Journal of
Sensor and Actuator Networks

MDPI

Article

Network Sliced Distributed Learning-as-a-Service for Internet of Vehicles Applications in 6G Non-Terrestrial Network Scenarios

David Naseh [1,*], Swapnil Sadashiv Shinde [1,2] and Daniele Tarchi [1]

[1] Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi", University of Bologna, 40126 Bologna, Italy; swapnil.shinde2@unibo.it (S.S.S.); daniele.tarchi@unibo.it (D.T.)
[2] CNIT—University of Bologna Research Unit, 40126 Bologna, Italy
* Correspondence: david.naseh2@unibo.it

Abstract: In the rapidly evolving landscape of next-generation 6G systems, the integration of AI functions to orchestrate network resources and meet stringent user requirements is a key focus. Distributed Learning (DL), a promising set of techniques that shape the future of 6G communication systems, plays a pivotal role. Vehicular applications, representing various services, are likely to benefit significantly from the advances of 6G technologies, enabling dynamic management infused with inherent intelligence. However, the deployment of various DL methods in traditional vehicular settings with specific demands and resource constraints poses challenges. The emergence of distributed computing and communication resources, such as the edge-cloud continuum and integrated terrestrial and non-terrestrial networks (T/NTN), provides a solution. Efficiently harnessing these resources and simultaneously implementing diverse DL methods becomes crucial, and Network Slicing (NS) emerges as a valuable tool. This study delves into the analysis of DL methods suitable for vehicular environments alongside NS. Subsequently, we present a framework to facilitate DL-as-a-Service (DLaaS) on a distributed networking platform, empowering the proactive deployment of DL algorithms. This approach allows for the effective management of heterogeneous services with varying requirements. The proposed framework is exemplified through a detailed case study in a vehicular integrated T/NTN with diverse service demands from specific regions. Performance analysis highlights the advantages of the DLaaS approach, focusing on flexibility, performance enhancement, added intelligence, and increased user satisfaction in the considered T/NTN vehicular scenario.

Keywords: distributed learning; vehicular networks; network slicing; edge intelligence; integrated terrestrial non-terrestrial networks

Citation: Naseh, D.; Shinde, S.S.; Tarchi, D. Network Sliced Distributed Learning-as-a-Service for Internet of Vehicles Applications in 6G Non-Terrestrial Network Scenarios. *J. Sens. Actuator Netw.* **2024**, *13*, 14. https://doi.org/10.3390/jsan13010014

Academic Editors: Ayman Radwan, Maria de Fátima Domingues and Abd-Elhamid Taha

Received: 12 December 2023
Revised: 26 January 2024
Accepted: 5 February 2024
Published: 7 February 2024

1. Introduction

Recently, Machine Learning (ML) techniques, especially those belonging to the Distributed Learning (DL) class, have gained huge popularity in dynamic wireless scenarios such as the Internet of Vehicles (IoV) with their added advantages in terms of learning efficiency, reliability, and data security [1]. With this, various DL methods, such as Federated Learning (FL), Multi-Agent Learning, and Collaborative Learning, are considered in vehicular domains [2]. Additionally, various ML tools and techniques have been considered to form suitable DL methods, such as multi-agent FL, DL with model split, DL with meta-Learning, and DL with swarm learning. In this way, a rich ecosystem of DL methods with specific characteristics, performance, and demand is formed and made available to serve users [3].

From a networking point of view, several new advances have recently been introduced, especially with the innovations of 5G and B5G technologies. Different computing paradigms, such as Edge/Cloud Computing, have been introduced to implement new services and applications with better performance [4]. Technologies, such as network softwarization through Network Function Virtualization (NFV), Software Defined Networking

(SDN), and Network Slicing (NS), have revolutionized the networking process and opened the doors to a multitude of applications and services with different demands and additional flexibility [5]. Furthermore, distributed computing and communication technologies, such as the edge-to-cloud continuum [6], and joint Terrestrial and Non-Terrestrial Networks (T/NTN) [7], have gained huge popularity in terms of capacity, coverage, and reliability for serving end users.

In the realm of advanced 6G technologies for NTN and satellite–terrestrial integrated networks, the contributions are notable as follows. Ref. [8] delves into secrecy–energy-efficient hybrid beamforming, proposing robust schemes for single and multiple earth stations. The authors in [9] focused on a destructive beamforming design, introducing low-complexity schemes for known and unknown malicious reconfigurable intelligent surfaces (RIS). Ref. [10] addresses joint beamforming for hybrid satellite–terrestrial relay networks, optimizing power while ensuring user rate requirements. Lastly, the authors in [11] proposed a multilayer RIS-assisted secure integrated terrestrial–aerial network architecture, maximizing system energy efficiency and defending against attacks.

Shifting focus to broader aspects of wireless communication, some of the latest key technologies that contribute to the applications of different 6G techniques in Intelligent Transportation Systems (ITS) and IoV, and the vision for 6G-enabled smart cities are as follows. Ref. [12] emphasizes cooperation between cognitive users for better cognitive radio network performance. The authors of [13] explored the role of 3GPP in the evolution of cellular communication, highlighting the sharing of resources in the context of the Internet of Vehicles. Ref. [14] delves into interference in cognitive radio networks, discussing terminologies and mitigation techniques. Finally, Ref. [15] provides a visionary perspective on 6G-enabled Internet of Things (IoT) networks for sustainable smart cities, incorporating artificial intelligence, machine learning, and novel architectures.

These technologies, along with different ML methods, are creating a new paradigm, known as Edge Intelligence, to enable distributed near-user intelligent wireless networks in different domains [16]. However, there are still some gaps between the potential of these innovative technologies and their possible uses to create a safe, reliable, and intelligent vehicular system. When focusing on the IoV scenario, users demand an increased number of services, each tailored to specific requirements [17]. For example, autonomous driving may require huge data processing and low latency, while infotainment applications may require ultra-broadband connections. The need to cope with several applications has a large impact on the need to create an intelligence-at-the-edge environment that can adapt to varying demands and different requirements proactively. To this end, NS is a perfect tool, enabling the possibility of logically managing network resources from both the communication and processing points of view, thus the possibility of providing several services in a flexible way at the same time [18].

In this environment, we focus on the proposed framework, where different DL algorithms, each tailored to specific requirements, can be deployed, allowing the possibility of managing heterogeneous intelligent services simultaneously. Although in the landscape of advanced 6G technologies for integrated T/NTN the literature highlights advancements, a comprehensive integration of DL techniques with NS in the IoV context is underexplored. Our motivation stems from the identified gaps in recent works, leading us to propose a DL-as-a-Service (DLaaS) framework for vehicular environments within 6G NTN scenarios. The existing literature focuses on specific aspects, but a holistic framework that seamlessly integrates DL, NS, and distributed computing/communication methods for proactive delivery of intelligent services is missing.

Our main contributions include:

- **Comprehensive Analysis:** We conduct an in-depth exploration of key technological innovations and various DL methods, laying the groundwork for subsequent developments.

- **DLaaS Framework:** We introduce an innovative DLaaS framework, representing a paradigm shift that seamlessly integrates DL, NS, and distributed computing/communication methods.
- **Adaptation for IoV:** We elucidate the adaptation of the framework specifically for IoV applications, revealing its potential in shaping the future of intelligent vehicular networks.
- **Case Study and Performance Analysis:** We provide a detailed case study and performance analysis, offering empirical evidence of the efficacy and practical implications of DLaaS in real-world scenarios.

Finally, we will see that our proposed DLaaS approach offers advantages in terms of enhanced performance, flexibility, scalability, and intelligence, thus addressing the identified gaps and contributing to the evolution of intelligent vehicular communication systems.

To elaborate more on our proposal, Section 2 delves into a meticulous analysis of key technological innovations alongside various DL methods, laying the foundation for our subsequent developments. Section 3 introduce the innovative DLaaS framework. This framework represents a paradigm shift that seamlessly integrates DL, NS, and distributed computing/communication methods. Integration empowers the proactive delivery of intelligent services to users, fostering a dynamic environment that caters to diverse and evolving requirements. Furthermore, we elucidate the adaptation of this framework specifically for IoV applications, unveiling its potential in shaping the future of intelligent vehicular networks. This section also provides a nuanced examination of both the challenges and advantages associated with the DLaaS framework. The culmination of our exploration is presented in Section 4, where we present a detailed case study accompanied by a performance analysis. This case study serves as a tangible demonstration that illustrates the advantages that the DLaaS approach offers in terms of flexibility, performance enhancement, and infusion of added intelligence into vehicular communication systems. Through this detailed study, we not only contribute to the theoretical framework but also provide empirical evidence of the efficacy and practical implications of DLaaS in real-world scenarios.

2. IoV Distributed Intelligence

2.1. Edge/Cloud Computing

Cloud-based infrastructures with abundant resources to meet end-user requirements were one of the popular solutions in the early part of the last decade. However, over time, several issues occurred when considering cloud-based infrastructures to compute end-user data. Longer transmission distances and the corresponding communication costs, data security threats, and backhaul congestion were among the main issues that reduced the impact of cloud technology over time. Furthermore, with the new advanced technologies, such as 5G, new services with limited latency and high data rate requirements were enabled, further placing additional burdens on cloud facilities. However, with new technologies, the end devices also evolved to have powerful on-board computation capabilities, and with that, abundant computation power distributed over the network area was added. This gives birth to Fog/Edge computing technologies, bringing cloud computing facilities closer to end users [19]. Over the years, edge computing has achieved great success in terms of providing end users with high-quality services with limited latency [20].

