

Article

Towards a Novel Air–Ground Intelligent Platform for Vehicular Networks: Technologies, Scenarios, and Challenges

Swapnil Sadashiv Shinde [†] and Daniele Tarchi ^{*,†}

Department of Electrical, Electronic and Information Engineering “Guglielmo Marconi”, University of Bologna, 40136 Bologna, Italy; swapnil.shinde2@unibo.it

* Correspondence: daniele.tarchi@unibo.it

† These authors contributed equally to this work.

Abstract: Modern cities require a tighter integration with Information and Communication Technologies (ICT) for bringing new services to the citizens. The Smart City is the revolutionary paradigm aiming at integrating the ICT with the citizen life; among several urban services, transports are one of the most important in modern cities, introducing several challenges to the Smart City paradigm. In order to satisfy the stringent requirements of new vehicular applications and services, Edge Computing (EC) is one of the most promising technologies when integrated into the Vehicular Networks (VNs). EC-enabled VNs can facilitate new latency-critical and data-intensive applications and services. However, ground-based EC platforms (i.e., Road Side Units—RSUs, 5G Base Stations—5G BS) can only serve a reduced number of Vehicular Users (VUs), due to short coverage ranges and resource shortage. In the recent past, several new aerial platforms with integrated EC facilities have been deployed for achieving global connectivity. Such air-based EC platforms can complement the ground-based EC facilities for creating a futuristic VN able to deploy several new applications and services. The goal of this work is to explore the possibility of creating a novel joint air-ground EC platform within a VN architecture for helping VUs with new intelligent applications and services. By exploiting most modern technologies, with particular attention towards network softwarization, vehicular edge computing, and machine learning, we propose here three possible layered air-ground EC-enabled VN scenarios. For each of the discussed scenarios, a list of the possible challenges is considered, as well possible solutions allowing to overcome all or some of the considered challenges. A proper comparison is also done, through the use of tables, where all the proposed scenarios, and the proposed solutions, are discussed.

Keywords: smart cities; vehicular networks; edge computing; machine learning; network softwarization; aerial platforms



Citation: Shinde, S.S.; Tarchi, D. Towards a Novel Air–Ground Intelligent Platform for Vehicular Networks: Technologies, Scenarios, and Challenges. *Smart Cities* **2021**, *4*, 1469–1495. <https://doi.org/10.3390/smartcities4040078>

Academic Editors: Kolomvatsos Kostas and Christos Anagnostopoulos

Received: 15 November 2021

Accepted: 7 December 2021

Published: 9 December 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

If the city is rapidly moving toward a Smart City, the traditional transportation systems are rapidly converging into an Intelligent Transportation System (ITS) mainly through the integration of innovative technologies, e.g., Internet of Things (IoT), and wireless technologies (e.g., 5G) [1]. With new connected vehicles, having different communication and computing technologies, Vehicular Networks (VNs) are radically moving toward the Internet of Vehicles (IoVs) paradigm, where new applications and services are growing. With the integration of modernized technologies such as IoT and various new communication modes, including Vehicle-to-Vehicle (V2V), Vehicle-to-Road Side Units (V2R), Vehicle-to-Infrastructure of Cellular Networks (V2I), Vehicle-to-Sensors (V2S), and Vehicle-to-Person (V2P), IoV can expand the traditional Vehicular Ad hoc Networks (VANETs) capability by adding several new services and applications. IoV is able to support various functions, including intelligent traffic management, traffic safety enhancements, dynamic information services, intelligent vehicle control for improving the overall road experience

of users [2–5]. However, vehicle onboard computation capacity is falling short when new latency-critical and data-intensive applications are considered; hence, new challenges are arising in VNs. The Cloud computing facilities are able to reduce the computation burden of new services; therefore, Vehicular Users (VUs) can transmit the portion/complete task to the cloud servers having enormous computation and communication power. In general, the cloud facilities are located far from the users, in the core network, and introduce some drawbacks, e.g., huge transmission costs, backhaul link congestions, data security threats due to long-distance communications. Such issues can be addressed by integrating the Edge Computing (EC) facilities into VNs, bringing cloud computing resources in the proximity of end-users [6,7].

In the case of VNs, EC facilities can be integrated through the deployment of several EC servers alongside the road network co-located with Roadside Units (RSUs). Such an approach is known as Vehicular Edge Computing (VEC) and has achieved lots of success by enabling new latency-critical services into VNs [8]. However, the limited capacity and coverage of EC servers/RSUs is the main bottleneck while exploiting VEC advantages. VEC-based VN can also be complemented by adding additional EC servers located on the terrestrial cellular base stations (i.e., 5G-gNB). Such ground-based multi EC platforms can compliment VUs by providing additional services and reducing the overall burden of the VEC servers.

However, in recent times, Terrestrial Networks (TNs) are becoming more and more used, with many new users requesting services with specific demands. Limited coverage into rural and remote areas, unreliable service during natural disasters like tsunamis and earthquakes, new security challenges, poor link budget with additional interferences are some of the main challenges that need to be considered while utilizing TN-based EC platforms into VNs. In addition, VUs are not the only user group requesting services from the TN-based EC platforms. With the presence of different users, the dynamically changing resources of EC servers with vehicular mobility adds additional challenges while integrating the TN-based EC services into VNs. With this limitation, integrating TN-based EC platforms into VN alone can not be a sufficient solution for the new futuristic vehicular services and applications, which will have more stringent requirements in terms of latency and computation resources.

Encouraged by the new technological developments and the additional interest shown by several tech giants (i.e., Facebook, Google, etc.), the Non-Terrestrial Networks (NTN), including space and air networks are growing these days mainly for providing global connectivity [9]. Several new platforms such as new satellite constellations, unmanned aerial vehicles (UAVs) swarms, small fueled aircraft, balloons have been deployed at different heights from the ground users to achieve the global connectivity challenge [10]. Better connectivity, scalability, reliability are some of the advantages of NTN based communication platforms. With the addition of modern communication technologies, such as multi-beam antennas, the NTN platforms can also provide EC-based services with an onboard computing server [11–13]. Such NTN-based EC platforms can complement the VNs for solving several problems, including RSUs' limited capacity and coverage. However, the higher transmission delay, introduced by space networking platforms (i.e., satellite constellations), is a major challenge while considering NTN platforms for serving latency-critical VNs. Compared with space networks, new aerial platforms such as Low Altitude and High Altitude Platforms (LAPs and HAPs) have a considerable advantage with reduced transmission distances, low deployment time and costs, and reduced communication channel losses. Therefore LAPs and HAPs can be excellent solutions for complimenting the VN for providing new innovative and intelligent services to the end-users.

Every connected vehicle in a VN is equipped with several sensors, able to generate tones of data (i.e., big data) in real-time. These data can be analyzed and exploited for improving the quality of VNs services and applications [14]. Recently machine learning (ML) techniques are used for solving challenging problems over wireless networks [15,16]. With new hardware technologies and the availability of a large amount of data through

IoT devices, ML research is grown fast. Several new ML tools and techniques have been developed and utilized for solving real-world problems on a daily basis. The use of new innovative ML-based solutions for analyzing the vehicular data for improving the vehicular environment can be beneficial [17]. However, proper infrastructures with communication and computing resources are required, for embedding the ML-based solutions into VNs, failure of which can introduce higher process costs (i.e., cost induced by ML model training, inference, etc). With limited resources and dynamic movements in VNs, it is challenging to implement ML techniques. The EC resources can be exploited for integrating the ML-based solution techniques into VNs.

Traditional ML approaches such as centralized ML require VUs to transmit their data to the centralized, more powerful servers [18]. However, such an approach can introduce higher costs in terms of communication latencies and energy requirements. With resource limitations and critical service requirements performing centralized ML over VN can be challenging. On the other hand, new learning approaches such as Federated Learning (FL), VUs are able to perform the training operations by themselves [19]. In each FL round, VUs perform the training operations to learn local ML model parameters, to be then sent towards the centralized server. After receiving all VUs parameters from its coverage area, the server is able to perform an averaging operation for creating a global model to be broadcast back to the VUs. Thus, the VUs can reduce the data communication cost by performing local training operations and also exploit the presence of other VUs, by learning experiences/data through the averaging. For these reasons, recently, FL-based solutions are preferred for VN application when solving challenging problems [19–21]. For benefiting from the collaborative learning into FL, a server platform with better coverage characteristics and channel conditions is required. NTN platforms, such as HAPs, can assist VN in implementing an efficient FL process with their better coverage characteristics, moderate transmission distances, and better channel conditions. FL and other ML techniques will be described in more detail in Section 3.

Therefore, there is clear scope for integrating NTN layers, especially air network-based EC platforms, with ground-based EC resources, into VNs. Such an approach can solve the limited capacity and coverage problem of conventional ground-based EC platforms and can provide more intelligent services and applications with stringent requirements. Therefore, we aim to provide a novel multiple EC platforms-enabled VN architecture by integrating ground and aerial network-based EC resources. Different enabling technologies, corresponding challenges, and opportunities are discussed in detail. We further analyze several vehicular scenarios and the benefits of using the proposed VN architecture for solving the challenges associated with them. In the following, we list the main contributions of this work:

- In Section 2, a review of the technological background is given by resuming the most important technical reports, projects, and papers addressing scenarios similar to the one we consider.
- In Section 3, we first describe the main enabling technologies that can be crucial for shaping the next generation VNs. In particular, we focus on ML, VEC, and network softwarization technologies, their characteristics, corresponding benefits, and challenges while used in vehicular environments.
- In Section 4, for efficient utilization of the previously introduced key technologies over the vehicular environment, we provide a joint air-ground network-based VN architecture (for better clarity, in the following the term air-ground refers to the architecture while T-NTN to the terrestrial and non-terrestrial network issues and deployment). We highlight the important characteristics of individual layers of the proposed architecture, including capacity, coverage, and applications. With multiple EC platforms, the proposed architecture can be useful for implementing various intelligent vehicular services and applications that can ease the vehicular experience.
- In Section 5, we further describe three main VN scenarios including the implementation of an efficient ML platform for VN services and applications, enabling the

computation offloading based EC services in VN, and a network function placement for the network slicing-based vehicular services. Important challenges, and the solution methodologies that can be used while considering the proposed network architecture over these important vehicular scenarios, are analyzed.

- In Section 6, we summarize the important characteristics, challenges, and proposed solutions for the scenarios discussed in the previous section.
- In Section 7, we conclude our work by highlighting the proposed solutions and possible future research directions that can be followed.

2. Scenario Background

The Smart City vision can be achieved through a proper integration of several key technologies (e.g., ML, EC). In [22], the authors have surveyed different EC technologies that can be useful for realizing the Smart City vision. Various Smart City scenarios and corresponding challenges are reported. Furthermore, in [23], the influence of big data and ML technologies in Smart City developments is presented. Developing a proper ITS is a key to creating sustainable Smart City environments. Different technologies are merging with the goal of forming sustainable, safe, and intelligent VNs for serving VUs. In [24], the authors surveyed several ML-based solutions employed in VNs communication and networking parts. Differently, the importance of a network softwerization technology in the VN and corresponding challenges is surveyed in [25]. In [26], the authors have proposed a software-defined collaborative EC platform for the vehicular scenario. They have mainly focused on ground-based EC technologies, including mobile edge computing (MEC), fog computing, cloudlet, etc. However, limited capacity and coverage issues of ground-based EC platforms limit the performance of traditional VNs.

Recently, the importance of NTN platforms in wireless communication is highlighted in different projects and research works. Several ongoing projects including EdgeSAT [27], SATis5 [28], Expanse [29] are aimed at integration (with terrestrial networks (TNs)), softwerization, and expansion of EC facilities over NTN platforms. Several study items such as IoT over NTN, new radio (NR) over NTN, satellite components in 5G architecture, and unmanned aerial systems are part of 3rd generation partnership project (3GPPs) Release 17 studies, which will be finalized at the end of the first quarter of 2022 [30]. It is also expected that 3GPP will reconsider many of these items in an upcoming release (i.e., Release 18) [31].

