This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

Computation Offloading for Vehicular Environments: A Survey

ALISSON B. DE SOUZA[1], PAULO A. L. REGO[1], (MEMBER, IEEE), TIAGO CARNEIRO[2], JARDEL DAS C. RODRIGUES[3], P. P. REBOUÇAS FILHO[3], (MEMBER, IEEE), JOSÉ N. DE SOUZA[1], (SENIOR MEMBER, IEEE), VINAY CHAMOLA[5], (SENIOR MEMBER, IEEE), VICTOR HUGO C. DE ALBUQUERQUE[4], (SENIOR MEMBER, IEEE), BIPLAB SIKDAR[6], (SENIOR MEMBER, IEEE)

[1]GREat, Federal University of Ceará, Fortaleza, CE, Brazil. (e-mails: alisson@ufc.br, pauloalr@ufc.br, neuman@ufc.br)
[2]INRIA Lille - Nord Europe, Lille, France. (e-mail: tiago.carneiro-pessoa@inria.fr)
[3]Lapisco, IFCE, Fortaleza, CE, Brazil. (e-mails: rodrigues.jardel@ifce.edu.br, pedrosarf@ifce.edu.br)
[4]Universidade de Fortaleza, Fortaleza, CE, Brazil. (e-mail: victor.albuquerque@unifor.br)
[5]Department of Electrical and Electronics Engineering, Birla Institute of Technology & Science (BITS), Pilani, Rajasthan, India. (e-mail:vinay.chamola@pilani.bits-pilani.ac.in)
[6]Department of Electrical and Computer Engineering, National University of Singapore, Singapore. (e-mail: bsikdar@nus.edu.sg)

Corresponding author: Biplab Sikdar (e-mail: bsikdar@nus.edu.sg).

ABSTRACT With significant advances in communication and computing, modern day vehicles are becoming increasingly intelligent. This gives them the ability to contribute to safer roads and passenger comfort through network devices, cameras, sensors, and computational storage and processing capabilities. However, to run new and popular applications, and to enable vehicles operating autonomously requires massive computational resources. Computational resources available with the current day vehicles are not sufficient to process all these demands. In this situation, other vehicles, edge servers, and servers in remote data centers can help the vehicles by lending their computing resources. However, to take advantage of these computing resources, computation offloading techniques have to be leveraged to transfer tasks or entire applications to run on other devices. Such computation offloading can lead to improved performance and Quality of Service (QoS) for applications and for the network. However, computation offloading in a highly dynamic environment such as vehicular networks is a major challenge. Therefore, this survey aims to review and organize the computation offloading literature in vehicular environments. In addition, we demystify some concepts, propose a taxonomy with the most important aspects and classify most works in the area according to each category. We also present the main tools, scenarios, subjects, strategies, objectives, etc., used in the works. Finally, we present the main challenges and future directions to guide future research in this active research area.

INDEX TERMS Computation Offloading, Vehicular Networks, Vehicular Cloud Computing, Vehicular Edge Computing, Vehicular Fog Computing.

I. INTRODUCTION

In the latest update from the World Health Organization (WHO), in January 2020, there were more than 2 billion registered vehicles around the world [1]. That number continues to increase, theoretically generating more comfort, convenience and efficiency for people, but also generating traffic congestion, accidents and pollution [2]. Significant efforts have been made to improve the situation. Among these efforts, the Vehicular Ad Hoc Networks (VANETs) that are used in Intelligent Transportation Systems (ITS) stand out. VANETs provide connectivity to vehicles through vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I) and vehicle-to-everything (V2X) communications. VANETs allow the exchange of information even in places without infrastructure [3], [4], [5]. Through these connections, vehicles can send alerts of collisions, accidents, overtaking, congestion, etc., and improve road safety [2], [6], [7]. In this way, the National Highway Traffic Safety Administration (NHTSA) of the United States estimates that adoption of only two V2X safety applications would save about 1,000 lives and prevent about half a million crashes a year [8]. In addition, these connections allow vehicle journeys to be more pleasurable for drivers and passengers by allowing the execution of entertainment applications [9]. Thus, different players have developed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

and adopted vehicular communications into academia and manufacturing of devices and vehicles. In fact, this market has been moving millions of dollars and is expected to move billions of dollars in the coming years [10], [11].

In addition to advancing communications, vehicles have also become intelligent and evolved in cameras, embedded systems, sensors, Advanced Driver-Assistance Systems (ADAS) and computing power. In fact, vehicles are increasingly being produced with powerful on-board computing capabilities. In this way, they can act as servers to assist in the processing of other vehicles, mobile devices of passengers or of people walking on the sidewalks, and even to relieve the computational load of other devices on the network [2].

However, new and popular applications have emerged, such as applications based on artificial intelligence, augmented reality (AR), image-aided navigation, intelligent vehicle control, gaming, etc. Such applications demand massive computation and storage resources to handle complicated data processing and storage operations, and still have critical latency requirements [12], [13]. In addition, it is expected that with hundreds of sensors in future vehicles generating an enormous amount of data, there will be a lot of pressure on the computational resources of the vehicles. The processing power required to process data from future vehicles will easily deplete the vehicles' on-board resources. For example, a study predicts that each vehicle will need approximately 10^6 DMIPS of computational power and must be able to process real-time traffic conditions within a latency of 100 ms to enable autonomous vehicle steering [14], [2]. Also, approximately 1 GB of data will need to be processed every second on real-time operating systems of self-driving cars [15].

Unfortunately, vehicles do not yet have enough on-board computing resources to handle all of these requirements and are still limited compared to the scale of data that needs to be stored and processed. In fact, it is a challenge to guarantee the proper quality of service (QoS). Even if more powerful processors such as Graphics Processing Units (GPUs) are installed, it can lead to high energy consumption due to the greater power and the need for cooling to meet thermal restrictions. In this way, fuel efficiency of the vehicle and driving range can be significantly affected [2].

In is sense, computing paradigms such as Cloud Computing will certainly be an important way to help intelligent vehicles, but may not be enough. While computing power, cost and power consumption are the main limiting factors for on-board computing, long latency and massive data transmission are the bottlenecks of cloud-based processing. In addition, this latency depends on the condition of the wireless channel, network bandwidth, and traffic congestion. Therefore, real-time processing and reliability cannot be guaranteed. In this way, cloud resources can be used for heavier processing, for long-term storage and for non-real-time processing [2].

To overcome the limited resources of in-vehicle computing, communication, storage and power while avoiding excessive latency in cloud computing, deploying computing

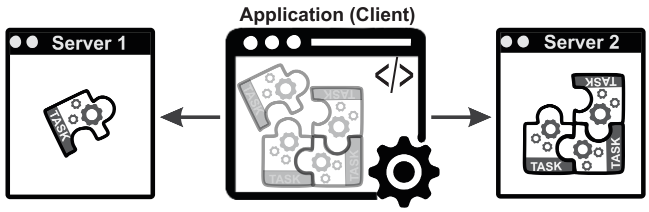


FIGURE 1. Figurative depiction of computation offloading technique.

resources at the edge of the wireless network has received significant attention from academia and industry. In fact, it is possible to deliver cloud services directly from the edge of the network and support delay-sensitive applications and also meet the low-latency requirement for mission-critical tasks [16], [17]. For example, highly complicated tasks such as the powerful Convolutional Neural Network (CNN) for feature extraction and vision-based perception tasks can run on nearby edge servers with more computing resources to help intelligent vehicles [2]. Another way to provide cloud services at the edge of the network is through the vehicles themselves, i.e., a cloud composed of vehicles. In this way, groups of vehicles can have better computing power and can provide services such as processing, connectivity, data collection, and storage, among others. The resources of these vehicles can be used for data preprocessing to reduce the use of bandwidth [18], [19].

With these computational resources available at different levels of the network, in the traditional cloud, in groups of vehicles, or in edge servers coupled to roadside infrastructures or base stations, it is necessary to take advantage of them to meet the requirements of the applications. In this sense, a technique that can help is called computation offloading, also known as computing offloading, computational offloading, computation as a service, computing as a service, task offloading, workload offloading, or cyber foraging [20], [21]. This technique consists of partitioning tasks of an application and sending them to be performed on other devices, as seen in Figure 1. In this case, these device have accessible and possibly idle or underutilized computing resources [22]. Computation offloading can be used to increase the computational capabilities of devices, for performance gains, or to decrease the processing load on an overloaded device by migrating parts of an application (or the whole application) to a remote computing device that may lend their computing power [23], [24], [25], [26], [27], [28], [29].

Existing computation offloading solutions have applied various strategies and mechanisms to manipulate steps in the offloading process, such as device discovery, resource profiling, application partitioning, and offloading decision [30]. Computation offloading can be classified as static when the developer or system defines before execution (in the project or at the time of installation) which parts of the application should be downloaded and where. It can also be classified as dynamic when the framework/system decides at run-time

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

which parts of the application should be downloaded and where to download, based on metrics related to the current network condition, mobile devices, and remote server.

However, it is very important to use computational offloading systems that can handle the challenges, of vehicular networks, have reliable communications and improve the performance of applications [31], [32], [33]. Therefore, the integration of computation offloading techniques, computing paradigms, and vehicular networks will play a vital role in the development of intelligent vehicles.

Thus, our survey aims to demystify concepts, aggregate and classify state-of-the-art research on computation offloading for vehicular environments and present its challenges. Important subjects are covered and we propose a complete taxonomy of the works in this domain.

A. EXISTING SURVEYS AND TUTORIALS

Although there are several surveys that deal with computation offloading and others that deal with vehicular networks, there is still a lack of a work that completely analyzes the integration of the two themes.

A comprehensive review of opportunistic offloading is presented in [26]. Such type of offloading can refer to both traffic/data offloading for content distribution and computation offloading for task distribution. In computation offloading, the classification is subdivided into with wired networks, with/without node selection, single/multi-objective(s) and tasks from inside/outside of the cluster. Discussions about the subject are also presented, as well as research directions, offloading strategies and future research problems. However, this survey shows opportunistic offloading in general and, in most cases, focuses on mobile devices and both traffic and computation offloading. In this way, it is not specific for computation offloading or for vehicular scenarios.

The survey by Zhou *et al.* [34] investigated data offloading techniques through Vehicular Ad Hoc Networks. Based on communications standards between vehicles and infrastructures, the authors classify data offloading through vehicle-to-vehicle communications, vehicle-to-infrastructure communications, and vehicle-to-everything communications. The pros and cons for some data offloading techniques in VANETs are also discussed. Then, challenges and open research problems are presented. Nonetheless, although the paper is specific to VANETs, it does not deal with computation offloading.

Lin *et al.* [35] analyzed computation offloading for edge computing. An insight into the architecture and types of edge computing nodes was given. Based on this overview, the paper reviews the challenges of computing offloading in terms of application partitioning, task allocation, and task execution, focusing on features for edge computing. Then, some application scenarios for edge computing are presented, such as real-time video analytics, smart “things”, vehicular applications, and cloud gaming. Subsequent, opportunities and future research directions are discussed. Nevertheless, the work presents computation offloading for edge computing in general and not specific for vehicular networks.

In 2019, Jiang *et al.* [36] discussed computation offloading in edge computing. Some aspects of computation offloading related to edge computing are surveyed such as: partitioning, what/when/where to offload, energy consumption minimization, Quality of Services guarantee, and Quality of Experiences enhancement. Case studies of cooperation between edge and cloud, resource scheduling approaches, gaming and trade-off among system performance and overheads for computation offloading decision making are also reviewed. Even so, the paper focuses on computation offloading related to mobile devices and edge computing, and not to vehicular scenarios.

In 2019, the authors of [2] provided partial coverage of the subject. The survey analyzes the latest developments in edge computing for Internet of Vehicles. It also presents important design problems, methodologies, typical use cases, hardware platforms and open research problems for computing in vehicular networks. Cases where vehicles act as clients and servers, and cases where edge servers coupled to base stations are used are described. Edge caching cases for content distribution and artificial intelligence models for use on the edge are also investigated. However, the article does not focus on the various aspects of computation offloading for vehicular scenarios and presents several issues not directly related to this.

In 2020, a survey focusing on computation offloading and vehicular networks was presented at [12]. The analyzed scenarios are those where the vehicles act as edge nodes for mobile devices with restriction of computational resources. In addition, the work presents some challenges in the area and focuses on solutions of computation offloading and classifying them according to the techniques of partitioning, scheduling and data retrieval. However, this survey focuses only on the part of algorithms and data retrieval, in cases where mobile devices (smartphones, tablets, wearables, etc.) act as clients, in cases where the servers are on the edge of the network, and cases where the task is completely offloaded. Our survey shows several types of clients and different servers, presents a complete taxonomy, and focuses on all aspects of offloading works, from technology to the types of experiments.

Therefore, our survey investigates, deepens and classifies in a complete way all topics related to the intersection between computation offloading and vehicular networks. A taxonomy is also proposed to classify these topics in several categories. In Table 1, we show the summary comparison of the above-mentioned surveys based on features, such as year, focus on computation offloading and vehicular networks, and complete deepening on the intersection between computation offloading and vehicular networks.

B. CONTRIBUTIONS

According to Table 1, although some surveys deal with parts of the subject addressed here in this paper, they do not focus on the intersection between computation offloading and vehicular networks, or when they do, they do not present a

TABLE 1. Comparison of surveys of Computation Offloading and Vehicular Environments. (√) indicates that topic is covered. (×) indicates that the topic is not covered.

Surveys	Year	Focus		Complete Deepening
		Computation Offloading (CO)	Vehicular Enviroments (VE)	CO + VE
Ref. [26]	2018	×	×	×
Ref. [34]	2018	×	√	×
Ref. [35]	2019	√	×	×
Ref. [36]	2019	√	×	×
Ref. [2]	2019	×	√	×
Ref. [12]	2020	√	√	×
Our survey		√	√	√

complete comprehensive outlook of all aspects of computation offloading. To the best of our knowledge, this work is the first to present and classify all the most important details of the computation offloading works in VANETs, from the technologies used, to details of the experiments used to validate proposals. Thus, we highlight our main contributions below.

- Demystification of important related concepts that are often confused in literature papers.
- Taxonomy and complete classification of the main aspects of computation offloading for vehicular environments, including communication standards, problems and experiments.
- Organization, description and identification of the main tools, subjects, scenarios, strategies, objectives, etc., used in most works in the area, including graphics to facilitate visualization.
- Presentation of the main challenges and problems in the area to guide future research.
- Tables with the categorization of most articles in the area of computation offloading in VANETs.

C. SURVEY ORGANIZATION

The rest of this survey is organized as follows: Section II presents some concepts widely used in the field of computing in vehicular environments and their differences; in the Section III and in the Tables at the end of this paper, we provide a taxonomy, classify several articles according to the proposed categories and approach the main subjects used in the works; Section IV identifies the open issues and challenges to guide future research; finally, in Section V, we present the conclusion.

II. DEMYSTIFYING RELATED CONCEPTS

In the literature, there are several concepts related to cloud, edge and fog and computing in vehicles and other devices that have subtle differences from each other. Therefore, below, we analyze the six main concepts. The first three deal with types of clouds (or remote execution environments) called the Vehicular Cloud (VC), Edge and Traditional Cloud (TC). The other three concepts are the main paradigms related to the subject of our survey: Vehicular Cloud Computing (VCC), Vehicular Edge Computing (VEC) and Vehicular Fog Computing (VFC).

A. VEHICULAR CLOUD (VC)

The concept of Vehicular Cloud (VC) emerged to make better use of the computing, communication and storage resources of vehicles [37]. Although there are several similar terms as Vehicular Micro Cloud (VMC) [38], [39], Mobile Vehicular Cloudlet (MVC) [40], [41], Vehicular Cloudlet (Vc) [41], V-Cloud [42], and Vehicular Ad hoc Cloud (VACloud) [22], we use the term Vehicular Cloud. In this type of cloud, vehicle owners can lend or rent the surplus computing resources on board, similar to what traditional cloud providers do. The computational resources of two or more vehicles, stationary or in motion, can be gathered, coordinated, allocated and offered, through V2X (Vehicle-to-Everything) connections, as a real-time service for other vehicles or customers to use [43], [31], [44].

A Vehicular Cloud (VC), as seen in Figure 2, is a pool of vehicular computing resources that can be dynamically coordinated and to offer services on demand, through V2V connections (orange lines), as in the cloud computing model. These VCs can be integrated with remote clouds, as well as be isolated, self-organized, autonomous, smaller, mobile, and ad hoc clouds, based on the availability of neighboring vehicles, and can be formed anywhere using the computational capabilities of on-board vehicles. In addition, VCs have advantages such as low-cost computing, support for decision-making in scenarios without infrastructure, use without energy limitations (with vehicles with large capacity batteries and recharged by engine operation) and guarantee of real-time services removing network delays involved in accessing traditional clouds. However, this type of cloud suffers from extreme mobility and dynamism of the environment and resources [45], [43], [18], [42], [31], [44], [46].

B. EDGE

In the context of vehicular networks, we use the term Edge to refer to the set of edge servers deployed in the vicinity of streets, avenues and roads (e.g., servers attached to RSUs or base stations) by service providers. These servers provide processing and storage for one or a few hops of vehicles through isolated servers or small data centers. This set of servers offers advantages such as greater computing, communication and storage capacities, reduced latency due to faster processing and being close to vehicles (eliminating excessive network hops), reduced traffic congestion in links between the core and the periphery network, and QoS improvement of vehicular applications with strict time requirements. However, the computing resources available at Edge are moderate compared to the traditional cloud [47], [32], [38], [48], [49], [50], [51].

C. TRADITIONAL CLOUD (TC)

We use the term Traditional Cloud (TC) to refer to a set of large-scale centralized data centers that are fixed on the facil-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

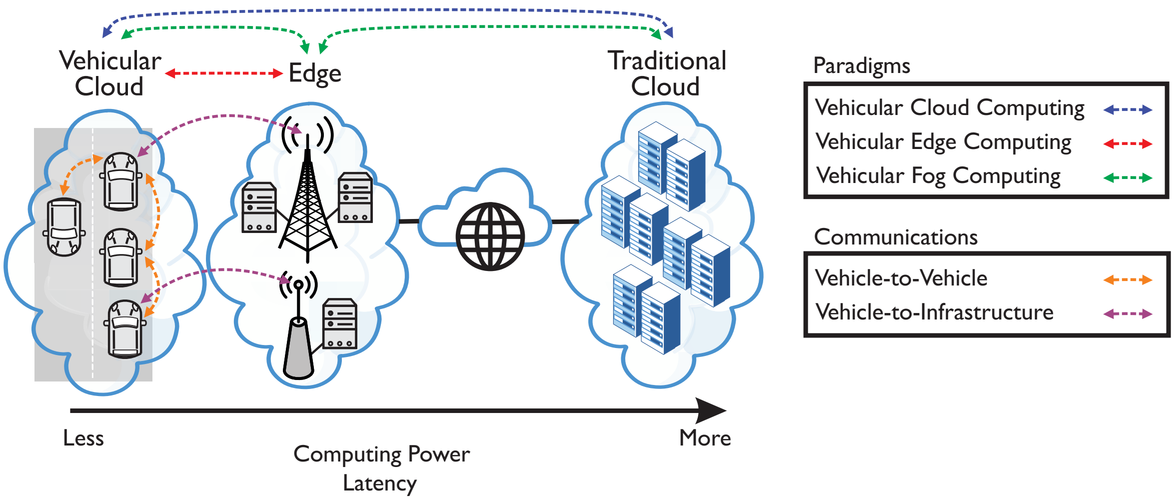


FIGURE 2. Cloud related paradigms in vehicular environments.

ities of a cloud provider. This cloud infrastructure is made available to the general public and offers advantages such as greater accessibility and availability to computational resources, aid in the execution of computationally intensive applications, unlimited and powerful computational resources to serve different customers, and absence of restriction of energy consumption due to the constant supply of energy in the data center. However, the time required or the latency to access the TC can be very high and may not be practical for some mission critical applications or applications with ultra-low latency requirements, in addition to requiring an internet connection at all times, presenting challenges of connectivity [52], [38].