Although edge computing has solved some of the cloud computing problems, its size limitations place additional restrictions on the computation, communication, and storage resources of edge facilities [21]. In recent times, as we move toward 6G, it has been seen that edge computing facilities are overwhelmed, and new solutions are required to fulfill the demands of new services. With this, different distributed networking infrastructures, such as the edge-cloud continuum and the Integrated T/NTN infrastructure, are considered and expected to play an essential role in the near future IoV scenarios [7,22]. These distributed networking infrastructures can have large sets of diverse networking nodes with on-board

computing/storage resources. Furthermore, with different communication technologies, these devices can communicate effectively with each other and with end users. Thus, a huge amount of distributed computing and communication power is available for the implementation of DL methods in such distributed networking infrastructures.

2.2. Network Slicing

Network softwarization is an important emerging trend in 5G and B5G-based networking systems aimed at creating a flexible network architecture with a reduced time to market for new services. NS is one of the major enabling technologies of the network softwarization realm allowing one to support diverse sets of services. NS has been introduced in the context of 5G, allowing mobile operators to create and customize their networks to provide optimized solutions for different market scenarios with diverse requirements [5]. Among others, Automation, Isolation, Customization, Elasticity, Programmability, End-to-End, and Hierarchical abstraction are the main principles of NS technology. NS provides dynamic resource management by enabling efficient resource sharing by considering various key performance indicators (KPIs) for each slice. NFV and SDN are two of the main technologies that enable NS over a common networking infrastructure. With NFV, network functions can be decoupled from proprietary hardware and run as software instances over virtualized environments, allowing them to overcome the lack of flexibility of traditional hardware-based network functions. On the other hand, SDN can help to create a fully softwarized wireless network by logically separating the data and control plane.

Recently, the vehicular community has shown great interest in NS technology for providing emerging services with complex structures to end users in a limited time [23]. Thus, the implementation of vehicular services over a distributed Vehicular Network (VN) through the deployment of several logical slices is gaining importance. Within this scenario, DL is a fundamental element required to support advanced IoV services. In the context of the DL ecosystem, a set of functions with different interdependencies is required to be executed. For example, in the case of centralized FL, functions such as data acquisition, data cleaning, hyperparameter settings, learning technique selection, data training, the transmission of learning data from devices to servers, data processing at the server, the averaging process performed by the server, and broadcasting of global model data from the server to devices are required to be implemented for completing one single learning iteration. Different forms of FL with advanced learning tools and technologies (i.e., FL with transfer learning) can require additional sets of functions. Several of these functions can be implemented as virtualized learning functions on network infrastructures. With the availability of distributed computing and communication infrastructures, such as the edge-cloud continuum and integrated T/NTN, along with virtualization technologies, these learning functions can be implemented at different locations based on their characteristics and requirements.

2.3. Overview of Distributed Learning Methods

In this part, we will discuss the fundamental structures and the distinctions between centralized learning and various distributed learning approaches, which are illustrated in Table 1. For more explanation, see [3,24].

Centralized Learning (CL): In the case of Centralized Learning (CL), a set of distributed wireless nodes (e.g., vehicles) needs to communicate their collected local datasets over an unreliable communication channel to the centralized entities (i.e., base stations, clouds, etc.) for the ML model training. This process often results in higher data transmission costs, training latency, data security issues, etc. In the case of resource-constrained nodes, such as the IoV scenario, these issues become more critical. For this reason, DL is preferred to latency-critical VNs to perform various learning tasks.

Table 1. Advantages, challenges, conditions and KPIs for different learning paradigms.

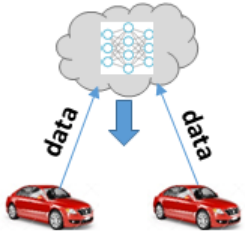
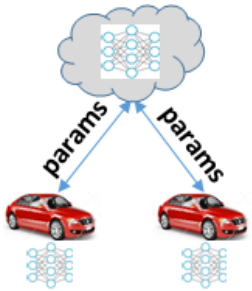
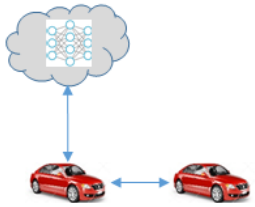
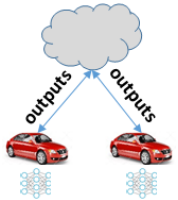
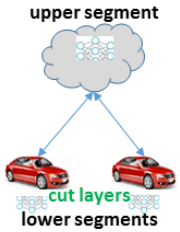
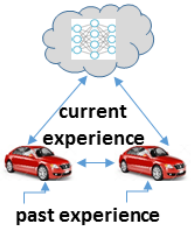
Learning Method	Advantages	Challenges	Conditions	KPIs
Centralized Learning (CL) 	• Able to train complex ML models • Better performance for non-convex applications • Less impact of communication link imperfections on training	• Sharing private data with a centralized controller • Significant transmission overhead for data collection • Difficulty of implementation for edge devices with limited resources and energy • Additional delays brought on by far-reaching transmission to a centralized server	• Devices' readiness to share their private data • Possibility of devices to transmit data • Tasks with relaxed latency requirements • ML models with complex Training Process	• Delay: High • Privacy: Low • Mobility: Low • Processing: High
Federated Learning (FL) 	• Elevated Sensitive Data Privacy • Learning a common ML model in a distributed way • Training ML models at the edge/device level	• Heterogeneous nodes with different amounts of data • Third-party attacks on parameter server (single point of failure) • Unreliable Communication Environments between devices and server • Large communication overhead proportional to the number of parameters and/or training iterations	• Reliable communication between nodes and parameter servers • Training tasks with limited complexity ML models • End devices capability to train ML model locally	• Delay: Medium • Privacy: Medium • Mobility: Low • Processing: Medium
Collaborative Federated Learning (CFL) 	• Elevated Sensitive Data Privacy • Capability to include more training data • Scalability to large-scale systems	• Impact of transmission inefficiency on training • Lower convergence rate when compared to FL • Difference in convergence and accuracy of different devices' model	• Reliable communication links between devices • Ability of devices to train/aggregate the local/received ML	• Delay: Low • Privacy: Low • Mobility: Medium • Processing: Medium
Group ADMM (GADMM) 	• Competition of only half of the devices at every communication round • Limiting communication to the two neighbouring devices • Reduced communication energy	• High communication payload • Limited scalability • Hampering the exchange of DNN parameters, especially when communication resources are limited	• Stable connections in high dynamicity environments • Limited coverage of nodes • Abundant communication and computation resources	• Delay: Low • Privacy: Low • Mobility: High • Processing: Low

Table 1. *Cont.*

Learning Method	Advantages	Challenges	Conditions	KPIs
Federated Distillation (FD) 	• Less communication burden due to only exchanging the models' outputs • Suitable for limited wireless resources • Capable of coping with heterogeneous models • Extensibility to an RL application by averaging operations across neighbouring states	Vulnerable to the problem of non-IID data distributions	• All the devices should have the same output task • Having IID data distribution	• Delay: Medium • Privacy: Medium • Mobility: Low • Processing: Medium
Split Learning (SL) 	• Fitting large-sized DNN into edge devices' small memory by dividing a single NN into multiple segments and distributing the lower segments across multiple workers • Robustness against non-IID data distributions	• Less communication efficiency because of instantaneous exchanging model updates in feedforward and backward propagation • Dependency of communication cost on the NN architecture and how to cut the layers	• Stable communication link between associated vehicular edge devices • Determining how to cut layers and NN structure based on the vehicular scenario	• Delay: Medium • Privacy: High • Mobility: Low • Processing: High
Multiagent Reinforcement Learning (MARL) 	• Capabilities of both exploration and exploitation • Agents can learn the dynamics of the environment and adapt their strategies through the experience obtained from their interactions with the environment and other agents	• Not guaranteeing the equilibrium of the constituted policies of individual agents • Requiring additional communication to guarantee the convergence	• Stable communication link between users and the learning environment • Being assured of the convergence to the equilibrium • Enough communication resources available	• Delay: High • Privacy: Low • Mobility: Medium • Processing: High
Transfer Learning (TL) 	• Improved quality and quantity of training data • Increased learning rate • Less computational demands • Less communication overhead • Preserved data privacy	• Determining source task • Specifying what to transfer • Determining the TL parameters • Choosing an appropriate number of layers for fine-tuning	• Existence of past experience available for all the devices • Availability of current/online data • Having enough memory to store past knowledge/experience	• Delay: Low • Privacy: Medium • Mobility: Low • Processing: High

In the following, the main DL methods are discussed in terms of characteristics, requirements and suitability to solve vehicular problems [25], where the main difference in DL versus CL is due to the fact that the ML model is trained at different locations and proper information exchange is performed among nodes.