In [32], the authors presented the new opportunities and challenges ahead for integrating the NTN technologies into upcoming 6G networks. The work in [33] analyzes the importance and challenges of integrating NTN networks into VN. A space-air-ground integrated VN architecture for supporting different vehicular services in diverse scenarios is proposed. Furthermore, in [34], the authors have presented a space-air ground integrated VN architecture highlighting the key features of each platform layer. It is possible to enable several EC services over different NTN layers by deploying proper computing resources. In [12], authors proposed EC-enabled UAV platforms for improving computation performance and reducing the execution latency of MEC systems. Recently in [35], authors studied the energy performance of an offloading strategy designed over EC-enabled satellites and HAP networks for ground-based users. In [36], the importance of the UAV-assisted VEC system and corresponding implementation issues are highlighted.

3. Enabling Technologies and Challenges for Futuristic Vehicular Networks

A suitable VN architecture able to serve users with new innovative services and applications has to exploit together several key technologies. However, with limited EC capacities and coverage restrictions, it is challenging to integrate these high-level technologies into the vehicular environment. Here, we introduce the main technologies for designing futuristic VNs and corresponding challenges.

3.1. Network Softwarization

Network softwarization is a key trend that uses Software-Defined Networking (SDN), Network Function Virtualization (NFV), and network slicing techniques for providing additional programmability, flexibility, and modularity in different parts of the wireless communication networks [37]. Network softwarization allows a flexible deployment and control of different vehicular services once adapted into VNs. Here, we introduce the main technologies that allows to create a flexible and programmable VN.

3.1.1. Network Function Virtualization (NFV)

NFV is a key technological trend to tackle the flexibility and scalability problems associated with traditional hardware-based Network Functions (NFs). In the past, different NFs such as firewalls, Content Delivery Networks (CDNs), Network Address Translation (NAT) were installed as dedicated hardware-based appliances. With this, the implementation of new services and applications were restricted by the deployment of specific hardware-based elements. NFV decouples this network functions from proprietary hardware appliances and runs them as software instances in Virtual Machines (VMs)/containers as Virtual Network Functions (VNFs) [38]. Through NFV, standard network resources such as compute, storage, and network functions can be virtualized and kept on Commercial Off-the-Shelf (COTS) hardware like x86 servers. In addition, multiple VMs/containers, through the proper assignments of virtualized resources, can run on a single server for improving server resource utilization. With the NFV technology, different network functions can be placed in different locations of the networks elements such as data centers, EC servers, etc.

In the case of a Multiple EC (Multi-EC) platforms-enabled-VN architecture, NFV can potentially bring several benefits, including reduced network cost, less time-to-market for new services, higher resource efficiency, and better scalability. However, each VNF demands specific computation resources based upon the service types. Furthermore, since VNFs can be deployed at multiple locations, it is important to find proper function placement strategies for implementing many hybrid vehicular services over a resource-constrained VN.

3.1.2. Software-Defined Networking

With the unprecedented increase in demand for heterogeneous services in VN, there is a need for a platform that can dynamically adapt the network and service according to the demand request. Software-Defined Networking (SDN) technology can create a more flexible and programmable VN for supporting the critical requirements of the services and applications [39,40]. The SDN forms a fully programmable wireless network by logically separating the data and control plane, where all control operations are performed through the centralized controller unit. The controller can have a global view of network topology, traffic load, network states, and link failures. In the case of an EC-enabled VN, having different EC layers of heterogeneous hardware having a proper centralized controller assisted by the individual layers local controller can be useful for providing flexible VN services. In the past several studies have shown the importance of SDN-based VNs, where SDN technology is integrated into VN to manage different parts with improved performance in terms of network flexibility, throughput, and flexible deployments of new services and applications. However, various issues need to be handled carefully during the integration of SDN into VN including, possible security attacks over the data plane, the vulnerability of a centralized controller in terms of single-point failures, issues related to the heterogeneous hardware of different EC platforms, various access network technologies, etc. Over the years, researchers have provided numerous solutions for these issues [26,40–43]. As an example, in [41], a 5G software-defined vehicular network having integrated SDN technology is proposed with clear separation of data, control, and application planes. In another study [26], the authors proposed a collaborative EC-based software-defined VN with multiple EC platforms. Furthermore, in [40], the authors

have experimented with the different SDN controllers over a complete software-defined vehicular networking on hardware. Recently in [42], the authors have proposed a novel IoV automation and orchestration system using SDN for connected autonomous vehicles. In [43], the authors study different VN architectures and propose a localized intelligence augmented highly reconfigurable software-defined heterogeneous vehicular networking architecture for avoiding single-point failures.

3.1.3. Network Slicing

The network slicing technique takes advantage of NFV and SDN for creating multiple logical networks or network slices over a common physical infrastructure of VN [44,45]. Multiple network slices can be configured over the same physical infrastructure to provide different services with diverse requirements. It provides dynamic resource management by enabling efficient resource sharing by considering various key performance indicators (KPIs) for each slice and can be an efficient technology over the VN with limited resources. Inherently, the network slice contains a chain of physical and virtual network functions that can be placed at different locations based upon the service requirements. In the case of an EC-enabled VN with users requesting services with different KPIs, each slice function needs to be placed carefully for satisfying the user demands. The problem of network slicing and corresponding network function placement over a Multi-EC multi-service VN is discussed later in Scenario 3 in Section 5.

3.2. Vehicular Edge Computing

With the development of IoT and new wireless communication technologies, many new services and applications have been enabled into the VN. Users are demanding services with tighter requirements in terms of latency, bandwidth, computational capability; with limited onboard resources VUs alone are not capable of providing a satisfying Quality of Service (QoS). One way to solve this issue is to use cloud computing, where each VU can transmit their workloads towards the computationally rich cloud servers located deep inside networks. Thus, cloud servers perform the computation operations on behalf of the VUs and send back the results. With high computation resources, the cloud server can serve many VUs with negligible computation latency. However, with a long transmission distance between VU and cloud platforms, this approach introduces large communication delays. Furthermore, user data can be exposed over a large transitional distance can be prone to the security challenges such as third-party attacks. With limited backhaul resources, issues like network connections are likely to happen when many VUs from a given service area request cloud computing services. Thus, providing a satisfying QoS can be challenging over the cloud computing platform.

For solving cloud computing problems, the MEC approach was introduced [46]. MEC brings cloud computing services closer to the end-users by deploying several edge servers in proximity. Thus, users can transmit their workload to these servers without incurring large delays, security issues, or network congestion problems. MEC has gained lots of attention in the recent past and enabled several new latency-critical applications and services over wireless networks [47,48]. In the vehicular scenario, the MEC framework is configured as the deployment of several RSUs along with the road networks and equipping them with the edge servers deployed in the proximity [8]. This approach is called vehicular edge computing (VEC) and has a huge potential for enabling latency-critical and data-intensive VN services [49]. Figure 1, shows the main elements of a reference VEC system which includes distributed VUs, several RSUs along with the road network, and EC servers located nearby RSU nodes. VUs can access VEC services provided by EC servers through RSU nodes with V2I communication links, while they can communicate among them through V2V communication links [50]. RSUs can also act as a gateway for uploading vehicular data to the BS and Cloud facilities.

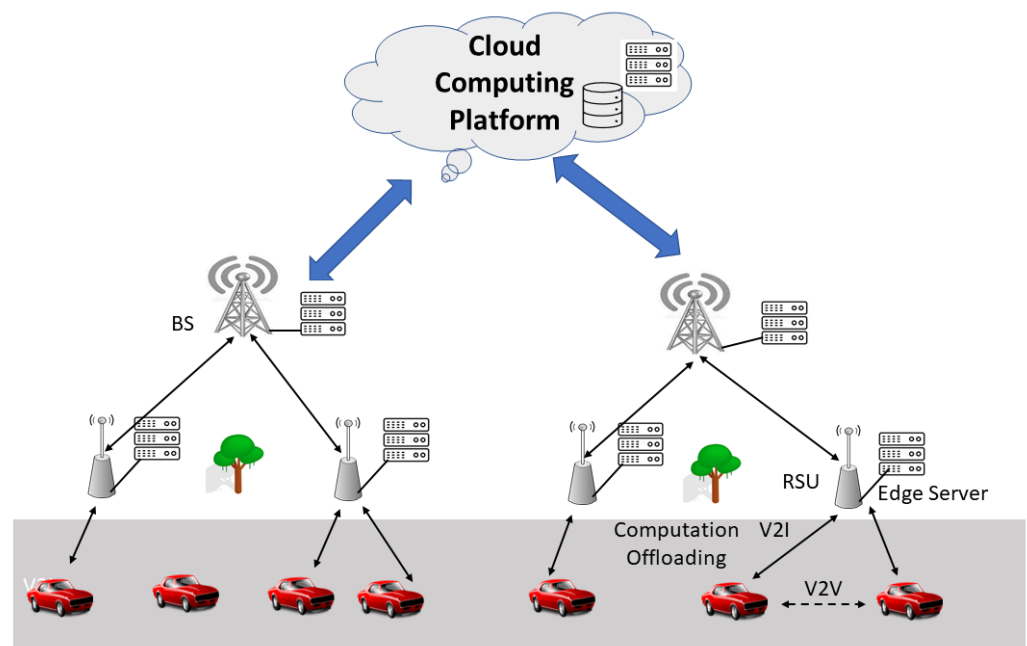


Figure 1. Vehicular edge computing framework.

The limited available resources and RSU coverage are the main bottlenecks for reducing the performance of a VEC system. The resource limitation often increases the VEC system cost in terms of computation latency when multiple VUs request services from the same EC server. The RSU coverage limitations, along with VUs mobility, can add an extra cost in terms of handover latencies. In some works, authors have considered hybrid system models in which both VEC and Cloud computing facilities are considered together for solving the capacity and coverage issues in VEC systems [6,51]. However, such approaches can have some serious drawbacks, such as long transmission distances for accessing cloud facilities. A VN with multiple EC layers of different air and ground networks in proximity (e.g., base station, LAPs, HAPs) can be a better approach compared with a hybrid VEC-cloud system. It can reduce the transmission latencies or network congestion caused by multiple VUs requesting cloud resources and solve the coverage and capacity problems of VEC systems alone.

Within Multi-EC multi-service VN, selection of proper EC platforms and the amount to be offloaded can improve VNs performance. The offloading operations of the surrounding VUs can be useful for other vehicles for a proper offloading decision, such as selection of EC platform, and amount to be offloaded. Such network selection and computation offloading problem have been discussed in the Scenario 2 in Section 5. Table 1 lists the advantages and disadvantage of different EC platforms in vehicular services.

3.3. Machine Learning

The traditional approaches used for solving VNs problems include convex optimization, game-theoretic approaches, and several other metaheuristics. These techniques mainly suffer from the heavy computational burden with exponentially increasing search space in large-scale scenarios; this is even more enforced when the considered VN system employs multiple EC nodes, and network softwarization solutions, enabling more flexible implementations. Heuristic approaches used for solving the VN problems with their NP-hardness are not able to adapt to the increasing complexity of the new applications.

Table 1. Advantages and disadvantages of different EC platforms for VN services and applications.

