 40, 60, 80, 100) the traditional cache algorithms provide almost the same QoE (approximately 0.045 in all file classes), while the proposed solution achieves higher QoE in almost all cases except the highest number of classes (~0.09 for 40 file classes, ~0.07 for 60 classes, ~0.055 for 80 file classes and ~0.045 for 100 file classes). This behavior is consistent, considering that as the number of classes increase, the files in each class decrease, thus, reducing the predictability of the algorithm. The method seems promising; however, it could be tested in more complex vehicular scenarios to examine its scalability and robustness.

The authors [72] present a multi-actor critic solution in a multi-server environment. The actors are distributed over the edge servers and there is a centralized critic, that informs the actors whether their selected actions are suitable. The main objective is to minimize the overall task execution delay. The system state represents the Edge servers' environments and includes the task execution delay on the server, the remaining available resources and cache resources, as well as the cache policy on the server. The proposed solution is compared to: i) an independent multi-actor critic variation, where the actor-critic pairs are independent with each other, ii) a random cache polic, and iii) a no-cache policy. Results show that the proposed solution achieves consistently lower task execution delay than the other baselines with increasing edge caching size and edge computation capability (for 600 MB file: ~0.50 sec against 0.40 sec of Actor Critic variation and ~0.78 sec of random cache policy) and computation capability (for 3 GHz: ~0.30 sec against ~0.40 sec of Actor Critic variation and ~0.88 sec of random cache policy). In the Centralized Critic proposed solution, the Critic module accepts information from all edge servers, and thus, makes decisions that perform better than its independent variation.

The authors in [73] propose another DRL solution based on Actor Critic for minimizing the downloading delay and the caching cost in mobile edge networks. The solution includes a centralized Actor-Critic pair where the system space represents the user's content demand matrices and the action space includes the set of caching decision matrices. The authors reduce the computational complexity of estimating the policy function with a branching neural network technique. They separate the action space in several dimension and assign each neural network branch on each dimension. The proposed solution outperforms the SoTA and other baselines for varying arrival rate of requests, number of content items requested and number of edge clouds in terms of total cost (12% improvement over SoTA for caching 2500 content items), average downloading delay (8% gain over SoTA), improving the cache hit rate and provides results closer to the upper bound.

Lessons learned: Centralized architectures are more popular in caching among the surveyed solutions. This is a result of the caching problem nature, where edge cloud servers retrieve information from vehicular users and train centrally a model to perform the necessary actions. FL was used only in one surveyed paper, but considering the issues of privacy, storage capacity and network overhead, it is anticipated that more publications will explore its application on caching.

Regarding the learning techniques, RL based solutions are more prominent than supervised and unsupervised learning ones. This can be attributed to the versatility of the reward functions, used for training the RL agents, where one can formulate such functions considering the constraints and target KPIs of the designated problem. Then, the agents are learning to select optimal actions according to this reward function. In addition, as we mentioned earlier, in RL, the agent is learning directly from the interactions with its environment, without requiring a pre-processed dataset. This is a significant advantage of RL compared to supervised learning techniques. Regarding the optimization objectives in caching management, the most commonly used objectives are to minimize the service latency or maximize the profits of mobile networks operators, according to Table 3.

As indicated in [62] and [67], RL can help reduce the content transmission delay and system overhead in joint optimization of bandwidth allocation and content caching or minimize the latency of caching services. Following [63], [64] and [65], DRL has been effectively used in joint caching and computing schemes for maximizing profits and reducing costs of communication and computation. In [66], neural networks can minimize the delay of retrieving infotainment contents in multi-access edge environment, when

Table 3
Publications on AI-enabled caching in vehicular networks.

Ref.	Scenario	ML method	Objective	Architecture	Conclusion
[62]	High-speed, free-flow road segment	(RL) • DDPG	Weighted average transmission delay minimization	Centralized	Reduces the minimum content transmission delay and minimizes the system overhead, compared to non-ML baselines.
[63]	HetNet	(DRL) • DQN	Communication, storage and computation cost minimization, with deadline constraints on the requested content	Centralized	Higher success probability in delivering the requested content within the imposed deadlines and achieves higher cost gains compared to other baselines.
[64]	Real traffic data	(DRL) • DDPG	Maximize MNO's profits	Centralized	Higher MNO's profits in different scenarios of increasing number of vehicles, computing charging time, caching charging price, vehicular velocity and convergence speed.
[65]	Randomly distributed vehicles	(DRL) • DQN	Maximize the MVNO's revenues	Centralized	Higher utility compared to other baselines at varying content size and charging prices for connecting to the cache servers.
[66]	Multiple dispersed RSUs	(Supervised) • CNN • MLP (Unsupervised) • k-Means	Minimize the total delay for retrieving infotainment contents	Distributed	Reduces the backhaul traffic with high accuracy of the contents that are requested across the RSUs.
[67]	Intersection	(Supervised) • LSTM for mobility prediction (RL) • Q-learning for caching	Minimize the latency of caching services	Centralized	LSTM Prediction accuracy: 87.6%. The Q-learning scheme achieves highest reward compared to baseline schemes with increasing number of available caches and prediction accuracy.
[68]	Urban scenario	(Unsupervised) • Adversarial autoencoder	Maximize cache-hit ratio	Federated	Best cache-hit ratio compared with other solutions.
[70]	Grid-like Road structure with intersections.	(Supervised) • Kernel Ridge Regression	Minimize the content retrieval time	Centralized	Outperforms undifferentiated caching in low differences between the target and actual retrieval times even for highly mobile topologies.
[71]	Straight road	(DRL) • DDPG	Maximize QoE	Centralized	Higher QoE for different file classes (∼0.09 for 40 file classes, ∼0.07 for 60 classes, ∼0.055 for 80 file classes) and ∼0.045 for 100 file classes.
[72]	5 Edge Servers, 1 Cloud Server Random UE trajectories	(RL) • Actor-Critic	Minimize the average task delay of the multi-server task caching model	Centralized	The solution achieves lower task latency with increasing edge caching size (for 600 MB file: ∼0.50 sec against 0.40 sec of another Actor Critic variation and ∼0.78 sec of random cache policy) and computation capability (for 3 GHz: ∼0.30 sec against ∼0.40 sec of Actor Critic variation and ∼0.88 sec of random cache policy).
[73]	5 Edge clouds co-located with the BSs 1000 users	(DRL) • Actor Critic	Minimize downloading delay and caching cost	Centralized	The proposed solution outperforms other baselines in terms of total cost (12% improvement over SoTA for caching 2500 content items), average downloading delay (8% gain over SoTA), improving the cache hit rate.

considering the passengers' features as additional training features. In [68], it is learned that autoencoders (unsupervised learning) can help in learning the latent representations of users' contextual information through federated learning, so that the contents stored in the RSUs caches are updated accordingly. It is also shown that synchronous federated learning is too slow for dynamic vehicular environments, suggesting that asynchronous federated learning be investigated in future work. Finally, it is learned from [70] that it is possible to spit the edge cache into different sections, to hold data of different traffic type and QoS by using a predictor that takes into account current network conditions.

4.3. Resources management

This Section examines two separate fields of resource management in vehicular networks: i) Radio Access Network (RAN) Resource Allocation, addressing power, spectrum and channel access optimization for reducing interference, power consumption and network congestion, and ii) Computation resources management, addressing service, task and traffic offloading in Mobile Edge Computing (MEC) environments with edge cloud nodes deployed in base stations, leading to faster response times than central deployments.

4.3.1. Radio Access Network resource allocation

RAN resource allocation refers to power, spectrum and channel access control targeting to reduce interference between users, thus increase the overall throughput. Table 4 summarizes supervised, reinforcement learning, and deep reinforcement learning strategies used to optimize power and/or spectrum over vehicular networks.

In [74], Gao et al. employ a centralized Deep Neural Network (DNN)-based power allocation scheme on an intersection of high mobility for V2I and V2V links, formulating a sum rate maximization problem with power constraints on the V2X links. The input features to the algorithm are the channel power gains of V2I and V2V links and the result is the optimal power allocation. Simulations show that the proposed DNN based solution performs close to the Weighted Minimum Mean Square Error (WMMSE) minimization approach. Although training the DNN is time-consuming, the trained network can provide a real-time solution. In [75], the authors propose a centralized bandwidth allocation scheme and scheduling queue management via a wireless slice manager. They employ a CNN, LSTM and DNN to classify Software Defined Networking (SDN) flows as safety flow or not. Simulations show that the proposed framework optimizes resource allocations according to the changing demands of a vehicular network, achieving high accuracy in flow predictions. LSTM (99.36% accuracy) outperforms CNN (95%) and DNN (92.58%).

Gyawali et al. [76] employ graph theory for channel allocation by a centralized BS and DRL for distributed power control by the vehicular users. Each V2V transmitter is considered as an agent that learns whether the channel state is above a minimum threshold required for successful V2V communication. The agent senses the state and selects an appropriate action, e.g., select the optimal transmission power level and maximize the reward. The reward is the aggregate capacities of the V2V links in a cluster. The objective is to maximize the long-term rewards through Q-learning based on Q-table. However, this method does not work well with large state-action space. To address this issue, the authors use Deep Q-Learning, which although presents a time-consuming training process, the execution is very fast and performs well in large state-action spaces. Results show that the proposed scheme increases V2V link capacity and reduces the interference among vehicles.

In [77], Koshimizu et al. proposed a distributed VANET clustering scheme applied on moving vehicles within a C-V2X network. They employ affinity propagation for clustering vehicles, where the cluster head is decided upon message exchanging among the vehicles through the PC5 interface of C-V2X network, according to 3GPP specifications. In addition, the clusters granularity, e.g., whether a cluster is small, medium or large, is identified through a soft-margin SVM algorithm, considering: *i*) the communication volume generated by each vehicle, *ii*) the preference of the vehicle in establishing a session in clustered or individual mode, *iii*) the traffic density (about 60 vehicles in a single direction over 2 km span), and *iv*) the PLMN congestion status (normal or congested state). Simulations took place using real traffic data recorded over a US highway during one day. The authors identified leading and car-following vehicles based on the GHR car-following model. Simulated data achieved satisfactory results in predicting appropriate clusters for different granularity parameters, while the overall access latency in a 5G network achieved values as low as 3.9 ms, highlighting that a stable access system can provide smaller access latency. Affinity propagation does not need the number of clusters beforehand, compared to other clustering schemes, as k-Means, posing a significant advantage for high mobility environments with rapidly changing vehicle “formations”, like vehicular networks. Cluster formation provides stability on the cellular network, as it reduces the number of direct connections when the number of moving vehicles is large, avoiding the so-called “access stagnation”.

Ye et al. [78] developed a distributed resource allocation for effective channel sharing between V2V and V2I communications in unicast and broadcast scenarios. Such distributed approaches provide lower transmission overhead during information sharing among vehicles than centralized approaches, where this overhead is large and does not scale effectively with increasing network size. The authors employ a DRL algorithm based on deep neural networks, because traditional Q-learning requires larger times to converge in problems with huge state-action spaces, as in V2V communications. The agents' objective is to minimize the V2V interference on the V2I links, while meeting the imposed latency requirements regarding messages exchanging.

The reward in both scenarios includes the V2V and V2I capacities and the latency constraints. In the unicast scenario, the V2V links observe the system space consisting of the channel V2I and V2V information, the previous interference power to the link, the selected sub-channel of neighbors in the previous time slot and the remaining load of the vehicular UE to transmit, and the remaining time to meet the latency requirements. The agents select a sub-channel and power level for transmission. In the broadcast scenario, each vehicle is an agent that re-broadcasts the messages that have been received to improve reliability. The state space includes the features of the unicast scenario, the number of times a message has been received by a vehicle, and the minimum distance to the vehicles that have broadcasted the message. The simulation environment is based on the Manhattan case in 3GPP TR 36.885 where vehicles are dropped in lanes according to a Poisson process. Simulations highlight a higher probability of satisfied V2V links and power level selection for various vehicles compared with other baselines. However, one weak spot is the computation complexity for selecting the power levels and sub-channels, which can be alleviated using known methods in the literature.

Khan et al. [79] use LSTM on vehicles to predict transmissions of neighboring vehicles to avoid packet collisions. The learning vehicle monitors the received packets of visible and hidden neighbors and relies on the packet reception history to predict future transmissions. The input features to the LSTM model include the time interval between the currently received packet and the previous packet from a particular neighbor, vehicle dynamics and their gradients (position, degree, speed), while the output is the time interval for the next packet transmission. The packet reception ratio is higher in cases incorporating learning, compared to the case without learning. The case with learning from both hidden and visible nodes presents lower Packet Reception Rate (PRR) compared to learning only from hidden nodes. This is due to the high number of nodes that the learning vehicle is tracking in the first case and thus, cannot find sufficient vacant windows of channel of low channel activities and transmits immediately. This is a promising approach on the MAC in V2X, but several open issues need further investigation, such as the effect of transmit rate control, the coordination of multiple learning vehicles and hybrid scenarios involving both learning and non-learning vehicles.

Xiang et al. [80] followed a distributed approach using DRL and multiple agents, i.e., the vehicles, to jointly optimize power and channel selection with the objective of maximizing the V2I sum throughput under latency and reliability constraints of the V2V links. They applied a double dueling deep recurrent Q-network (D3RQN) that incorporates dueling DQN to increase the approximation of the value estimation of the original DQN. They also employed hysteretic Q-learning to coordinate multi-agent training. Moreover, the authors utilize the approximate regretted reward to address the changing environment dynamics. The observation space of each agent comprises the experienced interference power over all sub-channels, the remaining payload size to be transmitted and the remaining time to finish the transmission. The Channel State Information (CSI) is not included, because it is computationally complex to estimate and is not used in schemes like NR Mode 2, where the UEs select resources using the sidelink reference signal received power (SL-RSRP). Results show that the proposed scheme presents higher robustness in the packet delivery ratio in V2V links against payload size (90% compared to approximately 73% of NR Mode 2 scheme) and increasing vehicle velocity (approximately 90% for 10-15 m/s to approximately 86% for 25-30 m/s). The authors also conduct an ablation study to investigate the contribution of each ML technique in the overall performance. They concluded that the approximate regretted reward is significant for training in dynamic environments, since its absence leads to performance degradation of the V2V packet delivery ratio.