D. VEHICULAR CLOUD COMPUTING (VCC)

Despite the fact that there are several similar terms such as Vehicle-Assisted Cloud Computing (VACC) [53], Cloud Computing in VANETs (CC-V) [54], Vehicle using Cloud (VuC) [54], Hybrid Vehicular Cloud (HVC) [54], Vehicle-to-Cloud (V2C) [42], Vehicular Cloud Networking (VCN) [55], and VANET-Cloud [42], we use the term Vehicular Cloud Computing (VCC) to refer to the paradigm that allows two types of cloud to be used in an isolated or integrated way: VC and TC, as shown in Figure 2. In this way, computation offloading can be done for both vehicles nearby and for servers in a remote data center. Thus, if vehicles need more access to services, greater computing capacity, energy and storage and greater server stability (with geographically fixed data centers), traditional cloud servers can be used [43], [31], [44].

E. VEHICULAR EDGE COMPUTING (VEC)

Even if some similar terms are used as Vehicular Mobile Edge Computing (VMEC) or Vehicular MEC [56], Vehicular Multi-access Edge Computing [57] and Vehicular Edge Network (VEN) [58], we use the term Vehicular Edge Computing (VEC) to refer to the paradigm that allows two cloud types to be used in an isolated or integrated way: VC and Edge, through V2I connections (purple lines), as shown in Figure 2. In this way, computation offloading can be done for both vehicles nearby and for edge servers. In this paradigm, TC can be used to manage the resources of edge servers [59]. However, in general, TC is not used to perform services requested by vehicles [47], [32], [48]. Thus, computing tends to be limited at most to edge servers with in a few hops from vehicles. In VEC, these servers are computing platforms isolated from the rest of the network available only to users within the Radio Access Network (RAN), can operate with little or no Internet connectivity, and can be in a remote location connected to or disconnected from the data centers of the traditional cloud [38], [60].

F. VEHICULAR FOG COMPUTING (VFC)

Though there are some similar terms such as Fog Vehicle Computing or Fog Vehicular Computing (FVC) [61], we use Vehicular Fog Computing (VFC) to refer to the paradigm that allows to use three cloud types in an isolated or integrated way: VC, edge and TC, as shown in Figure 2. In this way, computation offloading can be done for vehicles nearby, edge servers, and for servers in a remote data center. VFC extends the fog computing paradigm to vehicular networks. Thus, VFC is also hierarchical and provides computational resources and a continuity of traditional cloud services anywhere, from the cloud, through the core and the edge of the network, to the end devices, instead of performing computations only at the edge of the network. VFC is capable of handling applications with a variety of QoS requirements, as applications can be run at a hierarchy level that provides adequate processing capacity and meets latency requirements [38], [60], [62], [63], [64], [65], [66], [67], [68].

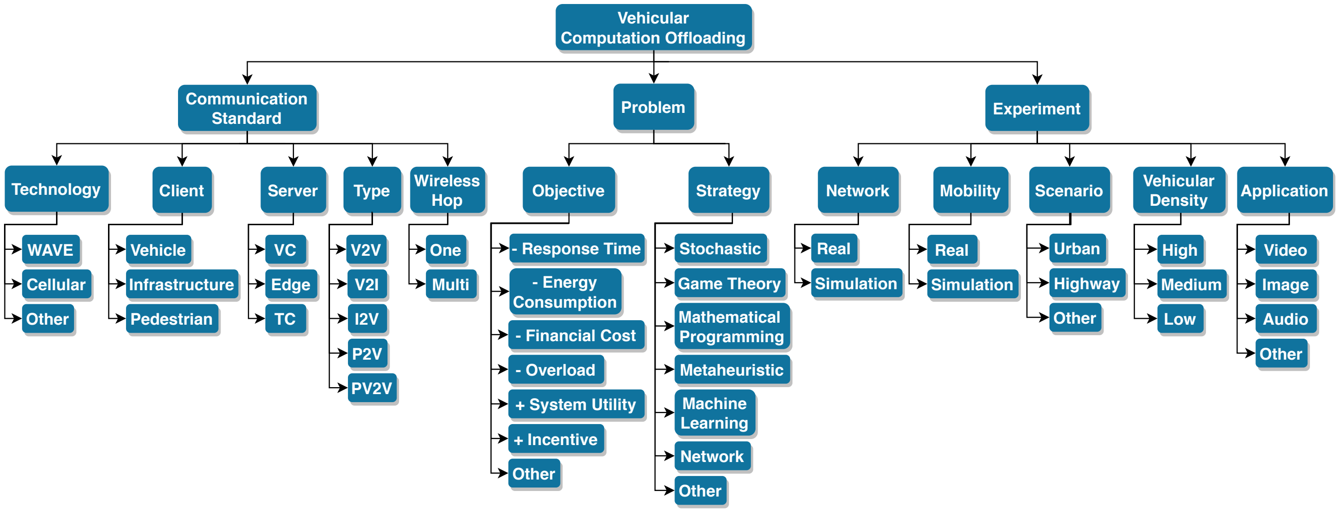


FIGURE 3. Proposed taxonomy of computation offloading in vehicular networks. The nodes of this taxonomy tree guide the reading of Section III. They indicate the subsection where the sub-category is discussed. Thus, nodes on the second level of the tree indicate subsections identified by uppercase letters; nodes on the third level of the tree indicate the subsections identified by numbers; nodes on the fourth level of the tree indicate subsections identified by lowercase letters.

III. TAXONOMY

In this section, we present a taxonomy of research in computation offloading in vehicular environments, that is the core of this survey. This taxonomy allows a classification of the different research articles in this area. The taxonomy is presented in Figure 3 and the articles of computation offloading in vehicular scenarios are classified in different categories in Tables 2-4 on the final pages of this survey. In addition, in this section, we describe the main subjects covered in the articles and categorize the works according to the proposed taxonomy. We also synthesize the classification data for these articles and present them through graphics.

According to Figure 3 and Tables 2-4, the main taxonomy categories are: "Communication Standard", "Problem", and "Experiment". "Communication Standard" categorizes communication technologies, client and server, type of communication in relation to uploading tasks and number of wireless hops. In "Problem", the objectives sought by the works are presented as well as strategies for formulating the problem, algorithms and techniques to solve the problem. Finally, in "Experiment", the details of the experiments carried out by the works are described as type of network, mobility, scenario and application, in addition to a classification in relation to vehicle density. Each of these categories has subsections that further detail the subject and present examples of articles that are in each of the subcategories.

Below, we detail each category and subcategories of computation offloading in vehicular environments.

A. COMMUNICATION STANDARDS

In computation offloading on vehicular networks, choosing how to communicate is an important decision. In this section, we summarize the details of the communication standards that previous works used to provide computation offloading on VANETs. For further details on the works, see Table 2 at the end of the paper.

1) Technology

The Internet of Vehicles (IoV) is a paradigm that has attracted growing interest from the scientific community, government agencies and the automotive industries. This paradigm has been proposed to provide collaboration between vehicles and reliable Internet services, and to improve the experience of drivers and passengers [2], [69], [70]. To enable these communications, different technologies have been proposed, as seen in Figure 4(a). These communication technologies are presented below.

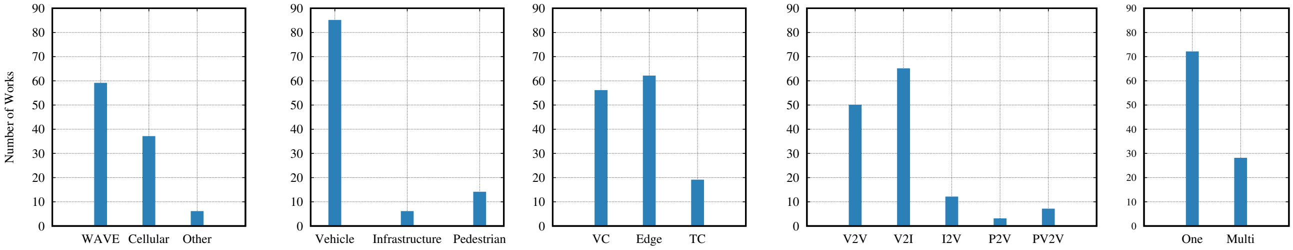
a: Wireless in the Vehicular Environments (WAVE)

The WAVE architecture is a family of protocols standardized by the IEEE for communications in vehicular networks. This architecture was created to offer safe and convenient communications in Intelligent Transportation Systems (ITS) and provide vehicles with direct connectivity to other vehicles (V2V) or infrastructures (V2I) [71], [72], [73], [74].

Some protocols of the WAVE architecture stand out. For example, IEEE 1609.1 is responsible by synchronization of the On-board Units (OBUs) and Roadside Units (RSUs). IEEE 1609.3 is responsible for network and transport layers. Through the Logical Link Control (LLC) layer, it can choose to use WAVE Short Message Protocol (WSMP) (to provide lower latency) or the TCP/UDP/IP stack. IEEE 1609.4 enables the multi channel operation, and prioritization of packets. IEEE 802.11p defines the Physical Layer (PHY), and the Media Access Control (MAC) layer. Some of its features are: operating ranges of up to 1000 meters, random MAC address and wildcard Service Set Identifications (SSIDs) [72].

With respect to the band spectrum of the physical layer of WAVE, a well accepted standard is called Dedicated Short

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access



(a) Technology. (b) Client. (c) Server. (d) Type. (e) Wireless hop.

FIGURE 4. Number of works according each classification of communication standard. These numbers do not refer to percentages, but to the number of articles that used each item in the categories. It is also possible that a work has used more than one item in each category (e.g. an article that used the WAVE and cellular items of the technology category in the same work). These numbers were counted according to the 100 works listed in Table 2 at the end of this paper.

Range Communications (DSRC). DSRC has an allocation of spectrum in the range of 5.85-5.925GHz. It is structured in 7 channels of 10 MHz, having one control channel (for security and control through WSMP messages) and the others being service channels available for different uses [72].

As the WAVE architecture connects vehicles with other devices via wireless communication, some works use it to enable computation offloading in vehicular scenarios, such as: [75], [76], [77], and [78].

b: Cellular networks

Cellular networks are high-capacity and high-speed communication networks and can be defined as a radio network in which the coverage area is divided into cells. Each cell contains a base station (BS) comprising transceivers and control units, and operates in its own frequencies. Each BS can serve several user equipments (UEs) operating within that cell [79]. Over the years, different generations and technologies of cellular networks have been proposed. Although third generation networks (3G) have been used in some works (as in [80]), below we will focus on the most used cellular networks in computing offloading in vehicular environments.

- Fourth generation networks (**4G**) are cellular networks based mainly on standards developed by the 3GPP and codified in ITU called Long Term Evolution (LTE) and LTE Advanced. These networks use Orthogonal Frequency Division Multiple Access (OFDMA) and provide low latency, higher throughput, and improved QoS. Typical data rates of 4G systems are 3-5 Mbps. Enhanced versions of 4G networks incorporate new technologies such as Multiple Input Multiple Output (MIMO) and carrier aggregation that allow higher data rates [81], [82]. Some works use 4G to provide computation offloading in vehicular environments as [83] and [84].
- Fifth generation networks (**5G**) are systems for cellular wireless networks standardized by 3GPP and ITU-R. Some features of 5G are: uses new radio technologies such as millimeter wave (mmWave), Massive MIMO channels, and beamforming, has small cells with lower coverages and high frequencies, allows device-to-device (D2D) communications and suffers with obstacles on the way. 5G systems have the potential to improve current systems and achieve massive data rates (up to 20 Gbps) with lower latency, better mobility, and include new sets of application such as connected vehicles and Internet of Things [85], [81], [86], [87], [88], [89]. 5G was used for computation offloading on VANETs in papers such as [90].

c: Hybrid

Other papers used more than one technology at the same time. For example, [91] used WAVE and 4G to perform computational offloading; [92] used WAVE, 4G and 5G; while [93] used WAVE and 5G.

d: Other

Other works used different technologies to execute computation offloading in VANETs. For example, [94] used Worldwide interoperability for Microwave Access (WiMAX). WiMAX is a technology standardized as IEEE 802.16 and provides wireless internet access with a reasonable data rate through base stations with a range of up to a few kilometers [95]. Other case is in [96] that used a standard for broadband access employing the TV white space (TVWS) band in low population density regions called IEEE 802.22 as technology to provide offloading.

2) Client

In VANETs, a device may need to run a computationally complex application and not have enough resources to run it in a viable time. In this case, the device can apply the computation offloading technique, that allows devices to send tasks to other devices to be executed in order to reduce the task's execution time and save energy [26], [23]. In this approach, such communication generally follows a client-server model, in which the client is the device that sends the offloading request, and the server is the device that receives the request and processes the task. It is also possible that the same device may act as a client and as a server. Several works have chosen different types of clients to request computing offloading, as seen in Figure 4(b). In the presented taxonomy, we identify the offloading clients most commonly found in the literature.

a: Vehicle

The most common approach is when a vehicle, (e.g., bus, car or truck) starts the offloading process, both in ad hoc and infrastructure mode. In this case, the vehicle may be overloaded, not have enough resources to run the application, or want to speed up the response time. Many research works carry out offloading starting in vehicles [58], [22], [97], [98]. The vehicle can initiate the offloading process in various ways. In one of them, the vehicle sends a broadcast request to find servers willing to perform the tasks [99], [75]. Another approach is to take advantage of beacon messages, exchanged periodically between devices, to know in advance who can perform tasks [100], [92].

b: Infrastructure

In this approach, the infrastructure (RSU or BS) starts the offloading process. The requests can come from a service running in the remote cloud or on an edge server. For instance, in cases of missing children (e.g., Amber Alert), vehicles in a certain region may receive offloading requests of several photos to perform face recognition tasks and collaborate to identify the missing child [101]. In other cases, the infrastructure may be overloaded and act as client needing the help of vehicular clouds or other devices [102], [103], [104].

c: Pedestrian

In this case, a pedestrian initiates the offloading process acting as clients. Pedestrians leverage unused resources of nearby vehicles to speed up the execution of tasks and save the battery life of their mobile devices. They can be inside vehicles [105], [106], or be stationary, or be standing/running/walking on sidewalks, roads, and squares [107], [108]. This type of scenario brings challenges. For example, a pedestrian sitting by the roadside can choose to offload to a vehicle that is passing on the road, and depending on the speed of the vehicle, it may be far away when the task processing is finished.

d: Hybrid

This is when the work considers more than one of the types of clients mentioned above. For instance, the works in [109] and [110] use vehicles and infrastructures as clients, while in [111] the authors consider vehicles and pedestrians as offloading clients.

3) Server

As with offloading clients, several entities may act as offloading servers. Several works have chosen different types of servers to execute tasks of computing offloading, as seen in Figure 4(c). Below, we provide a description of the types of servers used.

a: Vehicular Cloud

In the present case, offloading requests are received and executed in vehicles that are part of Vehicular Clouds. In this approach, as seen in Section II-A, the resources of one or more vehicles, whether moving or parked, are grouped and treated as computational resources that can be used to provide services, just like in the cloud computing paradigm. Several works create VC and divide the offloading tasks among participating vehicles in a cooperative way such as [112], [113], and [114].

b: Edge

This is when offloading requests are executed on servers located at the edge of the network, close to base stations and RSUs. As seen in Section II-B, due to the proximity of client devices, the edge typically has lower latency than the traditional cloud, but it also has less resources available. The edge can contain one or more servers or even mini data centers to serve clients of the most varied types. As seen in the Figure 4(c), most works have chosen to offload to the edge servers such as [58], [47], and [115].

c: Traditional Cloud

This case is when offloading requests are sent and processed on the traditional cloud. As seen in Section II-C, the traditional cloud allows applications to leverage features such as elasticity, availability, and unlimited resources to speed up execution of tasks. Besides, remote clouds are supposed to have enough resources to attend to requests from several clients and are used in several works as the offloading destination as in [116], [117], and [118].

d: Hybrid

The hybrid approach happens when more than one of the server types, previously mentioned, act as offloading servers, usually with a hierarchical environment. For instance, an edge server can expand its computing resources by using remote cloud servers when the demand is high, and there are not enough resources on edge. In [119], [120], and [121], the edge was used together with the vehicular cloud to provide more computing resources to clients. In [122] and [123], the vehicular cloud was used in conjunction with the traditional cloud. In [116] and [124], the traditional cloud was used along with the edge to increase its computational power. Lastly, in [125] and [92], the three types of servers were used: vehicular cloud, edge and traditional cloud.

4) Type

As mentioned earlier, VANETs have two main types of communications:- V2V and V2I. However, given the variety of entities that can start an offloading request, we argued that offloading may be initiated by vehicles (e.g., bus, car, taxi, etc.), infrastructures (e.g., RSU, base station, etc.), or pedestrians (e.g., person carrying a smartphone, tablet, notebook). Likewise, a variety of entities can act as offloading servers and therefore receive and perform offloaded tasks. Thus, as seen in Figures 4(d) and 5, we observed various types of recommendations in the literature, from the point of view of uploading tasks. We describe these types below.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

a: Vehicle to Vehicle (V2V)

In the case of direct transmission, ad hoc communication is used by the vehicles to offload tasks onto other vehicles using WAVE or other technologies that allow D2D communication. If it is an indirect transmission, when there are several hops in the path, V2V can be used as a way to forward the task to the final destination. The following works used V2V communication: [78], [127], and [128].

b: Vehicle to Infrastructure (V2I)

This type of communication allows vehicles to communicate directly with RSUs or base stations to send tasks of offloading to edge servers, micro data centers, traditional clouds or even other vehicles (passing through the infrastructure) via cellular, WAVE or other networks. Several works use this approach to upload a offloading task as in [129], [130], and [76].

c: Infrastructure to Vehicle (I2V)

In this type of communication, traditional cloud servers or edge servers offload tasks to vehicles through infrastructures such as base stations and RSUs. As seen in Figure 4(d), few works use this approach. Some papers that use this type of communication are [102], [103], and [131].

d: Pedestrian to Vehicle (P2V)

This is when a pedestrian, carrying smartphones, tablets or notebooks, offloads tasks directly from its mobile device or user equipment (UE) to vehicles passing on the road or parked. This type of communication can have the vehicle as the final destination or just as a task relay. Some works that use this communication are: [119] and [108].

e: Pedestrian inside of Vehicle to Vehicle (PV2V)

In this case, a pedestrian, through its mobile device, offloads tasks to the vehicle that he/she is inside. This vehicle can process the task or forward it via ad hoc network, to another vehicle which can either process the task or forward it via cellular or other network, to a RSU or base station and then to an edge or cloud server to process the task. This type of communication has been used in some papers as in [132] and [105].

f: Hybrid

This approach happens when more than one of the aforementioned types of communications are used. For instance, several works have developed approaches based on V2V and V2I communications [133], [91], [134], while other works consider I2V and P2V communications [107], and others consider V2V, V2I and PV2V [122], [123] and so on (for more details refer to Table 2 at the end of this paper).