EC Platform	Advantages	Disadvantages
VEC	Reduced Transmission Distance with Line of Sight (LOS) Communication	Limited Resources, Coverage Range, Frequent Handovers
TN-EC (i.e., BS)	Higher Computation and Communication Resources, Better Coverage Range	High Transmission Delay with Degraded Channel Quality (NLOS Communication)
Cloud	Unlimited Resources and No Coverage Issues	Huge Transmission Delay, Backhaul Network Congestion, Security Issued Due to Long Distance Communication Channels
LAP-EC	Reduced Transmission Distances with LOS communication, Reduced Deployment and Maintenance Time and Costs	Limited Resources, Low Flight Time
HAP-EC	Moderate Transmission Distances with LOS communication, Can Have High Resources, Solar Energy Source	High Deployment and Maintenance Time and Costs Compare with LAP, Communication can be Affected by Rain Fading
Satellite-EC	High Computation and Communication Resources	Large Transmission Distances (not suitable for latency critical VN), Large Deployment and Maintenance Cost

With the addition of IoT techniques, enabling what is usually referred to as IoV [5,50,52,53], vehicles become an excellent element for training data that can be utilized for solving vehicular problems (i.e., ML-based solutions). Several complex vehicular problems such as dynamic resource allocation, traffic predictions, cooperative congestion control, content caching, computation offloading, intrusion detection, anomaly detection can be solved by using popular ML methods such as supervised learning, unsupervised learning, reinforcement learning, etc. [24]. Among other ML techniques, Deep Reinforcement Learning (DRL) is a potential solution for many complex vehicular scenarios, which allows exploiting Deep Neural Networks (DNN) for analyzing VNs data without requiring any prior knowledge of the VN environment, which is hard to capture [54], e.g., correct state transmission matrix over VN states for Markov Decision Processes (MDP) based solutions. ML solutions outperform the heuristic and one-shot-based optimization techniques with better long-term performance.

Different approaches are available for the integration of ML-based solutions into vehicular environments. In each of these approaches, various communication, and computation strategies can be involved during the training process of an ML model. For example, a centralized ML model training approach requires each VN to send its data towards a centralized, more powerful server. In another case, a centralized ML server can further split the training data and corresponding operations with nearby servers to reduce the required computation time. A new collaborative learning-based approach such as federated learning (FL) can allow participating VUs to train ML models locally. The powerful central server can be employed for collecting and averaging the local training ML models parameters to combine the VUs learning experience. These technologies can have certain advantages, disadvantages and can face several new challenges in vehicular environments. In the following, we describe each of these approaches in detail.

3.3.1. Centralized ML

Even though VUs are capable of training fairly complex ML algorithms by themselves, it is yet not recommended due to several reasons. First, with their limited resources, training complex ML algorithms over Vehicles Onboard Units (OBUs) will be computationally expansive and can introduce large latency and energy costs. Complex ML techniques such as DNN require a large amount of data during training, which individual VUs are not capable of providing. Furthermore, dynamic environments like VNs are continuously changing, and the surrounding environments can affect the VUs performance while performing ML model training.

In the centralized learning approach, several randomly distributed VUs transmit their raw data towards a powerful centralized server (ML-server) with rich communication and computation resources. With this approach, fairly complex ML techniques like DNN can be adapted to solve VNs problems with better performance. With these advantages, some challenges need to be considered while implementing centralized ML-based solutions over vehicular environments. Though centralized computation servers are rich with computational resources, ML model training costs can grow exponentially with the increasing complexity of ML techniques (DNN with a high number of layers), which ensures large computation delays at servers. Furthermore, the transmission of the whole dataset towards centralized servers can be challenging with VUs limited communication resources and dynamic channel environments. In the case of VNs, with changing environment dynamics and corresponding renewed datasets, it is important to update the trained ML models for a short duration of time to avoid issues like model drift. Thus, centralized ML model training approach can have several issues while considering it for solving VNs problems.

Figure 2 shows an example of a centralized ML model training over VN where centralized ML server collects training data from individual VUs for performing training operations.

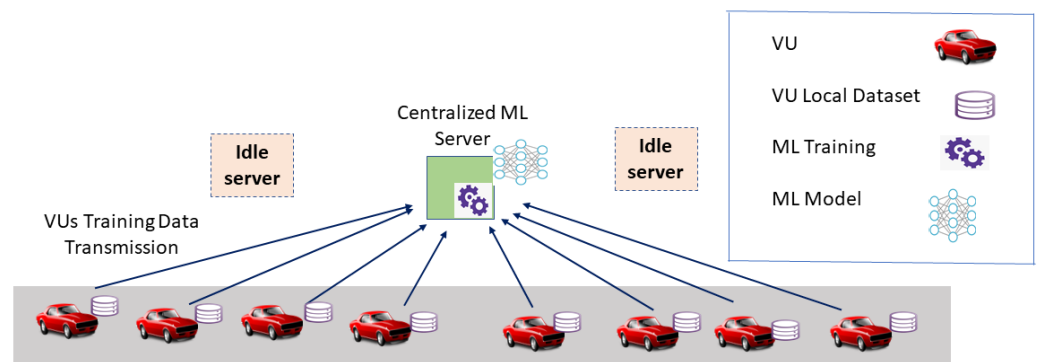


Figure 2. Centralized machine learning.

3.3.2. Distributed ML

For reducing the model training cost at a centralized ML server, the workload distribution methods can be adapted, in which the main server can select a set of powerful computing servers around it and allocate the model training tasks. This approach is known as a distributed learning approach in which multiple powerful servers collaboratively perform the training process. This allows to limit the model training latency at the centralized server and allows to train fairly complex ML models with acceptable latency. However, this approach does not solve the communication overhead problem faced by the individual VUs and thus can have limited performance over VNs.

Figure 3 shows the distributed ML model training over VN. In the case of distributed learning, the centralized server employs the nearby idle server resources for reducing the overall training latency.

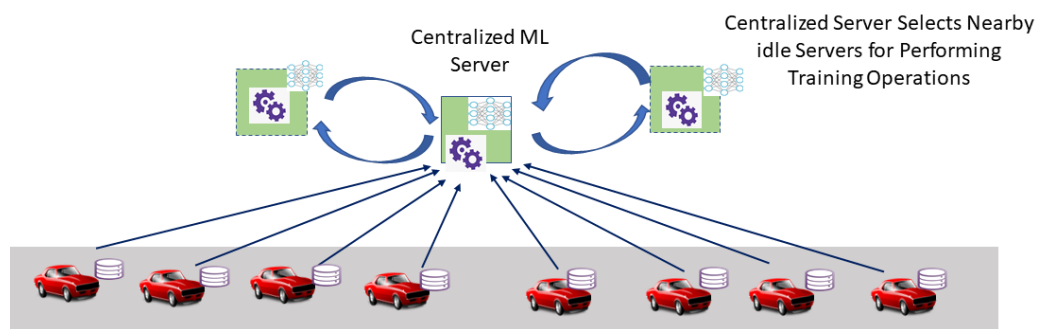


Figure 3. Distributed machine learning.

3.3.3. Federated ML

For overcoming the problems of the centralized and distributed approaches, the FL technique is proposed, where individual devices can perform model training by themselves exploiting local data and transmit only the ML model updates towards a centralized server. In some articles, authors do not distinguish between Distributed Learning and Federated Learning. However, here we have considered these two as separate ML model training techniques with different features [55]. The main differences between FL and distributed learning approaches are listed in Table 2. Once receiving updates from all vehicles in the coverage area, a centralized server performs an averaging operation (i.e., Federated Averaging (FedAvg)) for creating a centralized global model, whose parameters are then transmitted back to the vehicles. The averaging process allows individual vehicles to take advantage of other vehicles’ data and learning experiences for improving their ML models while the local device training process improves the time and energy efficiency of the model training process. Thus, a single FL process communication round includes several steps, such as individual VUs performing the local training operations, the transmission of the local model parameters towards the centralized server, after receiving parameters from an individual VUs, the averaging operation performed at centralized server for creating a global model that allows VUs to take advantage of other VUs training experiences and retransmission of model parameters back to VU. Such communication rounds can be performed for creating a suitable ML model with lesser estimation errors. The complete FL process over VN is shown in Figure 4.

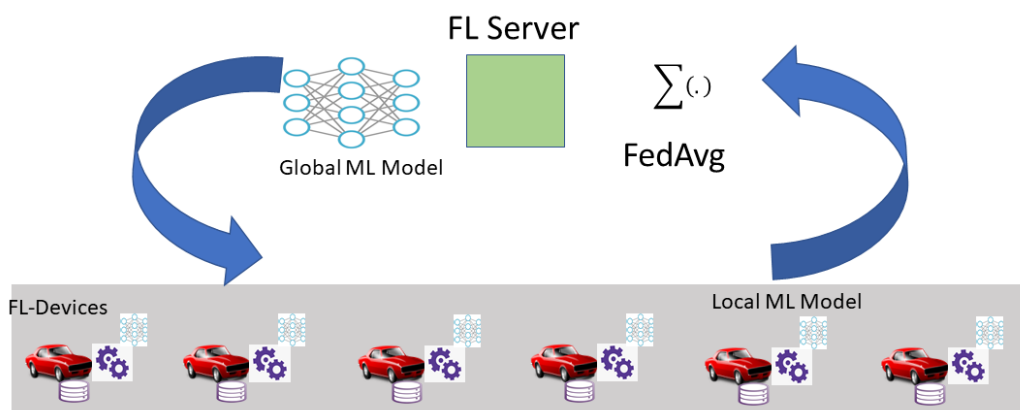


Figure 4. Federated learning.

Algorithm 1 lists the possible steps involved during the FL process. FL process requires several input parameters, including the number of FL devices participating (M), their datasets ($\{D_m\}$), and the maximum number of FL communication rounds ρ . It should be noted that the FL communication rounds limit can also be replaced by any other stopping criteria such as model convergence parameter, loss function value, etc. The initialization

step begins the FL process by defining the initial value of a global FL model parameter and FL communication round to zero (Line 1). At the beginning of each round, local model parameters are updated by the previous rounds' global parameter values (Line 3). Next, FL devices perform the ML model training for generating local ML model w_m^{it} at it^{th} iteration in parallel (Lines 4–5), where η is a learning rate. After completing the local training process, each device forwards its parameters to the FL server (Line 6), where it performs the FedAvg operation to create a new global model (Line 8) which is then used by VUs in next round of communication. FL process continues over ρ communication rounds.

Algorithm 1. Federated Learning.

Input: $M, \rho, \{D_m\}$
Output: w_G^ρ

```

1: Initialize  $w_G^0 \wedge it = 0$ 
2: for each  $it = 1, \dots, \rho$  do
3:   initialize  $w_m^{it} = w_G^{it-1}$ 
4:   for each  $m = 1, \dots, M$  do in parallel
5:     Update  $w_m^{it}$  based upon the  $\eta$  and the local device learning process.
6:     send  $w_m^{it}$  to FL Server
7:   end for
8:   FL server collects all the  $w_m^{it}$  and performs averaging
9:    $w_G^{it} = \frac{1}{M} \sum_{m=1}^M w_m^{it}$ 
10: end for
11: return  $w_G^\rho$ 

```

In general FL based solutions can have many benefits in VN environments, including reduced communication overheads compared with centralized/distributed learning models. However, in the case of a Multi-EC enabled VN, selecting a proper FL-server, FL devices, and the proper number of FL communication rounds can be beneficial in terms of training costs. We will discuss these issues in detail in the Scenario 1 in Section 5.

Table 2. Characteristics of the ML Models over VNs.

Characteristics	Centralized Learning	Distributed Learning	Federated Learning
Learning Entity	Centralized Server	Distributed Servers	On Device (VUs)
Communication Cost/Latency	High	High	Limited
Computation Cost/Latency	High	Limited	Limited
VUs Sensitive Data Privacy	Less	Less	High
Useful for	Training ML models (i.e., DNN with limited number of layers)	Training Models	Complex ML Models with Limited Complexity

4. Multiple Edge Computing Platforms Enabled Air-Ground Network Architecture for Vehicular Scenarios

In this section, we propose a VN architecture with multiple EC layers to serve VUs with diverse service requests. We characterize different EC platform layers in detail and highlight their importance for serving VUs.