Banitalebi et al. [81] propose a multi-agent Q-learning distributed solution to improve the energy efficiency of C-V2X network. The authors formulate the problem as joint and disjoint power and subcarrier allocation. In the joint approach, the learning agents are the vehicles and D2D pairs and their actions include selecting a subcarrier and/or the power level. In the disjoint approach, the learning agents are the base stations that select the subcarriers. In this case, the distributed Q-learning method allocates the subcarriers and then, based on this action, the power is allocated as a result of a non-cooperative game. The joint approach achieves higher gains in energy efficiency (36%) compared to the disjoint approach at the expense of using more memory.

The authors in [82] present a DRL-based distributing solution for automatically adjusting the message broadcasting rate of the vehicles in a C-V2X network. The solution employs a Double Deep Q-learning network with risk awareness; reduce the broadcasting rate if the vehicles maintain a safe distance and increase the rate when the distance reaches a risky distance. The vehicles are the learning agents that observe their environment, consisting of the channel busy ratio, the SINR and their position. The action space includes the adjustment of the broadcasting rate. The proposed solution reduces redundant data up to 16%, increasing 22% the packet reception rate compared with baselines.

Lusvarghi et al. [83] propose a supervised learning approach to predict future CAM generation times in LTE with kNN. The vehicles use kNN to predict the next CAM generation time. The input features to the kNN are the vehicle's and preceding vehicle's trajectories, velocities and positions. Then, based on the predicted value, the vehicles select autonomously the radio resources, i.e., the resource reservation interval and the reselection counter, for broadcasting. The proposed solution achieves a Packet Reception Rate performance close to the ground truth and consistently higher than the Semi-Persistent Scheduling algorithm (SSPS) used in LTE Release 14, Mode 4 [24].

Lessons learned: In Radio network allocation, both distributed and centralized architectures are used. The nature of the problem is such, that the optimal radio resources allocation can be decided either by central entities (base stations) or by dispersed entities (vehicular users). The most popular objective is to maximize the average throughput of the vehicular users. The underlying learning techniques include all three types, supervised, unsupervised and RL/DRL. Supervised learning techniques are used when formulating the problem as a classification problem, e.g., enforcing specific thresholds to select appropriate power controls or as regression, for predicting upcoming transmissions, using LSTM. Unsupervised learning is used a first step in hierarchical solutions, for clustering moving vehicles. RL/DRL are also popular, considering the advantages we mentioned in handover and caching operations, such as versatility of the reward function and the direct training without labeled datasets.

Various solutions seem promising. [74], [79], [76] indicate that deep neural networks in supervised learning and deep reinforcement learning applications can provide fast predictions online once they have been trained, but exhibit high computational complexity during their training/exploratory phases. A solution to this is to conduct training offline. Following [75], deep neural networks can help in accurate traffic classification and management. In [76], [78], classic Q-learning is not appropriate for problems with many state-action spaces, therefore deep reinforcement learning approximations are preferred. From [77], it is inferred that vehicle clustering can lead to a stable system with small access latency, since it reduces the number of direct connections on the cellular network, while affinity propagation is a clustering algorithm that does not require the definition of the number of clusters in advance. Finally, distributed approaches provide advantages compared with centralized ones, since they reduce the overhead of messages exchanged between vehicular users.

4.3.2. Computation resources management

Computation resource management includes all tasks revolving around service, task and traffic offloading in MEC environments. MEC is an emerging architecture where edge nodes provide services closer to the users, minimizing the overall latency compared to centralized approaches. Table 5 summarizes the most important points of publications on AI/ML for computations resources management.

Dai et al. in [84] simulate a three-layer MEC architecture comprised of the infrastructure layer (e.g. RSUs), the edge computing layer comprised of MEC edge servers and the cloud computing layer, comprised of cloud resource pools. The authors formulate a distributed task assignment problem to minimize the average service delay and employ MAB (RL) to find the optimal solution. Each vehicle submits a task via V2I, which is inserted into a submission list. The MEC server acts as an agent that either assigns the task to the cloud resource pool or proceeds with processing the task itself. The agents observe the estimated pending delay, for which the task is halted before being processed. The action space is the set of available MEC servers, which are candidates for assigning the newly submitted task. Simulations are based on the traffic simulator SUMO [85] using real world traffic in Chengdu, China. The proposed scheme achieves better performance in terms of Average Service Delay, Average Pending Delay and Ratio of Tolerating Penalty Delay under different processing rates, varying workloads, MEC numbers and increasing computation resources compared to other baselines. The authors also provide an estimation on the computational complexity of the algorithm which is in the order of the number of newly submitted tasks during each scheduling period. One major point that will be investigated further is the effect of packet loss and interference which are not considered in the system model.

In [86], Fan et al. propose a C-V2X network architecture with separated control and data planes. In the control plane, a central SDN controller designs the traffic offloading strategy according to the dynamic network state in order to maximize the throughput of the access points and the vehicles. The SDN controller uses a DNN model that is trained offline on historical traffic offloading strategies regarding the number of users or vehicles associated with the access points and predicts the future traffic offloading strategies. The output of the DNN is utilized in an online search algorithm that optimizes the association between the vehicles and the delay-insensitive users, as well as the cellular APs and the users, in order to reduce its complexity. The evaluation of the framework uses real vehicle traces recorded in Beijing, which are used for constructing the dataset. The proposed scheme improves the network throughput, load balance and user service ratio compared against existing schemes. The joint wireless resource allocation and the SDN computation resources will also be examined as a future direction.

Table 4
ML-based radio network resource allocation in vehicular networks.

Ref.	Scenario	ML method	Objective	Architecture	Conclusion
[74]	High mobility intersection	(Supervised) • DNN	Power optimization for maximizing throughput	Centralized	DNN achieves similar performance to WMMSE, albeit at the cost of training.
[75]	Vehicular network	(Supervised) • CNN • LSTM • DNN	Bandwidth allocation optimization	Centralized	The framework optimizes bandwidth allocation according to network dynamics.
[76]	Cellular network	(DRL)	Power optimization for data rate maximization	Distributed	Increased V2V link capacity and reduced interference among vehicles.
[77]	Inter City highway	(Supervised) • SVM • Decision Tree	VANET Clustering for reducing direct connections to cellular networks	Distributed	By adjusting the granularity parameters related to the clustering, satisfactory results were obtained in relation the average access latency.
[78]	Cellular Vehicular network	(DRL) • DQN	Spectrum and power optimization for minimizing V2V interference	Distributed	Each agent can learn how to satisfy the V2V constraints while minimizing the interference to V2I communications.
[79]	Dense highway	(Supervised) • LSTM	A learning vehicle predicts the packet transmissions of neighboring vehicles	Centralized	An intelligent vehicle can learn and predict the transmit patterns of its neighbors. This knowledge can then be used to orchestrate its own transmissions during periods of low channel activity.
[80]	Urban environment with uniform motion of vehicles	(DRL) • Double dueling deep recurrent Q-networks	Power and sub-channel selection optimization to maximize sum throughput of V2I links considering latency and reliability constraints of V2V links	Distributed	Higher robustness in the packet delivery ratio in V2V links against payload size (90% compared to approximately 73% of NR Mode 2 scheme) and increasing vehicle velocity (approximately 90% for 10-15 m/s to approximately 86% for 25-30 m/s).
[81]	Single-cell system with vehicles and D2D pairs	(RL) Q-learning	Subcarrier and power allocation to maximize energy efficiency	Distributed	The multi-agent Q-learning joint algorithm achieves 36% gain in energy efficiency compared to a disjoint power and subcarrier allocation approach at the expense of more memory.
[82]	2 km road segment with six lanes	(DRL) DDQN	Automatically adjust the message broadcasting rate of a transmitter vehicle	Distributed	The proposed solution reduces redundant data up to 16%, increasing 22% the packet reception rate compared with baselines.
[83]	Suburban setting Located in Modena (Italy)	(Supervised) kNN	Predict future CAM generation times, where each vehicle selectes autonomously the radio resources for message broadcasting	Distributed	The proposed solution achieves a Packet Reception Rate performance close to the ground truth and consistently higher than the allocation algorithm used in LTE, Release 14, Mode 4 [24].

Ke at al. [87] employ DRL and the DDPG (based on a neural network) for offloading tasks on a heterogeneous vehicular environment with MEC. A centralized agent decides how much power and bandwidth should be allocated to a service equipment for task offloading, considering the task's size, the buffer queue lengths, the channel vector and SINR for the uplink transmission of the particular service equipment. The agent receives a reward that contains a trade-off between energy consumption, bandwidth allocation and execution delay. The authors conduct simulations on varying channel state and available bandwidth scenarios, where the proposed solution outperforms other baselines in terms of consumed energy, convergence and cumulative rewards.

In [88], Liu et al. propose a vehicular edge computing network architecture where vehicles provide computation services for UEs and traditional edge servers. The authors employ a DRL-based solution with CNN to find the policies for computation offloading and resource allocation to maximize the network's total utility. A central agent considers the number of available vehicular edge servers (VES), the data rate of the UEs served by the VES and fixed edge servers (FES), the computation resources of the VES assigned to the UEs. The agent decides whether to offload the task on a VES or edge server and the percentage of spectrum and computation resources to be allocated to the UE. The DRL-based method achieves higher utility in terms of computation offloading and maintains lower delay for an increasing number of vehicles. It also requires lower computation resources required by a task, compared to executing tasks locally by other VES or only FES. In addition, the proposed method converges faster to a solution than the traditional Q-learning approach.

In [89], the authors propose a two-stage ML-based Vehicular Orchestrator for task offloading. The first stage consists of a Multi-Layer Perceptron (MLP) that predicts whether the offloaded task to a particular server will be successful or not, based on the WLAN upload/download delay, the short-term load of the servers, the task length, the vehicle density and the average edge utilization. In the second step, a regression model estimates the service time of the offloaded task, based on the task length, the average edge utilization, the WAN and Core network delay. The vehicular edge orchestrator is compared to other baselines (random moving average, game theory-based and MAB-based), considering an increasing number of vehicles from 100 to 1800 with varying speeds, outperforming them in terms of average task failure rates.

Yuan et al. [90] employ DRL to jointly optimize the migration of services over edge servers or other vehicles and the mobility planning of the vehicles. The main objective is minimizing the migration and traffic costs while satisfying service delay requirements. Considering that a single agent is not able to cope with the curse of dimensionality in such a joint optimization approach, the authors formulate the problem as a multi-agent deep reinforcement learning, reducing the state and action spaces. Thus, each service entity is regarded as one agent, which decides on the optimal policy based on a convolutional deep Q-network. Simulations indicate that the proposed method effectively reduces the vehicle's system cost and service delay, under varying number of vehicles or different types of computation tasks.

In [91], the authors propose a hierarchical framework for horizontal and vertical task offloading for software-defined vehicular based fog computing, based on Deep Q-Learning. The objectives include maximizing resource utilization at the fog layer (i.e., RSUs, BSs) and the minimization of the average end-to-end delay of time critical applications. A critical challenge in this architecture is the synchronization of the distributed fog nodes for optimizing the QoS. For this reason, the problem is formulated as a MDP function, solved by RL. A centralized AI-based SDN controller is an agent that explores its environment (state space), consisting of the traffic load probability of a fog node, the next estimated queue state of a fog node and the end-to-end delay of a task and selects the best available node for offloading a particular task. Compared to baselines, the proposed model achieves lower latency, less energy consumption and lower energy shortfall.

In [92], the authors employ "meta-learning" to minimize the consumer's cost when consuming edge node resources and simulate the Manhattan roadmap using Unity 3D Engine [93], incorporating round-abouts, intersections, highways and bridges. The authors define a ML algorithm space comprising various LSTM-based models. The proposed meta-learning framework consists of two stages. On the first stage, a DNN decides, which model should be selected from the ML algorithm space, according to previous experience and meta-features, which are parameters describing the internal representations of a dataset. The selected ML model predicts the edge resource consumption on the second stage. Results highlight that simple LSTM architectures perform better in most scenarios, when comparing to stacked- and dropout-based architectures, which can be attributed to the time dependency of the data. In stacked architectures, wrong predictions are propagated throughout the layers, leading to enlarged errors. Dropout layers do not provide better performance, because erasing randomly hidden units, may lead to erasing important information. In addition, the meta-learning framework consistently makes more economical decisions in all vehicular scenarios compared to "non-meta" methods.

Islam et al. [94] propose an intelligent task scheduler for vehicular edge server to maximize the successful task completion rate, while prioritizing critical tasks (e.g., safety related tasks) over non-critical ones (e.g., infotainment). The authors employ RL and use Extreme Learning Machine (ELM) to approximate the function that maps states into actions. ELM is based on neural networks, the hidden neuron weights of which are assigned randomly, -instead of using gradient descent-, towards faster training. The solution is distributed and each vehicular server includes a local agent that schedules the task assignment. The agents observe vehicular information, including location, speed and direction, along with information regarding the task to be executed (QoS, timestamp). Vehicles move on a unidirectional straight road with speeds of 30-55 km/h. Then, the agent decides whether to forward the task to other servers or rejects the task, if the scheduler predicts that the required QoS cannot be addressed. Simulations show that the proposed solution achieves 96% completion rate, even at high task arrival rates (35tasks/sec) against round robin static scheduling policy (that achieves below 20% completion rate for 35 tasks/sec). ELM is a promising method that can reduce training times, however, it should be compared with neural networks trained with gradient descent in terms of time complexity and task completion rate for a range of road traffic flows, to quantify the differences between these two approaches.

Dinh et al. [95] present a DRL-based solution with DQN to minimize the average task latency in an urban area with gNBs, RSUs and vehicles. The authors follow a distributed approach where vehicles and RSUs are the learning agents of the solution. The vehicles use their own small DQN for computational efficiency and the RSUs use a larger DQN to make task offloading decisions. The vehicles decide whether to offload tasks in RSUs or gNBs, while RSUs decide whether to offload tasks on gNBs or process them locally. The state space of the problem includes parameters related to computational efficiency (e.g., packet size, CPU cycles needed), amount of time to offload jobs and processing related features, such as time needed to complete a task. Results indicate that the distributed DQN solution reduces the average task latency from 9.5% to 68.3% compared to a Multi-armed Bandit Approach and a Constant Offloading Scheme for various scenarios, including increasing task arrival rate, vehicle arrival rate, number of RSUs, and packet size.