5) Wireless Hop

To further extend the possibilities of communications, we can also classify the number of wireless hops between clients and servers during offloading operations. We divided the works into two types: those that use only one wireless hop and those that use more than one wireless hop. We can see the number of works that use this in Figure 4(e). Below we describe this division.

a: One-hop

This is when the offloading task goes directly (one-hop) from clients to servers. This one-hop communication can be V2I, V2V, I2V, P2V, and PV2V, for example. This approach avoids unnecessary delays with multiple successive task forwardings and is less susceptible to offloading failures. This is the most commonly used approach and several works use it such as [135], [136], and [137].

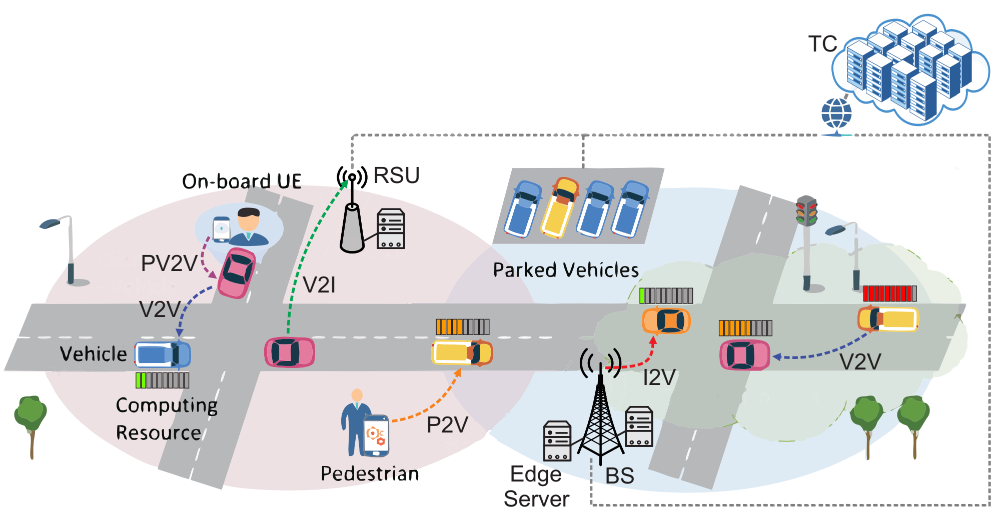


FIGURE 5. Communication types in vehicular networks (adapted from [126]).

b: Multi-hop

This case is when more than one hop (multi-hop) is required for the offloading to go from clients to servers, requiring the use of forwarder or relay nodes/devices. This approach increases the range of communication between client vehicles and servers (vehicles or infrastructure) and can use more powerful computational resources that are further away. For instance, an offloading request that needs to reach the remote cloud may pass through several vehicles until it arrives at a vehicle that has a direct connection with a base station or RSU to forward the task to the desired server. Some disadvantages can be, a higher latency and a higher rate of offloading failures. Some works that used this approach are [90], [121], and [126].

6) Discussion

The communication standards described in this section greatly affect the computation offloading performance of VANETs. For example, on the issue of technology, it may be a good option to use more than one technology to increase the data rate and bandwidth of devices. In addition, it is important to ensure a good adaptation of the new technological trends of communication (e.g. 5G and IEEE 802.11bd [138]) to the vehicle environments. Different technologies are also important for better integration with other network environments such as Internet of Things (IoT), Internet of Healthcare Things (IoHT), Industrial Internet of Things (IIoT), Smart Cities, etc. [139], [140], [141]. However, using more than one technology on devices can make them more expensive and cause problems of heterogeneity. On the issue of clients, it is very important to create solutions that meet and adapt to the specifics of each type, such as: high mobility of vehicles, lower battery capacity of mobile devices of pedestrians, and fixed and limited geographical range of base stations and RSUs. In the matter of servers, there is an important trade-off between latency and computational power. In the case of devices closer to the edge, such as vehicles and pedestrians, there is less communication latency, but also less computational power. In the case of traditional cloud servers, the communication latency is greater, but the computational power is also greater and can compensate. So it is important to carefully consider where to perform the tasks, whether in VC, Edge or TC. In the case of types of communication, as there are few studies that perform P2V, PV2V and I2V, we believe that these cases can be better explored and studied. Finally, in relation to wireless hops, there is also an interesting trade-off. Using one-hop is a more conservative approach and less likely to fail, but it may not exploit the full potential of the network's computing resources. On the other hand, using multi-hop can exploit this potential, but it can generate more failures and delays in the offloading process. After all, if ensuring that two devices are connected is already difficult, ensuring that three or more are connected is even more difficult.

B. PROBLEM

The computation offloading works in vehicular networks propose algorithms, solutions or improvements for the problems with only one objective or several different objectives simultaneously. The objectives range from reducing the response time, improving the usefulness of the system, to reducing financial costs. The proposed solutions or strategies also range from simple algorithms and heuristics to complex mathematical modeling and use of intricate machine learning and metaheuristic algorithms. Below we present the objectives and strategies of the algorithms used in the computation offloading works in VANETs.

1) Objective

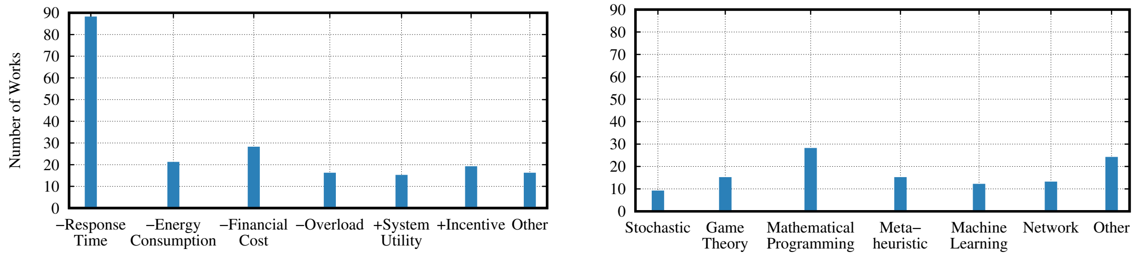
In offloading works on VANETs, several algorithms have been proposed for different objectives. Most works have more than one purpose, so they are multi-objective. Others use only one objective, but they all aim to improve some of the offloading systems and applications while dealing with the challenging scenarios of vehicular wireless networks. The next subsections present the main objectives studied by the works listed in this survey, as shown in Figure 6(a).

a: Decrease Response Time

Some vehicular applications, although they have become popular, are computationally complex, intensive, and of real time [93]. Taking too long to process an application's tasks can compromise its performance, data validity and even the safety of humans in a vehicle. Thus, reducing response time of applications (also called task processing time and computation overhead) is the main objective of the computation offloading technique. However, sending tasks to be processed on other devices on the network can be quite challenging in vehicular scenarios and it can also have its delays of transmission/reception and processing (e.g., a bad decision would be to send tasks to already overloaded devices). Thus, in fact, according to Figure 6(a), this is the most researched objective in this area. Several studies have proposed algorithms to decrease the response time of applications [93], [100], [121], [90].

b: Decrease Energy Consumption

Computationally intensive applications with critical time constraints also pose the challenge of excessive energy consumption [142]. This can happen with user equipment (UE) such as notebooks, wearable devices, tablets and smartphones as well as with vehicles. In the case of vehicles, large amounts of computational operations are performed on their on-board computers. The expectation is that the demand for vehicle computing resources will continue to increase exponentially with the development of autonomous vehicles [143]. Thus, it is important to analyze whether energy consumption for complex applications can compromise the maximum distance traveled by vehicles, especially for electric vehicles [142]. For this reason, some works perform



(a) Objective.

(b) Strategy.

FIGURE 6. Number of works according each classification of problem. These numbers do not refer to percentages, but to the number of articles that used each item in the categories. It is also possible that a work has used more than one item in each category (e.g. an article that used the stochastic and metaheuristic items of the strategy category in the same work). These numbers were counted according to the 100 works listed in Table 3 at the end of this paper.

computation offloading in order to reduce the energy consumption of vehicles [130], [144], [142].

In the case of in-vehicle UEs, they may have limited battery capacity and may be able to save energy by offloading tasks to other devices. However, as with vehicles, it is necessary to know the right portion of tasks to be transferred. This is because transferring tasks to other devices also consumes energy [106]. Some works made it their objective to reduce the energy consumption of UEs that are inside vehicles [106], [47].

c: Decrease Financial Cost

Another objective stated by some papers is to reduce the financial cost when offloading. For example, vehicles may have to pay a fee for computation and communication services. In the case of cellular communications, the vehicle may have to pay for the transmission and reception of data. In the case of computing, it may be that the server device (vehicle or edge server) also charges a fee to process the client's tasks. From the server's point of view, it can benefit economically by providing computing services. However, it may have to pay for electricity from grid operator and for renting wireless bandwidth from network operator [96]. Some of these costs may vary depending on hardware, technology, location and time and can depend on energy consumption [145], [146].

Several papers on VANETs have proposed algorithms to reduce the financial cost of computation offloading [96], [145], [147].

d: Decrease Overload

Some offloading works on vehicular systems also focus on reducing computational loads on overloaded devices. Such devices end up needing offloading to alleviate their loads, meet the QoS requirements of the running applications, and maintain connectivity and system stability. These devices can be both vehicles and edge servers. This overload can be caused by factors such as multiple clients choosing the same server to process their tasks, low computational capacity, or many computationally intensive application tasks being performed simultaneously. Some papers have the specific objective of reducing the overhead of devices such as [131] and [148]. Decreased overload can also refer to the reduction of network overload, in some cases [148].

e: Increase System Utility

In computation offloading in vehicular networks, it is very important that there is a good balance in the use of system resources. That is, the ideal scenario is when there are no overloaded devices and not many idle devices. Thus, the objective of many works is to increase the system utility by better balancing workloads between devices that can act as servers. Thus, tasks will be better distributed, resulting in better QoS for applications and less overloaded devices. Some works have proposed solutions to maximize the system utility of offloading in vehicular networks, such as [135] and [118].

f: Increase Incentive

Some previous work assumes that all devices will share their resources unconditionally. However this assumption is very optimistic for practical implementations. Due to increased delays and processing overload, selfish vehicles may be reluctant to act as servers unless they are rewarded in some way. Without proper encouragement, device owners are not motivated to share their computing resources. In addition, issues such as preference of shared resources, amount of available resources and transparency of information are not the same for all devices on the network. Thus, some works also note the need for more incentive mechanisms to have more devices acting as servers in the network and optimizing the economic benefits of those involved (e.g. the reward being paid in money) [119]. Thus, incentive mechanisms encourage voluntary devices to contribute their computational resources and to mitigate eventual overloads or lack of resources. However, care must be taken that these devices do not become greedy to maximize their profits even if they are already overloaded [149]. Therefore, some studies have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

proposed algorithms to increase incentives, with restrictions, in vehicular networks such as [119] and [149].

g: Other

Other objectives found in the works include: reduce the loss of quality of service experienced by the user [136], increase offloading reliability [150], and enrich the user experience [151].

2) Strategy

Following the taxonomy, in this section we deal with the algorithmic strategies used for solving the problem at hand. The following strategies were found: stochasti, game theory-related, mathematical programming, heuristics/metaheuristics, and machine learning methods.

a: Stochastic Methods

Stochastic processes have been used for problem modelling in several works. The Markov decision process and semi-Markov decision processes are the ones that were highlighted in [111], [120], [122], [123], [152]–[155].

b: Game Theory

Game theory is the science that studies the interaction between cooperating or competing individuals, and it is usually used in computer science for modelling applications such as optimization problems [156]. The game theory approaches that were employed the most were Stackelberg game [130], [144], [149], [157], [158], contract theory [119], [159], [160], and matching theory [161], [162].

c: Mathematical Programming

In our taxonomy, we consider mathematical programming techniques for solving optimization problems. In short, an optimization problem is a problem of finding the maximum (minimum) value of a function called objective function, subject to different restrictions, and can be formulated as follows [163].

$$
\begin{aligned}
min\ F(x),\ x \in \mathbb{R}^n \\
Subject\ to: \\
h_i(x) \geq 0, i \in 1, ..., m \\
j_i(x) = 0, i \in 1, ..., p,
\end{aligned}
\tag{1}
$$

where x are the decision variables, which can take either discrete or continuous values.

There are several classes of optimization problems, and for each class different algorithmic techniques are employed. For instance:

- In linear programming, the optimization problem has linear objective function, linear equality and linear inequality constraints. The works in [91], [96], [98], [114], [118], [164], [165] formulated their problem as a linear programming one, and Lagrangian relaxation is the technique used in majority of the works.
- In integer programming, the decision variables must be integers. There is also a variant of the integer programming for which some, but not all, variables must present integer values, called mixed-integer programming. Some works also formulate the problem as integer and mixed-integer programming [103], [120], [136], [148], [166], [167] and well know algorithmic approaches, such as branch-and-bound [168], cutting planes [169], and dynamic programming [150] can be seen.
- Some authors also formulate the problem as a convex optimization problem [47], [106], and mathematical programming techniques, such as alternating direction method of multipliers (ADM) are among the most commonly employed ones.

d: Metaheuristic

This subsection of our taxonomy is a direct consequence of the last one, as the decision version of diverse optimization problems is NP-complete. Therefore, it might be costly to solve bigger instances of such problems to optimality. In this sense, metaheuristics are algorithms that do not guarantee to deliver a proved optimal solution to a given instance of a problem, but usually return a good solution in a feasible time [170]. More specifically, metaheuristics are general and higher-level algorithms that incorporate operators designed to avoid getting stuck in a local optimum, called intensification and diversification operators [171].

Among the selected works, a great majority employ metaheuristics for solving optimization problems. The highlights are particle swarm optimization (PSO) [114], [115], [128], ant colony optimization (ACO) [75], [92], and genetic algorithm (GA) [77], [102], [149], [151]. Other examples of metaheuristics are bat algorithm [83] and iterated local search (ILS) [154].

e: Machine Learning

According to Mitchell [172], machine learning is a field of artificial intelligence (AI) that aims at improving a given algorithm automatically though experience. Among the collected works, support vector machine (SVM) [129], adaptive learning [100], reinforcement and deep reinforcement learning [14], [104], [108], [117], [146], [173], [174] are the machine learning techniques that stand out.

f: Network

Network-related strategies are approaches to network management through configurations and arrangements adopted to improve performance or achieve specific objectives. In this sense, two strategies were frequently adopted by offloading works on VANETs: Software-Defined Networks (SDN) and clustering. SDN is an architectural approach that simplifies and optimizes network operations by bringing interactions between applications and devices (real or virtual) and network services closer together, making them programmable. This is achieved by employing a central logical control

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

point, also called an SDN controller, which orchestrates and mediates interactions between applications and network elements. For this, the controller makes use of interfaces and the separation of data and control plans [175], [176]. The clustering strategy is a management technique that organizes nodes into a set of groups called clusters based on pre-defined criteria such as network load balancing, affinity, etc. Each cluster has one or more leaders called Cluster Heads that collect data from other nodes in the cluster called members and send the data (usually merged with various other data) to other devices on the network. This can decrease interference, cost, energy consumption and inefficiency [177].

Some works have used SDN as a strategy to improve offloading performance such as [153], [148], and [178]. Others have used clustering such as [46], [83], and [179].

g: Other

Few works have also employed graph theory algorithms [113] and fuzzy logic [127] to solve the problems. In addition, many works proposed problem-specific algorithms [120], [131], [142], [180].

3) Discussion

The formulation of the problem and the way to solve it is also of fundamental importance for computation offloading in vehicular networks. In general, the most common objective in problem formulations is to decrease the response time of applications. In addition, other objectives are also widely used, such as reducing energy consumption and financial cost. However, if trying to achieve a objective in a dynamic environment as vehicle networks is already challenging, trying to achieve more than one objective becomes even more complicated. Therefore, considerable efforts are still needed to obtain solutions that are suitable for vehicular environments and that achieve multiple objectives.

In this sense, several proposals have been published in the literature, such as: algorithms of mathematical programming, stochastic models, metaheuristics, game theory, machine learning, etc. However, producing valid and feasible complete solutions that minimize the objective function while achieving the objectives formulated in the offloading problems still requires major research efforts. One of the approaches that has stood out is machine learning. In this respect, as they are algorithms that need a lot of computational resources, there is also an important trade-off. For example, using an algorithm with deep learning can result in better choices for the offloading process, but it can take a lot of time and computational resources to reach the best result. On the other hand, using weak learning algorithms can give the result in less time and spend less computational resources, but it may not generate a good decision or result for the offloading process. In addition, it is important to analyze the use of the mixture of learned features offline/statically along with learned features online/dynamically, transfer learning and domain adaptation.

C. EXPERIMENT

To validate new proposals (algorithms, schemes, frameworks, systems, etc.) in the area of computation offloading in vehicular environments, it is very important to make good choices regarding test tools, scenarios and applications. In this section, we review the details of the experiments that previous works have used in their proposals related to offloading in vehicular networks. For more details of the works used, see Table 4 at the end of the paper.

1) Network

New systems, applications, protocols, etc., appear with increasing frequency as technologies evolve. This leads to the need to be able to quickly test these research and development proposals, so that they can be validated as quickly as possible [181]. To carry out and validate these experiments at the network and computation level, the works use different approaches (7(a)). Below, we detail the main approaches.

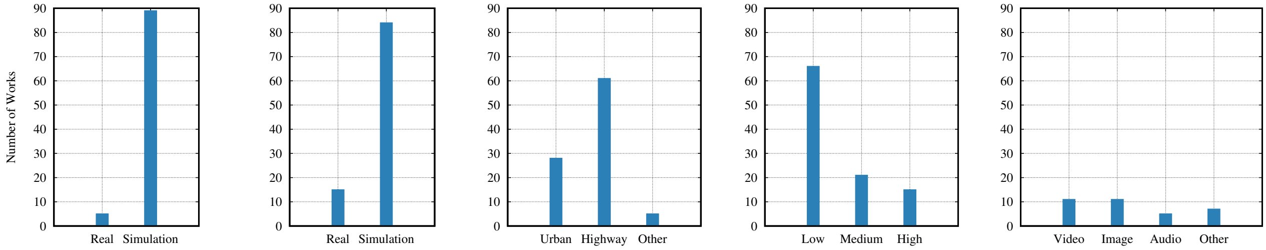
a: Real

Real network experiment is a technique that employs an experimental setup that consists entirely of real network systems, equipments, protocols, and applications so that various parameters can be quantified and the performance of the system can be assessed [181]. Real experiments can be performed by using the developed prototypes in the vehicular driving environments so that implemented services can be better evaluated. Performance metrics obtained in real scenarios provide more reliable values for analysis due to constraints such as power consumption, volume data processing, high-latency links, among others. Real experiments, however, have disadvantages such as a high financial cost incurred, and less control over the assessed environment. Few works have used real network experiments to evaluate their proposals of offloading in vehicular networks. Among these are [180], [76], and [182].

b: Simulation

In order to reduce the high costs in vehicular networks, the use of simulators becomes a suitable alternative. Network simulation is an experimentation technique that employs a setup that consists entirely of computer models of network systems, applications and protocols. Although this type of experiment allows controlled and reproducible tests, the lack of real equipment and components in a simulation leads to a lack of realism in the results. Even so, simulation is the most widely used technique for carrying out network experiments during the development and research stage [181], as seen in Figure 7(a).