TN-based infrastructures are playing an important role in creating a fully functional intelligent VN for futuristic Smart City environments. However, they alone are not capable of providing adequate services to dynamic VUs, which often request services with extremely low latency and high reliability. TN is vulnerable to ground-based security

attacks due to its fixed positions. Furthermore, in the case of natural disasters such as tsunamis, earthquakes users often fail to connect to the services of TN. Low accessibility into remote areas mainly because of the unwillingness of mobile operators to provide services into low revenue parts needs to be considered while integrating TN into VNs. NTN such as Air networks has a considerable advantage over TN in terms of availability, reliability, scalability, and low deployment costs. With these advantages, they can play an important role in complimenting TN for providing better quality services to VUs. Both TN and NTN platforms can enable EC-based services through placements of edge computing servers along with their distributed infrastructures. Therefore, a Multilayered joint T-NTN constituted by different EC platforms over TN and NTN can be utilized for providing heterogeneous services requested by VUs with demanded quality.

In Figure 5, we propose an air-ground network having multiple EC platform layers and jointly exploiting TN and NTN for serving VUs.

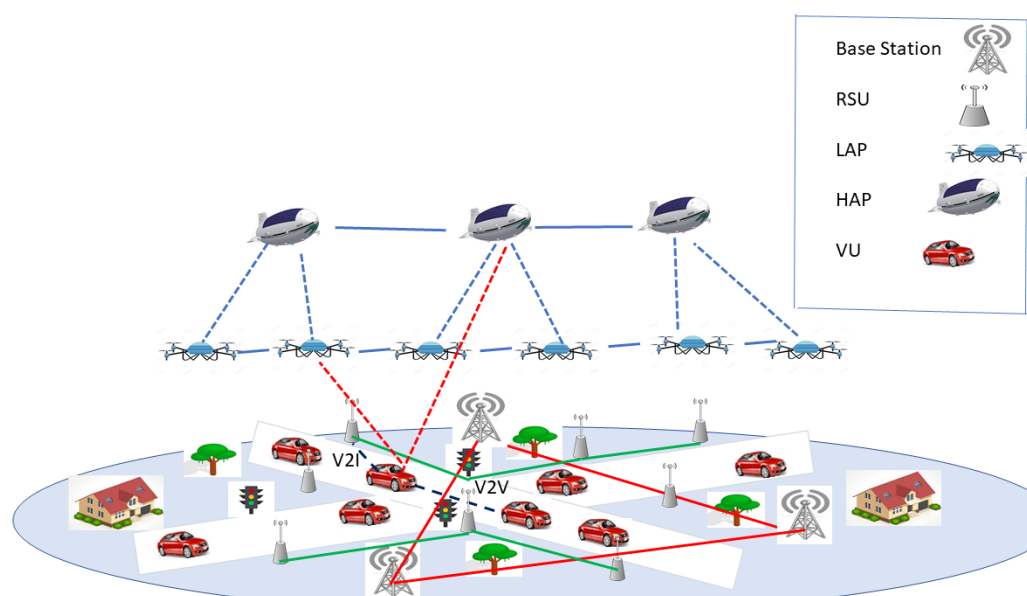


Figure 5. Multiple EC enabled VN.

Several VUs are randomly distributed in the considered service area and demand many latency-critical and data-intensive services. The proposed network architecture is constituted by several elements as shown in Figure 5: a set of VUs, RSU, BS elements, LAPs, and HAP. For avoiding redundancies, in the following, we omit the legend from the figures. A set of RSUs is deployed alongside roads having a limited capacity EC servers for providing computation services to the VU. Each RSU can serve a limited set of VUs in its coverage range. VUs are also supported by the BS elements equipped with EC services. BS elements can serve VUs over larger coverage areas compared with RSUs and can have powerful EC servers. However, the required roundtrip time for sending and receiving back the VUs task can be much higher. Thus, RSUs and BC servers jointly forms a multi-layered TN-based EC service platform for supporting VUs. VUs are also complemented by a swarm of LAPs (i.e., UAVs) deployed on top of them. UAVs can enable limited EC services with a pre-installed EC server. UAVs can have limited coverage and flight time that often limits their service range. VUs are also covered by multiple decentralized HAPs having better coverage and computing capacity compared with UAVs. Thus, Jointly, UAVs and HAPs form a multilayered NTN based EC platform for supporting VUs with better quality services and intelligent applications. Jointly both TN and NTN based EC platforms serve VUs with their available computation and communication resources.

As shown in Figure 6, the considered network infrastructure can be split into several layers of ground-based and areal networking infrastructures. The TN is constituted by several connected VUs in Layer 1 which can form a small vehicular cloud, RSUs equipped

with the EC servers in Layer 2, and multiple BS with EC facilities in Layer 3. The NTN has two layers of LAPs and HAPs belonging to the areal networks. LAP nodes are located at a relatively low distance from the ground compared with the HAPs and have limited computation and communication resources. Though HAPs are at a long distance from the VUs layer they can serve large coverage areas. Below we describe each of these networking layers in detail.

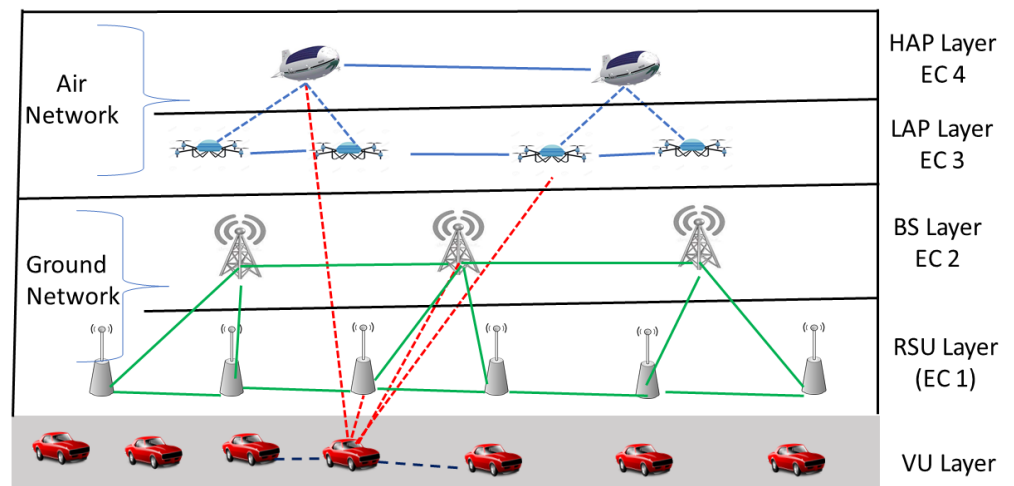


Figure 6. The Multi-EC framework for the vehicular scenario.

Layer 1—Connected VUs layer

Several connected VUs having communication and computation capabilities are grouped into Layer 1. Each VU can communicate with its neighboring VUs for possible information sharing through V2V communication technologies and use V2I links for interacting with other EC layers. VUs can communicate over a limited distance to share important information for enabling several safety-related, traffic flow management services that make drivers life easy on the road. Through V2I communication, VUs can share their workloads towards EC servers in proximity for enabling latency-critical and data-intensive applications. VUs often generate different tasks requests with specific requirements of computation, communication, and storage resources. These task requests often come with additional requirements (i.e., critical latency requirements), for which VUs often need assistance from the EC platforms in the proximity.

Layer 2—RSU-Edge Computing (RSU-EC) Layer

For enabling the EC facilities into VNs, several RSUs have been deployed alongside road infrastructures. RSUs can have communication technologies installed for communicating with VUs and other higher lever networking layers (i.e., cellular BSs, cloud infrastructures, etc.). EC servers having limited computation and storage resources can be deployed alongside RSUs to integrate the EC services into VNs. Thus, RSUs equipped with EC servers and communication technologies constitute an EC layer in the proximity of VUs for providing low latency services. However, with their limited resources and coverage, RSUs can serve a limited number of VUs. Furthermore, VUs mobility often restricts them from accessing RSU services for longer periods of time.

Layer 3—5G Base Station (BS) Layer

5G base stations (5G-gNB) have integrated modern communication and computation technologies that can be exploited as an EC platform. With larger coverage areas, they can serve a higher number of VUs. However, the BS-EC platform can have additional transmission delays due to longer communication distance compared with RSUs.

Layer 4—Low Altitude Platform (LAP) Layer

LAP platforms such as UAVs equipped with EC servers can add several benefits in terms of a reduced transmission time (less than 1 msec), reduced deployment and maintenance time, line of sight communications with better channel quality, etc. They can also act as a relay node for allowing VUs to transmit their information towards higher air networking layers such as HAP with reduced latency and energy costs. With the limited size and reduced flight times, UAVs can only have limited communication and computation resources onboard.

Layer 5—High Altitude Platform (HAP) Layer

A HAP network constituted by several nodes able to communicate amongst themselves and other infrastructures is forming another EC layer in the air network. Each HAP node can have a powerful EC server and the communication resources for serving VUs under its coverage. Better coverage, additional renewable energy sources, better stability are some of the HAPs main advantages over LAPs. However, HAP performance can be reduced by longer communication distances, additional channel loss in terms of rain fading high deployments, and maintenance time and costs. The HAPs coverage area can depend upon its altitude. In general HAP platform can provide coverage between a few 10 s of Kms to up to several 100 s of Km [56,57].

Communication, Computation and Storage Characteristics

Each EC platform layer of the proposed network architecture can adapt different computation and communication strategies. Several virtualization techniques (i.e., Virtual Machines (VMs), containers) can be used for the efficient utilization of EC resources. Furthermore, an SDN-based centralized control approach can be applied for managing the computation and storage resources of individual EC platforms. Multiple operators based communication technologies can be adapted for enabling the communication between EC nodes of the same and distinct layers. Table 3 lists the most important characteristics of individual EC layers considered in the proposed network architecture.

Table 3. EC platforms' characteristics.

EC Platform	VU Cloud	RSU EC	LAP EC	5G-BS EC	HAP EC
Computation Resources	Low	Limited	Limited	High	High
Communication	Low	Limited	Limited	High	High
Storage	Low	Limited	Low	High	Limited
Coverage	few 10 s m	few 100 m	few km	few km	up to 200 km
Energy Source	Electric/Fuel Cells	Electric Grid	Fuel Cells	Electric Grid	Fuel Cells/Solar

5. Proposed Multi-EC Enabled Air-Ground VN Scenarios

In this section, we describe the most prominent vehicular scenarios gaining more from the presence of a Multi-EC enabled air-ground architecture, by considering to serve VUs with different service requests. In particular, we have considered three main aspects able to enhance actual vehicular scenarios, i.e., the use of efficient ML platform, the exploitation of computation offloading approaches, and the optimal network function placement focusing on a slice-based Multi-EC multi-services VN architecture. The prominent scenarios are described in detail, by listing the important challenges ahead and the possible solution methods involving ML-based approaches for addressing them over VN environments.

5.1. Cost Efficient Federated Learning Platform for Vehicular Applications

5.1.1. Scenario Description

For the case of VN, each connected vehicle is equipped with multiple sensor devices that can generate tons of data in real-time. These data, through proper analysis, can be utilized for providing a better QoS to the end-users. Several ML tools are available for analyzing data and inferring important information from them. Furthermore, there are multiple ways for performing the ML model training over dynamic wireless environments, like the VN. A centralized ML model training approach is not much suitable technique to perform over VNs, mainly because of high communication and computation overheads. If a centralized training approach is adapted over VNs, VUs need to transmit their complete data towards the centralized server which performs training operations. Such an approach adds significant cost in terms of data transmission and training latency at the server. Thus, other distributed and collaborative training approaches like FL can be a better approach for ML model training over VUs data.