Federated Learning (FL): FL is one of the most widely used DL techniques, where wireless nodes perform distributed training operations with the help of a centralized parameter server. The FL process includes two main steps. First, devices with their local datasets perform the model training operation and communicate parameter updates to a server without the need to share their sensitive raw data. In the second step, the parameter server collects and aggregates the model updates from all devices to create a global learning model that has the benefit of aggregated training experiences from all devices. This global model is then used again by the nodes in the next FL iteration, allowing them to learn from the other devices' training experiences. It is especially helpful in IoV, where FL enables collaborative model training while respecting individual data ownership and security [26]. In some situations, the traditional centralized FL is not convenient and further optimization is needed. For this reason, in the recent past, different forms of FL have been proposed, especially to optimize FL performance according to learning environments [27]. Hierarchical FL with FL process distributed over several layers of edge devices [28], Federated Distillation (FD) [29], FL with transfer learning and FL with split learning [30] are examples of updated FL-based techniques.

Collaborative Federated Learning (CFL): In reality, devices may not be able to connect to the central node due to energy constraints or possibly high transmission latency. To address this issue and make FL more accessible in real-world scenarios, the concept of CFL has been introduced, which allows vehicles to participate in FL without communicating with the central unit [31]. Devices that cannot connect directly to the central node can interact with adjacent vehicles. In this paradigm, each device can be connected to its nearest vehicle. This learning method is also trained iteratively. First, each device sends its trained local FL model to its connected devices or to the central node. The central node then produces the global FL model and sends it to the corresponding devices. Finally, each device changes its local FL model depending on the FL parameters received from other devices or the BS. In FL, each device may train its local FL model using gradient descent (GD) techniques, while the BS aggregates the local FL models. In CFL, however, each device must both aggregate the local FL models received from other devices and train its own local FL model.

With the presence of high-quality computation hardware such as multi-core Central Processing Units (CPUs), Graphical Processing Units (GPUs), and Tensor Processing Units (TPUs), vehicular nodes can themselves train the learning models without the need for parameter servers. In some cases, with reduced mobility, and through V2X technology, vehicular nodes can collaborate to solve learning tasks. Such a CFL approach can be highly efficient in terms of training. Without the presence of a third party in the learning process, this can also further strengthen vehicular data security and improve model convergence and efficiency, making it applicable to IoV scenarios with diverse and dynamic data sources [32].

Federated Distillation (FD): FD leverages outputs from models rather than the parameters in FL. Since the output dimensions are significantly smaller than the model sizes, it is much more communication efficient [29]. For example, each device in a classification task performs local iterations while saving the average model output for each class. These local average outputs, which aggregate and average the local average output among agents in each class, are sent to the central node regularly. Each device downloads the results that constitute the resultant global average. Finally, each agent runs local iterations with their loss function in addition to a regularizer that measures the difference between its prediction output of a training sample and the global average output for the given class of the sample, which is called knowledge distillation (KD), to translate the downloaded global knowledge into local models. FD can also be beneficial in IoV to improve model performance in a network of vehicles [33].

Group alternate direction method of multipliers (GADMM): The FL central node may not be able to communicate with remote edge devices. It can also be vulnerable to failure or act as a single point of attack. To this aim, GADMM intends to provide distributed learning without a central entity by using the alternate direction method of multipliers (ADMM) technique and interacting exclusively with surrounding devices by splitting the devices into head and tail groups. Only two devices from the tail/head group are selected and create a chain, with each device from the head or tail group exchanging variables. With GADMM, only half of the agents compete for the restricted bandwidth during each communication cycle. Furthermore, by restricting communication to two nearby agents, the communication energy may be greatly reduced [34]. In IoV, GADMM can be used for collaborative decision making, traffic flow optimization, or other distributed tasks, such as energy-efficient resource allocation [35].

Split Learning (SL): SL is a technique that allows resource-limited wireless devices to train complex models such as Deep Neural Networks (DNN). During the DNN training process, the model can be split vertically or horizontally, allowing multiple nodes to train a portion of the model with limited data samples and training latency. SL combined with different forms of DL can be useful in dynamic vehicular settings for producing reliable complex learning models. Moreover, since SL does not exchange raw data, data privacy is somewhat maintained [30]. This approach is particularly relevant for IoV applications where model inference can occur locally on vehicles.

Multi-Agent Reinforcement Learning (MARL): When environmental dynamics influence agents' decisions, they must learn about these dynamics and adjust their methods based on experience gained via agent-to-environment and agent-to-agent interactions. In this regard, Reinforcement Learning (RL) with exploration and exploitation abilities is critical. Exploring in RL allows agents to understand the dependencies of their decisions on the environment and other agents (policy) and on the consequences (value), which may then be used to improve long-term rewards. Even in single-agent instances, the data necessary to understand policy and value might be dispersed over several agents acting as helpers. FL, FD, and GADMM can improve learning policies and value over distributed helpers despite communication and privacy constraints. Within the MARL paradigm, the interactions of several agents in the same environment while making decisions based on local observations are investigated [36]. MARL is classified into centralized/decentralized and cooperative/competitive frameworks based on the presence of a central controller and the sorts of interactions. Centralized MARL frameworks assume a central controller that learns decision-making rules by gathering all agents' experiences, which include observed states, actions taken, and rewards received. Exchanging such information may use a significant amount of communication and memory resources, while jeopardizing data privacy. Decentralized MARL without a central controller does not have these disadvantages, but it does not ensure individual agents' convergence to equilibrium policies, even in cooperative MARL where all workers aim for the same objective. In IoV, MARL can be applied to edge caching [37], cooperative navigation [38], traffic optimization [39], and other scenarios in which vehicles interact with each other.

Transfer Learning (TL): With the involved dynamicity, resource limitations, and latency constraints, performing full-scale model training is not always feasible in vehicular environments. To this end, TL can be very useful for performing model training. In the case of TL, learning agents can utilize past learning experiences through knowledge transfer (KT) to perform new learning tasks. TL approaches can increase the convergence rate, minimize reliance on labeled data, and improve the robustness of machine learning techniques in different vehicle settings [40]. There are various forms of TL based on KT strategies. For example, the experiences gathered in terms of learning data, e.g., data features and learning data scope, can be transferred for efficient learning of target tasks. On the other hand, knowledge depending on a trained model, for example, the structure and parameters of the model, can also be shared with a target task to improve training performance [41].

The advantages, challenges, conditions, and Key Performance Indicators (KPIs) of these fundamental structures are illustrated in Table 1 in order to distinguish them in different applications. The main KPIs, which are used later in the case study section, including delay, preserved privacy, mobility of handled vehicles, and gained processing capability, are ranked in the last column of the table.

2.4. Intelligence at the Edge for IoV

2.4.1. Applications

The fusion of IoT and Artificial Intelligence (AI), known as AIoT, is transforming IoV to improve road safety, efficiency, and mitigate traffic issues [42]. The IoV landscape comprises three main categories: Autonomous Driving (AD), Safe Driving Monitoring Systems, and Cooperative Vehicle Infrastructure Systems (CVIS).

Autonomous Driving: To address the challenges of massive data generation (i.e., 4000 TB per day) and the need for real-time decision making, edge computing emerges as a viable solution [43]. AD signifies a shift toward intelligent vehicles capable of AI-driven decision making. Edge computing, exemplified by vehicles such as HydraOne [44] and HydraMini [45], addresses the challenges of massive data generation and allows real-time decision making in critical scenarios. Furthermore, edge-based systems such as EdgeDrive improve safety through real-time Advanced Driver Assistance Systems (ADAS) applications [46].

Safe Driving Monitoring Systems: Driver monitoring systems, crucial for safe driving, combat issues such as drowsiness. In [47], a Raspberry Pi 3-based system was implemented that uses a DL algorithm for real-time alerts by analyzing facial features captured in both day- and night-drive scenarios.

Cooperative Vehicle Infrastructure Systems: CVISs establish real-time road information networks by connecting vehicles, pedestrians, and infrastructure. Using distributed infrastructure, including vehicles, base stations (BSs), and roadside units (RSUs), edge computing reduces transmission delays for timely communication. In [48], the authors proposed a you only look once (YOLO) DL model for car accident detection (YOLO-CA) system that uses 5G networks detects accidents promptly, using the CAD-CVIS dataset for improved accuracy.

In conclusion, the integration of edge intelligence into IoV brings notable advances in AD, safe driving monitoring, and CVIS, addressing data challenges and fostering cooperative systems for improved road safety and traffic management. The reviewed literature emphasizes the transformative impact of AI at the edge in the IoV landscape.

2.4.2. Aspects and Advantages

Intelligence at the edge in the context of the IoV refers to the deployment of computational and analytical capabilities directly within the vehicles or at the network's edge, rather than relying solely on centralized cloud-based processing. In IoV applications, as mentioned above, the need for real-time decision-making, decreased latency, increased efficiency, and improved privacy is what motivates this strategy. Here are key aspects and advantages related to intelligence-at-the-edge in IoV:

1. **Real-time Decision-Making:** By embedding intelligence at the edge, vehicles can make local, real-time decisions without relying on a centralized cloud server. This is critical for applications such as emergency braking and collision avoidance that require quick reactions [49].
2. **Reduced Latency:** Edge computing minimizes the delay in processing the data, since computations occur closer to the source of the data. This is particularly crucial in the IoV, where accurate and timely responses to dynamic traffic conditions depend on low-latency communication [50].
3. **Bandwidth Efficiency:** Processing data at the edge reduces the need to transmit large amounts of raw data to a central server for analysis. Instead, only relevant or summarized information can be sent, optimizing bandwidth usage in IoV networks [51].