Simulation tools provide multi-access control, resource management, and measurements of packet delivery ratio and delay, as well as, different models to evaluate networks such as ad hoc, sensors, and optical networks. Some simulators used to do computation offloading in VANETs were: OMNeT++ [183], ns-3 [184], and MATLAB-based environments [185]. For example, in [97], [75], and [131] OMNeT++ was



(a) Network. (b) Mobility. (c) Scenario. (d) Vehicular density. (e) Application.

FIGURE 7. Number of works according each classification of experiment. These numbers do not refer to percentages, but to the number of articles that used each item in the categories. It is also possible that a work has used more than one item in each category (e.g. an article that used the urban and highway items of the scenario category in the same work). These numbers were counted according to the 100 works listed in Table 4 at the end of this paper.

used; in [83], [94], and [128] ns-3 was used; and MATLAB-based environments were used in [96], [157], and [186].

2) Mobility

An important feature in VANETs is the mobility of the nodes. The nodes can move at high speeds because they are the vehicles themselves. These nodes follow the driver's behavior, being able to bend on a street, change direction or perform other sudden maneuvers. This can cause frequent changes in the network topology and a short connection time between the nodes, since they have a limited communication range. In fact, in VANETs, very few nodes remain connected for a long time [187]. However, there is a certain predictability of the movement of the nodes because they have to follow the patterns of the traffic routes, such as: direction of the road, traffic lights, speed limits, physical limits of road width etc. [188] [189]. Experimenting to replicate these characteristics is a challenge. Below, in this sense, we present what the computation offloading works have used, as can be seen in Figure 7(b).

a: Real

Some configuration parameters of road traffic simulator and mobility generator are often hard to set in simulated environments and may not provide adequate realism. In contrast to this, mobility studies in real scenarios provide more accurate results. Real experiments using 4G network in Aalborg, Denmark can be found at [190]. Some real world environments were also created for testing at Aldenhoven Testing Center (ATC) and Mcity (a test facility by the University of Michigan, USA) [191]. Some works used real mobility for offloading, such as [180], [182], and [84].

Although we have much more reliable data in real experiments, the main disadvantages of the mobility experiments in real scenarios are the high cost and the risks involved. For example, the costs of an autonomous vehicle can exceed US$ 300,000 without taking into account the additional expenses with scenery and other edge devices [192]. Furthermore, real experiments demand multiple individuals, equipment and vehicles and the situation depends on the other vehicles. Therefore, another alternative to have a real mobility of the vehicular offloading environment is to collect GPS data from previously chosen vehicles [174], [155] (e.g., taxis in a city) [136], [133] or traces of known vehicle routes (e.g., buses) [125], [151].

b: Simulation

Due to economic issues, logistic difficulties, and technology limitations, simulation tools are a widely adopted choice for validation of experiments in VANETs. A critical aspect is to approximate the data generated by simulators with data which reflects the real world behaviour of vehicular traffic as closely as possible. This behavior must be both macro-mobility (with macroscopic aspects such as road topology, per-road speed limits, number of lanes, safety rules, traffic signs, etc.) and micro-mobility (with drivers' individual behavior as interacting with other drivers, speed and acceleration in different traffic conditions, overtaking criteria, etc.) [193]. Thus, simulators are becoming increasingly robust to overcome this gap with realistic mobility models for the feasibility and validity of the research [194]. Another challenge related to the use of simulators is the compatibility with network simulators, since mobility simulators cannot be used to validate network experiments in isolation [195]. However, realistic mobility simulators as SUMO [196], VanetMobiSim [193], and PTV Vissim [197] already have ways to integrate with different network simulators.

In order to provide vehicular mobility and help in the generation of scenarios to validate the offloading experiments, the mobility simulator most used by the works was SUMO [160], [107], [92]. VanetMobiSim was also used [128].

Some works used real and simulated mobility [136], [133], [182]. However, they were not simultaneous experiments as if it were an emulation.

3) Scenario

Realistic vehicular scenarios are of utmost importance for providing reliable metrics in experiments. Some scenarios are more frequent in computation offloading for VANETS, such as urban and highway. Others are less frequent such as parking lots and university campuses. Defining a vehicular scenario is also important to evaluate models that can be used in specific situations. In this way, computation offloading algorithms can be designed to adapt more efficiently

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

to different scenarios. The following subsections present a description of the vehicular scenarios, as shown in Figure 7(c).

a: Urban

Urban areas are regions in which relevant traffic is evidenced. The presence of many segments with intersections makes communication and routing decisions more complex to deal with. Besides that, obstacles such as trees and buildings, high density of vehicles, 2D vehicular mobility, presence of pedestrians, viaducts and tunnels are difficult factors in this regard [198]. A scenario widely used as an urban environment is the Manhattan mobility model (Figure 8), which has streets organized in the form of a grid [199].

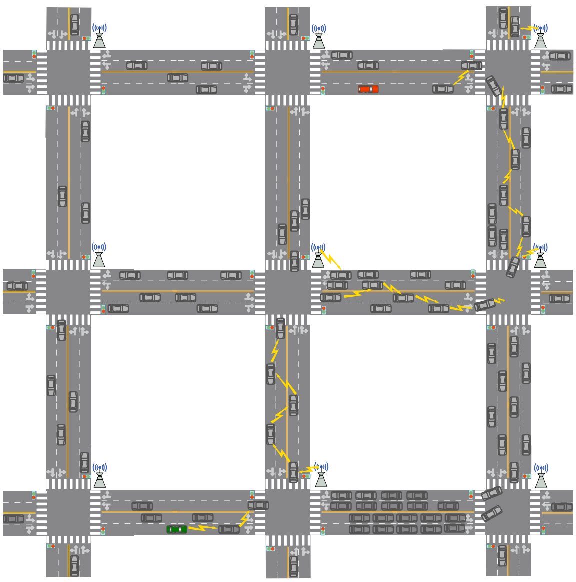


FIGURE 8. Urban scenario in the Manhattan model (adapted from [200]).

The urban scenario was used in several works of computation offloading, such as: [153], [90], and [106].

b: Highway

Highway scenarios (Figure 9), also called freeway scenarios, are generally characterized by a single road (with one or more lanes in each direction), 1D vehicular mobility, high speed vehicles, few obstacles, stable connection if the vehicles travel in the same lane or in the same direction (e.g., platooning) and unstable connection if the vehicles travel in opposite directions [201]. The provision of continuous connectivity or coverage is a major challenge in highway scenarios. High speeds and long distances require RSUs to be deployed efficiently for cost reduction [202]. Constant modifications of network topology in highways with no fixed structure creates a challenging technical issue. Frequent interruptions are also an obstacle in V2V communications due to diverse velocities of vehicles and short connection times for vehicles in opposite directions [203].

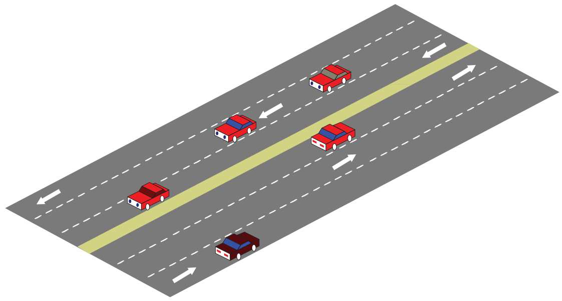


FIGURE 9. Highway scenario (adapted from [204]).

According to Figure 7(c) and Table 4, the scenario most used by the works to carry out offloading experiments was the highway [100], [46], [149]. Some works also used the highway and urban scenarios in the experiments such as [160] and [155].

c: Other

Other scenarios used less commonly are parking lots, university campuses, and industrial parks. In the case of parking lots, parked vehicles can be used to process tasks or to share communication resources [42]. Parked vehicles can be seen as static infrastructure, as RSUs, and help other nodes in the network. Since RSU infrastructure can significantly increase costs, demanding high maintenance overhead, parked vehicles offer a possibility to mitigate this problem [205]. These parked vehicles generally do not change location for long periods of time. With the help of power supplies, such as rechargeable vehicle batteries built into vehicles, parked vehicles can process tasks when their engines are turned off. In fact, the energy consumption for processing tasks can be very small, compared to other moving vehicle activities [66]. Some computation offloading works use scenarios of vehicles parked in the experiments [80], [66].

In the case of university campuses and industrial parks, these are scenarios characterized by low vehicle density, low vehicle speed and traffic generally limited to authorized vehicles. Some of these scenarios were used in [180] and [182].

4) Vehicular Density

VANETs rely heavily on having vehicles nearby so that they are able to exchange information and messages, especially on networks without infrastructure. Therefore, vehicular density is an important factor in these networks [206]. Thus, in the next subsections we describe the types of vehicular density and we show some works that used each type, as shown in Figure 7(d).

a: Low

Low vehicular density scenarios, also called sparse scenarios, have few vehicles to exchange information with each other and maintain good network connectivity. Thus, low densities can cause loss of messages and network packets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

due to reduced communication capabilities [206]. Vehicular densities are considered low when the vehicular density is approximately 11 vehicles/km in a highway scenario and 25 vehicles/km^2 in an urban scenario [207], [206]. As shown in Figure 7(d), low was the vehicular density most used in the offloading experiments [129], [96], [13].

b: Medium

Medium vehicular density scenarios are intermediate scenarios between low and high density scenarios. They have a larger number of network nodes, with better connectivity, and generally without traffic jams. Vehicular densities are considered medium when the vehicular density is approximately 55 vehicles/km in a highway scenario and 120 vehicles/km^2 in an urban scenario [207], [206]. This approximate density has been used in experiments in some offloading computation work [103], [173], [115].

c: High

High vehicular density scenarios have better connectivity because there are more vehicles in the network that are more likely to be within the communication range of others. However, these scenarios suffer with traffic jams, mainly during peak hours, and can provoke reduced message delivery due to packet collisions, redundancy, and contention at MAC and physical layers, caused by simultaneous forwarding, also known as broadcast storm [206]. Vehicular densities are considered high when the vehicular density is approximately 120 vehicles/km in a highway scenario and 250 vehicles/km^2 in an urban scenario [207], [206]. According to Figure 7(d), the high density was the least used in the offloading experiments [153], [165], [112].

Some works have also used more than one type of vehicular density in offloading experiments, such as: [135], [116], and [178].

5) Application

Applications in VANETs are most commonly divided into safety applications and comfort applications. Safety applications are responsible for preventing accidents, improving road safety, saving people's lives, and increasing the driver's ability to react in various situations during the trip. They are the most important applications and, therefore, a priority. Comfort applications, or non-safety applications, aim to make travel more pleasant for drivers and passengers, through information and entertainment, opening up the possibility of commercial activity to VANETs applications and increasing traffic efficiency [9], [208]. Although this classification is important and systems differentiate applications by their priority, the type of data that will be processed has a greater impact on application performance and offloading systems. For example, the size of the data affects the offloading transfer time and the complexity of operations on that data affects the offloading processing time. Thus, we classify the applications according to the type of data that will be processed: video, image, audio and others [209], [210]. In addition, most of these applications consider tasks independent of each other [100], [164], [92], although some consider the dependency between tasks [102], [91].

Next, we will see a description of these applications, as shown in Figure 7(e).

a: Video

Video-related applications in vehicular networks can be used for both safety and comfort. For example, they can increase safety through applications such as 3D scene reconstruction, augmented reality, emergency video call, real-time navigation, overtaking assistance and surveillance systems, in addition to a wide range of possibilities. This type of application can also be used in comfort applications such as online games, video streaming, and tourist information. Although video file sizes vary (depending on quality, amount of time, etc.), they are usually larger than files of other data types [209]. In addition, video applications can cause a long processing delay and consume a lot of energy (e.g., augmented reality) [211]. For this reason, offloading this information between two or more devices in VANETs, with the desired quality, delay and resolution, is a big challenge for this type of application [212], [213].

Some works have tested application tasks related to video in computation offloading systems on VANETs. For example, [46] tested a 3D scene reconstruction video application; [102] used a video navigation application in the experiments; and [114] used a video streaming application.

b: Image

Image-related applications are used for safety as identification of stolen vehicles through the license plate, systems that warn of hazards, searching for drivers on the road, and recognition of traffic light, gestures, faces, and objects through cameras installed in the vehicle [214], [215], [216], [217], [218], [219]. They are also used for comfort as social networks and contextual images of interest [201]. Since images are generally smaller in size and require less processing time when compared to videos, using images instead of videos can allow a drastic reduction in the data load circulating the network and the processing time of the applications [209]. Even so, some applications related to images have great challenges in offloading schemes in vehicular environments. Therefore, these applications were used in computation offloading experiments. For example, [93] used an application for one vehicle to recognize license plates from other vehicles; [125] used a facial recognition application; and [182] used an object recognition application.

c: Audio

Audio-related applications are also used for safety (e.g., emergency calls, theft detection, etc.) and comfort (e.g., voice chat, guided tour, etc.) [201] and can provide a large amount of information to drivers in a short time and with minimal deviation from attention to driving [220]. Audio applications are based on sound, typically of the human voice. The size of the audio files depends on the quality and amount of time that was recorded. In general, these files are also smaller than video files and require less processing time, depending on the application [209]. In offloading systems, this type of application also has transmission/reception and processing challenges. Some works have used audio applications in their computation offloading experiments on vehicular networks. The most used application was voice recognition [221], [134], [149].

d: Other

The development of other types of applications is also interesting in vehicular environments. Besides that, such applications can benefits from computational offloading for performance improvement. For example, text messaging or character-oriented data offers a lightweight and low operating cost alternative for both safety and comfort applications. It is possible to send critical information to vehicles with less delay and less chance of a bottleneck. Warnings about floods, accidents, and emergency vehicles passing by can be passed quickly and to a large number of vehicles, especially when the network can not handle the sending of this information by other mediums. Also, there is a possibility of commercial exploitation of this type of application, with tourist information, blockchain [35], [222], [223], [224], [225], [226], business suggestions, and events nearby [227]. In general, this type of character-oriented applications or other data types has smaller files, depending on the application, than other types of files [209]. Some types of applications used in computation offloading were: recommendation based on location [137], traffic information, online chatting [149], and real-time financial trading [165].

6) Discussion

The experiments are of great importance for the reliability of new proposals and solutions for computation offloading processes in vehicular scenarios. In fact, if an experiment is badly done or carried out without due scientific rigor, it can compromise the credibility of the results and the proposal may not be valid. Accordingly, the topics presented in this section should be analyzed carefully. For example, consider the question of choosing the testing platform related to networks and mobility. The ideal approach would be to experiment in real environments so that the results are reliable and realistic. However it may require a large amount of financial resources and the environment may need careful verification and configuration. On the other hand, using network and mobility simulators may be cheaper and better to control and replicate, but the results may be less realistic. In terms of scenarios, the type is also very important. Using an urban type scenario implies different street and avenue layouts, usually close to the grid model. A highway scenario usually consists of just one road with lanes with different directions and other scenarios have other configurations. Thus, the scenario impacts the mobility of vehicles, and, in turn, the performance of applications and offloading systems. Vehicular density is another factor that impacts offloading processes. The center of a large city at peak times can have many vehicles traveling and generating and exchanging data. This can cause many network packet collisions, make the network congested and offloading more difficult. In turn, a rural road with very little vehicular traffic may not even have adequate connectivity between the few network devices. Finally, the execution of different applications also affects the computation offloading processes. Generally, video applications have larger files to be transferred and require powerful computational resources. Audio, image and text applications may involve smaller files to be transferred. Thus, it is important that offloading systems adapt to these issues and maintain good performance regardless of which configuration/application is used.

IV. CHALLENGES AND FUTURE DIRECTIONS

Despite significant and recent advances in the field of computation offloading in vehicular networks, there are still important research challenges and open issues. Next, we present the challenges that we have intensively examined in the literature. Such topics can guide future research in the area. These open challenges and issues need further investigation and have been organized in relation to six key topics: network, mobility, security and privacy, incentive, experiment, and algorithm.

A. NETWORK

Network-related challenges greatly affect computation offloading in vehicular scenarios. For example, if the network is very congested, with a lot of contention, collision, noise and interference, the offloading may not happen or happen without success. This can happen due to factors such as a lot of offloading happening at the same time, data being exchanged, broadcast storm, vehicles within the same geographic region, control data or signaling messages (e.g., beacon messages from IEEE 802.11p protocol), transmission of large data (e.g., movies), etc. Therefore, it is very important to periodically monitor the status of the network (e.g., situation and whether the queues at the network interfaces are empty or full). In addition, many of these messages exchanged can be redundant or repeated, which could be resolved using techniques such as multicast or geocast (based on the location of the nodes).

Other network-related challenges are signal attenuation problems, lack of a central coordination point, hidden terminal, obstacles hindering communication (e.g., buildings and trucks) or even little connectivity due to a low number of nodes in the network. If there is multi-hop transmission, a problem that can also hinder is the increased delay as there will be more transmissions, receipts and processing of the packets.

Finally, the heterogeneity of technologies used can increase the network bandwidth but may need better management. For example, devices using WAVE may have different ranges than cellular technologies and not all devices on the network can use the same technologies. Furthermore, newer technologies need to be better studied to be fully adapted into vehicular scenarios such as 5G/mmWave [57], [85], [86] and IEEE 802.11bd (considered the evolution of IEEE 802.11p) [138]. We believe that these network-related challenges need to be better studied in future works to obtain better performance in the computation offloading processes in VANETs. In addition, the new network technologies mentioned offer good research opportunities.

B. MOBILITY

Vehicular mobility also creates some challenges that need to be solved for the offloading to be successful. For example, the fast speed of the nodes causes wireless links and paths to be constantly broken or fragmented, causing vehicles to move out of each other's communication range and connections to be short-lived. Also, since the network may have few nodes and in a matter of seconds or minutes it may already have a large number of nodes, there may be problems with scalability and it will be difficult to maintain a good performance regardless of the number of nodes in the network. In addition, these rapid and frequent changes in vehicle topology and variable node density can lead to a high packet loss rate when density is high due to a greater contention over wireless channels and when density is smaller because connectivity is low. If there is a multi-hop transmission, the probability of success in the transmission ends up being lower since the probability of all the nodes in the path remaining connected is also less.

One solution to these challenges is to use mechanisms for predicting the mobility of network nodes. Some works use an estimated lifetime of links to predict how long they will be active or nodes will be within range of each other. Even so, these estimates are not always correct. This is because drivers' behavior is not always predictable and, generally, nothing prevents them from choosing to turn around the next corner or make a return on a highway. Other approaches use predefined public route information (e.g., buses). However, this information is only available for a very small percentage of the vehicles in the network and even then it is not completely reliable as the vehicle can make extra stops or have other unexpected behavior. Thus, although some papers have investigated this topic, future works can still be directed to better adapt the computation offloading processes to mobility of vehicles.