The importance of FL-based approaches for solving VUs problems is highlighted in several recent works [19–21]. Here we aim at proposing an efficient FL-based platform for generic VN applications such as optimal pathfinding, multimedia streaming, collision avoidance, etc. The model training cost of the FL process should be analyzed for implementing the optimal FL-based platform to solve the vehicular problem.

5.1.2. Possible Challenges

Here we describe the important challenges that need to be considered for implementing an efficient FL platform over a Multi-EC enabled VN.

- **Proper FL-server selection** The FL process training cost can depend upon the number of users participating in the training process, channel environment, the transmission distance between FL devices and the server, type of ML model, types and quality of datasets, etc. Selecting a proper FL-server can limit the training cost. The FL server should have better coverage, high computation capacity, a short transmission distance, and better channel environments. However, most of these properties are contradictory and thus require tradeoffs while selecting the FL server node. For example, RSU EC nodes can be considered as an FL-server for reducing transmission distances and better quality channel environments but with coverage limitations, only a few VUs can participate in the training process. Similarly, considering the LAPs/UAVs as FL-server can be costly due to their limited coverage, shorter flight times, etc. On the other hand, HAP-EC nodes can have better coverage for allowing many VUs to participate in the training process; however, long transmission distance and poor quality channel environment can be challenging.
- **FL Device Selection** It is not always possible and advisable to consider all VUs data while performing the FL training process. It is often the case that some of the VUs might not be able to transmit their ML model parameters to the FL server because of poor transmission medium. Some VUs might have a poor quality dataset which results in poor training performance. Considering such VUs into the FL training process can degrade the overall training process performance. Therefore a suitable strategy is required to be implemented for selecting a proper set of VUs for the training process.
- **Training Cost Vs Convergence in FL** During FL, the training process can be continued till the loss function value converges to some predefined value. For having a reduced FL training process, the number of FL rounds performed by each VU can be limited based upon its available resources.

5.1.3. Proposed Solutions

We have considered two different approaches for implementing an efficient FL platform over VN assisted by multiple EC layers.

Hierarchical FL Platform over Multi-EC VN Architectures

In this approach, we develop a hierarchical learning-based FL platform over the Joint T-NTN (Figure 7). The HAP with larger coverage and resources acts as FL-server, while VUs (FL-devices) with local datasets perform model training. For the conventional FL process, VUs transmit the local model parameters to the HAP. This can result in higher transmission times, high energy costs, and often VUs dropouts from the training process mainly because of unreliable long-distance communication channels between VUs and HAP. In the case of the proposed hierarchical learning approach, instead of transmitting their model parameters to the HAP server over long distance/unreliable communication links, VUs resort to sending them to the nearby EC layers that includes LAP and RSU based EC servers. After receiving data from VUs, nearby EC servers perform the averaging operation to create a sub-global model parameter and transmit them to the HAP server. In this way, the amount of connections over long-distance links is limited and, as a result, saves the training process time and energy for each FL communication round. Furthermore, BC-EC nodes are used as a backup for avoiding service failures when HAP fails to connect with other platforms, mainly because of security issues or environmental conditions. In this case, nearby ECs transmit their model parameters to the BS instead of HAP.

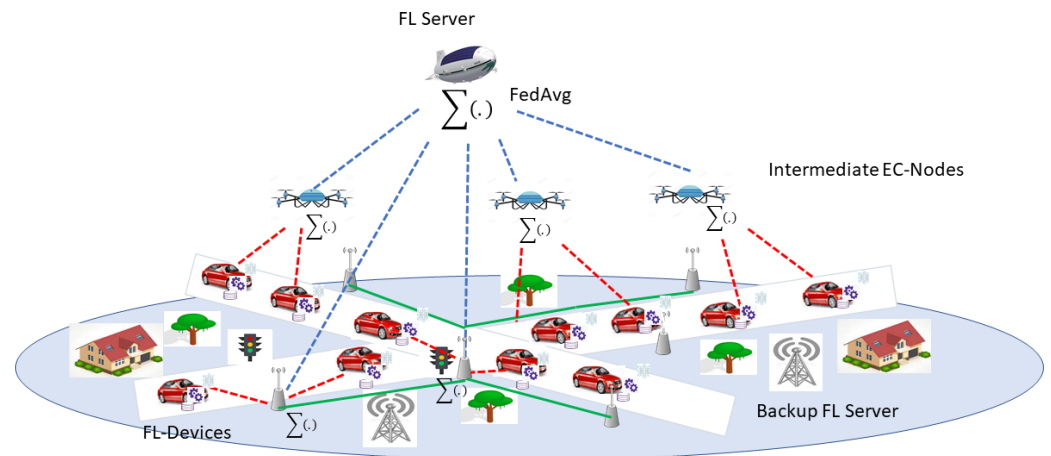


Figure 7. The hierarchical FL platform over a Multi-EC VN architecture.

Computation Offloading Based FL Platform over Multi-EC VN Architectures

In the case of a conventional FL process, many VUs drop out from the training process mainly because of limited computation/communication resources onboard. If a VU has limited computation resources, performing a local training process can result in a larger computation time and may not be able to send their updates to the FL server in time. For avoiding such issues during FL, a computation offloading-based FL platform can be used, exploiting the computation offloading services from the nearby EC servers for allowing VUs with scarce resources to participate in the FL process (Figure 8). In this approach, VUs mobility, limited coverage range of nearby EC servers, the security threats of EC platforms are required to be taken into account.

In Table 4, the most important characteristics of the proposed solutions for the cost-efficient FL platform implementation over Multi-EC enabled VN architecture are summarized.

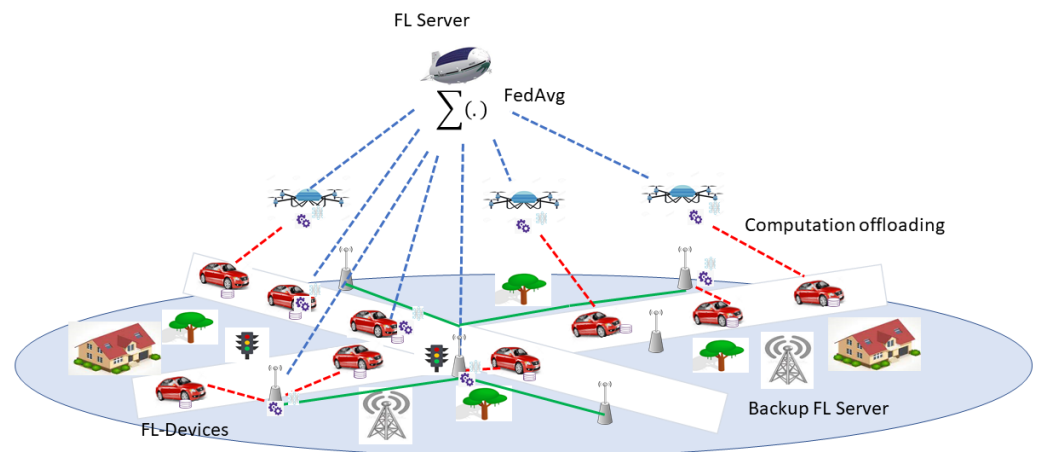


Figure 8. Computation Offloading for FL.

Table 4. The characteristics of the proposed solutions for the cost efficient FL platform.

FL Platform	VUs Computation Latency/Energy	VUs Communication Latency/Energy	Data Privacy	VUs Participation	Possible Challenges
Hierarchical FL	High on-device computation cost	Reduced VUs communication delay/energy due to short-distance communication between VUs and LAP/RSUs	FL model training process is performed by VUs alone without sharing data with any other nodes (privacy-preserving).	High on device training latency and energy cost can limit the number of VUs participating in the FL process.	Selection of proper LAP/RSU node (for mid-layer processing) is required for avoiding additional hand-over costs.
Computation Offloading based FL	Offloading process reduces the computation time and energy during the training process,	High Communication Delay/Energy Cost Due to Long Distance Communication	Large Number of VUs can Participate into FL Process.	VUs Data Needs to be Shared with the Outside World.	Selection of a Proper Edge Node and Offloading Amount During Offloading, Requires VUs data to be Send to the Outside World.

5.2. ML for Computation Offloading in Multi-Service Multi-EC Vehicular Environments

5.2.1. Scenario Description

In multi-service multi-vehicular environments, several VUs randomly distributed over large coverage areas generates different kinds of vehicular services requests. In general a service request can be represented by a tuple $\langle D_{s,m}, \Omega_{s,m}, \bar{T}_{s,m} \rangle$ where, s is the particular service type, m is the index VU requesting s , $D_{s,m}$ is the service task size in Byte, $\Omega_{s,m}$ are the requested CPU execution cycles and $\bar{T}_{s,m}$ is the maximum latency of the requested service. With limited onboard resources, VUs alone cannot handle such requests and require computation offloading services from different EC platforms over joint T-NTN, allowing them to offload a complete/certain portion of their workload towards EC nodes. This allows VUs to share their workload with EC nodes for halving the overall task processing costs (e.g., task processing latency energy). However, in the case of a Multi-EC multi-service vehicular environment, for handling a large number of service requests of different kinds, the selection of a proper EC platform and its individual EC node/nodes for offloading can be a challenging problem to be solved. This is due mainly because of different natures of services, varying coverage limits transmission distances, and available resources of EC platform layers. Moreover, selecting the wrong EC platform can add larger latency and energy overheads or might even create service failures. For example, if the requested service is a critical latency service, such as autonomous driving or drivers safety-related services (i.e., small $\bar{T}_{s,m}$), selecting a BS-EC or HAP-EC platform can not be a practical solution due to longer transmission delays (see Table 3). On the other hand, if the requested service is data-intensive, requiring larger computation resources (i.e., large $\Omega_{s,m}$), selecting

RSU-EC or LAP-EC might not be a good idea. With their limited computation capabilities, overall computation latency can become very high. During computation offloading, VUs might need to transfer sensitive data towards EC-node. Therefore, a proper assessment of EC-nodes security is required to be done in advance.

Furthermore, individual EC nodes of EC platforms can only have a limited number of services available with them mainly because of resource limits. Therefore, selecting a proper EC node of an EC platform is required to avoid offloading service failures. VUs mobility limits are also needed to be considered before making offloading decisions. For example, if a certain VU offloads its content towards an RSU-EC node, it has to complete the offloading procedure before it passes through the coverage range of RSUs. If failed, additional latency and energy costs in terms of RSU-handovers can be unbearable for the critical latency services. Thus, over a Multi-EC multi-services vehicular environment, finding a proper EC platform and selecting a proper EC-node along with the amount to be offloaded based upon VUs service request constraints and mobility limitation is an important problem to be solved.

5.2.2. Possible Challenges

The main challenges that need to be considered during computation offloading over a Multi-EC multi-service vehicular environment are described below.