Lin et al. [96] employ a contextual Multi-armed Bandit approach for minimizing the total task offloading energy in a two-lane freeway. The freeway consists of RSUs, VUEs and VECs. Each VUE is an agent that interacts with its environment and selects the optimal VEC at each iteration. Since this is a Multi-armed Bandit application, each VEC represents the "arm" that each agent selects. The authors extend the solution by introducing a contextual clustering of the bandits, i.e., the task preferences of neighboring VUEs are included in the problem formulation. The bandits of VUEs belonging to the same clusters, collaborate in the estimation of the maximum reward, since they share similar preferences. This contextual information improves the average success offloading ratio satisfying delay constraints ($\sim$1300 seconds against $\sim$1800 seconds of another context-aware benchmarker and $\sim$6600 seconds against a random offloading policy). Moreover, the proposed solution achieves the highest rewards among all solutions for different popularity coefficients, while it outperforms the benchmarkers in terms of convergence (the proposed solution converges at the 300th iteration, another context-aware benchmarker converges at the 1200th iteration, while a benchmarker that knows the input size of the task size converges at the 2800th iteration).

Lessons learned: Centralized architectures are more prominent than distributed ones, while federated learning approaches were not used in the surveyed papers. Unlike handover and caching management, there are extensive objectives used for formulating a computations recourses management problem, ranging from minimization of service latency to maximization of network utility. Regarding the AI/ML methods preferred, there is not a clear winner between supervised and RL/DRL methods. Both have been used extensively, considering the variety of features that comprise computation resources management problems, as presented in the previous analysis.

All approaches contribute to optimizing the provided edge resources. From [92], it is learned that meta-learning can perform better than non-meta methods in vehicular scenarios, while time dependency of the data may lead to simpler LSTM architectures performing better than more complex ones, like stacked or dropout-based approaches. In addition, the Geoffrey E. Havers (GHE) statistic is an effective measurement for traffic analysis, where a smaller GHE indicates a better regression of observed flows. From [90], multi-agent DRL can reduce the state and action spaces, copying effectively with the curse of dimensionality presented in single agent systems. In [86], [89], DNNs are used to improve throughput, user service ratio and task failure rates.

Table 5
ML-based computation resources management in vehicular networks.

Ref.	Scenario	ML method	Objective	Architecture	Conclusion
[84]	Real world map	(RL) • Multi-armed bandit	Minimize the average service delay	Distributed	Best average service delay under different conditions compared with baselines.
[86]	Real world vehicle traces	(Supervised) • DNN	Maximize the access point's and vehicles' throughput	Centralized	Improvement of network throughput, load balance and user service ratio compared against existing schemes.
[87]	Unidirectional road	(DRL) • DDPG	Minimize total system cost	Centralized	The solution converges faster and achieves higher average cumulative rewards and better average consumed energy compared to baselines.
[88]	Macro BS	(DRL) • CNN	Maximize total utility of network	Centralized	Achieves highest utility and lower delay for increasing number of vehicles compared to baselines.
[89]	Circular route	(Supervised) • MLP (classification) • Linear regression	Task offloading	Centralized	Outperforms other baselines in terms of average task failure rates and QoE.
[90]	Hexagonal grids	(DRL) • CNN	Minimize the migration and traffic costs while satisfying service delay requirements	Centralized training Distributed execution	The proposed method effectively reduces the system cost and the service delay of vehicles.
[91]	Urban IoV	(DRL) • Deep Q-learning	Maximize resource utilization. Minimize the average end-to-end delay of time-critical applications	Centralized	Lower latency and energy consumption compared to baselines.
[92]	Manhattan roadmap	(Supervised) • Meta-learning • LSTM • DNN	Minimize the expenses to consume edge computing resources	Centralized	Most economical decisions in roadmap scenarios compared with fully real time requests.
[94]	Two-lane road	(RL) • Extreme Learning Machine neural network based on Q-learning	Maximize the successful task completion rate	Distributed	The proposed solution achieves 96% accuracy in successful task completion rate. It also achieves completion rate over 96% against round robin scheduling also for high task arrival rates (35 tasks/s).
[95]	Urban area with RSUs and gNBs	(DRL) • DQN	Minimize average task latency	Distributed	DQN solution reduces the average task latency from 9.5% to 68.3% compared to a Multi-armed Bandit Approach and a Constant Offloading Scheme for various scenarios, including increasing task arrival rate, vehicle arrival rate, number of RSUs, and packet size.
[96]	2 km two-lane freeway	(RL) • Contextual Multi-armed Bandit	Minimize the total task offloading energy	Distributed	Outperforms other benchmarkers in terms of convergence (300th iteration against 1200th and 2800th iteration), average success offloading ratio (∼1300 seconds against ∼6600 seconds a random offloading policy).

4.4. Routing

Efficient data dissemination among vehicles is a challenging field of vehicular networks and studies the problem of optimizing multi-hop path and relay selection. The main concern revolves around latency and reliability considering the increasing vehicle density in future networks. Table 6 summarizes the important points of publications that apply AI/ML for routing packets in vehicular networks.

In [97] and [98], the authors employ a two-level clustering-based approach, including fuzzy logic for selecting the cluster-heads and RL select the next-hop/gateway node. In [97], Wu et al. employ clustering to copy the MAC contention problem by reducing the sender nodes' transmission of the concurrent messages. Clustering is based on a fuzzy logic scheme, where the vehicle mobility, channel conditions and vehicle distributions are considered inputs to select the cluster head. In the second stage, they employ a distributed RL approach to select the next-hop node to deal with the degradation of multi-hop TCP transmissions. The network nodes are agents who learn the environment by exchanging messages. Results indicate that the proposed approach effectively maintains packet collisions at a low level and provides improved TCP throughput for an increasing number of flows compared to static approaches, leading to lower packet delay.

In [98], Khan et al. extend this idea by using link reliability status, k-connectivity and relative velocity factor as input to the fuzzy logic scheme. Each vehicle calculates the leadership value for itself and its one-hop neighbors through exchanging messages. In addition, they use an improved Q-learning (IQL) approach for selecting the gateway, where the Q-values are updated only if the best action is available, reducing time and space complexity compared to the classical Q-learning (CQL) approach. The authors show that the proposed IQL algorithm converges faster to a solution compared to CQL (for 400 vehicles, CQL reaches a solution with 18000 iterations, while IQL needs only 6000), and produces a smaller number of request route messages for an increasing number of cluster-heads (for 30 cluster-heads/agents, IQL produces 60 messages and CQL ∼120). In terms of performance, the proposed two-level clustering scheme maintains higher throughput for increasing vehicle speeds and reduces the probability of route change compared to other DSRC-based cluster schemes.

Unlike the previous cluster-based approaches, the authors in [99] do not select a fixed cluster head for forwarding the route messages. Instead, they select the optimal next-hop grid using Q-learning. The agents learn based on the different grids, without including any observations from other vehicles, reducing the convergence speed because of the smaller number of available learning states. After selecting the optimal grid, the agents select the best relay vehicle for forwarding messages from the previously selected grid using a greedy or Markov prediction method. Higher priority in vehicle selection is given to buses, considering that they have fixed routes and schedules, improving the performance of the proposed scheme. An important parameter is the timeslot, during which two vehicles are considered neighbors if they are in each other's transmission range. Simulations highlight the improved performance of the proposed hierarchical routing scheme in terms of delivery ratio and throughput, at the cost of similar or slightly increased delay, hop count and number of times packets are forwarded, compared to alternative position-based routing protocols for different timeslots. The authors also showcase that "bus-aided" schemes increase the throughput and delivery ratio, decrease the hop count and the iterations a packet is forwarded, at the cost of slightly increased delay for different timeslots.

The authors in [100] and [101] employ a two level approach, where a convolutional neural network is applied to images for extracting features and then, they employ RL and DRL to select the next relay node. In [100], the authors apply a CNN to classify satellite images into Buildings, Open fields and Streets and then, each vehicle acts as an agent that learns from the environment using Q-learning and decides upon which vehicle to hop next by minimizing the propagation loss experienced at its receiver. This approach extends the reachability of a V2X link around 66,7% compared to traditional shortest-distance approaches. Similarly in [101], the authors use a CNN to extract features from the traffic and communication topologies maps that act as the state space for a centralized agent to decide on the next virtual relay node, using Deep Q-learning to accommodate a large number of features. This approach is compared to greedy and random decision-making frameworks showing significant improvements in utility performance. However, the proposed scheme assumes that the operator has all the necessary information available.

In [102], the authors propose a model-based RL solution that uses fuzzy logic to model the links between the vehicles. The fuzzy system evaluates the stability and connection quality of the links and outputs a rule, which is used by the agents, i.e., the vehicles, to decide the forwarding destination of the packets. The authors compare their solution with a model free-based approach and conclude that their solution presents higher packet delivery ratio even at high velocities (∼67% against 53% at 80 km/h) and improved transmission delay, because the fuzzy logic model discovers more stable paths (560 ms against 490 ms). Moreover, the model-based solution presents improved packet delivery ratio (82% against ∼70%), control overhead ratio (30% against ∼40%) and transmission delay (310 ms against 450 ms) for 500 simulated vehicles against traditional routing benchmarks.

Luo et al. [103] propose a location-based packet forwarding scheme with Q-learning. The solution includes two steps: i) the RSUs select the optimal road segments at intersections, and ii) the vehicles select other vehicles as optimal relays at the road segments via V2V communication, while at the intersections, the RSUs select the relays, considering the decision in step i). There is a centralized server, i.e., agent, that monitors historical traffic flows on road segments and selects the optimal road segment adjacent to an intersection, where the packets will be forwarded. Then, the Q-table is distributed to the RSUs at the intersections. The vehicles are assumed to know their own positions and the destination vehicle's position by GPS and obtain the road topology via digital maps. The solution also provides a congestion mechanism that selects an alternate path in case the optimal path is congested. Vehicular speeds vary from 40 km/h to 60 km/h. Compared to other solutions, the proposed scheme shows improved average delay for increasing packet sending rates (∼1 sec against ∼1.40 sec and ∼2.50 sec for 6 packet/sec), improved packet delivery ratio for increasing packet sending rate (∼0.66 against ∼0.56 and ∼0.53 at 6 packets/sec), with a slightly increased average hop count (∼12hops against ∼11hops at 6 packets/sec) because of its congestion mechanism.

Kandali et al. [104] employ a routing protocol based on k-Means to locate the optimal cluster head. The authors use the Continuous Hopfield Network to assign the initial set of cluster heads. Then, the number of clusters is fed into k-Means, which assigns the vehicles into the appropriate cluster, considering a link reliability model between the cluster head and the designated node. Results show that the proposed scheme presents higher throughput for increasing number of vehicles compared to other baselines (∼800 Kbps against ∼750Kbps for 200 vehicles), higher throughput for increasing vehicle velocities (∼770Kbps against ∼760Kbps for 100 km/h average velocity), lower end-to-end delay for increased vehicle velocities (∼0.5 sec against ∼3 sec at 100 km/h) and higher packet deliver ratio for increasing number of vehicles (∼97% against 90% for 100 vehicles). However, the packet delivery ratio approaches that of the baselines for 200 or more vehicles.

Lessons learned: Following [97–104], distributed approaches have been used more extensively in routing schemes than centralized ones, since the selection of next hop nodes takes place by the vehicles themselves utilizing V2V. Following this, the most popular optimization objective is selecting the next optimal node for forwarding the packets.

Regarding the AI/ML methods used, RL/DRL methods prevail compared to supervised and unsupervised learning techniques, which are only used as first stages in hierarchical solutions based on RL solutions. From [97],[98], it is learned that two-level clustering schemes can be effective in maintaining satisfactory throughput even in high density scenarios, when used in conjunction with RL. In [99], it is learned that an observation space consisting of the possible grids that the vehicle can follow, can provide improved delivery ratio and throughput, at the cost of similar or slightly higher delay compared to traditional routing methods. Finally, [100],[101], present an alternative approach for optimal relay selection, where features are extracted from traffic topology maps using neural networks to reduce the number of available dimensions and are used as inputs to RL or DRL models for optimal path search. DRL is used again in cases where the number of dimensions is high and cannot be accommodated by traditional Q-learning approaches.

4.5. Beam selection optimization

Millimeter wave (mmWave) is the radio spectrum above 24 GHz and includes the new 5G NR radio bands that can provide higher data volumes than sub-6 GHz bands, but at significantly lower distances. Beam selection is a technique, where multiple antennas in the transmitter form narrow beams and help overcome the poor propagation characteristics of mmWave. However, the RSU must always select the optimal beam pairs to achieve the highest performance. In this context, ML has been used to increase successful beam alignments, improve the system's overall performance and

Table 6
ML-based routing in vehicular networks.

Ref.	Scenario	ML method	Objective	Architecture	Conclusion
[97]	Freeway and street scenario	(Unsupervised) • Clustering for selecting Cluster Head based on Fuzzy Logic (RL) • Q-learning for selecting the next-hop node	Select the route that maximizes network performance	Distributed	Notable improvement of TCP throughput over baselines. More effective in terms of throughput and delay when the number of hops increases.
[98]	Freeway scenario	(Unsupervised) • Clustering (Select Cluster Heads with Fuzzy Logic) (RL) • Q-Learning for gateway selection	Reduce the number of iterations in the gateway selection to LTE Base Station	Distributed	Good throughput in high-density vehicular network scenarios. In high dense topology, there will be many agents and actions for Improved Q-Learning, which will lead to excessive iterations in gateway discovery.
[99]	Real trace data	(RL) • Q-learning for selecting the next optimal grid Greedy/Markov methods for selecting the relay vehicle	Improve the delivery ratio in VANETs	Distributed	Improved delivery ratio and throughput at the cost of similar or slightly increased delay, hop count and number of times packets are forwarded, compared to alternative position-based routing protocols for different timeslots.
[100]	Satellite images from urban environment	(Supervised) • CNN for Satellite image segmentation (RL) • Q-learning for optimal path search	Multi-hopping path selection for lowest propagation loss	Distributed	The proposed method can improve environmental recognition and extend the reachability of multi-hop communications by up to 66.7%.
[101]	Simulated traffic density map	(Supervised) • CNNs for extracting traffic patterns (DRL) • Deep Q Learning with DQN for relay selection	Selection of vUEs as virtual relays for transmission range extension	Centralized	Compared with the greedy and random decision-making schemes, the proposed scheme improves utility performance, transmission rate, and achieved SINR dramatically.
[102]	Simulated city part	(RL) • Model-based RL with Fuzzy Logic	To improve packet delivery ratio, control overhead ratio, transmission delay	Distributed	The model-based solution presents improved packet delivery ratio (82% against ∼70%), control overhead ratio (30% against ∼40%) and transmission delay (310 ms against 450 ms) for 500 simulated vehicles against benchmarks.
[103]	3 km × 3 km urban area with 24 intersections	(RL) • Q-learning	Select the optimal road segment for message forwarding	Centralized training Distributed execution	Improved average delay for increasing packet sending rates (∼1 sec against ∼1.40 sec and ∼2.50 sec for 6 packet/sec), improved packet delivery ratio for increasing packet sending rate (∼0.66 against ∼0.56 and ∼0.53 at 6 packets/sec), with a slightly increased average hop count (∼12 hops against ∼11 hops at 6 packets/sec) because of its congestion mechanism.
[104]	Highway	(Supervised) • k-Means	Select cluster heads based on link reliability	Centralized	Higher throughput for increasing number of vehicles compared to other baselines (∼800 Kbps against ∼750 Kbps for 200 vehicles), higher throughput for increasing vehicle velocities (∼770 Kbps against ∼760 Kbps for 100 km/h average velocity), lower end-to-end delay for increased vehicle velocities (∼0.5 sec against ∼3 sec at 100 km/h) and higher packet deliver ratio for increasing number of vehicles (∼97% against 90% for 100 vehicles).

reduce the complexity of exhaustive beam search schemes, as shown in Table 7.