4. **Enhanced Privacy and Security:** Edge intelligence allows data processing to occur locally, addressing concerns related to privacy and security. Sensitive information can be processed within the vehicle, minimizing the exposure of personal data to external networks [52].
5. **Distributed Computing:** Edge computing in IoV involves a distributed computing paradigm where intelligence is distributed across vehicles and road infrastructure. This decentralized approach enables collaborative decision-making and more efficient utilization of resources [53].
6. **Scalability:** Edge computing supports scalability in IoV applications. As the number of connected vehicles increases, edge devices can handle processing tasks locally, preventing bottlenecks on centralized cloud servers [54].
7. **Adaptive learning:** Intelligent edge devices can employ machine learning algorithms to adapt and improve their performance based on the data they process. This adaptability is valuable in IoV scenarios where traffic patterns and conditions can change dynamically [55].
8. **Offline operation:** Edge intelligence allows vehicles to perform certain tasks even when not connected to the central network. This offline operation is beneficial in scenarios where network connectivity may be intermittent or unavailable [56].

In summary, deploying intelligence at the edge in IoV applications offers the above-mentioned advantages. This approach aligns with the dynamic and distributed nature of IoV, which contributes to more efficient and responsive connected vehicle systems.

3. Network Sliced Distributed Learning

As introduced, DL is a promising technology for designing intelligent vehicular networking systems. However, mapping different DL functions for different IoV services and requirements represents a challenge. We propose here a DLaaS concept that allows the implementation of multiple DL operations over the distributed VN through the deployment of specific learning slices, where each DL method can be seen as the composition of multiple virtual functions.

3.1. End-to-End Functional Decomposition of DL

As described in the previous section, different DL methods can be characterized by different sets of functions that must be implemented in distributed networks to have the proper benefits. Thus, each DL approach can be implemented as a set of functions coordinated through a chain characterized by functional dependencies. For this, an adequate functional decomposition providing a set of typical DL functions is needed. After having discussed the different learning techniques in distributed environments, here we propose a set of possible learning functions that are needed for implementing various DL methods:

- **Data Acquisition Function (DAF):** Generally, distributed training operations involve several learning devices collaboratively performing the learning process. In the case of vehicular scenarios, this can be geographically distributed sets of vehicles moving across road networks. For the case of DL, learning devices need their own datasets to perform the training process, which can be collected through a data acquisition mechanism that involves a set of sensory nodes, processing devices, and data collection devices. The process that allows the composition of the learning dataset can be defined through a typical data acquisition function. Note that such functions can only be implemented on nodes/devices with typical hardware settings. With new vehicular nodes equipped with several sensory nodes, they can collect large amounts of data samples over time through DAF that can be exploited for a successful implementation of DL.
- **Data Preprocessing Function (DPrF):** In general, the learning data acquired through DAF can be in different formats, e.g., texts, images, videos, etc. Based on the learning tasks, the selected learning method, and their requirements, these data need to be pre-processed in a typical form. This can be achieved through learning data prepro-

cessing methods implemented through a Data Preprocessing Function (DPrF). This function can have methods for data cleaning, data dimensionality reduction, data normalization, etc. DPrF function can help reduce the overall size of the original datasets, and thus reduce the communication overhead for some typical DL methods where data parallelization techniques involving learning data transfer are needed. Thus, preferably, this function needs to be implemented alongside the DAF function to avoid possible communication overheads.

- **Distributed Learning Function (DLF):** In DL frameworks, the learning process can be performed on different nodes, e.g., end devices or edge nodes, according to the learning frameworks adopted. The end devices can do the learning process themselves for some simplified learning tasks. In some cases, collaborative learning frameworks can be adopted for complex learning models such as DNN, allowing different devices to collaboratively train the models (e.g., through different model split techniques). In another case, a data parallelism approach can be adopted, allowing struggling end devices with limited computational resources to send their data to the nearby devices/edge nodes to complete the training process in time. Therefore, a distributed learning function (DLF) is needed that adopts the selected learning strategy for the successful implementation of DL. Typical learning steps, such as learning model selection, hyperparameter settings, stochastic gradient descent (SGD), and backpropagation, can be part of a holistic DLF that can be implemented on distributed nodes.
- **Data Post Processing function (DPsF):** In a typical DL process, after performing the learning steps through DLF, the parameter updates must be sent to the parameter server or other learning nodes based on the adopted learning strategy. Often, data processing is needed to avoid communication overhead, limit data security risks, and add the appropriate weighting coefficients to the learning process results. This method can be implemented through a Data Post Processing Function (DPsF) that processes the learned data before its transmission to the outside world.
- **Data Collection Function (DCF):** In each DL cycle, parameter servers are required to collect learning updates from the devices and create a global update that can be used for the next round of communication. The data received by the servers may have additional information, encryption, noise, etc., and are required to be processed before taking into account the global model update. The Data Collection Function (DCF) includes the steps to collect data from learning devices and prepare them for the global update to be performed.
- **Global Model Update Function (GMUF):** The Global Model Update Function (GMUF) performs the updates of the learning model based on learning data. The DCF function results are further processed with some mechanisms, i.e., the averaging process for creating a global model. These model parameters are sent back to the devices or upper layers for further processing. GMUF function can have methods for generating, preprocessing, and transmitting global model parameters over different distributed nodes.
- **Distributed Model Inference Function (DMIF):** Model inference is an important step that must be considered for the successful implementation of AI applications based on DL. End users can adopt various forms of inference mechanism for the successful implementation of DL applications based on resource availability and application requirements. These methods and processes can be included in the Distributed Model Inference Function (DMIF). Based on resource availability and application performance requirements, different model inference strategies can be adopted. If a model in question is simple and requires limited computations, inference can be performed on the device itself, increasing the data security. However, in the case where the model requires a large number of parameters with a large computation cost, inference operations can be performed at edge or cloud layers. In some cases, joint strategies (e.g., device-edge, device-edge-cloud) can also be adopted with model-split operations. This creates different possible deployment options for the DMIF function.

This set of functions can be used to implement various DL methods in the IoV scenario using the NS principle, aiming to logically deploy multiple intelligent services at the same time.

3.2. DL-as-a-Service for IoV Applications

3.2.1. Proposed Methods

The IoV scenario considered is implemented through an integrated T/NTN equipped with edge computing platforms. Different networking nodes, such as Road Side Units (RSUs), Low-Altitude Platforms (LAPs), High-Altitude Platforms (HAPs), and Satellite nodes, are distributed throughout the service area. The system is able to take advantage of different DL methods to cope with the requirements of heterogeneous users. Since the scenario considered includes a massive amount of computation, communication, and storage resources distributed over the ground, air, and space networks, these resources can be utilized to create an intelligent VN through proper deployment of required DL methods, where each network device is able to host the virtual functions enabling the different DL execution. In such a system, a slice-based approach is considered, where each slice is a logical entity that enables the interconnection of different functions to build a specific DL method.

Without loss of generality, in Figure 1, four DL methods implemented in the form of slices are represented. The first slice aims to deliver an FL service performed on different layers of edge devices. In particular, end devices have their datasets to perform the learning process. For this, the DAF, DPrF, and DLF functions are deployed on a VU layer to enable the learning process. The learned parameters are then transmitted to nearby edge nodes (i.e., RSUs) through DPsF, limiting communication costs. The RSU node then collects the data from the VUs and performs the aggregation operations, for which DCF and GMUF are placed over it. The GMUF results are then transferred to the upper layer, where appropriate functions are present. The second slice aims to deliver DL with collaborative learning frameworks. The learning part is performed collaboratively over the user devices, and appropriate learning functions are placed over the VUs cloud. The upper layers are used to create a generalized global model by aggregating the local models. The third slice is for the case of split learning, where a data parallelization technique is adapted to split the learning process over the device and edge layers. Thus, learning functions are implemented both on the device and on the edge layer. Additionally, the HAP layer is used to create a global model. For latency-critical applications, a transfer learning-based DL slice can be considered, where past learning experiences are integrated into current learning cycles to limit the learning process costs. The learning process is distributed over different edge layers.

Figure 2 shows a more detailed view of the DLaaS concept, where the virtual learning functions of a single slice are reported. A multi-layer FL is considered representative, involving data collection and learning at the VU nodes, local/intermediate model updates collections and processing, i.e., averaging at the intermediate layers of RSUs, LAPs, HAPs, satellites, and model inference operation at VUs.

3.2.2. DLaaS Advantages

The proposed DLaaS approach introduces several advantages in terms of performance enhancement, flexibility, scalability, and intelligence. With the presence of multiple DL slices, the IoV system has *better performance* in terms of latency, energy costs, and overall learning performance. DLaaS allows different DL functions to be implemented on distributed platforms according to their specific requirements, local network conditions, and resource availability. This can improve performance by allowing several users/devices to participate in the learning process efficiently. Implementing various DL methods as slices on distributed computing platforms can provide additional *flexibility* in terms of resource sharing and slice function deployments. The NS approach also allows for better *scalability*,

as it can enable a higher number of DL methods. This approach can also boost vehicle *intelligence* through the deployment of several possible DL slices simultaneously.