- **Resource limitations of EC nodes:** Each EC node can have limited communication, computing, and storage resources through which they can only serve a limited number of users with acceptable costs. Furthermore, they can hold a limited number of services that can be accessed by users mainly because of resource limits.
- **Transmission Distance vs. Capacity Tradeoff:** For the case of Multi-EC multi-service VNs, an important coverage-vs-capacity tradeoff needs to be considered, while selecting an EC platform and an amount to be offloaded. The nearby EC platforms, such as RSUs and UAVs, can have a limited capacity and coverage range. Therefore they can serve only a few users with acceptable latency. On the other hand, 5G BS and HAP can have powerful EC servers with a better coverage range. However, they are located at further distances from VUs and can have longer transmission distances with poor quality channel environments. Therefore, a proper EC selection needs to be done for taking advantage of EC benefits from different platforms.
- **Vulnerability of EC servers towards Third-party Attacks:** It is often the case that during computation offloading VUs share a piece of sensitive information, including location, speed, possible directions/destination with the EC servers. If EC is vulnerable to third-party attacks, it can be dangerous to share such sensitive data with them, especially in the case of autonomous driving environments where an attacker can take control of vehicles and can cause unimaginable damages.
- **VUs Mobility and a Handover Latency:** In dynamic environments, like VNs, VUs mobility limits the computation offloading performance. The offloading process should be completed before VUs pass through the coverage range of an EC node, failure of which can add additional costs in terms of handover latency. VUs mobility can also impact the channel characteristics in terms of unstable channel behaviors, Doppler spread, etc, which can seriously degrade the offloading performance. In the past, researchers have adopted different approaches for modeling VUs mobility. In some cases, mobility characteristics are analyzed with mathematical frameworks (i.e., mobility models) with certain assumptions over VUs speed, directions, surrounding environments, etc. In some other cases, researchers have resorted to the trajectory predictions of VUs movements. In particular, the VUs mobility parameters can be used for trajectory prediction; thanks to this we are able to predict how the system resources can be used and in case reserve them. In the past, some authors have considered such approaches [49].

5.2.3. Proposed Solutions

We have considered two different approaches for efficiently manage the computation offloading challenges in a vehicular scenario.

V2V Technology for Finding the Proper VN-EC Pair during Offloading

With recent advances in communication technologies, connected VUs are able to communicate with each other over short distances by using V2V communication links. Such communication capabilities allow them to share important information with each other. Through the V2V communication links, VUs can share their offloading experiences, helping to select proper EC platforms and their nodes for computation offloading. For example, if a particular EC server is compromised in terms of security or has a low resource pool for serving new VUs, new VUs can be warned over V2V communication links for avoiding such EC for offloading data.

It is not practical for VUs to communicate with other vehicles, which are far away from them due to the short distance communication capabilities. Furthermore, individual VUs can not always collect information from each other and analyze it in real-time for taking decisions. By taking this consideration into account, here we propose a cluster-based approach, where a cluster head is selected to perform the data collection, data analysis, and sharing of findings with requesting VUs. The selection of a cluster head can be based upon several criteria, such as available resources, its distance from other VUs, etc. For solving the data analysis problem to find and forward the adequate set of EC nodes, various solution techniques can be adapted.

However, conventional solution methods such as convex optimization, heuristic, and metaheuristic methods need to be avoided since implementation costs can be huge. An ML-based solution technique, where a pre-trained ML model deployed over a cluster aimed at analyzing the real-time input data from other VUs over a cluster head can be an excellent solution to be considered. Once analyzed, the findings can be either broadcast to individual VUs or can also be provided in terms of request-response basis to avoid additional costs. Figure 9 shows the important steps considered.

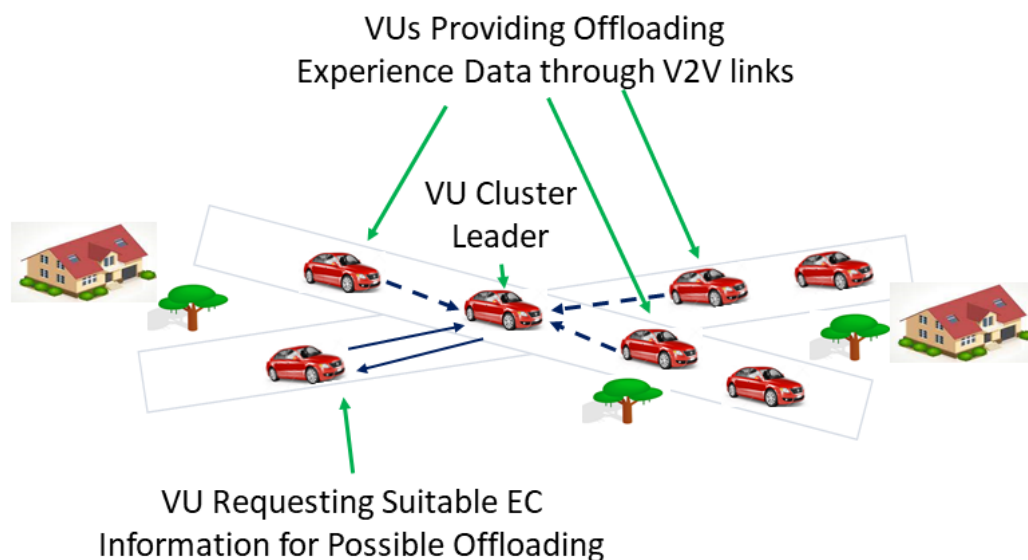


Figure 9. Intelligent solution for the offloading problem.

Multiple Edge Connectivity Mode for VU-EC Server Task Offloading

By resorting to one of the possible management solutions for managing EC nodes in 5G, we introduce here the possibility of integrating the presence of logical anchor points in our network [58]. Different anchor point-based connectivity strategies can be adapted for computation offloading between VU and EC platforms.

A simple distributed anchor point-based strategy allows VUs to select different EC servers from multiple EC platforms available for offloading. In another case, a primary-secondary server approach can be adapted. In this case, VU transfers its data to the selected EC server, which acts as a primary anchor point. After receiving VUs data, it can then further offload to the nearby secondary EC servers for reducing its workload. This strategy can reduce the overall computation time required for processing the overall data. In another case, VU can offload its data to multiple EC platforms simultaneously.

Figure 10 shows these three strategies adapted over a proposed Multi-EC enabled VN architecture. VU1 uses a distributed anchor point-based strategy and selects a BS-EC platform for computation offloading. VU2 uses a primary-secondary server-based strategy and offloads data to the BS-EC server. Being a primary anchor point, the BS-EC server selects two UAVs as secondary anchor points for reducing the overall computation time. This approach allows EC servers that are not able to server VUs in the first place mainly because of coverage limitations to act as a secondary anchor point for improving VUs operations. In the third case, VU3 itself selects multiple anchor points for data offloading. With multiple splits, it offloads a portion of the computation load to the selected anchor points.

In Table 5, we evaluate the different features of the proposed solutions for the computation offloading problem in multi-service Multi-EC vehicular environments.

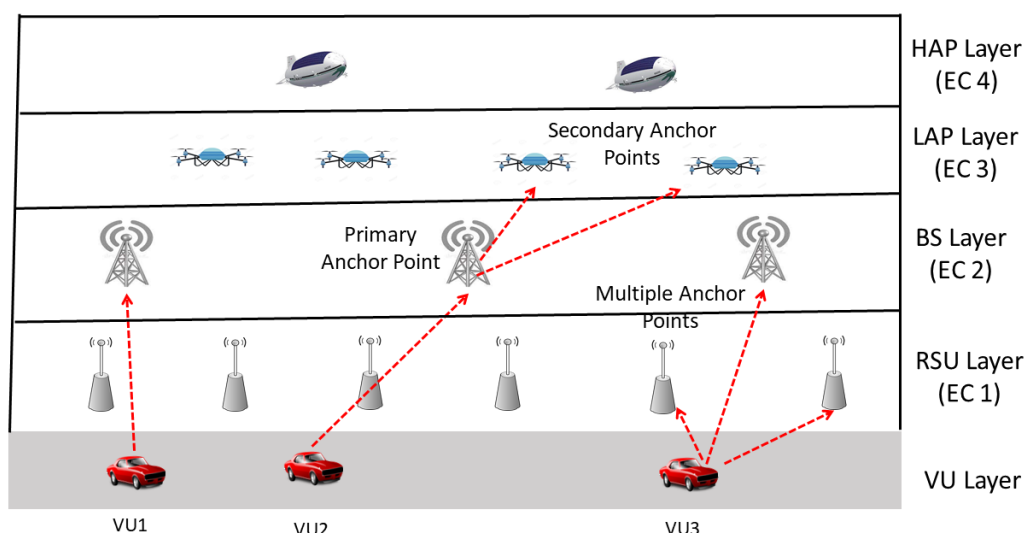


Figure 10. Anchor point based strategies.

Table 5. Computation offloading in multi-service Multi-EC vehicular environments.

Computation Offloading Solution	Offloading Servers	Number of Offloading Servers	Flexibility	Link Failures	Task Scheduler
V2V Technology for Finding the Proper VN-EC Pair During Offloading	VUs can offload data to the limited number of servers	Only One	Limited	For the case of link failure, the complete task needs to be re-transmitted	Scheduler complexity can be limited
Multiple Edge Connectivity Mode for VU-EC Server Task Offloading	VUs can offload data to the servers beyond it's coverage range (i.e., secondary servers)	Multiple	More Flexible Strategy	Link failure damage can be limited due to the decentralized data processing	A proper task scheduling is required for having better performance

5.3. ML-Based Network Function Placement Approaches for Vehicular Services over Multi-Service Multi-EC Vehicular Environments

5.3.1. Scenario Description

Network Slicing is considered an important enabling technology for the next generation 5G network architecture for multiple logical networks over a common physical infrastructure. Several VNFs can be embedded together to form a network slice associated with a particular networking service. VNF embedding and placement over several virtualized networking platforms is an important problem to be solved. Different services can have heterogeneous requirements and VNF placement needs to consider such requirements before performing VNF placement.

In this scenario, we consider the Network Function Placement (NFP) problem for multiple slices of VN services with heterogeneous requirements over the Multi-EC enabled VN. Based upon each slice requirement, VNF placement is performed over different EC platforms by considering their limited resources. The scenario is depicted in Figure 11, where two possible slices are deployed on network infrastructure.

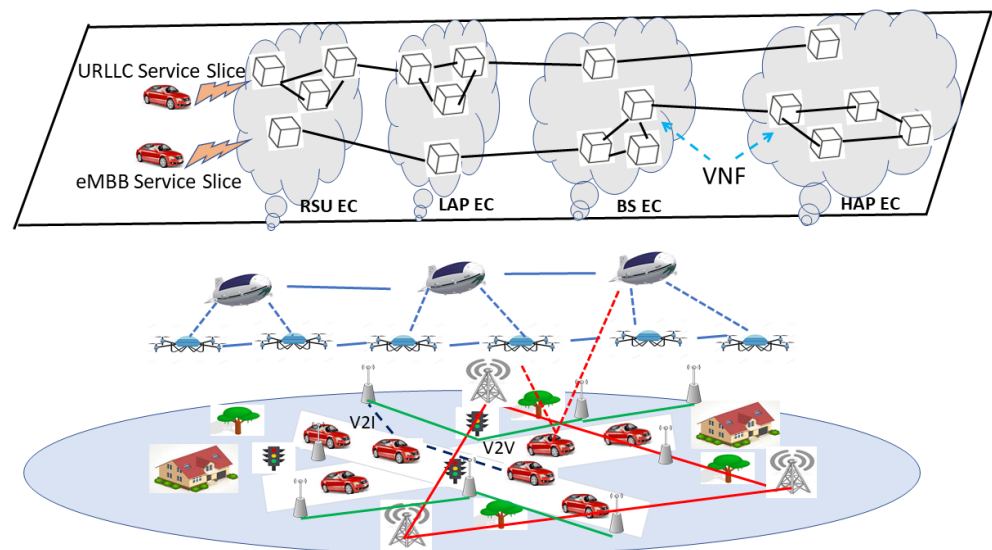


Figure 11. Network Slicing based VNF placement.