In [105], the authors apply a contextual MAB algorithm, where a mmBS observes the direction of arrival of the vehicles and selects a subset of the best beams over time to maximize the data successfully received by the bypassing vehicles in the coverage area. A significant advantage of this algorithm is that it does not require accurate location information or statistical information on traffic and environmental changes. Compared with baselines, the proposed solution achieves near-optimal performance in terms of

Table 7
ML-based beam selection in vehicular networks.

Ref.	Scenario	ML method	Objective	Architecture	Conclusion
[105]	HetNet Cellular with macro and mmBS	(RL) • Contextual multi-armed bandit	Maximize the data successfully received by bypassing vehicles	Centralized	The proposed solution achieves near optimal performance on 33 minutes compared to baselines.
[106]	HetNet Cellular with macro and mmBS	(RL) • Multi-armed bandit	Maximize the SNR of vehicles belonging to the same beam broadcasting cluster	Centralized	The proposed scheme increases the amount of aggregate received data by the vehicle. Social preferences provide robustness against blockages even under increasing vehicle arrival rate.
[107]	Two-lane urban street	(Supervised) • Random Forest	Beam selection based on situational awareness	Centralized	Probability of successful alignment is 60% with reduced overhead.
[108]	Two-lane straight street in urban canyon	(Supervised) • Random forest	Beam selection based on situational awareness	Centralized	Random Forest achieves the highest alignment probability (84.16%) compared to RBF-SVM, gradient boosting, and deep feedforward neural network.
[109]	At least 10 VUES TXs randomly distributed in the network.	(Supervised) • SVM	Select the best analog beam that can achieve the highest Average Sum rate (ASR) with low complexity	Centralized	Performance is very close to the theoretical boundary at a very low complexity. Higher average sum rate of the proposed algorithm compared to traditional channel estimation methods Lower computational complexity at increasing V2V link density.
[110]	Small mmWave cell in the UK	(RL) • Contextual Multi-armed Bandit (C-MAB)	Select the beam that maximizes the average vehicle sojourn time within that beam	Centralized	The C-MAB algorithm achieved the highest average beam sojourn time (approximately 22 sec), outperforming the classic MAB by 40%. Meanwhile, the random and best SNR schemes achieved approximately 11 sec of average beam sojourn time.
[111]	mmBS with blockages at an urban area	(DRL) • DQN	Maximize total capacity	Centralized	The solution achieves connection probabilities and capacity levels close to the maximum capacity scheme, outperforming the random connection direct link connection schemes.
[112]	250 m road with one mmBS	(RL) • Contextual Multi-armed Bandit	Maximize received amount of data for all vehicles	Centralized	Compared to risk-free schemes, the risk aware solution reduces the percentage of vehicles that do not receive enough data from 3.5% to 2.2% at the expense of a performance penalty of 9% at worst, presenting high adaptation behavior in sudden statistical changes.

throughput for different arrival rates (for vehicles with velocities from 20 km/h to 70 km/h), learning autonomously from its environment on average 33 minutes. However, the simulation setup assumes that only one vehicle communicates with the mmBS, even though multiple vehicles may exist in the coverage area. The work can be extended to study this effect as well.

Similarly, the authors in [106] employ a contextual MAB to select appropriate beams and beam angles by maximizing the SINR of the vehicles of the same broadcasting beam cluster. In this case, the state space includes the direction of arrival and the social preferences of the vehicles' passengers, expressed through a preference factor for movies. The proposed algorithm is compared to an optimal algorithm and an algorithm that does not consider social preference in its features (CML). The proposed scheme provides an approximately 9.6% increase in the vehicle's aggregate received data, transmitting 13 Gb more data than CML. In addition, social property provides robustness against blockages, while the under increasing vehicle arrival rate, the performance of CSML is improved by 13,2% on average.

Following a different approach, Wang et al. in [107], [108] select the optimal beam pair index with Random Forest model, by considering the RSRP of the beams and the vehicle's location (including errors from localization). The authors employ a top-K classification scheme, where the labels for classification are the top-K beam indices with the highest RSRP. In [70], the probability of successfully identifying the optimal beam is 60%, while in [71], the model achieves 84.16%, highlighting that the additional information of vehicle's locations increases the situational awareness of the solution. In addition, they showcase that an auto-terminated beam search achieves comparable performance compared to the exhaustive search over the recommended beam sets, regarding successful alignment. The authors reference online learning as a promising approach to increase training accuracy, while they plan to study the use of temporally-correlated data.

In [109], the authors apply an SVM model to select the best analog beam to achieve the highest Average Sum rate (ASR) with low complexity. They consider transmitter's output power, path loss, azimuth angles of Angle of Arrival (AoA) and Angle of Direction (AoD) and channel gain as input features to the model and the output is the best analog beam. The authors model the large number of VUEs with the random distribution of heterogeneous Poison point process and verify that the proposed scheme's performance is very close to the theoretical boundary at a very low complexity. The average sum rate of the proposed algorithm is higher than that of traditional channel estimation methods while its computational complexity remains lower at an increasing V2V link density. The authors highlight that they plan to study the effect of the vehicle's speed into the problem formulation.

In [110], Kose et al. apply a Contextual Multi-Armed Bandit (C-MAB) that considers the vehicle's contextual information, including vehicle orientation with respect to the small cell (i.e., north-east, north-west, south-west, south-east) and the vehicle's range from the small cell (i.e., near, middle, far) to select a beam-vehicle pair that maximizes average vehicle sojourn time within that beam. The authors note that they did not include vehicle speed in the features, because its impact on the performance is low, given the limited speed range in the city area (30-50 km/h). The algorithm is compared to a classic MAB (that does not consider the vehicle's context), a random scheme and a best SNR method, where the base station selects the vehicle with the highest SNR. The C-MAB algorithm achieves the highest average beam sojourn time (approximately 22 sec), outperforming the classic MAB by 40%. Meanwhile, the random and best SNR schemes achieve approximately 11 sec of average beam sojourn time. In addition, the C-MAB is not affected by the street layout as the classic MAB does, which eventually selects specific beams affected by the street layout and the vehicle's mobility. C-MAB utilizes the vehicle's context differentiating the vehicle's orientation resulting from the street layout.

Ju et al. [111] employ a DRL-based beam allocation solution to maximize capacity in mmWave networks. The mmBS acts as an agent that learns the optimal policy with DQN, retrieving information on beams, potential relays, target vehicles and blockage conditions. The agents select the optimal link (direct or relay), potential relay vehicles and optional beams. The proposed solution achieves connection probabilities very close to the maximum capacity scheme under different capacity thresholds (80%) and outperforms the random selection (30%) and direct connection schemes (0%) at the highest capacity threshold. Moreover, the total capacity of vehicles is achieved with the DRL solution for different communication blocking levels, reaching almost always the optimal performance.

Wirth et al. [112] propose a risk-aware contextual Multi-armed Bandit approach for maximizing the received amount of data by vehicles in mmWave networks. The mmBS selects the best beams based on the vehicles located in its coverage area, considering the risk that some vehicles cannot fulfill their communication requirements. Risk is quantified through statistical metrics, such as the conditional value at risk (the higher this value is, the smaller the risk is for receiving few data) and the mean variance (the higher the variance, the higher the risk for receiving few data). The agents select a set of beams (representing the arms of the Multi-armed Bandit problem) in a 250 m long road with vehicle speeds from 30 km/h to 70 km/h. Compared to risk-free schemes, the risk aware solution reduces the percentage of vehicles that do not receive enough data from 3.5% to 2.2% at the expense of a performance penalty of 9% at worst, since it avoids beams with potentially poor performance. The solution also presents high adaptation behavior in sudden statistical changes of the underlying conditions, because it is able to converge back to acceptable performance in 25% of the time steps needed compared to a non-adaptive solution.

Lessons learned: We can summarize a few points regarding beam selection in vehicular networks, based on Table 7. Beam selection optimization solutions use mainly centralized architectures, directly resulting from the problem's nature. A central entity collects data from the vehicular users and the networks and decides on the most optimal beam for a specific cell. The objectives used for optimizing the decisions include various metrics, such as maximizing the SNR or the sum rate of users. In addition, RL/DRL and supervised learning techniques are equally used in formulating these scenarios, either as decision-making or classification problems. Unsupervised learning was not used in the surveyed papers, but it would have been interesting to investigate whether the clustering of vehicles would have contributed to optimizing specific network metrics, as shown in caching and handover management problems.

Therefore, all approaches have similarly contributed to optimizing beam selection strategies. It is learned from [105] that online learning can be used in conjunction with MABs to maximize the received data, while from [106], cross-layer data, such as the passengers' social preferences and the direction of the vehicle's arrival, can provide additional contextual information and increase the vehicle's received data in the presence of blockages and under increasing vehicle rate. Following [107],[108], Random Forests have shown the highest probability of beam alignment compared with other supervised learning algorithms, while the vehicle's location can increase the situational awareness of the solution. Finally, in [109], an SVM classifier can help select the best analog beam, but the model's training should take place offline because of the high computational complexity. However, the prediction phase of the trained model presents lower complexity than traditional channel estimation methods.

4.6. QoS prediction

In [113], the authors use supervised learning techniques to predict the end-to-end (E2E) delay over three commercial LTE networks by conducting a vehicle measurement campaign. Then, they use the expected E2E delay, the vehicle's speed, SINR, RSRP and RSSI of the recorded data to train an ML model that classifies the delay below or above a threshold of 50 ms and 100 ms. The authors compare four ML models, namely an MLP, a Recurrent Neural Network, Random Forest and SVM where the MLP and RNN achieve the highest performance in classifying the delay. Considering that simple MLPs present smaller complexity than RNNs, it is shown that MLPs are suitable for such application. The authors conclude that additional information coming from the UE and the network could increase the model's performance regarding QoS prediction.

Moreira et al. [114] use supervised learning to classify whether a packet that will be transmitted can or cannot be delivered within a required latency window from the base station to a vehicle. The authors use multiple input features to train the ML models, including previous recorded delays, vehicle's coordinates, best cell index, radio measurements (RSRP, RSRQ, SINR, CQI), BLER, average cell throughput. The ML models are a multilayer perceptron (MLP), Logistic Regression and Random Forest which are compared to the traditional ARIMA filter. Simulations under different network workloads and packet sizes highlight that supervised ML extracts reliable results when the class imbalance is handled, which must be considered a step in the pre-processing pipeline. When the class imbalance is handled, the ML models can predict the successful delivery a few seconds before.

Zhang et al. [115] propose a latency prediction framework using data recorded over four months in an urban scenario. The authors observed that data presented fluctuations, which led them to decompose the latency in two separate components, i) the baseline, which has a trackable trend over time, and ii) the residual, which behaves like random noise. For the baseline, the authors used k-medoids to separate the data into k-clusters, according to their LTE signal strength, RSRP and RSRQ. Then, they applied k independent LSTM networks to predict the latency of each cluster. The proposed scheme presented the lowest RMSE compared to non-ML methods, like long- and short-term sampling.

In [116], the authors develop an LSTM-based scheme that predicts the QoS that will be available in a specified time horizon to a tele-operated driving vehicle. The LSTM autoencoder is trained on vehicle and network related data, including vehicle's location and speed, distance from the network cell, cell load percentage, number of connected vehicles in the cell, and the uplink throughput of the vehicle. After the data are pre-processed, they are forwarded into an LSTM encoder that encodes them into a context vector. Then, an LSTM decoder reconstructs the context vector and produces the predicted uplink throughput of the vehicle for multiple timesteps into the future. Results show that the proposed scheme predicts the uplink throughput for the next 7 seconds with satisfactory performance.

Mendoza et al. [117] present a QoS prediction solution for activating/deactivating Dual Connectivity in automatic guided vehicles, depending on their QoS requirements. The solution employs a k-Means algorithm for clustering the input data, i.e., uplink physical resource blocks, SINR and modulation. Then, the clustered data are fed into dedicated logistic regression models (one for each cluster) for classification in specific QoS categories. The proposed solution reduces QoS outages to 31% in single-frequency networks and 11% in dual frequency networks, compared to two naive predictor baselines, while being more robust to changing radio and interference conditions.

Kousaridas et al. [118] propose a QoS prediction scheme with Random Forests in a Tele-Operated Driving (ToD) Use case over a 5G network. QoS prediction takes place at the NWDAF of the 5G core network and includes two phases: i) the offline training phase, where data are collected from various sources, and ii) online inference phase, where the trained model provides a prediction on the UL throughput for a specific predicted horizon. Input training features include the location of the teleoperated vehicle, its distance from the base station, the number of vehicles connected to each cell, the data rate demands of non-tele-operated vehicles, and the reciprocal of the sum of distances between the tele-operated vehicle's serving cell and the non-tele-operated vehicles attached to neighboring cells. The prediction accuracy of the Random Forests model depends on the features used for its training. The authors highlight that using features related both to the ToD service and the network, results in lower mean absolute error (45 kbps), standard deviation of the absolute error (21 kbps), and mean absolute percentage error (0.0005) compared to training with network-only features (607kbps, 1637 kbps, 0.070, respectively) or ToD-only features (1528 kbps, 2053 kbps, 0.140). Moreover, the authors evaluate the model's performance, when trained with imperfect estimates of the input features (using ARIMA) for a specific time horizon. The predicted UL throughput with imperfect input estimates follows closely the case with the ideal estimates for the first three seconds of the prediction horizon, after which the error increases. Finally, the authors evaluate the Random Forests model in an unknown scenario, where there are 80 NToD vehicles in the simulation, which were not part of the training dataset. In this case, the UL throughput predictions present high errors after the fourth seconds of the prediction horizon, due to increased uncertainty introduced int he scenario (Predicted throughput: $\sim$1.6x10^7 bps, Actual throughput $\sim$1.9x10^7 bps at 6 seconds prediction horizon). The authors conclude that proper identification of input features and prediction horizon are needed and advanced ML-related techniques, e.g., online learning, need to be investigated, that can deal with the high dynamics of V2X environments.