C. SECURITY AND PRIVACY

Security and privacy is a great concern for any network, and many recent works have pointed out various security issues and solutions that have been come up in the recent past [228], [229], [230]. Security and privacy are also very important topics in computing offloading in VANETs. In general, the principles of cloud computing for security and privacy also apply to devices at the edges of the network. However, the edge deals with much more private information. For example, vehicles may have data on their daily routes and the exposure of that information can be dangerous for personal security. In addition, computation offloading can send sensitive data to untrusted servers (from any person or company and can be a malicious node). This, coupled with the absence of a centralized control point, makes it difficult to create integrated security and privacy policies. A possible solution for this is to offload only non-confidential tasks. However, it can happen that most tasks are confidential and there is no improvement in performance in offloading a few tasks. Another solution is to add encryption or authorization/authentication certificates. However, this can compromise offloading efficiency, Quality of Experience (QoE), and system scalability because it will require more computational power or more time to process tasks [35].

There is also a risk that potential servers can deny service to clients they do not know and this may lead to offloading failure or underutilization of network resources. On the other hand, malicious nodes can upload tasks containing viruses, due to the lack of security strategies. There can also be a Distributed Denial of Service (DDoS) in which the nodes become infected and stop providing computing services. Thus, it is also necessary to ensure secure communication and avoid the spread of viruses or false information. As security and privacy is a priority, there is a need for more studies, research, solutions and mechanisms in this area [26], [231], [232], [233], [234].

D. INCENTIVE

Computation offloading requires collaboration by network devices to share resources such as CPU, storage space, battery, etc. For example, a vehicle can perform tasks for another vehicle or forward data to other nodes in the network using its own computational resources. Transmitting and receiving this information and leaving wireless network interfaces connected consumes energy. In addition, the owner of these resources could benefit more if his/her own applications used the borrowed resources. In this way, the network nodes can become selfish and refuse to share their resources without any compensation. So, the challenge is how to motivate network nodes to share their resources. Although some studies have suggested incentive mechanisms to reward nodes that share their resources (e.g., monetary payment) or analyzed the possibility of the device owners themselves voluntarily deciding, there is still a need for more incentive mechanisms that are appropriate for different scenarios and that are really attractive [26]. Therefore, research efforts must still be directed towards solving the challenges of this topic.

E. EXPERIMENT

To validate the computation offloading experiments in vehicular networks, reliable test environments are necessary. Realistic vehicular communication experiments have a significant impact on the credibility of the results. Although simulators have constantly improved in reproducing realistic traffic patterns and movements, including interaction between vehicles, these simulators can still improve the microscopic modeling of the individual movements of each element of the network

[235]. Replicating through simulations the exact movements of real vehicular traffic is still a challenge.

In addition, products and prototypes must necessarily be tested in real environments before they are deployed, even if simulations have already been used. This is because simulations may not provide adequate realism and real experiments provide more accurate results [181]. However, making these real experiments more accessible is a challenge due to the prohibitive cost for some research centers. In this manner, considerable future works are needed to perform and analyze computation offloading processes in vehicular environments in a reliable and realistic way.

F. ALGORITHM

A computation offloading system is a group of computational components and modules that interact with each other to improve the processing capabilities of a device by allowing the migration of task execution to other devices. Thus, some of the biggest challenges of computation offloading in VANETs are related to the algorithms of the systems. Some of these challenges are listed below.

1) Load Balancing

From the macro point of view of the network, ensuring load balance among all the devices on the network is still an open issue. For example, ensuring that all task processing requests are not addressed only to the most powerful device on the network (often an edge server attached to a BS) or that only a few nodes on the network receive most of the requests is still challenging. Thus, it is very important that the algorithms are designed to make the best possible use of computational and network resources. If not properly managed, few devices can be overloaded and many can be idle at the same time, thus impairing network and application performance.

2) Partitioning

Before performing computation offloading, an application partitioning procedure may be necessary, although not all applications can be partitioned. The purpose of this procedure is to divide the application into tasks that can be performed on different devices. This partitioning can be done through different models, techniques and levels of granularity. In addition, this partitioning can be done automatically by the system or manually by the developers of the application through code annotation or markup [236]. In this sense, an important challenge is how to partition a large and complex application workload in an optimal or near-optimal way. Thus, parameters such as the size and quantity of tasks must be taken into account in order to optimize the upload, download and processing time for the remote devices.

3) Failure Handling

As vehicle networks have dynamic topologies, there are constant disconnections between vehicles and devices. Such disconnections can affect computation offloading processes between clients and servers and result in failures. For example, as can be seen in Figure 10 , orange server vehicle is moving out of range of the red client vehicle without returning processing results. Thus, mechanisms and algorithms are needed to ensure that the application is not affected by the failures. In this respect, some important challenges concern an acceptable level of fault tolerance and ways of recovering from failures. For example, some tasks may be less important and may have the results of their processing discarded. In addition, copies of tasks can be sent to run on different devices, can be kept locally in case of failure detection [93], or, even if the client and server no longer have a direct connection, it can use multi-hop transmissions to deliver processing results or tasks. Thus, despite the need for fault tolerance and failure recovery schemes, it is not a topic covered by the great majority of the papers studied in the present survey and is a key topic of research.

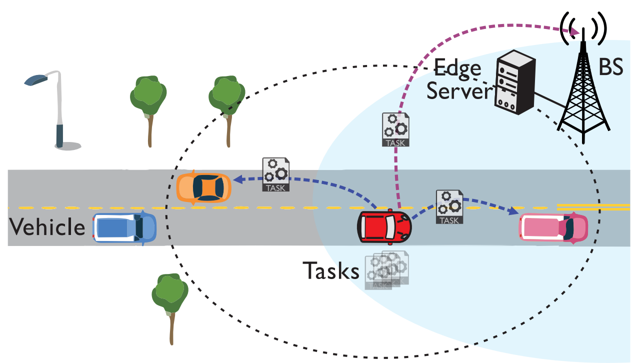


FIGURE 10. Task distribution process in vehicular networks.

4) Policies

Another challenge is related to the policies adopted by the offloading systems. For example, users or applications may have different priorities, Service Level Agreements (SLAs), be classified as gold, silver, bronze, etc., and have different QoS and QoE requirements. It can be a challenge to ensure that these policy of priorities and agreements are correctly applied and managed in the offloading systems. In addition, few studies have addressed this topic in computation offloading in vehicular environments.

5) Discovery and Resource Request

An important step in the offloading processes concerns the discovery and request of computational resources from other devices. Finding the best way to do this can be challenging. For example, using periodic beacon messages to carry contextual information about devices (e.g., location, speed, CPU capacity, etc.) can be a good solution. However, this can increase network overhead. Another solution would be to use traditional request/reply messages to decrease overhead. But that could increase the delay in the offloading process. In addition, at the time of the request, the contextual situation can be in a certain way, and when the tasks are actually sent, the situation may already be different and that information may be out of date. This can hinder the computation offloading process. Thus, further studies on this interesting topic are still lacking.

6) Task Distribution

Perhaps the most important and challenging step in the computation offloading process is the task distribution or task scheduling (the task distribution process can be seen in Figure 10). This step can have several parameters and metrics that need to be taken into account in order to obtain the best possible performance in offloading. In fact, finding the optimal way to distribute tasks in order to have the maximum reduction in application processing time and the minimum percentage of failures in the entire process has been described as a NP-hard problem [75], [94], [102], [114], [169]. Thus, distributing these tasks in an intelligent and optimal way can require considerable computational resources. Below, we list some topics that need to be analyzed to optimize task distribution:

- Dependence or independence of tasks;
- Multiple criteria, attributes, constraints, and objectives (e.g., reducing response time, energy consumption, and financial costs at the same time or reducing response time and still maintaining good data quality);
- Convergence time of distribution decision;
- Different degrees of priority and deadlines of users, applications and tasks;
- Servers with multiple and distinct characteristics;
- Tasks with different complexities, sizes and requirements;
- Reliability level so that the servers are within range of the client when returning the processing result (e.g., spreading replicas of the most important tasks for several servers to process and that at the same time does not congest the network);
- One or multi-hop scenarios;
- Specific technologies such as 5G/mmWave that need to have line of sight to transmit/receive;
- Contextual information;
- Scenarios with large amounts of data and processing (e.g., autonomous vehicles);
- Ideal number of server nodes;
- Whether offloading is worth doing or not;
- Best moment to do the offloading;
- Best number of tasks to be sent;
- Information about user behavior;
- Best place to process tasks (e.g., Local, VC, Edge, TC, or a combination of these).

Therefore, dealing with all or some of these algorithmic issues is a major challenge. Considerable research efforts are required for these topics. In addition, there are several research opportunities related to intelligent algorithms for computation offloading in VANETs, with emphasis on machine learning algorithms.

V. CONCLUSION

Computation offloading is a technique that has potential to improve application performance in vehicular networks. Although the vehicular scenarios are challenging, it is possible to apply computation offloading so as to benefit different types of clients and servers and, consequently, the end users of the applications. Furthermore, computation offloading can be applied in different paradigms such as Vehicular Cloud/Fog/Edge Computing. With the arrival of autonomous vehicles and their applications with large and complex processing requirements, applying this technique will be even more necessary. In this paper, we comprehensively covered the state-of-the-art of computation offloading in vehicular environments. We began with summarizing the various terms and paradigms used in the area. Next, we proposed a taxonomy for existing literature on computation offloading in vehicular environments. Furthermore, we classified a large number of works in this emerging area according to the proposed categories, describing the main concepts used in the works. Finally, we presented problems and challenges that can guide future research.

REFERENCES

[1] World Health Organization (WHO). (2020) Number of registered vehicles. [Online]. Available: https://www.who.int/gho/road_safety/registered_vehicles/number/en/
[2] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," Proceedings of the IEEE, 2019.
[3] A. A. Khan, M. Abolhasan, and W. Ni, "5g next generation vanets using sdn and fog computing framework," in 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2018, pp. 1–6.
[4] K. Wevers and M. Lu, "V2x communication for its-from ieee 802.11 p towards 5g," IEEE 5G Tech Focus, vol. 1, no. 2, pp. 5–10, 2017.
[5] A. B. Souza, J. Celestino, F. A. Xavier, F. D. Oliveira, A. Patel, and M. Latifi, "Stable multicast trees based on ant colony optimization for vehicular ad hoc networks," in The International Conference on Information Networking 2013 (ICOIN). IEEE, 2013, pp. 101–106.
[6] I. Ahmad, R. M. Noor, M. R. Zaba, M. A. Qureshi, M. Imran, and M. Shoaib, "A cooperative heterogeneous vehicular clustering mechanism for road traffic management," International Journal of Parallel Programming, pp. 1–20, 2019.
[7] A. Ullah, S. Yaqoob, M. Imran, and H. Ning, "Emergency message dissemination schemes based on congestion avoidance in vanet and vehicular fog computing," IEEE Access, vol. 7, pp. 1570–1585, 2018.
[8] National Highway Traffic Safety Administration (NHTSA). (2019) Traffic safety and the 5.9 ghz spectrum. [Online]. Available: https://www.nhtsa.gov/speeches-presentations/traffic-safety-and-59-ghz-spectrum
[9] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," Journal of network and computer applications, vol. 37, pp. 380–392, 2014.
[10] MarketsandMarkets. (2018) Automotive v2x market. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/automotive-vehicle-to-everything-v2x-market-90013236.html
[11] S. Zeadally, J. Guerrero, and J. Contreras, "A tutorial survey on vehicle-to-vehicle communications," Telecommunication Systems, vol. 73, no. 3, pp. 469–489, 2020.
[12] A. Boukerche and V. Sotoro, "Computation offloading and retrieval for vehicular edge computing: Algorithms, model and classification," ACM Computing Surveys (CSUR), 2020.
[13] Y. Liu, S. Wang, Q. Zhao, S. Du, A. Zhou, X. Ma, and F. Yang, "Dependency-aware task scheduling in vehicular edge computing," IEEE Internet of Things Journal, 2020.
[14] Y. Sun, J. Song, S. Zhou, X. Guo, and Z. Niu, "Task replication for vehicular edge computing: A combinatorial multi-armed bandit based

approach," in 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018, pp. 1–7.

[15] Intel. (2014) Self-driving car technology and computing requirements. [Online]. Available: https://www.intel.com/content/www/us/en/automotive/driving-safety-advanced-driver-assistance-systems-self-driving-technology-paper.html

[16] V. Chamola, C.-K. Tham, and G. S. Chalapathi, "Latency aware mobile task assignment and load balancing for edge cloudlets," in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2017, pp. 587–592.

[17] V. Chamola, C.-K. Tham, S. Gurunarayanan, N. Ansari et al., "An optimal delay aware task assignment scheme for wireless sdn networked edge cloudlets," Future Generation Computer Systems, vol. 102, pp. 862–875, 2020.

[18] M. U. Farooq, M. Pasha, and K. U. R. Khan, "A data dissemination model for cloud enabled vanets using in-vehicular resources," in Computing for Sustainable Global Development (INDIACom), 2014 International Conference on. IEEE, 2014, pp. 458–462.

[19] A. Coutinho, E. O. Carneiro, and F. G. P. Greve, "Computação em névoa: Conceitos, aplicações e desafios," Minicursos do XXXIV SBRC, pp. 266–315, 2016.

[20] H. Lin, S. Zeadally, Z. Chen, H. Labiod, and L. Wang, "A survey on computation offloading modeling for edge computing," Journal of Network and Computer Applications, p. 102781, 2020.

[21] M. Aazam, S. Zeadally, and K. A. Harras, "Offloading in fog computing for iot: Review, enabling technologies, and research opportunities," Future Generation Computer Systems, vol. 87, pp. 278–289, 2018.

[22] B. Li, Y. Pei, H. Wu, Z. Liu, and H. Liu, "Computation offloading management for vehicular ad hoc cloud," in International Conference on Algorithms and Architectures for Parallel Processing. Springer, 2014, pp. 728–739.

[23] P. A. Rego, P. B. Costa, E. F. Coutinho, L. S. Rocha, F. A. Trinta, and J. N. de Souza, "Performing computation offloading on multiple platforms," Computer Communications, vol. 105, pp. 1–13, 2017.

[24] L. Yang, J. Cao, S. Tang, D. Han, and N. Suri, "Run time application repartitioning in dynamic mobile cloud environments," IEEE Transactions on Cloud Computing, vol. 4, no. 3, pp. 336–348, 2016.

[25] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14–23, Oct 2009.

[26] D. Xu, Y. Li, X. Chen, J. Li, P. Hui, S. Chen, and J. Crowcroft, "A survey of opportunistic offloading," IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2198–2236, 2018.

[27] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," Mobile Networks and Applications, vol. 18, no. 1, pp. 129–140, 2013.

[28] V. Hassija, V. Saxena, and V. Chamola, "A mobile data offloading framework based on a combination of blockchain and virtual voting," Software: Practice and Experience, 2020.

[29] A. Yousafzai, I. Yaqoob, M. Imran, A. Gani, and R. M. Noor, "Process migration-based computational offloading framework for iot-supported mobile edge/cloud computing," IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4171–4182, 2019.

[30] M. Sharifi, S. Kafaie, and O. Kashefi, "A Survey and Taxonomy of Cyber Foraging of Mobile Devices," IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 1232–1243, 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6083230

[31] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," Journal of Network and Computer Applications, vol. 40, pp. 325–344, 2014.

[32] S. Raza, S. Wang, M. Ahmed, and M. R. Anwar, "A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions," Wireless Communications and Mobile Computing, vol. 2019, 2019.

[33] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1628–1656, 2017.

[34] H. Zhou, H. Wang, X. Chen, X. Li, and S. Xu, "Data offloading techniques through vehicular ad hoc networks: A survey," IEEE Access, vol. 6, pp. 65 250–65 259, 2018.

[35] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading towards edge computing," Proceedings of the IEEE, vol. 107, no. 8, pp. 1584–1607, 2019.

[36] C. Jiang, X. Cheng, H. Gao, X. Zhou, and J. Wan, "Toward computation offloading in edge computing: A survey," IEEE Access, vol. 7, pp. 131 543–131 558, 2019.

[37] I. Ahmad, R. M. Noor, I. Ali, M. Imran, and A. Vasilakos, "Characterizing the role of vehicular cloud computing in road traffic management," International Journal of Distributed Sensor Networks, vol. 13, no. 5, p. 1550147717708728, 2017.

[38] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," Journal of Systems Architecture, 2019.

[39] F. Hagenauer, C. Sommer, T. Higuchi, O. Altintas, and F. Dressler, "Vehicular micro clouds as virtual edge servers for efficient data collection," in Proceedings of the 2nd ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services. ACM, 2017, pp. 31–35.

[40] C. Wang, Y. Li, D. Jin, and S. Chen, "On the serviceability of mobile vehicular cloudlets in a large-scale urban environment," IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 10, pp. 2960–2970, 2016.

[41] L. Gu, D. Zeng, and S. Guo, "Vehicular cloud computing: A survey," in 2013 IEEE Globecom Workshops (GC Wkshps). IEEE, 2013, pp. 403–407.

[42] S. Bitam, A. Mellouk, and S. Zeadally, "Vanet-cloud: a generic cloud computing model for vehicular ad hoc networks," IEEE Wireless Communications, vol. 22, no. 1, pp. 96–102, 2015.

[43] A. Boukerche and E. Robson, "Vehicular cloud computing: Architectures, applications, and mobility," Computer networks, vol. 135, pp. 171–189, 2018.

[44] B. Ahmed, A. W. Malik, T. Hafeez, and N. Ahmed, "Services and simulation frameworks for vehicular cloud computing: A contemporary survey," EURASIP Journal on Wireless Communications and Networking, vol. 2019, no. 1, p. 4, 2019.

[45] S. Abdelhamid, R. Benkoczi, and H. S. Hassanein, "Vehicular clouds: ubiquitous computing on wheels," in Emergent Computation. Springer, 2017, pp. 435–452.

[46] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," IEEE Communications Magazine, vol. 56, no. 8, pp. 48–54, 2018.

[47] Z. Zhou, P. Liu, Z. Chang, C. Xu, and Y. Zhang, "Energy-efficient workload offloading and power control in vehicular edge computing," in 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). IEEE, 2018, pp. 191–196.

[48] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," IEEE Internet of Things Journal, 2018.

[49] G. Singh, A. Singh, M. Singh, S. Sharma, N. Kumar, and K.-K. R. Choo, "Blocked: Blockchain-based secure data processing framework in edge envisioned v2x environment," IEEE Transactions on Vehicular Technology, 2020.

[50] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art," IEEE Access, vol. 8, pp. 116 974–117 017, 2020.

[51] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, and M. Shoaib, "Bringing computation closer toward the user network: Is edge computing the solution?" IEEE Communications Magazine, vol. 55, no. 11, pp. 138–144, 2017.

[52] P. Mell, T. Grance et al., "The nist definition of cloud computing," 2011.

[53] H. Zhang, Q. Zhang, and X. Du, "Toward vehicle-assisted cloud computing for smartphones," IEEE Transactions on Vehicular Technology, vol. 64, no. 12, pp. 5610–5618, 2015.

[54] A. Aliyu, A. H. Abdullah, O. Kaiwartya, Y. Cao, M. J. Usman, S. Kumar, D. Lobiyal, and R. S. Raw, "Cloud computing in vanets: architecture, taxonomy, and challenges," IETE Technical Review, vol. 35, no. 5, pp. 523–547, 2018.

[55] E. Lee, E.-K. Lee, M. Gerla, and S. Y. Oh, "Vehicular cloud networking: architecture and design principles," IEEE Communications Magazine, vol. 52, no. 2, pp. 148–155, 2014.

[56] C. Wu, T. Yoshinaga, X. Chen, and Y. Ji, "Multi-access edge computing for connected vehicles," IEEE COMSOC MMTC Communications - Frontiers, 2019.