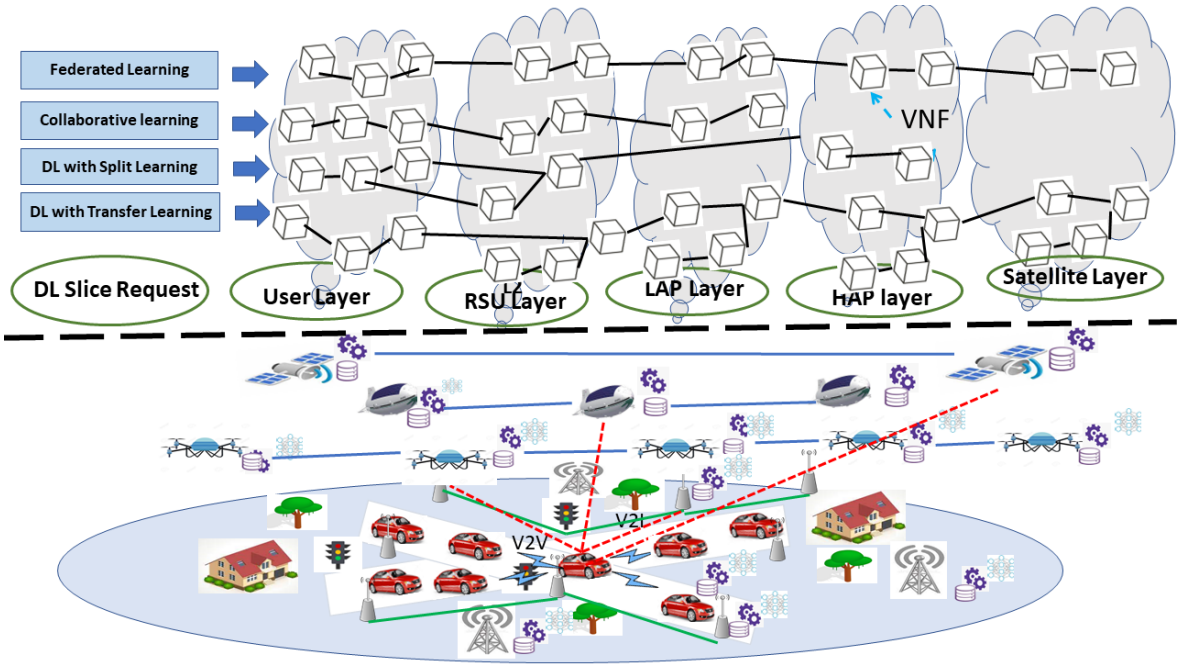


Figure 1. Distributed learning as a service over NS for IoV applications.

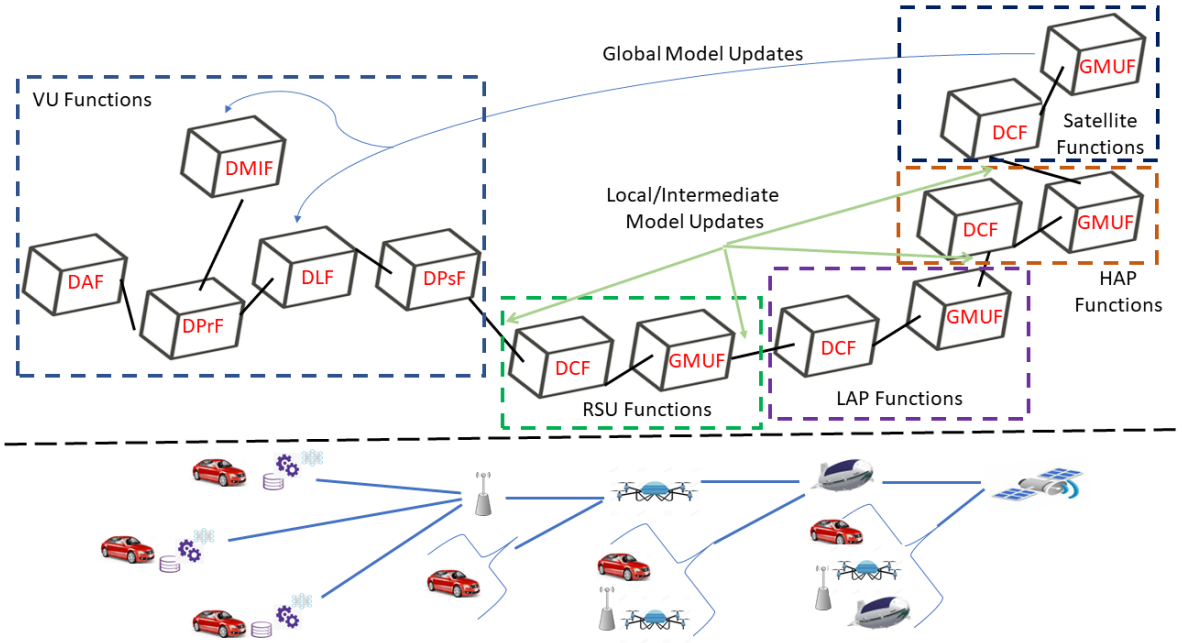


Figure 2. Multi-layer Federated Learning as a slice for IoV scenarios.

3.2.3. DLaaS Challenges

Though the proposed DLaaS method can have many advantages, several challenges must be considered while performing slice function deployments. These challenges can be based on application requirements, user-side demands, network restrictions, etc. The following key challenges should be considered in the proposed DLaaS method for proper benefits.

Learning Function placement: The DL functions can have specific requirements in terms of communication, computation, storage resources, and hardware dependencies.

When placing the functions, it is also important to take into account the functional dependencies to avoid excess costs; for example, DPrF and DAF should be placed together to avoid the communication burden. The mobility of VUs, application requirements, and limited edge resources can add additional challenges. Thus, several of these issues can make the placement of the slice functions quite challenging, and improper placement of the functions can lead to an inefficient learning environment.

Network Resource Allocations: Learning slices can be data-intensive, computation-intensive, or communication-intensive based on their higher demands for storage, computing, and communication resources, respectively. Edge nodes can have a limited set of available resources that can change over time with various demands. Additionally, each node may host several learning functions from different DL slices. Additional constraints, such as mobility, unstable communication environments, and application requirements, can make the network resource allocation problem quite challenging.

Multi-slice Implementations with Specific Demands: DL slices are characterized by different function chains, requirements, etc., and implementing them over a common infrastructure can be challenging. Each slice can have an impact on other slices' performance as a result of resource sharing. Multiple communication links enabled through a different set of slices can increase issues like noise, interference, etc.

Security Threats: Due to the presence of multiple DL slices with different user groups, the overall threat to data security can be elevated. Some slices/users can be more vulnerable to outside attacks and can end up impacting and compromising the security of other learning processes.

High-Speed Distributed Computing and Communication Environment: The challenge of resource management in a high-speed distributed computing and communication environment is crucial for the efficient functioning of the DLaaS framework. As vehicular networks operate in dynamic and high-speed environments, ensuring optimal resource allocation for fast-paced distributed computing becomes a significant challenge [57]. The need to manage resources such as computation, communication, and storage in real-time, considering the rapid movement of vehicles and the evolving nature of network conditions, adds complexity to the DLaaS implementation.

Thus, several of these challenges need to be properly addressed in order to have the additional benefits of DL as a Slice concept over a distributed vehicular networking environment.

3.3. DLaaS Architecture for IoV

Different IoV parameters, such as VUs' geographical positions, speed, edge node densities, application requirements, etc., can impact the VUs' demands for one of the available DL slices. Here, in Figure 3, we consider a case study of realistic IoV scenarios with different slice demands. A centralized orchestrator/manager considers the scenario requirements where, thanks to specific DL slice descriptors, is able to deploy the Learning Functions to the different nodes. To this aim, proper descriptions of T/NTN layers are considered, where specific communications and computing capabilities are mapped. When assigning DL slices to the different VUs, they can be considered logically organized in clusters, where each cluster is characterized by specific applications, environmental conditions, and/or vehicular characteristics.

As can be seen in Table 1, different DL algorithms can behave differently and, therefore, their selection depends on the selected IoV application, environmental conditions and characteristics of the IoV scenario, among others. For example, when using CFL, we must ensure that latency is not a critical factor, but we must also have a physical configuration that allows communication between nearby devices. To demonstrate the best approach, in Figure 3, we set up a configuration of vehicles, RSUs, LAPs, HAPs, and satellites with different physical conditions based on the three qualitative vehicle speed levels, vehicle/edge density and scenario dynamicity.