Lessons learned: Following [113–118], supervised learning techniques prevail in QoS prediction in vehicular environments. This is a direct result of the QoS prediction problem, where the prediction of certain network metrics is needed to assess forthcoming QoS of deployed services (using LSTM) or classify whether packets delay will violate imposed thresholds. Therefore, RL/DRL methods are not preferred since regression and classification procedures are more appropriate for this kind of problems. From an architecture perspective, centralized architectures are preferred compared to distributed ones, which is a direct result of the problem formulation.

While QoS prediction seems not as popular as previous network operations, we can extract some specific points regarding the contributions of the AI/ML approaches. For example, it is learned that LSTM based neural networks are suitable for QoS prediction, because the features are mostly time dependent, while supervised learning needs handling of the potential imbalance that may exist in the data to produce accurate predictions [114].

5. Discussion and open issues

This section reports observations based on the surveyed papers and discusses the most critical aspects regarding the application of AI/ML in automotive use cases, including the selection of the ML algorithms depending on the task, the time complexity and the impact on the training and response times, the features used for training ML models in vehicular networks, the data collection phase and the selected architecture (centralized, distributed, federated).

5.1. Learning types in vehicular networks

The selection of the appropriate ML algorithm depends on the problem formulation, i.e., whether it refers to regression, classification, decision making or clustering. Different approaches can be used in the context of the same network operation. In QoS prediction, the authors in [113] train an MLP, a RNN, Random Forest and SVM to classify whether a packet delay is below or above a specified threshold, while in [115], the authors formulate the QoS problem as time series forecasting/regression and apply LSTM networks to predict the latency of vehicle clusters. The same analysis can be applied to the other network operations we examined in this survey.

The supervised learning algorithms utilized in the surveyed papers include traditional approaches, such as kNN, Decision Trees, SVM, and neural networks. Tables 2–8 show that neural networks are more popular in vehicular environments than traditional approaches. This can be attributed to two reasons: *i*) the inherent characteristics of the neural networks architecture, that makes them appropriate for GPU acceleration to reduce training speed, and *ii*) the fact that neural networks are capable of building better data representations when the amount of data and features increases. We describe in detail the time complexity aspects in Section 5.2 – *Time Complexity*.

On the other hand, supervised learning presents some difficulties that need to be addressed before deploying ML models in production. First, most supervised ML algorithms need data that have been pre-processed prior to training. Pre-processing procedures include missing values management, feature transformation, normalization and standardization, in order to avoid problems due to different scaling between the features of a dataset. Xin et al. [119] highlighted that pre-processing accounts for almost 60% of the ML production pipeline, while training only for 20%. In addition, Moreira et al. [114] showcased that class imbalance can affect the accuracy of ML models in QoS Prediction for V2X, if not addressed. Therefore, going beyond the simulation environment, all these pre-processing operations must be considered before deploying ML models in production environments, and especially dynamic vehicular ones. Secondly, labeling datasets and producing high quality annotated data are time consuming, error prone and costly activities. To this end, self-supervised learning has emerged, where ML models are trained to predict the label of datasets, based on a part of the same dataset [120]. An additional technique is Few-Shot Learning, which rapidly generalizes to new tasks using prior knowledge based on a few samples in a supervised manner [121]. This technique was not considered in any of the surveyed papers.

Table 8
ML-based QoS prediction in vehicular networks.

Ref.	Scenario	ML method	Objective	Architecture	Conclusion
[113]	Highway	(Supervised) • MLP • RNN • Random Forest • SVM	Classify the E2E packet delay between specific thresholds	Distributed (although only one vehicle is used, the concept can be extended to multiple vehicles)	MLP achieved f1-scores of up to 88%. Better performance than Random Forest and SVM.
[114]	Manhattan grid urban scenario	(Supervised) • Logistic regression • Multilayer perceptron • Random Forest	Classify whether a packet can be delivered within a latency requirement	Centralized	For prediction accuracy for varying packet sizes, RF performs better, with MLP being close to it, while the ARIMA filter has the lowest accuracy. ARIMA model performs better for the "late" label (the case in which the packet delay exceeds the latency requirement).
[115]	Fixed-location scenarios (office and apartment) Mobile scenarios	(Unsupervised) • k-Medoids (latency clustering) (Supervised) • LSTM	Delay prediction	Centralized	Combination of LSTM and statistical approaches can result in low prediction errors for end-to-end latency.
[116]	Manhattan grid urban scenario	(Supervised) • LSTM	Uplink & Downlink latency and throughput prediction for Teleoperated Driving use case	Centralized	The use of LSTM for predicting metrics such as latency/throughput for relatively short time windows is a promising solution. Increasing the prediction horizon for more than a few seconds may result in high prediction accuracy errors.
[117]	2 km x 1 km area with macro and small cells	(Unsupervised) • k-Means (Supervised) • Logistic Regression	Predict QoS user requirements to activate/deactivate dual connectivity	Centralized	The proposed solution reduces QoS outages to 31% in single-frequency networks and 11% in dual frequency networks, compared to two naive predictor baselines.
[118]	Urban Grid Road with three macro sites	Supervised • Random Forests	Predict UL throughput in Tele-Operated Driving use case	Centralized	Using features related both to the ToD service and the network, results in lower mean absolute error (45 kbps), standard deviation of the absolute error (21 kbps), and mean absolute percentage error (0.0005) compared to training with network-only features (607kbps, 1637 kbps, 0.070, respectively) or ToD-only features (1528 kbps, 2053 kbps, 0.140). Prediction accuracy decreases as the prediction horizon increases in unknown environments.

RL is used when the problem formulation includes decision making. One or multiple agents explore their observation environment and select a suitable action according to a reward function. RL and DRL have been used extensively in V2X for handovers, beam selection, caching, physical and computations resources management, and routing. This can be attributed to the following: *i*) the reward function can be formulated to include a wide range of KPI metrics and constraints in line with various scenarios, such as maximization of mobile operators' revenues or minimization of migration and traffic costs, *ii*) the agent learns directly from its environment, so the data are available immediately to it without the need to conduct labeling, and *iii*) RL can be used in multi-agent distributed systems, reducing the state and action spaces and avoiding the curse of dimensionality that can occur in single agent systems [90].

However, RL presents some weak points that need to be addressed. First, the agent needs time until it reaches the state for selecting the optimal action, called convergence time. This can be time-consuming and may render an RL approach unsuitable for network operations that support automotive use cases with stringent latency constraints. Second, traditional tabular approaches in solving the Q-learning problem have presented improved performance in relatively small state spaces with a few features. However, when the state and action spaces increase, the convergence time of these algorithms increases as well [76], [78], [101]. In these cases, DRL is more appropriate, because it relies on neural networks for the function approximation and provides better data representations. DRL can benefit from GPU acceleration, as we discussed in supervised learning.

Finally, unsupervised learning has also been used in V2X, although to a smaller extent. Clustering is used as the first stage in hierarchical approaches, in order to reduce the amount of data that need to be processed by subsequent steps involving ML [66], [97], [98], [115]. In addition, autoencoders is a family of neural networks that do not need labeled datasets, since they are trained using the input features as labels. Autoencoders have been used in QoS Prediction [116] and Caching [68] vehicular applications for identifying latent data representations.

5.2. Time complexity

Due to the time-critical services that numerous V2X use cases involve, the time complexity of ML-based solutions must be carefully considered in dynamic vehicular environments to assess the feasibility and viability of the implementation. The time cost of AI/ML algorithms can be distinguished based on the training and the response time [7], [122]. The training time refers to the time it takes to train an ML algorithm, while the response time is defined as the time it takes a trained model to make a prediction. The training time can be computationally high when neural networks are used either in DRL or on their own. To mitigate this issue, the authors in [74], [76], [78], [79], [88] conduct training offline and make the prediction online. However, when training a model offline, the varying network, mobility and environmental conditions may be changing at rates much higher than the training time, and thus, the model may learn patterns that are not true after some time, leading to unsuitable predictions [122]. In such cases, online learning schemes are more appropriate.

The training/convergence time can be reduced with various approaches. Qi et al. [59] employed an online learning scheme with transfer learning for proactive handovers and showed that it is more effective compared to offline schemes in reducing the convergence speed of the ML model when the users' velocities changed. Khan et al. [98] used an improved Q-learning approach to select the gateway in a V2V routing application, where the Q-values were updated only if the best action was available, reducing the convergence speed, compared to the traditional Q-learning approach. Sim et al. [105] proposed an online MAB algorithm where mmWave base stations identify the best beam for V2I links using only the vehicle's direction of arrival. Results showed that the model was able to converge to the near-optimal state in 33 minutes on average (ranging between 7 minutes and 75 minutes depending on the state conditions), which is considered a satisfactory value compared to war-driving tests that need more than 75 minutes[105]. ELM [123] is another approach that assigns random weights in the hidden layers of feedforward neural networks leading to reduced training speed with excellent performance and was considered only in one of the surveyed papers [94]. ELMs have been considered in cache management in general wireless networks [122]. In addition, studying the model's generalization is also critical to deal with cases when the distribution of the testing data is different than that of the training data, which would be expected in highly dynamic vehicular environments in urban areas. Therefore, the training time is significant to report when applying ML algorithms in order to fully capture the dynamic changes of the vehicular environment.

The response time is also critical and is bound by the requirements of automotive use cases. ML algorithms with high response time cannot be applied in time-critical scenarios. Use cases, such as emergency trajectory alignment, impose an end-to-end latency requirement of 3 ms, *i.e*., the latency between the V2X application server and the vehicle [27]. This means that any underlying network operation that supports critical automotive use cases, e.g., handover, must preserve the connectivity of V2V/V2I links to support the imposed requirements.

Yan et al. [58] employed kNN to conduct proactive handovers in intersections, achieving a handover decision duration of 2.40 ms. The authors noted that this is lower than conventional beam training schemes which reach 40.96 ms. Ye et al. [78] proposed a DRL-based solution with deep neural networks that selects the optimal power level for transmission considering latency constraint of 100 ms. They reported 0.240 ms as the response time it takes to select a power level, running on a GPU. Xiang et al. [80] reported 0.670 msec for each agent's action selection on a CPU. Neural networks running on GPUs can provide response time in the order of milliseconds [124]. Therefore, these results show that there is potential for deep neural networks in optimizing V2X communications for critical automotive use cases.

5.3. Features to train ML models in vehicular networks

The features used in each paper to train the ML models were reported above in detail. These features can be categorized in seven distinct categories based on the source, from which they are retrieved, namely: Vehicle (e.g., position, velocity, degree of arrival), User (e.g., age, movies preferences), Content (e.g., content popularity, content size), RSU (e.g., availability, PLMN congestion status), Radio Access (e.g., RSRP, RSSI, SINR), Edge Servers (e.g., required computation resources) and Road (e.g., roadmaps).

Following the surveyed papers, the required features mainly depend on the assigned task. ML algorithms used in handovers are trained using mainly Vehicle-, RSU- and Radio-related features, such as vehicle's location, vehicle's throughput and the RSUs/vehicle's service beam indices or RSSI. Caching uses mainly Content-, User- and Edge-related information, including content popularity, user contextual information (age, gender, zip code), availability of cache units, rate of arrival of critical and non-critical data, and availability of resources in edge servers. Computation resources management tasks tend to use features from the Edge and Vehicle layers, such as buffer queues' length, queuing delay, assigned computation resources to vehicles and computation capabilities (i.e., CPU cycles). On the other hand, physical resources allocation relies more on Radio-related features like V2V and V2I links channel power gains and PLMN congestion status. Beam selection scenarios use Radio and Vehicle features, including location, transmission power, beam pairs indices and in one case, Content-specific information. Routing relies on Road information (grid roadmaps, satellite images, traffic topology), and vehicle location. Finally, QoS prediction uses Radio (RSRP, RSRQ, SINR, HARQ), Vehicle information and network delay.

In almost all surveyed papers, features come from multiple categories in order to improve the model's performance. Li et al. [106] verified that using cross-layer information, such as requested user content and channel state information, improves the model's performance (and the beam broadcasting strategy overall) compared to using information from the radio access layer only. On the other hand, the number of features impacts the time and space complexity of the algorithm and the training and response times, so this aspect should always be considered when performing feature selection.

5.4. Training datasets

The availability of training datasets is a significant aspect of modeling vehicular networks and extracting performance results. In the surveyed papers, there are three major pillars which are used to create a vehicular simulation environment: *i*) the road network, *ii*) the vehicle traffic flows, and *iii*) the communication model between vehicles-vehicles (V2V) and vehicles-RSUs (V2I).

The production of vehicle traffic flows takes place with road traffic generators, such as SUMO [85] and iTETRIS [125]. Regarding road network modeling, the same traffic generators can generate generic road scenarios, such as highways. Several papers [53], [64], [66], [67], [77], [84], [90], [92], [99], [105] also use real-word roadmaps with vehicle Global Positioning System (GPS) data that are imported in road traffic generators, to produce the vehicle traffic flows or they are used to calibrate traffic models, such as the Macroscopic Bureau of Public Roads function (MBPR) [90] and the Gazis-Herman-Rothery (GHR) car-following model [77]. Using real-world data leads to more representative simulation environments and more accurate performance analysis of ML algorithms under different traffic and road conditions. However, such real-world vehicle traces may include irregularities and special characteristics that need to be examined before using them.

Celes et al. [126] defined five quality criteria to assess existing real-world vehicle trace data: *i*) sampling granularity variation, because vehicles' positioning is made at different times and the sampling rate introduces gaps on vehicles' trajectories, *ii*) positioning errors, due to sensors, tunnels, urban canyons that may need to be filtered in advance, *iii*) variability, where vehicular environments are characterized by restricted mobility patterns, but exhibit different stop-and-go patterns (e.g., buses, taxis) that impact the network topology, *iv*) volume of mobility data, referring number of vehicles, trajectories and trace duration, and *v*) spatiotemporal observation window, considering that mobility patterns change at different time intervals and can result from the city's particularities, which could be used in designing context-aware solutions. Based on this, real vehicle traces need to be pre-processed accordingly before being used to simulations.