[57] Q. Hu, C. Wu, X. Zhao, X. Chen, Y. Ji, and T. Yoshinaga, "Vehicular multi-access edge computing with licensed sub-6 ghz, ieee 802.11 p and mmwave," IEEE Access, vol. 6, pp. 1995–2004, 2017.

[58] Y. Liu, S. Wang, J. Huang, and F. Yang, "A computation offloading algorithm based on game theory for vehicular edge networks," in 2018 IEEE International Conference on Communications (ICC). IEEE, 2018, pp. 1–6.

[59] V. Chamola, A. Sancheti, S. Chakravarty, N. Kumar, and M. Guizani, "An iot and edge computing based smart parking operations framework for v2g system," IEEE Transactions on Vehicular Technology, pp. 1–1, 2020.

[60] F. Lin, L. Yang, K. Xiong, and X. Gong, "Recent advances in cloud-aware mobile fog computing," Wireless Communications and Mobile Computing, vol. 2019, 2019.

[61] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, N. Zhao, M. K. Khan, and N. Kumar, "Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing," IEEE Vehicular Technology Magazine, vol. 12, no. 3, pp. 55–64, 2017.

[62] "Open fog architecture overview," http://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf, acessed on May, 2019.

[63] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. da Silva, C. Lee, and O. Rana, "The internet of things, fog and cloud continuum: Integration and challenges," Internet of Things, 2018.

[64] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," IEEE access, vol. 6, pp. 47 980–48 009, 2018.

[65] Z. Ning, J. Huang, and X. Wang, "Vehicular fog computing: Enabling real-time traffic management for smart cities," IEEE Wireless Communications, vol. 26, no. 1, pp. 87–93, 2019.

[66] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," IEEE Transactions on Vehicular Technology, vol. 65, no. 6, pp. 3860–3873, 2016.

[67] Y. Xiao and C. Zhu, "Vehicular fog computing: Vision and challenges," in 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2017, pp. 6–9.

[68] C. Huang, R. Lu, and K.-K. R. Choo, "Vehicular fog computing: architecture, use case, and security and forensic challenges," IEEE Communications Magazine, vol. 55, no. 11, pp. 105–111, 2017.

[69] A. H. Sodhro, J. J. Rodrigues, S. Pirbhulal, N. Zahid, A. R. L. de Macedo, and V. H. C. de Albuquerque, "Link optimization in software defined iov driven autonomous transportation system," IEEE Transactions on Intelligent Transportation Systems, 2020.

[70] A. H. Sodhro, Z. Luo, G. H. Sodhro, M. Muzamal, J. J. Rodrigues, and V. H. C. de Albuquerque, "Artificial intelligence based qos optimization for multimedia communication in iov systems," Future Generation Computer Systems, vol. 95, pp. 667–680, 2019.

[71] IEEE, "Ieee family of standards for wireless access in vehicular environments," in available from IEEE Standards, 2011.

[72] R. Uzcategui and G. Acosta-Marum, "Wave: a tutorial," Communications Magazine, IEEE, vol. 47, no. 5, pp. 126–133, 2009.

[73] Y. J. Li, "An overview of the dsrc/wave technology," in 7th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, 2010.

[74] H. Hartenstein and K. Laberteaux, "A tutorial survey on vehicular ad hoc networks," Communications Magazine, IEEE, vol. 46, no. 6, pp. 164–171, 2008.

[75] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Ave: Autonomous vehicular edge computing framework with aco-based scheduling," IEEE Transactions on Vehicular Technology, vol. 66, no. 12, pp. 10 660–10 675, 2017.

[76] A. Vladyko, A. Khakimov, A. Muthanna, A. A. Ateya, and A. Koucheryavy, "Distributed edge computing to assist ultra-low-latency vanet applications," Future Internet, vol. 11, no. 6, p. 128, 2019.

[77] J. Sun, Q. Gu, T. Zheng, P. Dong, A. Valera, and Y. Qin, "Joint optimization of computation offloading and task scheduling in vehicular edge computing networks," IEEE Access, vol. 8, pp. 10 466–10 477, 2020.

[78] A. B. de Souza, P. A. L. Rego, and J. N. de Souza, "Exploring computation offloading in vehicular clouds," in 2019 IEEE 8th International Conference on Cloud Networking (CloudNet). IEEE, 2019, pp. 1–4.

[79] D. T. Ngo, D. H. N. Nguyen, and T. Le-Ngoc, "Intercell interference coordination: Towards a greener cellular network," Handbook of Green Information and Communication Systems, pp. 147–182, 2013.

[80] A. Ashok, P. Steenkiste, and F. Bai, "Enabling vehicular applications using cloud services through adaptive computation offloading," in Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services. ACM, 2015, pp. 1–7.

[81] M. Kratsios, "Emerging technologies and their expected impact on non-federal spectrum demand," Executive Office of the President of the United States, 2019.

[82] M. U. Rehman and G. A. Safdar, LTE communications and networks: Femtocells and antenna design challenges. John Wiley & Sons, 2018.

[83] J. Sun, Q. Gu, T. Zheng, P. Dong, and Y. Qin, "Joint communication and computing resource allocation in vehicular edge computing," International Journal of Distributed Sensor Networks, vol. 15, no. 3, p. 1550147719837859, 2019.

[84] A. Ashok, P. Steenkiste, and F. Bai, "Vehicular cloud computing through dynamic computation offloading," Computer Communications, vol. 120, pp. 125–137, 2018.

[85] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5g: A tutorial overview of standards, trials, challenges, deployment, and practice," IEEE journal on selected areas in communications, vol. 35, no. 6, pp. 1201–1221, 2017.

[86] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-end simulation of 5g mmwave networks," IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2237–2263, 2018.

[87] C. R. Storck and F. Duarte-Figueiredo, "A 5g v2x ecosystem providing internet of vehicles," Sensors, vol. 19, no. 3, p. 550, 2019.

[88] A. H. Sodhro, S. Pirbhulal, G. H. Sodhro, M. Muzammal, L. Zongwei, A. Gurtov, A. R. L. de Macêdo, L. Wang, N. M. Garcia, and V. H. C. de Albuquerque, "Towards 5g-enabled self adaptive green and reliable communication in intelligent transportation system," IEEE transactions on intelligent transportation systems (Print), 2020.

[89] S. A. A. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5g for vehicular communications," IEEE Communications Magazine, vol. 56, no. 1, pp. 111–117, 2018.

[90] A. U. Rahman, A. W. Malik, V. Sati, A. Chopra, and S. D. Ravana, "Context-aware opportunistic computing in vehicle-to-vehicle networks," Vehicular Communications, vol. 24, p. 100236, 2020.

[91] X. Fan, T. Cui, C. Cao, Q. Chen, and K. S. Kwak, "Minimum-cost offloading for collaborative task execution of mec-assisted platooning," Sensors, vol. 19, no. 4, p. 847, 2019.

[92] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Mobile edge computing for the internet of vehicles: Offloading framework and job scheduling," ieee vehicular technology magazine, vol. 14, no. 1, pp. 28–36, 2018.

[93] A. B. d. Souza, P. A. L. Rego, P. H. G. Rocha, T. C. Pessoa, and J. N. d. Souza, "A task offloading scheme for wave vehicular clouds and 5g mobile edge computing," in 2020 IEEE Global Communications Conference (Globecom). IEEE, 2020, p. to appear.

[94] S. Midya, A. Roy, K. Majumder, and S. Phadikar, "Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach," Journal of Network and Computer Applications, vol. 103, pp. 58–84, 2018.

[95] D. Comer, Computer networks and internets. Pearson, 2015.

[96] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1079–1092, 2018.

[97] M. Pasha and K. U. R. Khan, "Architecture and channel aware task offloading in opportunistic vehicular edge networks," IJSSST, vol. 18, no. 4, pp. 15–1, 2015.

[98] T. Adhikary, A. K. Das, M. A. Razzaque, A. Almogren, M. Alrubaian, and M. M. Hassan, "Quality of service aware reliable task scheduling in vehicular cloud computing," Mobile Networks and Applications, vol. 21, no. 3, pp. 482–493, 2016.

[99] M. Pasha et al., "Opportunistic task offloading in vehicular networks," in 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB). IEEE, 2017, pp. 510–514.

[100] Y. Sun, X. Guo, J. Song, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Adaptive learning-based task offloading for vehicular edge computing systems,"

IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3061–3074, 2019.

[101] Q. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Distributed collaborative execution on the edges and its application to amber alerts," IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3580–3593, 2018.

[102] F. Sun, F. Hou, N. Cheng, M. Wang, H. Zhou, L. Gui, and X. Shen, "Cooperative task scheduling for computation offloading in vehicular cloud," IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 11 049–11 061, 2018.

[103] L. Aminizadeh and S. Yousefi, "Cost minimization scheduling for deadline constrained applications on vehicular cloud infrastructure," in 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE). IEEE, 2014, pp. 358–363.

[104] F. Sun, N. Cheng, S. Zhang, H. Zhou, L. Gui, and X. Shen, "Reinforcement learning based computation migration for vehicular cloud computing," in 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018, pp. 1–6.

[105] Z. Wang, Z. Zhong, D. Zhao, and M. Ni, "Vehicle-based cloudlet relaying for mobile computation offloading," IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 11 181–11 191, 2018.

[106] Z. Zhou, J. Feng, Z. Chang, and X. Shen, "Energy-efficient edge computing service provisioning for vehicular networks: A consensus admm approach," IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 5087–5099, 2019.

[107] A. Boukerche and V. Soto, "An efficient mobility-oriented retrieval protocol for computation offloading in vehicular edge multi-access network," IEEE Transactions on Intelligent Transportation Systems, 2020.

[108] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," IEEE Transactions on Vehicular Technology, vol. 68, no. 11, pp. 11 158–11 168, 2019.

[109] G. Luo, Q. Yuan, H. Zhou, N. Cheng, Z. Liu, F. Yang, and X. S. Shen, "Cooperative vehicular content distribution in edge computing assisted 5g-vanet," China Communications, vol. 15, no. 7, pp. 1–17, 2018.

[110] Y. Sun, L. Xu, Y. Tang, and W. Zhuang, "Traffic offloading for online video service in vehicular networks: A cooperative approach," IEEE Transactions on Vehicular Technology, vol. 67, no. 8, pp. 7630–7642, 2018.

[111] Z. Jiang, S. Zhou, X. Guo, and Z. Niu, "Task replication for deadline-constrained vehicular cloud computing: Optimal policy, performance analysis, and implications on road traffic," IEEE Internet of Things Journal, vol. 5, no. 1, pp. 93–107, 2017.

[112] H. Li, X. Li, and W. Wang, "Joint optimization of computation cost and delay for task offloading in vehicular fog networks," Transactions on Emerging Telecommunications Technologies, vol. 31, no. 2, p. e3818, 2020.

[113] M. LiWang, Z. Gao, S. Hosseinalipour, and H. Dai, "Multi-task offloading over vehicular clouds under graph-based representation," in ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020, pp. 1–7.

[114] C. Zhu, J. Tao, G. Pastor, Y. Xiao, Y. Ji, Q. Zhou, Y. Li, and A. Ylä-Jääski, "Folo: Latency and quality optimized task allocation in vehicular fog computing," IEEE Internet of Things Journal, 2018.

[115] L. T. Tan, R. Q. Hu, and L. Hanzo, "Twin-timescale artificial intelligence aided mobility-aware edge caching and computing in vehicular networks," IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3086–3099, 2019.

[116] H. Guo, J. Zhang, and J. Liu, "Fiwi-enhanced vehicular edge computing networks: Collaborative task offloading," IEEE Vehicular Technology Magazine, vol. 14, no. 1, pp. 45–53, 2018.

[117] Y. Cui, Y. Liang, and R. Wang, "Resource allocation algorithm with multi-platform intelligent offloading in d2d-enabled vehicular networks," IEEE Access, vol. 7, pp. 21 246–21 253, 2019.

[118] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," IEEE Transactions on Vehicular Technology, vol. 68, no. 8, pp. 7944–7956, 2019.

[119] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation resource allocation and task assignment optimization in vehicular fog computing: A contract-matching approach," IEEE Transactions on Vehicular Technology, vol. 68, no. 4, pp. 3113–3125, 2019.

[120] Z. Su, Y. Hui, and T. H. Luan, "Distributed task allocation to enable collaborative autonomous driving with network softwarization," IEEE Journal on Selected Areas in Communications, vol. 36, no. 10, pp. 2175–2189, 2018.

[121] H. Wang, X. Li, H. Ji, and H. Zhang, "Federated offloading scheme to minimize latency in mec-enabled vehicular networks," in 2018 IEEE Globecom Workshops (GC Wkshps). IEEE, 2018, pp. 1–6.

[122] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, and X. Shen, "An smdp-based resource allocation in vehicular cloud computing systems," IEEE Transactions on Industrial Electronics, vol. 62, no. 12, pp. 7920–7928, 2015.

[123] Z. Wang, Z. Zhong, and M. Ni, "Application-aware offloading policy using smdp in vehicular fog computing systems," in 2018 IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2018, pp. 1–6.

[124] J. Feng, Z. Liu, C. Wu, and Y. Ji, "Hvc: A hybrid cloud computing framework in vehicular environments," in 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud). IEEE, 2017, pp. 9–16.

[125] X.-Q. Pham, T.-D. Nguyen, V. Nguyen, and E.-N. Huh, "Joint node selection and resource allocation for task offloading in scalable vehicle-assisted multi-access edge computing," Symmetry, vol. 11, no. 1, p. 58, 2019.

[126] S. Zhou, Y. Sun, Z. Jiang, and Z. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks," IEEE Communications Magazine, vol. 57, no. 5, pp. 49–55, 2019.

[127] S. Buda, S. Guleng, C. Wu, J. Zhang, K.-L. A. Yau, and Y. Ji, "Collaborative vehicular edge computing towards greener its," IEEE Access, vol. 8, pp. 63 935–63 944, 2020.

[128] C. Chen, L. Chen, L. Liu, S. He, X. Yuan, D. Lan, and Z. Chen, "Delay-optimized v2v-based computation offloading in urban vehicular edge computing and networks," IEEE Access, vol. 8, pp. 18 863–18 873, 2020.

[129] S. Wu, W. Xia, W. Cui, Q. Chao, Z. Lan, F. Yan, and L. Shen, "An efficient offloading algorithm based on support vector machine for mobile edge computing in vehicular networks," in 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2018, pp. 1–6.

[130] Z. Zhou, H. Yu, C. Xu, Z. Chang, S. Mumtaz, and J. Rodriguez, "Begin: Big data enabled energy-efficient vehicular edge computing," IEEE Communications Magazine, vol. 56, no. 12, pp. 82–89, 2018.

[131] V. Soto, R. E. De Grande, and A. Boukerche, "Repro: time-constrained data retrieval for edge offloading in vehicular clouds," in Proceedings of the 14th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks. ACM, 2017, pp. 47–54.

[132] Z. Wang, Z. Zhong, D. Zhao, and M. Ni, "Bus-based cloudlet cooperation strategy in vehicular networks," in 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall). IEEE, 2017, pp. 1–6.

[133] Z. Ning, X. Wang, J. J. Rodrigues, and F. Xia, "Joint computation offloading, power allocation, and channel assignment for 5g-enabled traffic management systems," IEEE Transactions on Industrial Informatics, vol. 15, no. 5, pp. 3058–3067, 2019.

[134] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," IEEE Vehicular Technology Magazine, vol. 12, no. 2, pp. 36–44, 2017.

[135] M. LiWang, S. Dai, Z. Gao, Y. Tang, and H. Dai, "A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network," IEEE Internet of Things Journal, 2018.

[136] C. Zhu, G. Pastor, Y. Xiao, Y. Li, and A. Ylae-Jaeaeski, "Fog following me: Latency and quality balanced task allocation in vehicular fog computing," in 2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). IEEE, 2018, pp. 1–9.

[137] Q. Qi, J. Wang, Z. Ma, H. Sun, Y. Cao, L. Zhang, and J. Liao, "Knowledge-driven service offloading decision for vehicular edge computing: A deep reinforcement learning approach," IEEE Transactions on Vehicular Technology, vol. 68, no. 5, pp. 4192–4203, 2019.

[138] G. Naik, B. Choudhury, and J.-M. Park, "Ieee 802.11 bd & 5g nr v2x: Evolution of radio access technologies for v2x communications," IEEE Access, vol. 7, pp. 70 169–70 184, 2019.

[139] M. M. Hassan, M. R. Hassan, S. Huda, and V. H. C. de Albuquerque, "A robust deep learning enabled trust-boundary protection for adversarial industrial iot environment," IEEE Internet of Things Journal, 2020.

[140] A. H. Sodhro, S. Pirbhulal, Z. Luo, and V. H. C. de Albuquerque, "Towards an optimal resource management for iot based green and sustainable smart cities," Journal of Cleaner Production, vol. 220, pp. 1167–1179, 2019.

[141] L. Oliveira, J. J. Rodrigues, S. A. Kozlov, R. A. Rabêlo, and V. H. C. d. Albuquerque, "Mac layer protocols for internet of things: A survey," Future Internet, vol. 11, no. 1, p. 16, 2019.

[142] X. Li, Y. Dang, M. Aazam, X. Peng, T. Chen, and C. Chen, "Energy-efficient computation offloading in vehicular edge cloud computing," IEEE Access, vol. 8, pp. 37 632–37 644, 2020.

[143] A. Mehra, M. Mandal, P. Narang, and V. Chamola, "Reviewnet: A fast and resource optimized network for enabling safe autonomous driving in hazy weather conditions," IEEE Transactions on Intelligent Transportation Systems, pp. 1–11, 2020.

[144] D. Ye, M. Wu, J. Kang, and R. Yu, "Optimized workload allocation in vehicular edge computing: A sequential game approach," in International Conference on Communicatins and Networking in China. Springer, 2017, pp. 542–551.

[145] M. Tahmasebi and M. R. Khayyambashi, "An efficient model for vehicular cloud computing with prioritizing computing resources," Peer-to-Peer Networking and Applications, pp. 1–10, 2018.

[146] Y. He, N. Zhao, and H. Yin, "Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach," IEEE Transactions on Vehicular Technology, vol. 67, no. 1, pp. 44–55, 2017.

[147] Y. Wang, P. Lang, D. Tian, J. Zhou, X. Duan, Y. Cao, and D. Zhao, "A game-based computation offloading method in vehicular multi-access edge computing networks," IEEE Internet of Things Journal, 2020.

[148] S. Misra and S. Bera, "Soft-van: Mobility-aware task offloading in software-defined vehicular network," IEEE Transactions on Vehicular Technology, vol. 69, no. 2, pp. 2071–2078, 2019.

[149] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," IEEE Transactions on Intelligent Transportation Systems, 2020.

[150] J. Zhou, D. Tian, Y. Wang, Z. Sheng, X. Duan, and V. C. Leung, "Reliability-oriented optimization of computation offloading for cooperative vehicle-infrastructure systems," IEEE Signal Processing Letters, vol. 26, no. 1, pp. 104–108, 2018.

[151] D. Ye, M. Wu, S. Tang, and R. Yu, "Scalable fog computing with service offloading in bus networks," in 2016 IEEE 3rd International Conference on Cyber Security and Cloud Computing (CSCloud). IEEE, 2016, pp. 247–251.