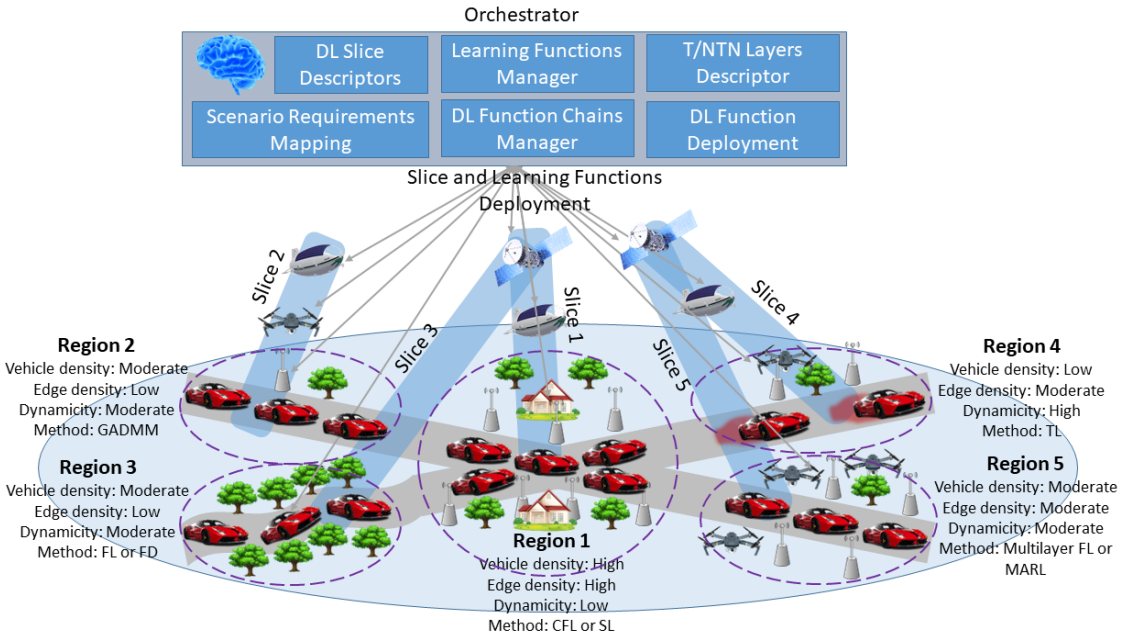


Figure 3. Slice and DL Function Deployment for regions with different IoV characteristics.

Different application requirements, i.e., latency, privacy, and reliability, can dictate the choice of the DL slice along with the local environmental parameters. The orchestrator node must consider all these parameters and requirements before assigning the DL slices to the users. This approach can improve the performance of the DL in terms of model training costs, reliability, and user satisfaction.

In Figure 3, Region 1 represents an area where a large group of VUs is moving slowly (e.g., city centers, ring roads, traffic hotspots in the city). In addition, the area is covered with a moderate number of edge nodes on the ground and in space, providing distributed computing/communication services. In such scenarios, VUs collaboration needs to be exploited for efficient DL processes. To this end, collaborative FL or SL can be the best choice.

Region 2 highlights a road scenario with typical road environments (i.e., highways) having a limited number of edge nodes, to provide server VUs with steady speed. In such cases, the density of VUs will be limited, and collaborative learning might be challenging. Furthermore, with the limited number of edge nodes, advanced learning techniques, such as hierarchical FL, might not be feasible. However, with low edge node densities, only a few VUs might be able to connect to the servers. Such scenarios can be adequate to explain the benefits of GADMM-based DL.

Region 3 lacks terrestrial/aerial edge nodes and stable connections between users due to challenging physical conditions (i.e., dense forests, and hillsides). Such cases can motivate the use of FL for privacy-preserving applications.

Region 4 shows extremely good road conditions with high-speed VUs. Frequent handovers, limited training data, and unstable channel conditions can be some of the main concerns in these regions. In such cases, advanced learning methods, such as TL, can be ideal.

Region 5 is similar to Region 2 with additional edge nodes. Such a high density of edge nodes can enable multilayer FL with hierarchical learning methods or MARL to improve latency and user data privacy. Additionally, collaborative learning between proximal edge nodes and VUs can also be available for strugglers, allowing more users to participate in DL training.

Note that, although Figure 3 shows only one case of each IoV scenario, such conditions can be replicated throughout the service areas, and thus the orchestrator needs to consider

the different groups of users simultaneously while assigning the slices. However, the scenarios are not limited to those stated and one may stumble upon a combination of the physical conditions illustrated in Figure 3. In general, we can use TL and GADMM for latency-critical tasks, SL and Hierarchical Learning for computationally intensive tasks, and FD and CFL for those that require privacy preservation. Subsequently, we can dedicate a slice to the scenario based on the conditions and applications the user demands. Note that the same physical infrastructure node (e.g., satellite) can be used for multiple logical slices at the same time.

4. Case Study

4.1. Scenario Description

The introduced DLaaS framework holds considerable potential in enhancing performance, bolstering flexibility, ensuring scalability, and fostering heightened intelligence within VNs. This study conducts a rigorous evaluation of the efficacy of the DLaaS concept in a resource-constrained Internet of Vehicles (IoV) scenario, depicted in Figure 3, within the Matlab environment. Assessment focuses on discerning performance improvements relative to conventional approaches. In this scenario, VUs are spatially distributed throughout a service area, seeking various DL slices aligned with their localized environmental conditions and application requirements. Five distinct regions are considered, each expressing a demand for a unique set of five DL slice types. As elucidated in the preceding section, the slice allocation encompasses SL for Region 1, GADMM for Region 2, FD for Region 3, TL for Region 4, and Multilayer FL for Region 5.

4.2. Function Deployment

Each requested DL slice is modeled through a chain of seven learning functions that include DAF, DPrF, DLF, DPsF, DCF, GMUF and DMIF. A multi-layered integrated T/NTN infrastructure including vehicular, RSU, UAV, HAP, and satellite layers is considered. From a functional deployment perspective, Region 5, corresponding to Slice 5, is configured with Multilayer FL, a methodology designed to execute FL across distinct layers of edge devices. Specifically, end devices use their data sets to perform the learning process. The deployment involves the placement of DAF, DPrF, and DLF on a VU layer, facilitating the learning process. Subsequently, the acquired parameters are communicated to neighboring edge nodes, represented by RSUs, via DPsF to optimize communication costs. RSUs collect data from VUs and perform aggregation operations with the assistance of DCF and GMUF. The GMUF results are then transmitted to the upper layer, where the relevant functions reside. The deployment of the function for the third slice utilizing FD mirrors the FL process described. For Slice 1, employing SL, a data parallelization technique divides the learning process between the device and edge layers, implementing learning functions on both levels. In addition, the HAP layer is employed to formulate a global model. In the case of Slice 2, which employs GADMM, collaborative learning occurs across user devices, with appropriate functions located on the VUs cloud. The upper layers contribute to the creation of a generalized global model by aggregating local models. Slice 4, which caters to latency-critical applications, adopts a Transfer Learning-based DL approach, integrating past learning experiences into current cycles to mitigate learning process costs. The learning process is distributed across various edge layers.

4.3. Network Resources and Simulation Parameters

Each layer has limited computational, storage, and communication resources available to implement the DL slices, which are indicated in Table 2. These values reflect the current and anticipated capabilities of NTN devices. For example, VUs are typically equipped with mobile processors that can provide up to 10,000 FLOPS. RSUs tend to be more powerful than VUs, boasting computational capabilities of up to 30,000 FLOPS. LAPs and HAPs are typically equipped with specialized networking hardware that can provide up to

50,000 FLOPS. Satellites, on the other hand, are constrained by available power and cooling, and their computational capacities typically range from 70,000 FLOPS to 100,000 FLOPS.

Table 2. Network Resource Allocation Across Layers.

Integrated T/NTN Layer	Computation Resources (FLOPS)	Communication Resources (Mbps)	Storage Resources (GB)
VU	10,000	20	10
RSU	30,000	40	30
LAP	30,000	30	10
HAP	50,000	50	50
Satellite	70,000	90	100

The communication bandwidth values represent the current capabilities of Long-Term Evolution (LTE) and 5G and beyond cellular networks. VUs are typically connected to 5G networks with bandwidths of up to 20 Mbps. RSUs can connect to higher speed networks with bandwidths of up to 40 Mbps. LAPs and HAPs can also connect to higher-speed networks, with bandwidths of up to 50 Mbps. Satellites, however, are limited by the available bandwidth of the satellite link, and their bandwidth typically ranges from 90 Mbps to 100 Mbps.

The storage resource values represent the current capabilities of flash memory and solid-state drives. VUs typically have storage capacities of up to 10 GB. RSUs can have storage capacities of up to 30 GB. LAPs and HAPs can have storage capacities of up to 50 GB. Satellites, on the other hand, are constrained by the available storage space in the satellite, and their storage capacities typically range from 100 GB to 200 GB.

These values were employed as a starting point for our simulations and analyses. Of course, the specific values that one utilizes will depend on the specific applications and scenarios that they are considering. Finally, the LAP, HAP, and LEO nodes are located at distances of 1.2, 20, and 1000 km from the Earth's surface, respectively. Shannon's channel capacity formula is adopted to model the channel between different layers [58].