Regarding the communication model, the environment consists of RSUs, vehicles with transmitters/receivers, which can interact via cellular (LTE, 5G) or IEEE-802.11p (DSRC) and channel fading models. The simulation of the communication models takes place with NS-2 [127], NS-3 [128], OMNET++ [129] and MATLAB [130]. The channel fading models follow 3GPP technical and reports, including TR 36.814 [131], TR 37.885 [132], or they use traditional Nakagami [133] and Rayleigh [134] channel models. The training datasets are generated through these three sources and they are used in ML packages, such as TensorFlow [135], Keras [136] and MATLAB for training and testing the ML algorithms. It is imperative to create more realistic synthetic data using the tools above while assessing the quality of the datasets regarding the proper representation of actual mobility patterns.

5.5. Data sampling granularity

In the surveyed papers, the features for training the ML algorithms are sampled in fixed time intervals. The authors focus on selecting some of the system features in order to train their models, but do not investigate how the granularity of the retrieved data may affect the model's performance. A vehicular environment undergoes dynamic changes in mobility patterns not only per day, but throughout the year, so this aspect may create the need to adjust the sampling intervals of the data. Adaptive data collection and storage is a logical step considering the massive amounts of data that are generated in multi-layer environments. Such intelligent strategies can also be used in vehicular environments according to contextual and environmental changes to optimize the network transmission overhead and available computation resources, while they could widen the prediction window whenever needed. A promising direction is investigating the impact of adaptive data collection on the performance of underlying ML algorithms used in network operations.

5.6. Vehicle velocity and positioning requirements

Vehicular environments include vehicle's velocity and positioning as two key parameters of the system model in the surveyed papers. Velocity is modeled as an average value with a typical deviation varying according to the road scenario, *i.e.*, whether it includes a highway or an urban setting. Typical values for urban settings vary between 30 km/h to 70 km/h [55], [64], [89], [97], [99], [105], [114], [116], while in freeways the velocities range from 20 km/h to 160 km/h [54], [58], [59], [79], [97], [108], [137]. Velocity values should be considered in light of the underlying automotive use case that the V2X solution wants to enable. 5GAA has defined a set of automotive use cases along with their Service Level Requirements including positioning accuracy and velocity [138], [139]. Velocity is defined as "the maximum absolute speed of a vehicle at which a defined QoS can be achieved" and that "there may be a need to capture the peak expected speed" [139]. In light of this, simulation models need to also consider the designated expected peak velocity in their simulation models depending on the underlying automotive use case and the road scenario. Typical velocity limits are 50 km/h in cities, 100 km/h in rural areas and 180 km/h in highways. However, there are some exceptions, such as the case of hazardous location warning in highways, where the peak velocity can go up to 250 km/h [139]. In summary, the velocities used for V2X simulation should address the requirement imposed by the underlying automotive use cases.

Similarly, positioning accuracy is another parameter of vehicular environment's modeling. City scenarios impose positioning accuracy down to 10 cm [138], [139]. This means that the localization of vehicles by the network must come with a great degree of accuracy. GPS receivers provide high accuracy, but not at the expected level of some critical use cases, as urban settings can affect their performance. GPS in smartphones is accurate to within 4.9 m under open sky, but its accuracy worsens near buildings, bridges and trees [140]. When complemented by dedicated base stations, the Global Navigation Satellite System (GNSS) can also provide accurate measurements to the centimeter level, but fail to work in dense urban environments under tree canopies and when blocked by buildings [141]. A promising solution to this issue is using mmWave communication that can provide accurate positioning in dense urban deployments and complement GNSS in cases where it fails to operate. 5G positioning along with GNSS and other sensors on top of autonomous vehicles can solve the issue of accurate positioning in V2X scenarios.

5.7. Federated, centralized & distributed architecture

The architecture largely depends on the designated network operation in question. Centralized architectures are used mostly in computations' resource management [86–92] and beam selection [105–109], where a central server receives all information from the radio, vehicle and edge layers and decides how to offload tasks to maximize the network's utility or to select the best beam for a vehicle-RSU pair. A disadvantage of centralized architectures is the high network transmission overhead that comes with retrieving data from multiple entities, which can be costly and complex in real-world environments.

Distributed architectures are more appropriate for routing applications [97–100], where vehicles participate in selecting the next hop nodes by exchanging messages and information with each other over V2V. Similarly, increasing the V2V links' capacity, a physical-resource allocation problem, can be easily formulated as a distributed application. However, distributed approaches have the disadvantage of strict synchronization between the different entities involved. An interesting variation was presented in [90], where the training took place in a central node and the execution was distributed.

FL has also been used in handover [59] and caching [68] to cope with the limited storage capacity of the vehicles and protect the privacy of the users when caching content. It has been effective in terms of performance, but a disadvantage is that synchronous FL is slow for the highly dynamic vehicular environments [68]. As the authors note, asynchronous FL can be applied to solve this issue and preserve the advantages that come along with it. Considering that FL addresses privacy issues, limited storage capacity of users, as well as the increased network overhead of centralized approaches, it is anticipated that more V2X applications will adopt it.

6. Research directions

AI/ML applications in next-generation networks have gained considerable attention by the research community and relevant industry stakeholders. Especially for B5G and 6G networks, the use of such algorithms in network planning, network diagnostics and network optimization and control is actively investigated [142]. Following our discussion in Section 5 on the implications of AI/ML in V2X communications, we present a set of key points and chal-

Table 9
Research directions on AI/ML in V2X communications.

Research Directions	Key Points and Challenges
Intelligent data acquisition and storage	• Big data require excessive amounts of storage • Adaptive acquisition and storage depending on network and road traffic conditions • Maintain the robustness of underlying algorithms regardless of the varying sampling granularity
Explanation of AI/ML-based decisions	• "Black box" models, like deep neural networks, do not provide direct explanations on how they reached a particular decision • V2X environments are highly dynamic, so decisions should not be outdated, following the current road traffic and network conditions • Evaluation of selected decisions based on feedback from the environment to increase the trustworthiness of the model for critical decision making
Context Awareness	• Contextual information on users' habits can be readily available in the network to optimize network and control procedures (*e.g.*, NWDAF in cellular networks) • Reduce prediction effort by integrating contextual information, as well as destination and path information about the vehicle, into the network's analytics functions
Decentralized and Collaborative AI	• Centralized approaches are not efficient for big data due to increased network overhead • V2X vehicles belonging to different mobile operators will need to exchange data • Federated Learning introduction in 3GPP Rel. 17 [144] for model sharing between NWDAF instances distributed in different areas
Energy Efficient AI/ML algorithms	• Increased use of AI/ML on connected low-power devices • Benchmarking AI/ML algorithms for energy efficiency • Identify computationally intensive algorithm parts and reduce overall energy consumption while maintaining performance

lenges and subsequent research directions for next-generation networks. Table 9 summarizes these open areas.

Intelligent data acquisition and storage refer to an adaptive strategy regarding data collection and storage based on changing network and road traffic conditions. All papers surveyed in our work used fixed sampling intervals for acquiring their data. On real-world deployments, however, network and road traffic congestion levels vary on a daily basis. Considering the vast amount of data of large-scale V2X deployments, collecting, processing, and storing these volumes results in costly operations. In this context, the intelligent adaptation of the sampling granularity based on current conditions could provide the means to lower operational costs. Another challenge is to maintain the robustness of the underlying AI/ML algorithms at acceptable levels under varying sampling granularities.

Explaining the decisions of AI/ML models is another active research area with social implications. Deep neural networks constitute the so-called "black box models", because the user cannot directly explain how the model reached a decision about a specific data point. Therefore, providing meaningful explanations to users (*e.g.*, Mobile Network Operators, Service Providers, drivers) in a comprehensible manner is imperative to increase trustworthiness about the underlying algorithms' decision-making capabilities. Considering also that vehicular environments are highly dynamic with sudden changes daily, the decisions made by the AI/ML models should reflect current network and road traffic conditions to ensure that they are not outdated. There are several approaches for extracting explanations from a deep neural network, such as gradient-based methods and layer-wise relevance propagation. A comprehensive review and taxonomy are provided in [143].

Another point worth considering is that contextual information regarding the users' trajectory patterns and preferences can be integrated into the network without the need for an increased complexity to predict their actions. For example, autonomous vehicles will follow specific trajectories based on the users' habits. In cellular networks, the NWDAF can readily include this information in their analytics to optimize specific network and control procedures without additional prediction effort. Moreover, information about the destination and the path of a vehicle available to GPS navigational systems could also be used to simplify the state space of the problems.

Traditional centralized training architecture presents shortcomings, such as increased network overhead due to raw data transfer, especially in large-scale V2X deployments. Distributed and federated learning approaches have emerged that can reduce this overhead. It is anticipated that such decentralized approaches will increase in the future due to several reasons: *i*) local AI/ML deployments on the vehicles themselves, *ii*) collaborative model/data sharing between vehicles and RSUs belonging to different Mobile Network Operators, *iii*) user's privacy, and *iv*) network congestion reduction. To highlight the trend towards decentralization, we refer to 3GPP's introduction of Federated learning for model sharing between multiple NWDAF instances [144]. In this use case, 3GPP considers a hierarchical NWDAF deployment in a single PLMN, where transferring all raw data centrally is inefficient. By keeping one NWDAF in a central location and distributing multiple NWDAF instances locally -possibly co-located with other 5G Core Network Functions- only model sharing takes place, while raw data are not exposed.

Finally, we reference energy-efficient AI/ML algorithms. Recently, the number of connected devices has increased considerably, introducing use cases where AI/ML runs on top of embedded devices. These devices present a low power footprint, while AI/ML can produce computationally intensive tasks unsuitable for such environments. To this end, benchmarking AI/ML algorithms with respect to energy efficiency is essential when selecting a particular solution for embedded devices.

7. Conclusion

This survey presented recent advances in AI/ML applications in V2X communications. We classified the related literature in handover management, beam allocation, caching, radio network allocation, computations resources management, routing and QoS prediction considering only vehicular environments. For each category, we surveyed the ML technique, the training features, architecture, and optimization objectives, and extracted results and observations concerning time complexity, performance and suitability of learning techniques according to the designated problem.