[152] C.-C. Lin, D.-J. Deng, and C.-C. Yao, "Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units," IEEE Internet of Things Journal, vol. 5, no. 5, pp. 3692–3700, 2017.

[153] M. Li, F. R. Yu, P. Si, H. Yao, and Y. Zhang, "Software-defined vehicular networks with caching and computing for delay-tolerant data traffic," in 2018 IEEE International Conference on Communications (ICC). IEEE, 2018, pp. 1–6.

[154] Q. Wu, H. Ge, H. Liu, Q. Fan, Z. Li, and Z. Wang, "A task offloading scheme in vehicular fog and cloud computing system," IEEE Access, vol. 8, pp. 1173–1184, 2019.

[155] X. Zhang, J. Zhang, Z. Liu, Q. Cui, X. Tao, and S. Wang, "Mdp-based task offloading for vehicular edge computing under certain and uncertain transition probabilities," IEEE Transactions on Vehicular Technology, vol. 69, no. 3, pp. 3296–3309, 2020.

[156] T. Roughgarden, "Algorithmic game theory," Communications of the ACM, vol. 53, no. 7, pp. 78–86, 2010.

[157] M. Liwang, J. Wang, Z. Gao, X. Du, and M. Guizani, "Game theory based opportunistic computation offloading in cloud-enabled iov," IEEE Access, vol. 7, pp. 32 551–32 561, 2019.

[158] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in 2017 IEEE International Conference on Communications (ICC). IEEE, 2017, pp. 1–6.

[159] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). IEEE, 2016, pp. 288–294.

[160] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," IEEE Transactions on Vehicular Technology, vol. 68, no. 9, pp. 8322–8335, 2019.

[161] P. Liu, J. Li, and Z. Sun, "Matching-based task offloading for vehicular edge computing," IEEE Access, vol. 7, pp. 27 628–27 640, 2019.

[162] B. Gu and Z. Zhou, "Task offloading in vehicular mobile edge computing: A matching-theoretic framework," IEEE Vehicular Technology Magazine, vol. 14, no. 3, pp. 100–106, 2019.

[163] J. Nocedal and S. Wright, Numerical optimization. Springer Science & Business Media, 2006.

[164] J. Wang, D. Feng, S. Zhang, J. Tang, and T. Q. Quek, "Computation offloading for mobile edge computing enabled vehicular networks," IEEE Access, vol. 7, pp. 62 624–62 632, 2019.

[165] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint offloading and resource allocation in vehicular edge computing and networks," in 2018 IEEE Global Communications Conference (GLOBECOM). IEEE, 2018, pp. 1–7.

[166] C. Yang, Y. Liu, X. Chen, W. Zhong, and S. Xie, "Efficient mobility-aware task offloading for vehicular edge computing networks," IEEE Access, vol. 7, pp. 26 652–26 664, 2019.

[167] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," IEEE Internet of Things Journal, 2018.

[168] S. Li, S. Lin, L. Cai, W. Li, and G. Zhu, "Joint resource allocation and computation offloading with time-varying fading channel in vehicular edge computing," IEEE Transactions on Vehicular Technology, vol. 69, no. 3, pp. 3384–3398, 2020.

[169] M. Nabi, R. Benkoczi, S. Abdelhamid, and H. S. Hassanein, "Resource assignment in vehicular clouds," in 2017 IEEE International Conference on Communications (ICC). Ieee, 2017, pp. 1–6.

[170] E.-G. Talbi, Metaheuristics: from design to implementation. John Wiley & Sons, 2009, vol. 74.

[171] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," ACM computing surveys (CSUR), vol. 35, no. 3, pp. 268–308, 2003.

[172] T. M. Mitchell et al., "Machine learning. 1997," Burr Ridge, IL: McGraw Hill, vol. 45, no. 37, pp. 870–877, 1997.

[173] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," IEEE Transactions on Vehicular Technology, vol. 67, no. 11, pp. 10 190–10 203, 2018.

[174] Z. Ning, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, B. Sadoun, and R. Y. Kwok, "Joint computing and caching in 5g-envisioned internet of vehicles: A deep reinforcement learning-based traffic control system," IEEE Transactions on Intelligent Transportation Systems, 2020.

[175] T. D. Nadeau and K. Gray, SDN: Software Defined Networks: an authoritative review of network programmability technologies. " O'Reilly Media, Inc.", 2013.

[176] K. Benzekki, A. El Fergougui, and A. Elbelrhiti Elalaoui, "Software-defined networking (sdn): a survey," Security and communication networks, vol. 9, no. 18, pp. 5803–5833, 2016.

[177] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, "Clustering objectives in wireless sensor networks: A survey and research direction analysis," Computer Networks, p. 107376, 2020.

[178] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," IEEE Transactions on Vehicular Technology, vol. 69, no. 2, pp. 2092–2104, 2019.

[179] Z. Ning, X. Wang, and J. Huang, "Mobile edge computing-enabled 5g vehicular networks: Toward the integration of communication and computing," ieee vehicular technology magazine, vol. 14, no. 1, pp. 54–61, 2018.

[180] A. Ashok, P. Steenkiste, and F. Bai, "Adaptive cloud offloading for vehicular applications," in 2016 IEEE Vehicular Networking Conference (VNC). IEEE, 2016, pp. 1–8.

[181] R. Beuran, Introduction to network emulation. CRC Press, 2012.

[182] J. Tang, S. Liu, B. Yu, and W. Shi, "Pi-edge: A low-power edge computing system for real-time autonomous driving services," arXiv preprint arXiv:1901.04978, 2018.

[183] A. Varga and R. Hornig, "AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT," in Proceedings of the First International ICST Conference on Simulation Tools and Techniques for Communications Networks and Systems. ICST, 2008. [Online]. Available: https://doi.org/10.4108%2Ficst.simutools2008.3027

[184] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in Modeling and tools for network simulation. Springer, 2010, pp. 15–34.

[185] D.-N. Le, A. K. Pandey, S. Tadepalli, P. S. Rathore, and J. M. Chatterjee, Network Modeling, Simulation and Analysis in Matlab: Theory and Practices. John Wiley & Sons, 2019.

[186] Y. Sun, X. Guo, S. Zhou, Z. Jiang, X. Liu, and Z. Niu, "Learning-based task offloading for vehicular cloud computing systems," in 2018 IEEE International Conference on Communications (ICC). IEEE, 2018, pp. 1–7.

[187] S. Yousefi, M. Mousavi, and M. Fathy, "Vehicular ad hoc networks (VANETs): challenges and perspectives," in ITS Telecommunications Proceedings, 2006 6th International Conference on, 2006, pp. 761–766.

[188] A. Hanggoro and R. F. Sari, "Performance evaluation of the manhattan mobility model in vehicular ad-hoc networks for high mobility vehicle," in 2013 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT), 2013, pp. 31–36.

[189] V. Hassija, V. Gupta, S. Garg, and V. Chamola, "Traffic jam probability estimation based on blockchain and deep neural networks," IEEE Transactions on Intelligent Transportation Systems, pp. 1–10, 2020.

[190] L. C. Gimenez, M. C. Cascino, M. Stefan, K. I. Pedersen, and A. F. Cattoni, "Mobility performance in slow- and high-speed LTE real scenarios," in 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring). IEEE, may 2016. [Online]. Available: https://doi.org/10.1109%2Fvtcspring.2016.7504347

[191] T. Kopacz and D. Heberling, "Introduction of a new vehicular test environment for validation of communication based systems," in 2016 10th European Conference on Antennas and Propagation (EuCAP). IEEE, 2016, pp. 1–5.

[192] S. Liu, L. Li, J. Tang, S. Wu, and J.-L. Gaudiot, "Creating autonomous vehicle systems," Synthesis Lectures on Computer Science, vol. 6, no. 1, pp. i–186, oct 2017. [Online]. Available: https://doi.org/10.2200%2Fs00787ed1v01y201707csl009

[193] M. Fiore, J. Harri, F. Filali, and C. Bonnet, "Vehicular mobility simulation for vanets," in 40th Annual Simulation Symposium (ANSS'07). IEEE, 2007, pp. 301–309.

[194] Z. Ting, H. Jianjun, S. Li, L. Jianfeng, and M. Yan, "Study on mobility models in vehicular communication system," in 2009 2nd IEEE International Conference on Broadband Network Multimedia Technology, 2009, pp. 57–61.

[195] S. A. B. Mussa, M. Manaf, K. Z. Ghafoor, and Z. Doukha, "Simulation tools for vehicular ad hoc networks: A comparison study and future perspectives," in 2015 International Conference on Wireless Networks and Mobile Communications (WINCOM). IEEE, oct 2015. [Online]. Available: https://doi.org/10.1109%2Fwincom.2015.7381319

[196] "Simulation of urban mobility (SUMO)"," http://sumo.sourceforge.net. Acessed on June, 2020.

[197] "PVT Vissim," https://www.ptvgroup.com/en/solutions/products/ptv-vissim/. Acessed on June, 2020.

[198] L. Zhu, C. Li, B. Li, X. Wang, and G. Mao, "Geographic routing in multilevel scenarios of vehicular ad hoc networks," IEEE Transactions on Vehicular Technology, vol. 65, no. 9, pp. 7740–7753, sep 2016. [Online]. Available: https://doi.org/10.1109%2Ftvt.2015.2499205

[199] W. Viriyasitavat, O. K. Tonguz, and F. Bai, "Uv-cast: an urban vehicular broadcast protocol," IEEE Communications Magazine, vol. 49, no. 11, pp. 116–124, 2011.

[200] J. Wu, M. Fang, H. Li, and X. Li, "Rsu-assisted traffic-aware routing based on reinforcement learning for urban vanets," IEEE Access, vol. 8, pp. 5733–5748, 2020.

[201] C. Sommer and F. Dressler, Vehicular networking. Cambridge University Press, 2014.

[202] B. Aslam and C. C. Zou, "Optimal roadside units placement along highways," in 2011 IEEE Consumer Communications and Networking Conference (CCNC). IEEE, jan 2011. [Online]. Available: https://doi.org/10.1109%2Fccnc.2011.5766608

[203] C. Li, Y. Zhang, T. H. Luan, and Y. Fu, "Building transmission backbone for highway vehicular networks: Framework and analysis," IEEE Transactions on Vehicular Technology, vol. 67, no. 9, pp. 8709–8722, sep 2018. [Online]. Available: https://doi.org/10.1109%2Ftvt.2018.2844471

[204] M. Alam, M. Sher, and S. A. Husain, "Integrated mobility model (imm) for vanets simulation and its impact," in 2009 International Conference on Emerging Technologies. IEEE, 2009, pp. 452–456.

[205] D. Eckhoff, C. Sommer, R. German, and F. Dressler, "Cooperative awareness at low vehicle densities: How parked cars can help see through buildings," in 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011. IEEE, dec 2011. [Online]. Available: https://doi.org/10.1109%2Fglocom.2011.6134402

[206] J. A. Sanguesa, F. Naranjo, V. Torres-Sanz, M. Fogue, P. Garrido, and F. J. Martinez, "On the study of vehicle density in intelligent transportation systems," Mobile Information Systems, vol. 2016, 2016.

[207] N. Akhtar, S. C. Ergen, and O. Ozkasap, "Vehicle mobility and communication channel models for realistic and efficient highway vanet simulation," IEEE Transactions on Vehicular Technology, vol. 64, no. 1, pp. 248–262, 2014.

[208] A. M. Vegni, M. Biagi, R. Cusani et al., "Smart vehicles, technologies and main applications in vehicular ad hoc networks," Vehicular technologies-deployment and applications, pp. 3–20, 2013.

[209] W. Stallings, Business data communications. Pearson Prentice Hall, 2009.

[210] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can wifi deliver?" IEEE/ACM Transactions on networking, vol. 21, no. 2, pp. 536–550, 2012.

[211] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An edge-computing based architecture for mobile augmented reality," IEEE Network, vol. 33, no. 4, pp. 162–169, 2019.

[212] E. B. Smida, S. G. Fantar, and H. Youssef, "Video streaming forwarding in a smart city's vanet," in 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA). IEEE, 2018, pp. 1–8.

[213] A. Vinel, E. Belyaev, O. Lamotte, M. Gabbouj, Y. Koucheryavy, and K. Egiazarian, "Video transmission over ieee 802.11 p: Real-world measurements," in 2013 IEEE International Conference on Communications Workshops (ICC). IEEE, 2013, pp. 505–509.

[214] S. Badura and A. Lieskovsky, "Intelligent traffic system: Cooperation of manet and image processing," in 2010 First International Conference on Integrated Intelligent Computing. IEEE, 2010, pp. 119–123.

[215] V. John, K. Yoneda, B. Qi, Z. Liu, and S. Mita, "Traffic light recognition in varying illumination using deep learning and saliency map," in 17th International IEEE Conference on Intelligent Transportation Systems (ITSC). IEEE, 2014, pp. 2286–2291.

[216] Q. Ye, L. Yang, and G. Xue, "Hand-free gesture recognition for vehicle infotainment system control," in 2018 IEEE Vehicular Networking Conference (VNC). IEEE, 2018, pp. 1–2.

[217] A. A. Elngar and M. Kayed, "Vehicle security systems using face recognition based on internet of things," Open Computer Science, vol. 10, no. 1, pp. 17–29, 2020.

[218] J. Kim, D. S. Han, and B. Senouci, "Radar and vision sensor fusion for object detection in autonomous vehicle surroundings," in 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, 2018, pp. 76–78.

[219] Š. Toth, J. Janech, and E. Kršák, "Image recognition system for the vanet," in 2013 Federated Conference on Computer Science and Information Systems. IEEE, 2013, pp. 675–678.

[220] P. Bucciol, F. Ridolfo, and J. C. De Martin, "Multicast voice transmission over vehicular ad hoc networks: issues and challenges," in Seventh International Conference on Networking (icn 2008). IEEE, 2008, pp. 746–751.

[221] L. Li, H. Zhou, S. X. Xiong, J. Yang, and Y. Mao, "Compound model of task arrivals and load-aware offloading for vehicular mobile edge computing networks," IEEE Access, vol. 7, pp. 26 631–26 640, 2019.

[222] A. Jindal, G. S. Aujla, and N. Kumar, "Survivor: A blockchain based edge-as-a-service framework for secure energy trading in sdn-enabled vehicle-to-grid environment," Computer Networks, vol. 153, pp. 36–48, 2019.

[223] V. Hassija, V. Chamola, G. Han, J. J. Rodrigues, and M. Guizani, "Dagiov: A framework for vehicle to vehicle communication using directed acyclic graph and game theory," IEEE Transactions on Vehicular Technology, vol. 69, no. 4, pp. 4182–4191, 2020.

[224] V. Hassija, V. Chamola, V. Gupta, and G. S. S. Chalapathi, "A framework for secure vehicular network using advanced blockchain," in 2020 International Wireless Communications and Mobile Computing (IWCMC), 2020, pp. 1260–1265.

[225] V. Hassija, V. Chamola, S. Garg, D. N. G. Krishna, G. Kaddoum, and D. N. K. Jayakody, "A blockchain-based framework for lightweight data sharing and energy trading in v2g network," IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 5799–5812, 2020.

[226] V. Hassija, V. Saxena, V. Chamola, and F. Richard Yu, "A parking slot allocation framework based on virtual voting and adaptive pricing algorithm," IEEE Transactions on Vehicular Technology, vol. 69, no. 6, pp. 5945–5957, 2020.

[227] M. Noussaiba and R. Rahal, "State of the art: Vanets applications and their rfid-based systems," in 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2017, pp. 0516–0520.

[228] T. Alladi, V. Chamola, B. Sikdar, and K. R. Choo, "Consumer IoT: Security vulnerability case studies and solutions," IEEE Consumer Electronics Magazine, vol. 9, no. 2, pp. 17–25, 2020.

[229] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: Application areas, security threats, and solution architectures," IEEE Access, vol. 7, pp. 82 721–82 743, July 2019.

[230] G. K. Verma, B. Singh, N. Kumar, and V. Chamola, "CB-CAS: Certificate-based efficient signature scheme with compact aggregation for industrial internet of things environment," IEEE Internet of Things Journal, 2019.

[231] S. Garg, A. Singh, K. Kaur, G. S. Aujla, S. Batra, N. Kumar, and M. S. Obaidat, "Edge computing-based security framework for big data analytics in vanets," IEEE Network, vol. 33, no. 2, pp. 72–81, 2019.

[232] G. Bansal, N. Naren, V. Chamola, B. Sikdar, N. Kumar, and M. Guizani, "Lightweight mutual authentication protocol for v2g using physical unclonable function," IEEE Transactions on Vehicular Technology, 2020.

[233] G. Bansal, N. Naren, and V. Chamola, "Rama: Real-time automobile mutual authentication protocol using puf," in 2020 International Conference on Information Networking (ICOIN), 2020, pp. 265–270.

[234] S. Pirbhulal, W. Wu, K. Muhammad, I. Mehmood, G. Li, and V. H. C. de Albuquerque, "Mobility enabled security for optimizing iot based intelligent applications," IEEE Network, vol. 34, no. 2, pp. 72–77, 2020.

[235] K. Z. Ghafoor, L. Kong, S. Zeadally, A. S. Sadiq, G. Epiphaniou, M. Hammoudeh, A. K. Bashir, and S. Mumtaz, "Millimeter-wave communication for internet of vehicles: Status, challenges and perspectives," IEEE Internet of Things Journal, 2020.

[236] J. Liu, E. Ahmed, M. Shiraz, A. Gani, R. Buyya, and A. Qureshi, "Application partitioning algorithms in mobile cloud computing: Taxonomy, review and future directions," Journal of Network and Computer Applications, vol. 48, pp. 99 – 117, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804514002161

[237] M. Pasha and K. U. R. Khan, "Mobile edge computing by vehicular edge networks for efficient and scalable task offloading," International Journal of Pure and Applied Mathematics, vol. 118, no. 24, 2018.

[238] J. Zhou, D. Tian, Y. Wang, Z. Sheng, X. Duan, and V. C. Leung, "Reliability-optimal cooperative communication and computing in connected vehicle systems," IEEE Transactions on Mobile Computing, 2019.

[239] M. Pasha and K. U. Rahman Khan, "Scalable and energy efficient task offloading schemes for vehicular cloud computing," International Journal of Computer Networks & Communications (IJCNC) Vol, vol. 10, 2018.

[240] S. Raza, W. Liu, M. Ahmed, M. R. Anwar, M. A. Mirza, Q. Sun, and S. Wang, "An efficient task offloading scheme in vehicular edge computing," Journal of Cloud Computing, vol. 9, pp. 1–14, 2020.

[241] Q. Liu, Z. Su, and Y. Hui, "Computation offloading scheme to improve qoe in vehicular networks with mobile edge computing," in 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP). IEEE, 2018, pp. 1–5.

[242] V. H. Hoang, T. M. Ho, and L. B. Le, "Mobility-aware computation offloading in mec-based vehicular wireless networks," IEEE Communications Letters, vol. 24, no. 2, pp. 466–469, 2019.

[243] S. Batewela, C.-F. Liu, M. Bennis, H. A. Suraweera, and C. S. Hong, "Risk-sensitive task fetching and offloading for vehicular edge computing," IEEE Communications Letters, vol. 24, no. 3, pp. 617–621, 2019.