4.4. Key Performance Indicators

In the evaluation of our IoV scenario, we quantified user satisfaction across various Key Performance Indicators (KPIs)—Latency, Privacy, Mobility, and Computing Capacity. The Matlab simulation environment facilitated a comprehensive analysis of the performance of the DLaaS framework under various conditions. The definitions and formulas of the KPIs we use are as follows:

- **Latency:**
  - **Definition:** Latency measures the delay experienced by Vehicular Users (VUs) in obtaining responses from the DLaaS framework.
  - **Simulation:** The latency satisfaction, denoted as S_{Latency}, is calculated as the percentage of users for whom the latency requirements are met:

$$S_{\text{Latency}} = \frac{\text{Number of users meeting latency requirements}}{\text{Total number of users}} \times 100\%$$

- **Privacy:**
  - **Definition:** Privacy represents the level of data security and confidentiality maintained during DL processes.
  - **Simulation:** Privacy satisfaction, denoted as S_{Privacy}, is calculated similarly based on the percentage of users for whom privacy requirements are met.

$$S_{\text{Privacy}} = \frac{\text{Number of users meeting privacy requirements}}{\text{Total number of users}} \times 100\%$$

- **Mobility:**
  - **Definition:** Mobility reflects the ability of VUs to maintain seamless connectivity while traversing diverse geographical regions.
  - **Simulation:** Mobility satisfaction, denoted as S_{Mobility}, is calculated based on the percentage of users meeting mobility requirements.

$$S_{\text{Mobility}} = \frac{\text{Number of users meeting mobility requirements}}{\text{Total number of users}} \times 100\%$$

- **Computing Capacity:**
  - **Definition:** Computing capacity denotes the ability of the DLaaS framework to handle computational demands efficiently.
  - **Simulation:** Computing capacity satisfaction, denoted as $S_{\text{Computing}}$, is calculated similarly based on the percentage of users meeting computing capacity requirements.

$$S_{\text{Computing}} = \frac{\text{Number of users meeting computing capacity requirements}}{\text{Total number of users}} \times 100\%$$

4.5. Impact of Multiple Slices on User Satisfaction

In Figure 4, user satisfaction with respect to Latency, Privacy, Mobility, and Computing Capacity is illustrated. Different types of DL slices were used, each tailored to specific computational and communication characteristics, as detailed in Table 1. The results demonstrate that with five slices, all considered KPIs achieve high satisfaction levels for all users. Conversely, with only one slice, satisfaction levels drop significantly, underscoring the effectiveness of the proposed multi-slice DLaaS framework.

4.5.1. Mobility

In Figure 5, we observe the impact of varying the number of slices on user satisfaction with respect to mobility for different numbers of VUs. The results highlight a notable trend. As the number of slices increases, there is an evident improvement in mobility satisfaction. This improvement is attributed to the ability to cover larger areas of the ground efficiently. With multiple slices, the service infrastructure can span wider geographical regions, allowing VUs to traverse expansive areas without experiencing loss of connectivity to the server or higher-layer entities, such as UAVs.

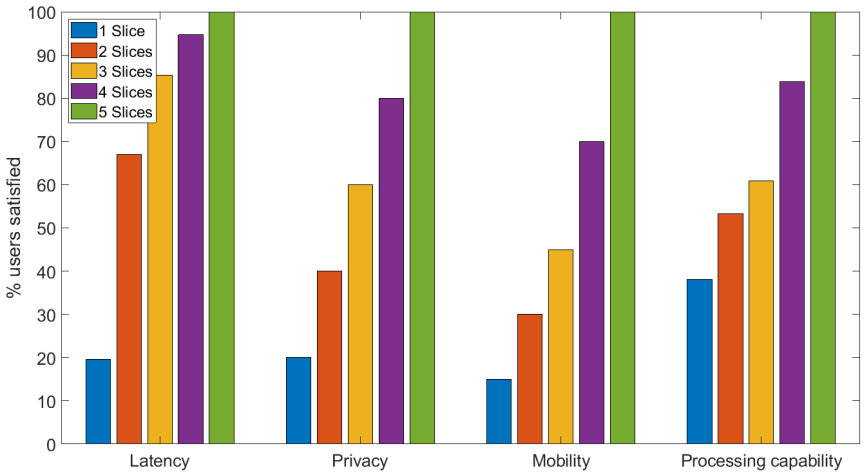


Figure 4. User satisfaction in different KPIs vs the number of slices.

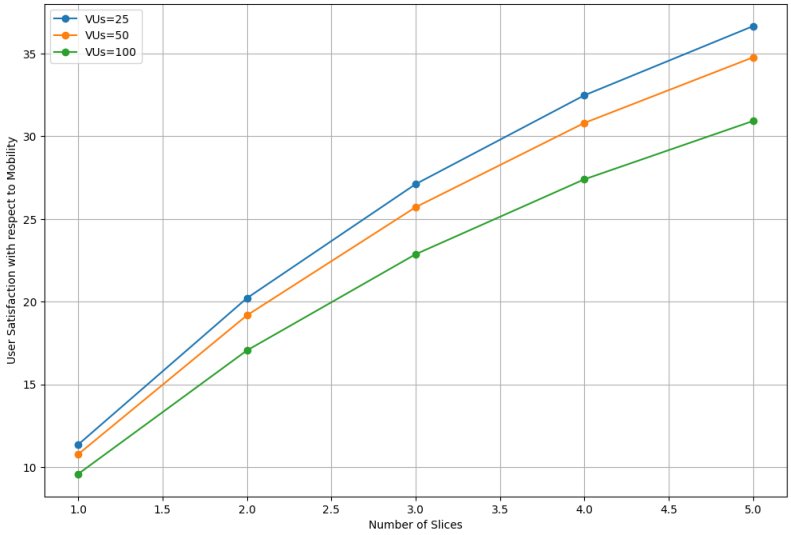


Figure 5. User satisfaction with respect to mobility versus the number of slices for different number of VUs.

In particular, when only one slice is utilized, the coverage of the entire area is based on a single slice and the FL method. This limitation presents challenges in efficiently serving diverse VU mobility patterns. However, as the number of slices increases, the system gains the ability to take advantage of different layers of NTN, leading to a significant increase in user satisfaction. This is particularly advantageous as the number of VUs increases, demonstrating the adaptability of the proposed framework to scale and accommodate growing demands without compromising user satisfaction in terms of mobility.

4.5.2. Processing Capacity

The insights derived from Figure 6 accentuate the positive impact of employing multiple slices on user satisfaction with the average processing capacity, especially in the presence of varying numbers of VUs. The figure manifests a clear trend in which the use of the slices contributes to a substantial improvement in the overall satisfaction of the VUs with respect to the processing capacity.

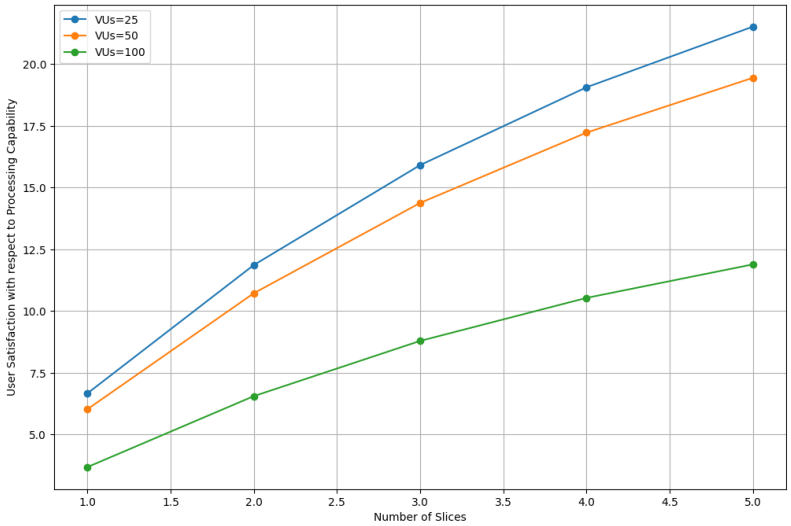


Figure 6. User satisfaction with respect to average processing capability versus the number of slices for different number of VUs.

As expected, the provision of more resources through the implementation of slices in each layer empowers the system to quickly meet the increasing demands of a growing VU population. This resilience is particularly noteworthy, as the figure demonstrates sustained improvements in processing capability satisfaction even with a higher influx of VUs. The results reiterate the superiority of the proposed DLaaS structure in efficiently catering to the computational needs of a dynamic and expanding user base.

4.5.3. Average Latency

Figure 7 delves into the realm of latency satisfaction, shedding light on the influence of the number of slices on the user experience, depending on the varying number of VUs. The results portray enhanced performance in terms of latency satisfaction as the number of slices increases. This improvement can be attributed to the increased bandwidth available for communication, resulting in reduced communication and transmission delays. Additionally, the increased resources in each slice and layer contribute to serving VU demands with reduced computation delay.

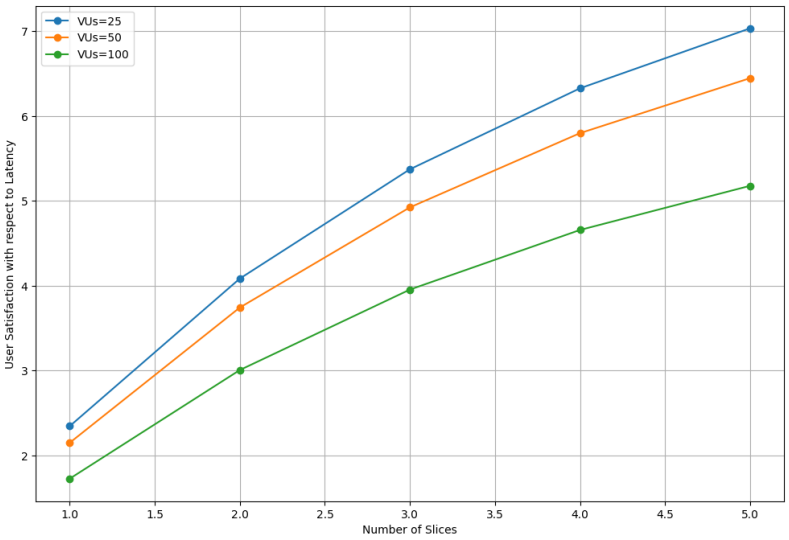


Figure 7. User satisfaction with respect to average latency versus the number of slices for different number of VUs.