Based on the surveyed publications, there is no "one size fits all" solution. Depending on the problem, different tasks require different formulations, including AI/ML algorithm selection, optimization objectives, architecture, and training features. Each family of AI/ML algorithms comes with their own advantages and disadvantages. The problem formulation must also consider the requirements of the underlying use case, which is largely affected by the training and response times of the selected AI/ML model. In addition, it is vital to explore approaches that reduce these times, while preserving the robustness of the AI/ML algorithm. AI/ML in V2X has already shown potential in optimizing network operations, but there are still open issues that need to be addressed due to the intricacies of both AI/ML and the highly dynamic vehicular networks. Based on the surveyed papers, vehicular networks, empowered by AI/ML and V2X communications as cooperative technologies, can be transformed into autonomous networks with self-configuration/optimization/healing capabilities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] SAE, Automated Driving - Levels of Driving Automation Are Defined in New Sae International Standard J3016, 2016.
[2] P. Sewalkar, J. Seitz, Vehicle-to-pedestrian communication for vulnerable road users: survey, design considerations, and challenges, Sensors (Switzerland) (2019), https://doi.org/10.3390/s19020358.
[3] 3GPP, TR 22.885 (V14.0.0). Study on LTE Support for Vehicle to Everything (V2X) Services Release 14.
[4] 3GPP, TR 22.886 (V16.2.0). Study on Enhancements and Support for 5G V2X Services; Release 16.
[5] Y.L. Morgan, Notes on DSRC & WAVE standards suite: its architecture, design, and characteristics, IEEE Commun. Surv. Tutor. 12 (4) (Dec. 2010) 504–518, https://doi.org/10.1109/SURV.2010.033010.00024.
[6] D. Jiang, L. Delgrossi, IEEE 802.11p: towards an international standard for wireless access in vehicular environments, in: IEEE Vehicular Technology Conference, 2008, pp. 2036–2040.
[7] P.V. Klaine, M.A. Imran, O. Onireti, R.D. Souza, A survey of machine learning techniques applied to self-organizing cellular networks, IEEE Commun. Surv. Tutor. 19 (4) (2017) 2392–2431, https://doi.org/10.1109/COMST.2017.2727878.
[8] OECD, OECD/Legal/0449 - Recommendation of the Council on Artificial Intelligence, https://legalinstruments.oecd.org/en/instruments/oecd-legal-0449. (Accessed 1 September 2022).
[9] P.M. Krafft, M. Young, M. Katell, K. Huang, G. Bugingo, Defining AI in policy versus practice, https://doi.org/10.1145/3375627.3375835, 2020.
[10] C.C. Aggarwal, Neural Networks and Deep Learning, 2018.
[11] H. Ye, L. Liang, G.Y. Li, J. Kim, L. Lu, M. Wu, Machine learning for vehicular networks: recent advances and application examples, IEEE Veh. Technol. Mag. (2018), https://doi.org/10.1109/MVT.2018.2811185.
[12] L. Liang, H. Ye, G.Y. Li, Toward intelligent vehicular networks: a machine learning framework, IEEE Int. Things J. 6 (1) (2019) 124–135, https://doi.org/10.1109/JIOT.2018.2872122.
[13] M. Noor-A-Rahim, Z. Liu, H. Lee, G.G.M.N. Ali, D. Pesch, P. Xiao, A Survey on Resource Allocation in Vehicular Networks, IEEE Trans. Intell. Transp. Syst. 23 (2) (2022) 701–721, https://doi.org/10.1109/TITS.2020.3019322.
[14] W. Tong, A. Hussain, W.X. Bo, S. Maharjan, Artificial Intelligence for Vehicle-to-Everything: a survey, IEEE Access (2019), https://doi.org/10.1109/ACCESS.2019.2891073.
[15] F. Tang, Y. Kawamoto, N. Kato, J. Liu, Future intelligent and secure vehicular network toward 6G: machine-learning approaches, Proc. IEEE (2020), https://doi.org/10.1109/JPROC.2019.2954595.
[16] F. Tang, B. Mao, N. Kato, G. Gui, Comprehensive survey on machine learning in vehicular network: technology, applications and challenges, IEEE Commun. Surv. Tutor. (2021) 1, https://doi.org/10.1109/COMST.2021.3089688.
[17] A. Mchergui, T. Moulahi, S. Zeadally, Survey on Artificial Intelligence (AI) techniques for Vehicular Ad-hoc Networks (VANETs), Veh. Commun. 34 (Apr. 2022) 100403, https://doi.org/10.1016/J.VEHCOM.2021.100403.
[18] ITU Radiocommunication Bureau, Intelligent transport systems (ITS) usage M Series Mobile, radiodetermination, amateur and related satellite services [Online], https://www.itu.int/dms_pub/itu-r/opb/rep/R-REP-M.2445-2018-PDF-E.pdf, 2018. (Accessed 1 September 2022).
[19] S. Gräfling, P. Mähönen, J. Riihijärvi, Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications, in: ICUFN 2010 - 2nd International Conference on Ubiquitous and Future Networks, 2010, pp. 344–348.
[20] IEEE, IEEE Std 802.11-2020: Part11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications, in: IEEE Standard for Information Technology— Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks—Specific Requirements, 2020 [Online], https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9363693. (Accessed 1 September 2022).
[21] M. Benveniste, 'Tiered contention multiple access' (TCMA), a QOS-based distributed MAC protocol, in: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, vol. 2, 2002, pp. 598–604.
[22] G. Bianchi, L. Fratta, M. Oliveri, Performance evaluation and enhancement of the CSMA/CA MAC protocol for 802.11 wireless LANs, in: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, vol. 2, 1996, pp. 392–396.
[23] G. Naik, B. Choudhury, J.M. Park, IEEE 802.11bd & 5G NR V2X: evolution of radio access technologies for V2X communications, IEEE Access 7 (2019) 70169–70184, https://doi.org/10.1109/ACCESS.2019.2919489.
[24] 3GPP, TR 21.914; Release 14 Description; Summary of Rel-14 Work Items (Release 14).
[25] 3GPP, TR 21.915; Release 15 Description; Summary of Rel-15 Work Items (Release 15).
[26] 3GPP, TR 21.916; Release 16 Description; Summary of Rel-16 Work Items; (Release 16).
[27] A. Kanavos, D. Fragkos, A. Kaloxylos, V2X communication over cellular networks: capabilities and challenges, Telecom 2 (1) (Jan. 2021) 1–26, https://doi.org/10.3390/TELECOM2010001.
[28] M.H.C. Garcia, et al., A tutorial on 5G NR V2X communications, IEEE Commun. Surv. Tutor. (2021), https://doi.org/10.1109/COMST.2021.3057017.
[29] Scikit-learn, 1.6. Nearest neighbors — scikit-learn 1.1.1 documentation#Brute Force, https://scikit-learn.org/stable/modules/neighbors.html#brute-force. (Accessed 1 September 2022).
[30] Scikit-learn, 1.6. Nearest Neighbors-scikit-learn 1.1.1 documentation#K-D Tree, https://scikit-learn.org/stable/modules/neighbors.html#k-d-tree. (Accessed 1 September 2022).
[31] K. Hassine, A. Erbad, R. Hamila, Important complexity reduction of random forest in multi-classification problem, in: 2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019, Jun. 2019, pp. 226–231.
[32] Scikit Learn, Support vector machines, https://scikit-learn.org/stable/modules/svm.html. (Accessed 1 September 2022).
[33] Scikit Learn, Neural network models (supervised), https://scikit-learn.org/stable/modules/neural_networks_supervised.html. (Accessed 1 September 2022).
[34] K. He, J. Sun, Convolutional neural networks at constrained time cost, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 07–12-June-2015 (Oct. 2015) 5353–5360, https://doi.org/10.1109/CVPR.2015.7299173.
[35] A. Vaswani, et al., Attention is all you need, Adv. Neural Inf. Process. Syst. 2017-December (2017) 5999–6009.
[36] Scikit Learn, Clustering with KMedoids and common-nearest-neighbors, https://scikit-learn-extra.readthedocs.io/en/stable/modules/cluster.html. (Accessed 1 September 2022).
[37] Scikit Learn, sklearn.cluster.KMeans — scikit-learn 1.1.2 documentation, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html. (Accessed 1 September 2022).
[38] Scikit Learn, Affinity propagation, https://scikit-learn.org/stable/modules/clustering.html#affinity-propagation. (Accessed 24 May 2022).
[39] J.E. van Engelen, H.H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (Feb. 2020) 373–440, https://doi.org/10.1007/S10994-019-05855-6/FIGURES/5.
[40] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, second edi, A Bradford Book, 1998.
[41] C. Sammut, G.I. Webb, Bellman equation, in: Encyclopedia of Machine Learning, Springer, 2011.
[42] H. Dong, et al., Deep Reinforcement Learning: Fundamentals, Research, and Applications, Springer Nature, 2020.
[43] G. Li, C. Cai, Y. Chen, Y. Gu, Y. Wei, Y. Chi, Is Q-learning minimax optimal? A tight sample complexity analysis, arXiv preprint, arXiv:2102.06548.
[44] Y. Wu, W. Zhang, Q. Gu, A finite-time analysis of two time-scale actor-critic methods, in: Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20), 2020, pp. 17617–17628.
[45] E. Even-Dar, S. Mannor, Y. Mansour, PAC Bounds for Multi-armed Bandit and Markov Decision Processes, Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), vol. 2375, 2002, pp. 255–270.
[46] ETSI, TS 123 501 - V15.2.0 - 5G; system architecture for the 5G system (3GPP TS 23.501 version 15.2.0 release 15) [Online], https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx, 2018. (Accessed 12 November 2020).

[47] 3GPP, TS 29.250 V17.6.0; 5G System; Network Data Analytics Services; Stage 3 (Release 17).
[48] ETSI, Experiential Networked Intelligence Industry Specification Group (ENI ISG), https://www.etsi.org/committee/eni. (Accessed 24 May 2022).
[49] ETSI, Industry Specification Group (ISG) Securing Artificial Intelligence (SAI), https://www.etsi.org/committee/sai. (Accessed 24 May 2022).
[50] ITU, Focus group on machine learning for future networks including 5G, https://www.itu.int/en/ITU-T/focusgroups/ml5g/Pages/default.aspx. (Accessed 24 May 2022).
[51] O-RAN Alliance [Online], https://www.o-ran.org/.
[52] 5GAA, Making 5G proactive and predictive for the automotive industry, [Online], https://5gaa.org/wp-content/uploads/2020/01/5GAA_White-Paper_Proactive-and-Predictive_v04_8-Jan.-2020-003.pdf, 2019.
[53] N. Aljeri, A. Boukerche, A two-tier machine learning-based handover management scheme for intelligent vehicular networks, Ad Hoc Netw. 94 (Nov. 2019) 101930, https://doi.org/10.1016/j.adhoc.2019.101930.
[54] Y. Lin, Z. Zhang, Y. Huang, J. Li, F. Shu, L. Hanzo, Heterogeneous user-centric cluster migration improves the connectivity-handover trade-off in vehicular networks, IEEE Trans. Veh. Technol. 69 (12) (2020) 16027–16043, https://doi.org/10.1109/TVT.2020.3041521.
[55] H. Khan, A. Elgabli, S. Samarakoon, M. Bennis, C.S. Hong, Reinforcement learning-based vehicle-cell association algorithm for highly mobile millimeter wave communication, IEEE Trans. Cogn. Commun. Netw. 5 (4) (Dec. 2019) 1073–1085, https://doi.org/10.1109/TCCN.2019.2941191.
[56] K. Tan, D. Bremner, J. Le Kernec, Y.A. Sambo, L. Zhang, M.A. Imran, Intelligent handover algorithm for vehicle-to-network communications with double-deep Q-learning, IEEE Trans. Veh. Technol. (2022) 1, https://doi.org/10.1109/TVT.2022.3169804.
[57] V.B. Souza, M.H. Pereira, L.H.S. Lelis, X. Masip-Bruin, Enhancing resource availability in vehicular fog computing through smart inter-domain handover, in: 2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings, December 2020.
[58] L. Yan, et al., Machine learning-based handovers for sub-6 GHz and mmWave integrated vehicular networks, IEEE Trans. Wirel. Commun. (2019), https://doi.org/10.1109/TWC.2019.2930193.
[59] K. Qi, T. Liu, C. Yang, Federated learning based proactive handover in millimeter-wave vehicular networks, in: International Conference on Signal Processing Proceedings, ICSP, vol. 2020-Decem, 2020, pp. 401–406.
[60] R. Karmakar, G. Kaddoum, S. Chattopadhyay, Mobility management in 5G and beyond: a novel smart handover with adaptive time-to-trigger and hysteresis margin, IEEE Trans. Mob. Comput. (2022), https://doi.org/10.1109/TMC.2022.3188212.
[61] C. Lee, J. Jung, J.M. Chung, Intelligent dual active protocol stack handover based on double DQN deep reinforcement learning for 5G mmWave networks, IEEE Trans. Veh. Technol. 71 (7) (Jul. 2022) 7572–7584, https://doi.org/10.1109/TVT.2022.3170420.
[62] M. Zhang, S. Wang, Q. Gao, A joint optimization scheme of content caching and resource allocation for internet of vehicles in mobile edge computing, J. Cloud Comput. 9 (1) (2020), https://doi.org/10.1186/s13677-020-00182-x.
[63] L.T. Tan, R.Q. Hu, Mobility-aware edge caching and computing in vehicle networks: a deep reinforcement learning, IEEE Trans. Veh. Technol. (2018), https://doi.org/10.1109/TVT.2018.2867191.
[64] Z. Ning, et al., Joint computing and caching in 5G-envisioned Internet of vehicles: a deep reinforcement learning-based traffic control system, IEEE Trans. Intell. Transp. Syst. (2020), https://doi.org/10.1109/tits.2020.2970276.
[65] Y. He, N. Zhao, H. Yin, Integrated networking, caching, and computing for connected vehicles: a deep reinforcement learning approach, IEEE Trans. Veh. Technol. 67 (1) (2018) 44–55, https://doi.org/10.1109/TVT.2017.2760281.
[66] A. Ndikumana, N.H. Tran, C.S. Hong, Deep learning based caching for self-driving car in multi-access edge computing, IEEE Trans. Intell. Transp. Syst. (2020) 1–16, https://doi.org/10.1109/tits.2020.2976572.
[67] L. Hou, L. Lei, K. Zheng, X. Wang, A Q-learning-based proactive caching strategy for non-safety related services in vehicular networks, IEEE Int. Things J. 6 (3) (2019) 4512–4520, https://doi.org/10.1109/JIOT.2018.2883762.
[68] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, M.S. Hossain, Mobility-aware proactive edge caching for connected vehicles using federated learning, IEEE Trans. Intell. Transp. Syst. 22 (8) (Aug. 2021) 5341–5351, https://doi.org/10.1109/TITS.2020.3017474.
[69] MovieLens 1M Dataset | GroupLens, https://grouplens.org/datasets/movielens/1m/. (Accessed 31 August 2022).
[70] V.S. Varanasi, S. Chilukuri, Adaptive differentiated edge caching with machine learning for V2X communication, in: 2019 11th International Conference on Communication Systems and Networks, COMSNETS 2019, May 2019, pp. 481–484.
[71] C. Song, W. Xu, T. Wu, S. Yu, P. Zeng, N. Zhang, QoE-driven edge caching in vehicle networks based on deep reinforcement learning, IEEE Trans. Veh. Technol. 70 (6) (Jun. 2021) 5286–5295, https://doi.org/10.1109/TVT.2021.3077072.
[72] H. Ge, et al., Multi-server intelligent task caching strategy for edge computing, in: 4th International Conference on Natural Language Processing (ICNLP), Sep. 2022, pp. 563–569.
[73] W. Jiang, D. Feng, Y. Sun, G. Feng, Z. Wang, X.G. Xia, Proactive content caching based on actor-critic reinforcement learning for mobile edge networks, IEEE Trans. Cogn. Commun. Netw. 8 (2) (Jun. 2022) 1239–1252, https://doi.org/10.1109/TCCN.2021.3130995.
[74] J. Gao, M.R.A. Khandaker, F. Tariq, K.K. Wong, R.T. Khan, Deep neural network based resource allocation for V2X communications, IEEE Veh. Technol. Conf. 2019-Sept. (Sep. 2019), https://doi.org/10.1109/VTCFALL.2019.8891446.
[75] S.K. Tayyaba, et al., 5G vehicular network resource management for improving radio access through machine learning, IEEE Access 8 (2020) 6792–6800, https://doi.org/10.1109/ACCESS.2020.2964697.
[76] S. Gyawali, Y. Qian, R.Q. Hu, Resource allocation in vehicular communications using graph and deep reinforcement learning, in: 2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings, Dec. 2019.
[77] T. Koshimizu, S. Gengtian, H. Wang, Z. Pan, J. Liu, S. Shimamoto, Multi-dimensional affinity propagation clustering applying a machine learning in 5G-cellular V2X, IEEE Access 8 (2020) 94560–94574, https://doi.org/10.1109/ACCESS.2020.2994132.
[78] H. Ye, G.Y. Li, B.H.F. Juang, Deep reinforcement learning based resource allocation for V2V communications, IEEE Trans. Veh. Technol. 68 (4) (Apr. 2019) 3163–3173, https://doi.org/10.1109/TVT.2019.2897134.
[79] M.I. Khan, F.X. Aubet, M.O. Pahl, J. Harri, Deep learning-aided resource orchestration for vehicular safety communication, IFIP Wirel. Days 2019-April (Apr. 2019), https://doi.org/10.1109/WD.2019.8734252.
[80] P. Xiang, H. Shan, M. Wang, Z. Xiang, Z. Zhu, Multi-agent RL enables decentralized spectrum access in vehicular networks, IEEE Trans. Veh. Technol. 70 (10) (Oct. 2021) 10750–10762, https://doi.org/10.1109/TVT.2021.3103058.
[81] N. Banitalebi, P. Azmi, N. Mokari, A.H. Arani, H. Yanikomeroglu, Distributed learning based resource allocation for self-organizing C-V2X communication in cellular networks, IEEE Open J. Commun. Soc. (2022), https://doi.org/10.1109/OJCOMS.2022.3211340.
[82] L.H. Nguyen, V.L. Nguyen, J.J. Kuo, Efficient reinforcement learning-based transmission control for mitigating channel congestion in 5G V2X sidelink, IEEE Access 10 (2022) 62268–62281, https://doi.org/10.1109/ACCESS.2022.3182021.
[83] L. Lusvarghi, M.L. Merani, Machine learning for disseminating cooperative awareness messages in cellular V2V communications, IEEE Trans. Veh. Technol. 71 (7) (Jul. 2022) 7890–7903, https://doi.org/10.1109/TVT.2022.3170982.
[84] P. Dai, et al., Multi-armed bandit learning for computation-intensive services in MEC-empowered vehicular networks, IEEE Trans. Veh. Technol. 69 (7) (2020) 7821–7834, https://doi.org/10.1109/TVT.2020.2991641.
[85] P.A. Lopez, et al., Microscopic traffic simulation using SUMO, in: IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, vol. 2018-November, Dec. 2018, pp. 2575–2582.
[86] B. Fan, Z. He, Y. Wu, J. He, Y. Chen, L. Jiang, Deep learning empowered traffic offloading in intelligent software defined cellular V2X networks, IEEE Trans. Veh. Technol. 69 (11) (2020) 13328–13340, https://doi.org/10.1109/TVT.2020.3023194.
[87] H. Ke, J. Wang, L. Deng, Y. Ge, H. Wang, Deep reinforcement learning-based adaptive computation offloading for MEC in heterogeneous vehicular networks, IEEE Trans. Veh. Technol. 69 (7) (2020) 7916–7929, https://doi.org/10.1109/TVT.2020.2993849.
[88] Y. Liu, H. Yu, S. Xie, Y. Zhang, Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks, IEEE Trans. Veh. Technol. 68 (11) (2019) 11158–11168, https://doi.org/10.1109/TVT.2019.2935450.
[89] C. Sonmez, C. Tunca, A. Ozgovde, C. Ersoy, Machine learning-based workload orchestrator for vehicular edge computing, IEEE Trans. Intell. Transp. Syst. 22 (4) (2021) 2239–2251, https://doi.org/10.1109/TITS.2020.3024233.
[90] Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo, X. Shen, A joint service migration and mobility optimization approach for vehicular edge computing, IEEE Trans. Veh. Technol. (2020), https://doi.org/10.1109/TVT.2020.2999617.
[91] M. Ibrar, et al., ARTNet: AI-based resource allocation and task offloading in a reconfigurable Internet of vehicular networks, IEEE Trans. Netw. Sci. Eng. (2020), https://doi.org/10.1109/TNSE.2020.3047454.
[92] D. Chen, Y.C. Liu, B.G. Kim, J. Xie, C.S. Hong, Z. Han, Edge computing resources reservation in vehicular networks: a meta-learning approach, IEEE Trans. Veh. Technol. 69 (5) (2020) 5634–5646, https://doi.org/10.1109/TVT.2020.2983445.
[93] Unity Real-Time Development Platform | 3D, 2D VR & AR Engine, https://unity.com/. (Accessed 31 August 2022).
[94] S. Islam, S. Badsha, S. Sengupta, Context-aware fine-grained task scheduling at vehicular edges: an extreme reinforcement learning based dynamic approach, in: Proceedings - 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2021, Jun. 2021, pp. 31–40.
[95] H. Dinh, N.H. Nguyen, T.T. Nguyen, T.H. Nguyen, T.T. Nguyen, P. Le Nguyen, Deep reinforcement learning-based offloading for latency minimization in 3-tier V2X networks, IEEE Wirel. Commun. Netw. Conf., WCNC 2022-April (2022) 1803–1808, https://doi.org/10.1109/WCNC51071.2022.9771583.