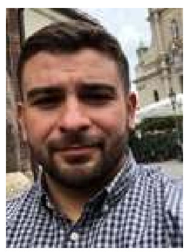
[244] T. Cui, Y. Hu, B. Shen, and Q. Chen, "Task offloading based on lyapunov optimization for mec-assisted vehicular platooning networks," Sensors, vol. 19, no. 22, p. 4974, 2019.

ALISSON B. DE SOUZA is Assistant Professor at Federal University of Ceará, Quixadá, Brazil and is currently pursuing the Ph.D. degree in Computer Science at Federal University of Ceará, Fortaleza, Brazil. He received the B.S. and M.S. degree in Computer Science from State University of Ceará, Fortaleza, Brazil, in 2009 and 2012, respectively. His main research interests are in Vehicular and Flying Ad Hoc Networks, Wireless Sensor Networks, Cloud, Edge and Fog Computing, Internet of Things, Computer Networks, Mobile and Wireless Networks, and Distributed Systems.

PAULO A. L. REGO [M'15] holds a PhD in Computer Science from the Federal University of Ceará (UFC) since 2016. He is currently an adjunct professor at Computing Department at UFC. He is an IEEE Member and works in the area of Computer Science, with emphasis on Computer Networks and Distributed Systems. His main research interests include Mobile and Cloud Computing, Vehicular Ad Hoc Networks, Edge and Fog Computing.

TIAGO CARNEIRO received the Bachelor's (2009) and Master's degree (2012) in Computer Science from the State University of Ceará, and the D.Sc degree in Computer Science from the Federal University of Ceará, Brazil. He is currently a Postdoctoral researcher at INRIA Lille - Nord Europe, France. His research interests include the use of heterogeneous architectures for solving combinatorial optimization problems.

JARDEL DAS C. RODRIGUES received his B.Sc. in Computing Science from the Federal Institute of Education, Science and Technology of Ceara (2014) where he has experienced studies in the computer architecture area. Also, he obtained an academic achievement award (first place) in 2012. In 2017, received his M.Sc in Computing Science from the Federal Institute of Education, Science and Technology of Ceara. Some topics of his research in this period were mathematical morphology and biomedical image segmentation, more precisely retinal images. Actually, is professor at Federal Institute of Education, Science and Technology of Ceara and Ph.D. student of the Post-Graduation Program in Teleinformatics Engineering from the Federal University of Ceara and his research interests include Image and Signal Processing, Pattern Recognition, Pose Estimation, and Biometric Applications.



PEDRO P. REBOUÇAS FILHO is graduated in Mechatronics Engineering from the Federal Institute of Ceará, Brazil (2008). In 2010, he obtained an M.Sc. in Teleinformatics Engineering, in the field of Biomedical Engineering, also at the Federal University of Ceará. In 2013, he earned a Ph.D. degree in Teleinformatics Engineering from the same University. He has been an Assistant Professor at the Federal Institute of Ceará since 2008. From 2015 to 2016, he was a postdoctoral researcher at the University of Porto, Portugal. He is co-author of more than 100 articles in national and international journals and conferences. He has also been involved in several research projects in the field of Computer Vision, Medical Image, and embedded systems.

JOSÉ N. DE SOUZA [SM'13] holds a PhD degree at Pierre and Marie Curie University (PARIS VI/MASI Laboratory), France, since 1994. He is currently working as a researcher full professor at the Federal University of Ceará in the Computer Science Department, and is IEEE Senior member. Since 1999, he has been the Brazilian representative at the IFIP TC6. His main research interests are Cloud Computing and Network Management.

VINAY CHAMOLA received the B.E. degree in electrical and electronics engineering and master's degree in communication engineering from the Birla Institute of Technology and Science, Pilani, India, in 2010 and 2013, respectively. He received his Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2016. In 2015, he was a Visiting Researcher with the Autonomous Networks Research Group (ANRG), University of Southern California, Los Angeles, CA, USA. He also worked as a post-doctoral research fellow at the National University of Singapore, Singapore where he worked in the area of Internet of Things. He is currently Assistant Professor with the Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus where he heads the Internet of Things Research Group / Lab. He has over 50 publications in high ranked SCI Journals including more than 33 *IEEE Transaction* and Journal articles. His works have been published in Journals like *IEEE Transactions on Communications*, *IEEE Transactions on Vehicular Technology*, *IEEE Journal on Selected Areas in Communications*, *IEEE Communications Magazine* etc. Furthermore, his works have been accepted and presented in reputed conferences like *IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC, IEEE PerCom* to name a few. His research interests include IoT Security, Blockchain, 5G network management and addressing research issues in VANETs and UAV networks. He has served as a reviewer for several IEEE/Elsevier Journals. He is a Guest Editor in Computer Communication, Elsevier. He serves as an Associate Editor for the Ad Hoc Networks, Elsevier; IET Quantum Communications and also Frontiers in Communications and Networks journal. He is a senior member of the IEEE.

VICTOR HUGO C. DE ALBUQUERQUE [M'17, SM'19] is a full professor and senior researcher at the University of Fortaleza, UNIFOR. He has a Ph.D in Mechanical Engineering from the Federal University of Paraíba (2010), an MSc in Teleinformatics Engineering from the Federal University of Ceará (2007) and Bachelor in Mechatronics Engineering from the Federal Center of Technological Education of Ceará (2006). He mainly researches IoT, Machine/Deep Learning, Pattern Recognition, and Robotics. He is an Associate Editor of IEEE Access.

BIPLAB SIKDAR received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology Kanpur, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His current research interests include wireless network, and security for Internet of Things and cyberphysical systems. Dr. Sikdar served as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS from 2007 to 2012. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING.

TABLE 2. (Appendix A) Communication standard classification of works about computation offloading in vehicular networks. (√) indicates that topic is covered.

Reference	Technology			Client			Server			Type					Wireless Hop	
	WAVE	Cellular	Other	Vehicle	Infrastructure	Pedestrian	VC	Edge	TC	V2V	V2I	I2V	P2V	PV2V	One	Multi
Ref. [58]				√				√			√				√	
Ref. [135]				√			√			√					√	
Ref. [180]		√		√				√	√		√				√	
Ref. [100]	√			√				√		√					√	
Ref. [145]	√			√			√		√	√	√				√	
Ref. [129]				√				√			√				√	
Ref. [122]				√		√	√		√	√	√			√	√	√
Ref. [123]				√		√	√		√	√	√			√	√	√
Ref. [97]	√			√			√			√					√	
Ref. [75]	√			√			√			√						√
Ref. [130]	√	√		√				√			√				√	
Ref. [132]						√	√			√				√		√
Ref. [46]	√	√	√	√			√	√		√	√					√
Ref. [221]	√			√				√		√	√					√
Ref. [96]			√	√				√			√				√	
Ref. [164]				√				√			√				√	
Ref. [22]	√			√			√	√		√						√
Ref. [119]		√				√	√	√					√		√	
Ref. [90]		√		√			√			√						√
Ref. [102]		√			√		√					√			√	
Ref. [103]	√	√			√		√					√			√	
Ref. [159]				√				√			√				√	
Ref. [76]	√			√				√			√				√	
Ref. [120]				√			√	√		√	√				√	
Ref. [166]	√			√				√			√				√	
Ref. [80]		√		√					√		√				√	
Ref. [106]	√					√		√			√			√		√
Ref. [47]	√					√		√			√			√		√
Ref. [126]	√	√		√		√	√			√	√	√				√
Ref. [121]	√	√		√			√	√		√	√					√
Ref. [116]	√		√	√				√	√	√	√				√	
Ref. [136]	√	√		√			√			√					√	
Ref. [114]	√	√		√			√			√					√	
Ref. [157]		√		√			√			√					√	
Ref. [124]	√	√		√				√	√		√				√	
Ref. [146]	√	√		√				√			√				√	
Ref. [83]		√		√			√			√						√
Ref. [133]	√	√		√				√		√	√				√	
Ref. [167]	√			√				√			√				√	
Ref. [125]		√				√	√	√	√			√			√	
Ref. [186]	√					√	√			√				√		√
Ref. [161]		√		√				√	√		√				√	
Ref. [91]	√	√		√			√	√		√	√					√
Ref. [237]	√			√			√			√					√	
Ref. [92]	√	√		√			√	√	√	√	√					√
Ref. [179]	√	√		√				√			√					√
Ref. [134]	√			√				√		√	√					√
Ref. [173]	√	√		√			√	√		√	√				√	
Ref. [94]			√	√			√	√	√		√	√			√	
Ref. [40]	√			√			√			√						√
Ref. [99]	√			√			√			√					√	
Ref. [158]				√				√			√				√	
Ref. [144]				√			√	√			√	√			√	
Ref. [182]				√				√	√		√				√	
Ref. [98]	√			√			√			√						√
Ref. [104]		√			√		√					√			√	
Ref. [238]	√	√		√			√		√	√	√					√
Ref. [150]				√				√			√				√	
Ref. [131]	√					√	√					√				√
Ref. [152]	√			√			√	√		√	√				√	
Ref. [169]				√			√			√					√	
Ref. [239]	√			√			√			√					√	
Ref. [151]					√		√					√			√	
Ref. [153]	√	√		√				√	√		√				√	
Ref. [111]	√			√		√	√				√	√			√	
Ref. [14]	√			√			√			√					√	
Ref. [115]	√	√		√				√			√				√	
Ref. [105]						√	√			√				√		√
Ref. [84]		√		√				√	√		√				√	
Ref. [66]	√			√			√			√						√
Ref. [147]				√				√			√				√	
Ref. [154]				√			√		√	√	√				√	
Ref. [107]	√				√	√	√					√	√			√
Ref. [240]	√	√		√			√	√		√	√				√	
Ref. [127]	√			√			√			√						√
Ref. [118]	√			√				√	√		√				√	
Ref. [241]	√	√		√				√			√				√	
Ref. [108]	√	√				√	√	√					√		√	
Ref. [128]	√			√			√			√					√	
Ref. [13]	√			√				√			√				√	
Ref. [142]		√		√				√			√				√	
Ref. [174]		√		√				√			√				√	
Ref. [165]	√			√				√			√				√	
Ref. [112]	√				√		√			√		√				√
Ref. [77]	√			√				√			√				√	
Ref. [168]	√			√				√			√				√	
Ref. [137]		√	√	√			√	√		√	√				√	
Ref. [155]		√		√				√			√				√	
Ref. [242]				√				√			√				√	
Ref. [113]				√			√			√					√	
Ref. [160]	√			√			√	√		√	√				√	
Ref. [117]				√				√	√		√				√	
Ref. [243]				√				√			√				√	
Ref. [148]	√			√				√			√				√	
Ref. [244]		√		√			√	√		√	√				√	
Ref. [178]	√		√	√				√	√	√	√				√	√
Ref. [162]				√				√			√				√	
Ref. [149]	√			√			√	√		√	√				√	
Ref. [78]	√			√			√			√					√	
Ref. [93]	√	√		√			√	√		√	√				√	

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3033828, IEEE Access

TABLE 3. (Appendix B) Problem classification of works about computation offloading in vehicular networks. (√) indicates that topic is covered.

Reference	Objective							Strategy						
	- Response Time	- Energy Consumption	- Financial Cost	- Overload	+ System Utility	+ Incentive	Other	Stochastic	Game Theory	Mathematical Programming	Metaheuristic	Machine Learning	Network	Other
Ref. [58]	√								√					
Ref. [135]	√		√		√	√				√				
Ref. [180]	√													√
Ref. [100]	√											√		
Ref. [145]	√		√											√
Ref. [129]	√						√					√		
Ref. [122]						√		√						
Ref. [123]	√					√		√						
Ref. [97]	√													√
Ref. [75]	√										√			
Ref. [130]		√					√		√				√	
Ref. [132]	√	√											√	
Ref. [46]	√												√	
Ref. [221]	√		√											
Ref. [96]	√		√							√				
Ref. [164]	√	√								√				
Ref. [22]	√													√
Ref. [119]	√				√	√			√					
Ref. [90]	√													√
Ref. [102]	√										√			
Ref. [103]	√									√				
Ref. [159]	√		√		√				√					
Ref. [76]	√			√									√	
Ref. [120]	√		√			√		√		√	√		√	
Ref. [166]	√		√							√				
Ref. [80]	√													√
Ref. [106]		√								√				
Ref. [47]	√	√								√				
Ref. [126]	√											√		
Ref. [121]	√													√
Ref. [116]	√													√
Ref. [136]	√						√			√				
Ref. [114]	√						√			√	√			
Ref. [157]	√		√						√					
Ref. [124]	√									√				
Ref. [146]	√			√		√						√		
Ref. [83]	√										√		√	
Ref. [133]	√			√									√	
Ref. [167]	√			√	√					√				
Ref. [125]	√		√			√				√				
Ref. [186]	√											√		
Ref. [161]	√								√					
Ref. [91]	√		√							√				
Ref. [237]	√			√										√
Ref. [92]	√										√			
Ref. [179]	√												√	
Ref. [134]	√		√						√					
Ref. [173]	√		√	√		√						√		
Ref. [94]	√	√									√			
Ref. [40]							√							√
Ref. [99]	√												√	
Ref. [158]	√		√			√			√					
Ref. [144]	√	√	√			√			√					
Ref. [182]	√	√												√
Ref. [98]	√						√			√				√
Ref. [104]	√											√		
Ref. [238]	√	√								√				
Ref. [150]							√			√				
Ref. [131]	√			√										√
Ref. [152]	√	√						√						
Ref. [169]	√									√				
Ref. [239]	√	√									√			
Ref. [151]				√	√	√	√				√			
Ref. [153]	√			√				√					√	
Ref. [111]	√						√	√						
Ref. [14]	√						√					√		
Ref. [115]	√		√	√		√					√			
Ref. [105]	√	√												√
Ref. [84]	√													√
Ref. [66]	√													√
Ref. [147]	√		√		√				√					
Ref. [154]	√	√	√			√		√			√			
Ref. [107]	√			√			√							√
Ref. [240]			√	√	√									√
Ref. [127]	√						√						√	√
Ref. [118]	√		√		√				√	√				
Ref. [241]	√	√	√		√		√							√
Ref. [108]			√		√	√		√				√		
Ref. [128]	√										√			
Ref. [13]	√									√				
Ref. [142]	√	√									√			
Ref. [174]		√	√			√						√		
Ref. [165]	√				√					√				
Ref. [112]	√			√		√				√				
Ref. [77]	√		√		√					√	√			
Ref. [168]					√					√				
Ref. [137]	√					√						√		
Ref. [155]	√							√						
Ref. [242]	√	√	√											√
Ref. [113]	√		√											√
Ref. [160]	√		√		√	√			√	√				
Ref. [117]	√											√		
Ref. [243]	√						√			√				
Ref. [148]	√	√	√	√						√			√	
Ref. [244]		√								√				
Ref. [178]	√			√					√				√	
Ref. [162]	√	√							√					
Ref. [149]		√	√	√	√	√			√		√			
Ref. [78]	√						√							√
Ref. [93]	√						√							√

TABLE 4. (Appendix C) Experiments classification of works about computation offloading in vehicular networks. (√) indicates that topic is covered.

Reference	Network		Mobility		Scenario			Vehicular Density			Application			
	Real	Simulation	Real	Simulation	Urban	Highway	Other	High	Medium	Low	Video	Image	Audio	Other
Ref. [58]		√		√						√				
Ref. [135]		√		√	√			√	√	√				
Ref. [180]	√		√			√	√			√	√	√		√
Ref. [100]		√		√		√								
Ref. [145]		√		√										
Ref. [129]		√		√		√				√				
Ref. [122]				√						√				
Ref. [123]		√		√						√				
Ref. [97]		√		√		√			√	√				
Ref. [75]		√		√	√	√			√	√				
Ref. [130]		√		√	√					√				
Ref. [132]		√		√		√				√				
Ref. [46]				√		√				√	√			
Ref. [221]		√		√		√				√			√	
Ref. [96]		√		√		√				√				
Ref. [164]		√		√		√								
Ref. [22]		√		√										
Ref. [119]		√		√		√				√				
Ref. [90]		√		√	√									
Ref. [102]		√		√		√				√	√			
Ref. [103]		√		√		√			√					
Ref. [159]				√		√		√	√					
Ref. [76]	√	√		√		√								
Ref. [120]		√		√		√				√				
Ref. [166]		√		√		√				√				
Ref. [80]	√		√			√	√			√	√	√		
Ref. [106]		√		√	√					√				
Ref. [47]		√		√		√				√				
Ref. [126]		√		√		√								
Ref. [121]		√		√		√								
Ref. [116]				√		√		√	√	√				
Ref. [136]		√	√	√	√			√			√			
Ref. [114]		√	√	√	√			√			√			
Ref. [157]		√		√		√				√				
Ref. [124]		√		√	√	√			√	√				
Ref. [146]		√		√							√			
Ref. [83]		√		√		√			√	√				
Ref. [133]		√	√	√	√					√				
Ref. [167]		√		√		√			√	√				
Ref. [125]		√	√		√					√		√		
Ref. [186]		√		√		√				√				
Ref. [161]		√		√	√	√				√				
Ref. [91]		√		√		√				√				
Ref. [237]		√		√		√				√				
Ref. [92]		√		√	√					√				
Ref. [179]			√	√	√									
Ref. [134]		√		√		√							√	
Ref. [173]		√		√					√					
Ref. [94]		√		√	√			√						
Ref. [40]			√		√					√				
Ref. [99]		√		√		√				√				
Ref. [158]		√		√		√								
Ref. [144]		√												
Ref. [182]	√	√	√	√			√					√	√	
Ref. [98]		√		√	√				√	√				√
Ref. [104]		√		√		√				√				
Ref. [238]		√		√		√		√	√	√				
Ref. [150]		√												
Ref. [131]		√		√	√					√				
Ref. [152]		√		√										
Ref. [169]		√	√	√	√				√	√				√
Ref. [239]		√		√		√				√				
Ref. [151]		√	√	√	√					√				
Ref. [153]		√		√	√			√						
Ref. [111]		√		√		√		√	√	√				
Ref. [14]		√		√		√				√				
Ref. [115]		√				√			√					
Ref. [105]		√		√		√				√				
Ref. [84]	√		√		√		√			√	√	√		√
Ref. [66]		√	√		√		√			√				
Ref. [147]				√						√				
Ref. [154]		√		√		√				√				
Ref. [107]		√		√	√	√				√				
Ref. [240]		√		√		√			√	√				
Ref. [127]		√				√								
Ref. [118]		√		√		√								
Ref. [241]		√								√				
Ref. [108]		√		√	√					√				
Ref. [128]		√		√		√				√				
Ref. [13]		√		√		√				√				
Ref. [142]		√		√						√				
Ref. [174]		√	√			√				√				
Ref. [165]		√		√		√		√			√	√	√	√
Ref. [112]		√		√		√		√						
Ref. [77]		√		√		√			√	√				
Ref. [168]		√		√		√								
Ref. [137]		√		√		√				√	√	√		√
Ref. [155]		√	√	√	√	√								
Ref. [242]		√		√		√				√				
Ref. [113]		√		√						√				
Ref. [160]		√		√	√	√				√				
Ref. [117]		√		√						√				
Ref. [243]		√			√							√		
Ref. [148]		√		√										
Ref. [244]		√		√						√				
Ref. [178]		√		√		√		√	√	√				
Ref. [162]		√				√				√				
Ref. [149]		√				√		√	√		√	√	√	√
Ref. [78]		√		√	√	√		√	√	√		√		
Ref. [93]		√		√		√		√	√	√		√		