5.3.2. Possible Challenges

- **Resource Allocation problem:** In the vehicular environment, VUs often request multiple slices with stringent requirements. As described before, each network slice can be embedded as multiple VNFs embedded together. Each VNF of a network slice can have specific computation demands. Allocating a sufficient amount of resources for each slice, considering their particular requirements is a challenging problem over Multi-EC-enabled VN and needs to be addressed carefully for reducing service failures.
- **VNF Placement Problem:** Each service request can have specific demands in terms of latency, data rate, etc. To satisfy such requests performing a proper VNF placement over multi-EC enabled VN is can be beneficial. For example, to fulfill the stringent latency requirements of Ultra-Reliable Low Latency Communications (URLLC) based slices, placing several VNFs into proximity of an end-user (i.e., over RSU-EC or LAP-EC) can be beneficial with reduced transmission delays. However, in the case of enhanced Mobile Broadband (eMBB) slice, requiring high data rates, it is needed to explore higher layers of EC platforms, (i.e., BS-EC, HAP-EC, etc.) for having sufficient resources.

5.3.3. Proposed Solution

Preference Weighted Network Function Placement for a Heterogenous Vehicular Service Slices

Each VUs service request comes with a specific demand. Required service latency and the data rate are two main characteristics of demanded services. The VN services can be differentiated into two main categories. A set of slices with high data rate requirements can be grouped as eMBB Service slices. Services with critical latency requirements are grouped into URLLC classes. Each of these services can be deployed as a chain of VNFs.

For avoiding the excess transmission delay between VNFs, to satisfy the critical latency requirement of service, it is preferred to place VNFs associated with URLLC services in proximity. On the other hand, for satisfying the user data rate requirements of eMBB, several VNFs of eMBB slices can be placed on platforms with large computation capabilities. Such an approach can have many benefits for serving many users with efficient utilization of EC resources. However, such placement can not be done with a random guess, and a proper strategy is required. In one of our past works, we have considered a network operator-biased approach for multi-service network function placement in a 5G network slicing architecture [59]. In this approach, we have defined a preference function that assigns a proper weigh value for each slice's functions based upon the slice category and the function types. A similar strategy can be adapted for VUs slices over a Multi-EC enabled VN. An ML-based intelligent strategy can be applied for designing a preference function for associating a proper weight for each VNFs placement over different platform layers.

Thus, by analyzing the individual VUs service request data and the available resources of EC platforms, an intelligent approach can be defined, for solving both VU slice function placement and resource allocation problem over a Multi-EC enabled VN. Table 6 presents the important features of a proposed solution for the NFP problem for vehicular services over multi-service Multi-EC vehicular environments.

Table 6. Network function placement over multi-service Multi-EC vehicular environments.

NFP Solution	Resource Allocation	Service Demands	Service Prioritization	Challenges
Preference Weighted NFP for a Heterogenous Vehicular service slices	Enables efficient allocation of computation and communication resources of various platforms to the different slices functions based upon their requirements.	Various services demands in terms of low latency, high data rates can be fulfilled by preference weighted flexible NFP.	It is possible to prioritize the services with stringent requirements by allocating their functions first.	A proper preference function mapping various services demands need to be defined. Various platforms having heterogeneous hardware and communication technologies need to work coherently to have benefits

6. Discussion

In the previous section, we have discussed various vehicular scenarios that can be considered for the proposed Multi-EC enable VN architecture.

In Scenario 1, we discuss the importance of a cost-efficient FL platform over a Multi-EC enabled Vn architecture. The selection of proper FL devices having high-quality datasets and FL servers with better coverage characteristics can help to reduce the overall FL process convergence cost. We have proposed two innovative solutions, including the hierarchical FL-based and the computation offloading enabled FL platform, to address the challenges of this scenario.

In Scenario 2, we have discussed the important characteristics and corresponding challenges of the efficient computation offloading process over Multi-EC enabled VN architecture for enabling latency-critical and data-intensive applications and services. Proper EC server selection and the amount to be offloaded can reduce the overall process cost. VUs can utilize other VUs offloading experiences for making better decisions. We

propose a V2V data sharing enabled ML approach and multiple edge connectivity-based solution approaches for addressing the challenges of a computation offloading problem.

In Scenario 3, we have addressed the NFP problem for the multi-service Multi-EC enabled VN architecture. Network slicing technology can realize heterogeneous vehicular services with different requirements over common VN infrastructure. Specific services are implemented as a chain of VNFs with multiple possible placement options. VNs limited resources and service requirements need to be considered during function placements over various platforms available in Multi-EC enabled VN. We propose a preference-weighted approach for the NFP problem where the preference function models the specific weighting associated with each VNF based upon its characteristics. Innovative ML-based solutions can be used for modeling the proper preference function based upon the service characteristics.

The most prominent characteristics, challenges, and the proposed solution methods for the three proposed scenarios are summarized in Table 7.

Table 7. VN Scenarios Characteristics.

VN Scenarios	Key Features	Challenges	Proposed Solutions
Cost Efficient Federated Learning Platform for Vehicular Applications	<ul style="list-style-type: none"> • A cost-efficient FL platform is required over a resource-constrained VN. • The communication and computation cost of each FL iteration needs to be analyzed. • FL servers with better coverage characteristics can reduce training costs. • FL devices with better quality datasets can reduce the FL process cost. 	<ul style="list-style-type: none"> • Proper FL-server selection • FL device selection • Training cost vs. convergence in FL 	<ul style="list-style-type: none"> • Hierarchical FL platform over Multi-EC VN architecture • Computation offloading based FL platform over Multi-EC VN architecture
ML for Computation Offloading in Multi-Service Multi-EC Vehicular Environments	<ul style="list-style-type: none"> • Computation offloading enables new latency-critical and data-intensive applications and services • Proper EC server selection can reduce the processing cost. • An optimal amount of data should be offloaded to have benefits. • VUs can share important data for making better offloading decisions. • Several EC server selection technologies can be adapted. 	<ul style="list-style-type: none"> • Resource limit of EC nodes • Transmission distance vs. capacity trade-off • Vulnerability of EC servers towards third-party attacks • VUs mobility and a handover latency 	<ul style="list-style-type: none"> • V2V technology for finding the proper VN-EC pair during Offloading • Multiple edge connectivity mode for VU-EC server task offloading
ML-Based NFP Approach for Vehicular Services over Multi-Service Multi-EC Vehicular Environments	<ul style="list-style-type: none"> • Network slicing technology can be adapted for providing multiple VN services with heterogeneous requirements. • Service-based proper resource allocation needs over resource-constrained VN. • It is possible to place VNF at different locations over a Multi-EC enabled VN. 	<ul style="list-style-type: none"> • Resource allocation problem • VNF placement problem 	<ul style="list-style-type: none"> • Preference weighted NFP for a heterogeneous vehicular service slices

7. Conclusions

Modern cities have to face with several challenges, and of the most important for the citizen-life is the management of an efficient ITS. In this paper, we aim at designing some possible scenarios where EC solutions allows to solve some of the challenges of VNs. Differently from other approaches we aim at integrating NTNs in the system so as to create a multi-level Air-Ground architecture. The paper discusses the possible exploitation of network softwarization technologies, vehicular EC and ML for creating an intelligent system able to fulfill the vehicular network requirements. The proposed solutions are discussed analyzing the main challenges and the solutions they are able to take into account. Our future works will be focused on implementing the considered scenarios for understanding their feasibility and how each one can cope with the given challenges.

Author Contributions: Conceptualization, D.T.; investigation, S.S.S.; writing—original draft preparation, S.S.S.; writing—review and editing, D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ICT	Information and Communication Technologies
EC	Edge Computing
VN	Vehicular Networks
RSU	Road Side Units
BS	Base Stations
VUs	Vehicular Users
ITS	Intelligent Transportation System
IoT	Internet of Things
IoV	Internet of Vehicles
V2V	Vehicle-to-Vehicle
V2R	Vehicle-to-Road Side Units
V2I	Vehicle-to-Infrastructure
V2S	Vehicle-to-Sensors
V2P	Vehicle-to-Person
VEC	Vehicular Edge Computing
TN	Terrestrial Networks
NTN	Non-terrestrial Networks
LAP	Low Altitude Platforms
HAP	High Altitude Platforms
ML	Machine Learning
MEC	Mobile Edge Computing
NR	New radio
3GPP	3rd generation partnership project
UAV	Unmanned Aerial Vehicle
NFV	Network Function Virtualization
SDN	Software-defined Networking
CDN	Content Delivery Network
NAT	Network Address Translation
VMs	Virtual Machines
COTS	Commercial Off-the-shelf
VNF	Virtual Network Function
KPI	Key Parameter Index
QoS	Quality of Service
DRL	Deep Reinforcement Learning
DNN	Deep neural Network
FL	Federated Learning
MDP	Markov Decision Process
LOS	Line of Sight
NLOS	Non Line of Sight
OBU	On-board Unit
FedAvg	Federated averaging
eMBB	enhanced Mobile Broadband
URLLC	Ultra-reliable Low Latency Communications