[96] Y. Lin, Y. Zhang, J. Li, F. Shu, C. Li, Popularity-aware online task offloading for heterogeneous vehicular edge computing using contextual clustering of bandits, IEEE Int. Things J. 9 (7) (Apr. 2022) 5422–5433, https://doi.org/10.1109/JIOT.2021.3109003.
[97] C. Wu, T. Yoshinaga, Y. Ji, Y. Zhang, Computational intelligence inspired data delivery for vehicle-to-roadside communications, IEEE Trans. Veh. Technol. 67 (12) (2018) 12038–12048, https://doi.org/10.1109/TVT.2018.2871606.
[98] Z. Khan, P. Fan, F. Abbas, H. Chen, S. Fang, Two-level cluster based routing scheme for 5G V2X communication, IEEE Access 7 (2019) 16194–16205, https://doi.org/10.1109/ACCESS.2019.2892180.
[99] F. Li, X. Song, H. Chen, X. Li, Y. Wang, Hierarchical routing for vehicular Ad Hoc networks via reinforcement learning, IEEE Trans. Veh. Technol. 68 (2) (2019) 1852–1865, https://doi.org/10.1109/TVT.2018.2887282.
[100] M.E. Morocho-Cayamcela, H. Lee, W. Lim, Machine learning to improve multi-hop searching and extended wireless reachability in V2X, IEEE Commun. Lett. 24 (7) (Jul. 2020) 1477–1481, https://doi.org/10.1109/LCOMM.2020.2982887.
[101] X. Du, H. Van Nguyen, C. Jiang, Y. Li, F.R. Yu, Z. Han, Virtual relay selection in LTE-V: a deep reinforcement learning approach to heterogeneous data, IEEE Access 8 (2020) 102477–102492, https://doi.org/10.1109/ACCESS.2020.2997729.
[102] O. Jafarzadeh, M. Dehghan, H. Sargolzaey, M.M. Esnaashari, A model-based reinforcement learning protocol for routing in vehicular Ad hoc network, Wirel. Pers. Commun. 123 (1) (Nov. 2021) 975–1001, https://doi.org/10.1007/S11277-021-09166-9.
[103] L. Luo, L. Sheng, H. Yu, G. Sun, Intersection-based V2X routing via reinforcement learning in vehicular Ad Hoc networks, IEEE Trans. Intell. Transp. Syst. 23 (6) (Jun. 2022) 5446–5459, https://doi.org/10.1109/TITS.2021.3053958.
[104] K. Kandali, L. Bennis, H. Bennis, A new hybrid routing protocol using a modified K-means clustering algorithm and continuous Hopfield network for VANET, IEEE Access 9 (2021) 47169–47183, https://doi.org/10.1109/ACCESS.2021.3068074.
[105] G.H. Sim, S. Klos, A. Asadi, A. Klein, M. Hollick, An online context-aware machine learning algorithm for 5 g mmWave vehicular communications, IEEE/ACM Trans. Netw. 26 (6) (Dec. 2018) 2487–2500, https://doi.org/10.1109/TNET.2018.2869244.
[106] D. Li, S. Wang, H. Zhao, X. Wang, Context-and-social-aware online beam selection for mmWave vehicular communications, IEEE Int. Things J. 8 (10) (May 2021) 8603–8615, https://doi.org/10.1109/JIOT.2020.3047676.
[107] Y. Wang, M. Narasimha, R.W. Heath, Towards robustness: machine learning for MmWave V2X with situational awareness, in: Conference Record - Asilomar Conference on Signals, Systems and Computers, vol. 2018-Octob, Feb. 2019, pp. 1577–1581.
[108] Y. Wang, A. Klautau, M. Ribero, A.C.K. Soong, R.W. Heath, MmWave vehicular beam selection with situational awareness using machine learning, IEEE Access 7 (2019) 87479–87493, https://doi.org/10.1109/ACCESS.2019.2922064.
[109] Y. Yang, Z. Gao, Y. Ma, B. Cao, D. He, Machine learning enabling analog beam selection for concurrent transmissions in millimeter-wave V2V communications, IEEE Trans. Veh. Technol. 69 (8) (Aug. 2020) 9185–9189, https://doi.org/10.1109/TVT.2020.3001340.
[110] A. Kose, C.H. Foh, H. Lee, K. Moessner, Profiling vehicles for improved small cell beam-vehicle pairing using multi-armed bandit, in: International Conference on ICT Convergence, vol. 2021-October, 2021, pp. 221–226.
[111] Y. Ju, et al., DRL-based beam allocation in relay-aided multi-user MmWave vehicular networks, in: INFOCOM WKSHPS 2022 - IEEE Conference on Computer Communications Workshops, 2022.
[112] M. Wirth, A. Klein, A. Ortiz, Risk-aware multi-armed bandits for vehicular communications, in: IEEE Vehicular Technology Conference, vol. 2022-June, 2022.
[113] L. Torres-Figueroa, H.F. Schepker, J. Jiru, QoS evaluation and prediction for C-V2X communication in commercially-deployed LTE and mobile edge networks, in: IEEE Vehicular Technology Conference, vol. 2020-May, May 2022, https://doi.org/10.1109/VTC2020-SPRING48590.2020.9129382.
[114] D.C. Moreira, I.M. Guerreiro, W. Sun, C.C. Cavalcante, D.A. Sousa, QoS predictability in V2X communication with machine learning, in: IEEE Vehicular Technology Conference, vol. 2020-May, May 2022, https://doi.org/10.1109/VTC2020-SPRING48590.2020.9129490.
[115] W. Zhang, M. Feng, M. Krunz, H. Volos, Latency prediction for delay-sensitive V2X applications in mobile cloud/edge computing systems, in: 2020 IEEE Global Communications Conference, GLOBECOM 2020 - Proceedings, vol. 2020-Janua, Dec. 2022, https://doi.org/10.1109/GLOBECOM42002.2020.9348104.
[116] S. Barmpounakis, L. Magoula, N. Koursioumpas, R. Khalili, J.M. Perdomo, R.P. Manjunath, LSTM-based QoS prediction for 5G-enabled Connected and Automated Mobility applications, 2021 IEEE 4th 5G World Forum (5GWF) (Oct. 2021) 436–440, https://doi.org/10.1109/5GWF52925.2021.00083.
[117] J. Mendoza, et al., Proactive dual connectivity for automated guided vehicles in outdoor industrial environment, IEEE Access 10 (2022) 54149–54163, https://doi.org/10.1109/ACCESS.2022.3176730.
[118] A. Kousaridas, et al., QoS prediction for 5G connected and automated driving, IEEE Commun. Mag. 59 (9) (Sep. 2021) 58–64, https://doi.org/10.1109/MCOM.110.2100042.
[119] D. Xin, H. Miao, A. Parameswaran, N. Polyzotis, Production machine learning pipelines: empirical analysis and optimization opportunities, Proc. ACM SIGMOD Int. Conf. Manage. Data 14 (21) (2021) 2639–2652, https://doi.org/10.1145/3448016.3457566.
[120] L. Jing, Y. Tian, Self-supervised visual feature learning with deep neural networks: a survey, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (Nov. 2021) 4037–4058, https://doi.org/10.1109/TPAMI.2020.2992393.
[121] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni, Generalizing from a few examples, ACM Comput. Surv. (CSUR) 53 (3) (Jun. 2020), https://doi.org/10.1145/3386252.
[122] Y. Sun, M. Peng, Y. Zhou, Y. Huang, S. Mao, Application of machine learning in wireless networks: key techniques and open issues, IEEE Commun. Surv. Tutor. 21 (4) (Oct. 2019) 302–3108, https://doi.org/10.1109/COMST.2019.2924243.
[123] G. Bin Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1–3) (Dec. 2006) 489–501, https://doi.org/10.1016/J.NEUCOM.2005.12.126.
[124] NVIDIA, NVIDIA Data Center Deep Learning Product Performance, https://developer.nvidia.com/deep-learning-performance-training-inference. (Accessed 7 April 2022).
[125] D. Krajzewicz, L. Bieker, J. Härri, R. Blokpoel, Simulation of V2X applications with the iTETRIS system, Proc., Soc. Behav. Sci. 48 (Jan. 2012) 1482–1492, https://doi.org/10.1016/J.SBSPRO.2012.06.1124.
[126] C. Celes, A. Boukerche, A.A.F. Loureiro, Mobility data assessment for vehicular networks, in: IEEE International Conference on Communications, vol. 2019-May, May 2019, https://doi.org/10.1109/ICC.2019.8761361.
[127] The Network Simulator - ns-2, https://www.isi.edu/nsnam/ns/. (Accessed 24 May 2022).
[128] T.R. Henderson, M. Lacage, G.F. Riley, Network simulations with the ns-3 simulator, SIGCOMM Demonstr. (2008).
[129] A. Varga, R. Hornig, An overview of the OMNeT++ simulation environment, in: Simutools '08: Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, No. 60, 2008, pp. 1–10.
[130] Mathworks, MATLAB, https://www.mathworks.com/. (Accessed 24 May 2022).
[131] 3GPP, TR 36.814 V9.0.0, Further Advancements for E-UTRA Physical Layer Aspects, Release 9, 2010.
[132] 3GPP, TR 37.885 V15.3.0, Study on evaluation methodology of new Vehicle-to-Everything V2X use cases for LTE and NR (Release 15), 2019.
[133] M. NAKAGAMI, The m-distribution—a general formula of intensity distribution of rapid fading, Stat. Methods Radio Wave Propag. (Jan. 1960) 3–36, https://doi.org/10.1016/B978-0-08-009306-2.50005-4.
[134] J.G. Proakis, M. Salehi, Digital Communications, 5th ed., McGraw-Hill, New York, 2008.
[135] M. Abadi, TensorFlow: learning functions at scale, in: Proceedings of the 21st ACM SIGPLAN International Conference on Functional Programming, 2016.
[136] F. Chollet, Keras [Online], https://keras.io, 2015.
[137] A. Ndikumana, C.S. Hong, Self-driving car meets multi-access edge computing for deep learning-based caching, https://doi.org/10.1109/ICOIN.2019.8718113, 2019.
[138] C-V2X Use Cases and Service Level Requirements Volume I – 5G Automotive Association, https://5gaa.org/news/c-v2x-use-cases-and-service-level-requirements-volume-i/. (Accessed 5 April 2022).
[139] C-V2X Use Cases Volume II: Examples and Service Level Requirements – 5G Automotive Association, https://5gaa.org/news/c-v2x-use-cases-volume-ii-examples-and-service-level-requirements/. (Accessed 5 April 2022).
[140] GPS Accuracy, https://www.gps.gov/systems/gps/performance/accuracy/. (Accessed 5 April 2022).
[141] H. Wymeersch, G. Seco-Granados, G. Destino, D. Dardari, F. Tufvesson, 5G mmWave positioning for vehicular networks, IEEE Wirel. Commun. 24 (6) (Dec. 2017) 80–86, https://doi.org/10.1109/MWC.2017.1600374.
[142] 5G PPP Technology Board, AI and ML - Enablers for Beyond 5G Networks, 2021.
[143] G. Vilone, L. Longo, Explainable Artificial Intelligence: a systematic review, https://arxiv.org/abs/2006.00093, 2020.
[144] 3GPP, TR 23.700-91 V17.0.0; Technical Specification Group Services and System Aspects; Study on enablers for network automation for the 5G System (5GS); Phase 2 (Release 17), 2020.