References

1. Shah, S.A.A.; Ahmed, E.; Imran, M.; Zeadally, S. 5G for Vehicular Communications. *IEEE Commun. Mag.* **2018**, *56*, 111–117. [[CrossRef](#)]
2. Contreras-Castillo, J.; Zeadally, S.; Guerrero-Ibañez, J.A. Internet of Vehicles: Architecture, Protocols, and Security. *IEEE Internet Things J.* **2018**, *5*, 3701–3709. [[CrossRef](#)]
3. Sharma, S.; Kaushik, B. A survey on internet of vehicles: Applications, security issues & solutions. *Veh. Commun.* **2019**, *20*, 100182:1–100182:46. [[CrossRef](#)]
4. Siddiqui, S.A.; Mahmood, A.; Sheng, Q.Z.; Suzuki, H.; Ni, W. A Survey of Trust Management in the Internet of Vehicles. *Electronics* **2021**, *10*, 2223. [[CrossRef](#)]
5. Cesarano, L.; Croce, A.; Martins, L.d.C.; Tarchi, D.; Juan, A.A. A Real-time Energy-Saving Mechanism in Internet of Vehicles Systems. *IEEE Access* **2021**, early access. [[CrossRef](#)]
6. Zhao, J.; Li, Q.; Gong, Y.; Zhang, K. Computation Offloading and Resource Allocation For Cloud Assisted Mobile Edge Computing in Vehicular Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 7944–7956. [[CrossRef](#)]
7. Wang, J.; Feng, D.; Zhang, S.; Tang, J.; Quek, T.Q.S. Computation Offloading for Mobile Edge Computing Enabled Vehicular Networks. *IEEE Access* **2019**, *7*, 62624–62632. [[CrossRef](#)]
8. Liu, L.; Chen, C.; Pei, Q.; Maharjan, S.; Zhang, Y. Vehicular edge computing and networking: A survey. *Mob. Netw. Appl.* **2021**, *26*, 1145–1168. [[CrossRef](#)]
9. Qiu, J.; Grace, D.; Ding, G.; Zakaria, M.D.; Wu, Q. Air-Ground Heterogeneous Networks for 5G and Beyond via Integrating High and Low Altitude Platforms. *IEEE Wirel. Commun.* **2019**, *26*, 140–148. [[CrossRef](#)]
10. Rinaldi, F.; Maattanen, H.L.; Torsner, J.; Pizzi, S.; Andreev, S.; Iera, A.; Koucheryavy, Y.; Araniti, G. Non-Terrestrial Networks in 5G & Beyond: A Survey. *IEEE Access* **2020**, *8*, 165178–165200. [[CrossRef](#)]
11. Xie, R.; Tang, Q.; Wang, Q.; Liu, X.; Yu, F.R.; Huang, T. Satellite-Terrestrial Integrated Edge Computing Networks: Architecture, Challenges, and Open Issues. *IEEE Netw.* **2020**, *34*, 224–231. [[CrossRef](#)]
12. Zhou, F.; Hu, R.Q.; Li, Z.; Wang, Y. Mobile Edge Computing in Unmanned Aerial Vehicle Networks. *IEEE Wirel. Commun.* **2020**, *27*, 140–146. [[CrossRef](#)]
13. Zhou, Z.; Feng, J.; Tan, L.; He, Y.; Gong, J. An Air-Ground Integration Approach for Mobile Edge Computing in IoT. *IEEE Commun. Mag.* **2018**, *56*, 40–47. [[CrossRef](#)]
14. Khattak, H.A.; Farman, H.; Jan, B.; Din, I.U. Toward Integrating Vehicular Clouds with IoT for Smart City Services. *IEEE Netw.* **2019**, *33*, 65–71. [[CrossRef](#)]
15. Ali, S.; Saad, W.; Rajatheva, N.; Chang, K.; Steinbach, D.; Sliwa, B.; Wietfeld, C.; Mei, K.; Shiri, H.; Zepernick, H.J.; et al. 6G White Paper on Machine Learning in Wireless Communication Networks. *arXiv* **2020**, arXiv:cs.IT/2004.13875.
16. Sun, Y.; Peng, M.; Zhou, Y.; Huang, Y.; Mao, S. Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3072–3108. [[CrossRef](#)]
17. Tang, F.; Kawamoto, Y.; Kato, N.; Liu, J. Future Intelligent and Secure Vehicular Network Toward 6G: Machine-Learning Approaches. *Proc. IEEE* **2020**, *108*, 292–307. [[CrossRef](#)]
18. Kato, N.; Mao, B.; Tang, F.; Kawamoto, Y.; Liu, J. Ten Challenges in Advancing Machine Learning Technologies toward 6G. *IEEE Wirel. Commun.* **2020**, *27*, 96–103. [[CrossRef](#)]
19. Du, Z.; Wu, C.; Yoshinaga, T.; Yau, K.L.A.; Ji, Y.; Li, J. Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues. *IEEE Open J. Comput. Soc.* **2020**, *1*, 45–61. [[CrossRef](#)] [[PubMed](#)]
20. Ye, D.; Yu, R.; Pan, M.; Han, Z. Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach. *IEEE Access* **2020**, *8*, 23920–23935. [[CrossRef](#)]
21. Elbir, A.M.; Soner, B.; Coleri, S. Federated Learning in Vehicular Networks. *arXiv* **2020**, arXiv:eess.SP/2006.01412.
22. Khan, L.U.; Yaqoob, I.; Tran, N.H.; Kazmi, S.M.A.; Dang, T.N.; Hong, C.S. Edge-Computing-Enabled Smart Cities: A Comprehensive Survey. *IEEE Internet Things J.* **2020**, *7*, 10200–10232. [[CrossRef](#)]
23. Ullah, Z.; Al-Turjman, F.; Mostarda, L.; Gagliardi, R. Applications of Artificial Intelligence and Machine learning in smart cities. *Comput. Commun.* **2020**, *154*, 313–323. [[CrossRef](#)]
24. Tang, F.; Mao, B.; Kato, N.; Gui, G. Comprehensive Survey on Machine Learning in Vehicular Network: Technology, Applications and Challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2027–2057. [[CrossRef](#)]
25. Cardona, N.; Coronado, E.; Latré, S.; Riggio, R.; Marquez-Barja, J.M. Software-Defined Vehicular Networking: Opportunities and Challenges. *IEEE Access* **2020**, *8*, 219971–219995. [[CrossRef](#)]
26. Wang, K.; Yin, H.; Quan, W.; Min, G. Enabling Collaborative Edge Computing for Software Defined Vehicular Networks. *IEEE Netw.* **2018**, *32*, 112–117. [[CrossRef](#)]
27. EdgeSAT—Edge Network Computing Capabilities For Satellite Remote Terminals. Available online: <https://artes.esa.int/projects/edgesat> (accessed on 11 November 2021).
28. SATis5—Demonstrator for Satellite-Terrestrial Integration in the 5G Context. Available online: <https://artes.esa.int/projects/satis5-0> (accessed on 11 November 2021).
29. Expanse—Leveraging Big Data Concepts in Future SatCom Networks. Available online: <https://artes.esa.int/projects/expanse> (accessed on 11 November 2021).
30. 3GPP—Release 17. Available online: <https://www.3gpp.org/release-17> (accessed on 12 December 2020).

31. Imadur, R.; Sara, M.R.; Olof, L.; Christian, H.; Henning, W.; Claes, T.; Paul, S.B.; Patrik, P.; Dirk, G. 5G evolution toward 5G Advanced: An overview of 3GPP releases 17 and 18. In *Ericsson Technology Review*; 2021. Available online: <https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/5g-evolution-toward-5g-advanced> (accessed on 13 October 2021).
32. Giordani, M.; Zorzi, M. Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities. *IEEE Netw.* **2021**, *35*, 244–251. [[CrossRef](#)]
33. Zhang, N.; Zhang, S.; Yang, P.; Alhussein, O.; Zhuang, W.; Shen, X.S. Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions. *IEEE Commun. Mag.* **2017**, *55*, 101–109. [[CrossRef](#)]
34. Niu, Z.; Shen, X.S.; Zhang, Q.; Tang, Y. Space-air-ground integrated vehicular network for connected and automated vehicles: Challenges and solutions. *Intell. Converg. Netw.* **2020**, *1*, 142–169. [[CrossRef](#)]
35. Ding, C.; Wang, J.B.; Zhang, H.; Lin, M.; Li, G.Y. Joint Optimization of Transmission and Computation Resources for Satellite and High Altitude Platform Assisted Edge Computing. *IEEE Trans. Wireless Commun.* **2021**, in press. [[CrossRef](#)]
36. Hu, J.; Chen, C.; Cai, L.; Khosravi, M.R.; Pei, Q.; Wan, S. UAV-Assisted Vehicular Edge Computing for the 6G Internet of Vehicles: Architecture, Intelligence, and Challenges. *IEEE Commun. Stand. Mag.* **2021**, *5*, 12–18. [[CrossRef](#)]
37. Afolabi, I.; Taleb, T.; Samdanis, K.; Ksentini, A.; Flinck, H. Network Slicing and Software-Defined Networking: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2429–2453. [[CrossRef](#)]
38. Yi, B.; Wang, X.; Li, K.; Das, S.; Huang, M. A comprehensive survey of Network Function Virtualization. *Comput. Netw.* **2018**, *133*, 212–262. [[CrossRef](#)]
39. Al-Heety, O.S.; Zakaria, Z.; Ismail, M.; Shakir, M.M.; Alani, S.; Alsariera, H. A Comprehensive Survey: Benefits, Services, Recent Works, Challenges, Security, and Use Cases for SDN-VANET. *IEEE Access* **2020**, *8*, 91028–91047. [[CrossRef](#)]
40. Sadio, O.; Ngom, I.; Lishou, C. Design and Prototyping of a Software Defined Vehicular Networking. *IEEE Trans. Veh. Technol.* **2020**, *69*, 842–850. [[CrossRef](#)]
41. Ge, X.; Li, Z.; Li, S. 5G Software Defined Vehicular Networks. *IEEE Commun. Mag.* **2017**, *55*, 87–93. [[CrossRef](#)]
42. Pokhrel, S.R. Software Defined Internet of Vehicles for Automation and Orchestration. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3890–3899. [[CrossRef](#)]
43. Mahmood, A.; Zhang, W.E.; Sheng, Q.Z. Software-Defined Heterogeneous Vehicular Networking: The Architectural Design and Open Challenges. *Future Internet* **2019**, *11*, 70. [[CrossRef](#)]
44. Campolo, C.; Molinaro, A.; Iera, A.; Menichella, F. 5G Network Slicing for Vehicle-to-Everything Services. *IEEE Wirel. Commun.* **2017**, *24*, 38–45. [[CrossRef](#)]
45. Mei, J.; Wang, X.; Zheng, K. Intelligent Network Slicing for V2X Services Toward 5G. *IEEE Netw.* **2019**, *33*, 196–204. [[CrossRef](#)]
46. Bozorgchenani, A.; Mashhadi, F.; Tarchi, D.; Salinas Monroy, S.A. Multi-Objective Computation Sharing in Energy and Delay Constrained Mobile Edge Computing Environments. *IEEE Trans. Mob. Comput.* **2021**, *20*, 2992–3005. [[CrossRef](#)]
47. Zhang, J.; Hu, X.; Ning, Z.; Ngai, E.C.H.; Zhou, L.; Wei, J.; Cheng, J.; Hu, B.; Leung, V.C.M. Joint Resource Allocation for Latency-Sensitive Services Over Mobile Edge Computing Networks With Caching. *IEEE Internet Things J.* **2019**, *6*, 4283–4294. [[CrossRef](#)]
48. Kovacevic, I.; Harjula, E.; Glisic, S.; Lorenzo, B.; Ylianttila, M. Cloud and Edge Computation Offloading for Latency Limited Services. *IEEE Access* **2021**, *9*, 55764–55776. [[CrossRef](#)]
49. Bozorgchenani, A.; Maghsudi, S.; Tarchi, D.; Hossain, E. Computation Offloading in Heterogeneous Vehicular Edge Networks: On-line and Off-policy Bandit Solutions. *IEEE Trans. Mob. Comput.* **2021**, in press. [[CrossRef](#)]
50. Ji, B.; Zhang, X.; Mumtaz, S.; Han, C.; Li, C.; Wen, H.; Wang, D. Survey on the Internet of Vehicles: Network Architectures and Applications. *IEEE Commun. Stand. Mag.* **2020**, *4*, 34–41. [[CrossRef](#)]
51. Khayyat, M.; Elgendy, I.A.; Muthanna, A.; Alshahrani, A.S.; Alharbi, S.; Koucheryavy, A. Advanced Deep Learning-Based Computational Offloading for Multilevel Vehicular Edge-Cloud Computing Networks. *IEEE Access* **2020**, *8*, 137052–137062. [[CrossRef](#)]
52. Zhou, H.; Xu, W.; Chen, J.; Wang, W. Evolutionary V2X Technologies Toward the Internet of Vehicles: Challenges and Opportunities. *Proc. IEEE* **2020**, *108*, 308–323. [[CrossRef](#)]
53. Martins, L.d.C.; Tarchi, D.; Juan, A.A.; Fusco, A. Agile optimization for a real-time facility location problem in Internet of Vehicles networks. *Networks* **2021**, in press. [[CrossRef](#)]
54. Lin, C.H.; Lin, Y.C.; Wu, Y.J.; Chung, W.H.; Lee, T.S. A Survey on Deep Learning-Based Vehicular Communication Applications. *J. Signal Process. Syst.* **2021**, *93*, 369–388. [[CrossRef](#)]
55. Posner, J.; Tseng, L.; Aloqaily, M.; Jararweh, Y. Federated Learning in Vehicular Networks: Opportunities and Solutions. *IEEE Netw.* **2021**, *35*, 152–159. [[CrossRef](#)]
56. Arum, S.C.; Grace, D.; Mitchell, P.D. A review of wireless communication using high-altitude platforms for extended coverage and capacity. *Comput. Commun.* **2020**, *157*, 232–256. [[CrossRef](#)]
57. ITU-R. *Methodology for Determining the Power Level for High Altitude Platform Stations Ground Terminals to Facilitate Sharing with Space Station Receivers in the Bands 47.2–47.5 GHz and 47.9–48.2 GHz*; Recommendation SF.1843; ITU-R: 2007. Available online: <https://www.itu.int/rec/R-REC-SF.1843-0-200710-I/en> (accessed on 8 December 2021).

-
58. Technical Specification Group Services and System Aspects. *Study on Enhancement of Support for Edge Computing in 5G Core Network (5GC)*; Technical Report 23.748 v17.0.0, 3GPP; 2020. Available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3622> (accessed on 8 December 2021).
 59. Shinde, S.S.; Marabissi, D.; Tarchi, D. A network operator-biased approach for multi-service network function placement in a 5G network slicing architecture. *Comput. Netw.* **2021**, *201*, 108598:1–108598:14. [[CrossRef](#)]