The observed decrease in total latency, resulting from optimized communication and computation resources, underscores the efficacy of the proposed DLaaS structure. Interestingly, the figure reveals that even with a doubling of the number of VUs, the latency KPI does not undergo a proportional decrease. This phenomenon signifies the strategic utilization of additional resources made available through multiple slices, showcasing the system's resilience to scalability challenges.

4.6. Average Response Time

On the other hand, we would like to show how slicing can be beneficial in reducing the time required to respond to the varying tasks demanded by users. To achieve this, we assume that the probability density function (PDF) of the requested tasks is Poisson distributed with a parameter λ, which represents the average frequency of requests for each VU. The expected time value required for the slices to reconfigure to a new state for the requested services is reported, showing how much time is needed between consecutive requests to deploy a slice. Figure 8 shows this variable versus the number of slices for five different values of λ in requests per second. This figure leads us to conclude that the more slices, the lower the reconfiguration frequency, and five slices are enough for the reconfigurability to go to zero.

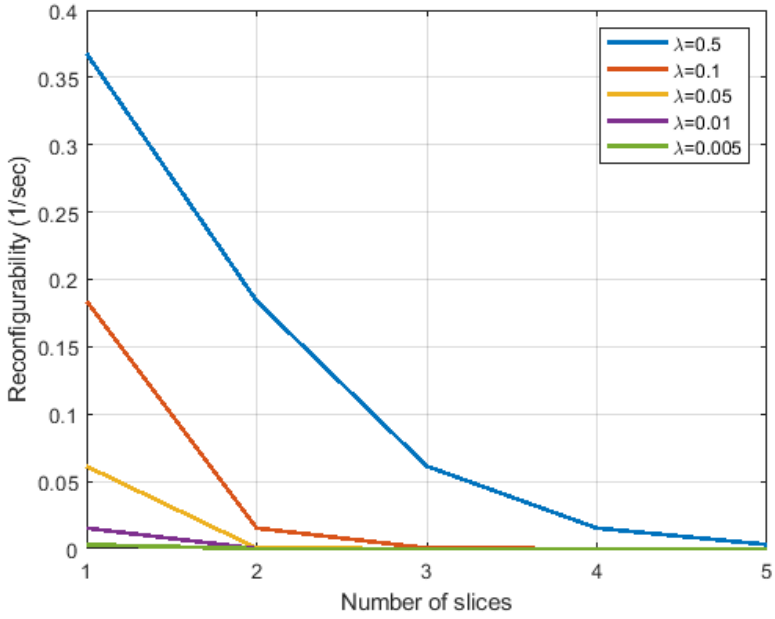


Figure 8. Reconfigurabilty vs the number of slices.

5. Conclusions

In this paper, we have proposed a DLaaS concept to implement various DL slices in distributed vehicular environments through virtualization on NTNs. We have analyzed various technologies, including edge computing, distributed computing/communication, network slicing, and different DL methods. An end-to-end functional decomposition of typical DL methods and their possible implementation as slices over the distributed networking platforms is discussed. A detailed case study of a typical vehicular scenario is considered. Through simulation, it has been shown that the proposed DLaaS concept can have several advantages in terms of user satisfaction, flexibility, scalability, performance boosts, and added intelligence. Some key challenges of the proposed DLaaS concept are also provided to motivate future research.

Author Contributions: Conceptualization, D.T.; methodology, D.N., S.S.S. and D.T.; software, D.N.; validation, D.N. and S.S.S.; formal analysis, D.N. and S.S.S.; investigation, D.N.; resources, D.T.; data curation, D.N.; writing—original draft preparation, D.N.; writing—review and editing, D.T., D.N. and S.S.S.; visualization, D.N.; supervision, D.T.; project administration, D.T.; funding acquisition, D.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was co-funded by the European Commission under the "5G-STARDUST" Project, which received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101096573 and by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001—program "RESTART"). The views expressed are those of the authors and do not necessarily represent the project. The Commission is not liable for any use that may be made of any of the information contained therein. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available on request due to restrictions e.g., privacy or ethical.

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

References

1. 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]
2. Posner, J.; Tseng, L.; Aloqaily, M.; Jararweh, Y. Federated Learning in Vehicular Networks: Opportunities and Solutions. *IEEE Netw.* **2021**, *35*, 152–159. [CrossRef]
3. Chen, M.; Gündüz, D.; Huang, K.; Saad, W.; Bennis, M.; Feljan, A.V.; Poor, H.V. Distributed Learning in Wireless Networks: Recent Progress and Future Challenges. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3579–3605. [CrossRef]
4. Duan, S.; Wang, D.; Ren, J.; Lyu, F.; Zhang, Y.; Wu, H.; Shen, X. Distributed Artificial Intelligence Empowered by End-Edge-Cloud Computing: A Survey. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 591–624. [CrossRef]
5. Afolabi, I.; Taleb, T.; Samdanis, K.; Ksentini, A.; Flinck, H. Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2429–2453. [CrossRef]
6. Cheng, Z.; Gao, Z.; Liwang, M.; Huang, L.; Du, X.; Guizani, M. Intelligent Task Offloading and Energy Allocation in the UAV-Aided Mobile Edge-Cloud Continuum. *IEEE Netw.* **2021**, *35*, 42–49. [CrossRef]
7. Shang, B.; Yi, Y.; Liu, L. Computing over Space-Air-Ground Integrated Networks: Challenges and Opportunities. *IEEE Netw.* **2021**, *35*, 302–309. [CrossRef]
8. Lin, Z.; Lin, M.; Champagne, B.; Zhu, W.P.; Al-Dhahir, N. Secrecy-Energy Efficient Hybrid Beamforming for Satellite-Terrestrial Integrated Networks. *IEEE Trans. Commun.* **2021**, *69*, 6345–6360. [CrossRef]
9. Lin, Z.; Niu, H.; An, K.; Hu, Y.; Li, D.; Wang, J.; Al-Dhahir, N. Pain without Gain: Destructive Beamforming from a Malicious RIS Perspective in IoT Networks. *IEEE Internet Things J.* 2023, *early access*. [CrossRef]
10. Lin, Z.; Niu, H.; An, K.; Wang, Y.; Zheng, G.; Chatzinotas, S.; Hu, Y. Refracting RIS-Aided Hybrid Satellite-Terrestrial Relay Networks: Joint Beamforming Design and Optimization. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3717–3724. [CrossRef]
11. Sun, Y.; An, K.; Zhu, Y.; Zheng, G.; Wong, K.K.; Chatzinotas, S.; Ng, D.W.K.; Guan, D. Energy-Efficient Hybrid Beamforming for Multilayer RIS-Assisted Secure Integrated Terrestrial-Aerial Networks. *IEEE Trans. Commun.* **2022**, *70*, 4189–4210. [CrossRef]
12. Thakur, P.; Singh, G. Cooperative Spectrum Monitoring in Homogeneous and Heterogeneous Cognitive Radio Networks. In *Spectrum Sharing in Cognitive Radio Networks: Towards Highly Connected Environments*; Wiley Telecom: Hoboken, NJ, USA, 2021; pp. 121–146. [CrossRef]
13. Thakur, P.; Singh, G. Radio Resource Management in Internet-of-Vehicles. In *Spectrum Sharing in Cognitive Radio Networks: Towards Highly Connected Environments*; Wiley Telecom: Hoboken, NJ, USA, 2021; pp. 311–338. [CrossRef]
14. Thakur, P.; Singh, G. Interference Management in Cognitive Radio Networks. In *Spectrum Sharing in Cognitive Radio Networks: Towards Highly Connected Environments*; Wiley Telecom: Hoboken, NJ, USA, 2021; pp. 255–279. [CrossRef]
15. Mishra, P.; Singh, G. 6G-IoT Framework for Sustainable Smart City: Vision and Challenges. In *Sustainable Smart Cities: Enabling Technologies, Energy Trends and Potential Applications*; Springer International Publishing: Cham, Switzerland, 2023; pp. 97–117. [CrossRef]
16. Deng, S.; Zhao, H.; Fang, W.; Yin, J.; Dustdar, S.; Zomaya, A.Y. Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence. *IEEE Internet Things J.* **2020**, *7*, 7457–7469. [CrossRef]
17. Manias, D.M.; Chouman, A.; Al-Dulaimi, A.; Shami, A. Slice-Level Performance Metric Forecasting in Intelligent Transportation Systems and the Internet of Vehicles. *IEEE Internet Things Mag.* **2023**, *6*, 56–61. [CrossRef]
18. Wu, Y.; Dai, H.N.; Wang, H.; Xiong, Z.; Guo, S. A Survey of Intelligent Network Slicing Management for Industrial IoT: Integrated Approaches for Smart Transportation, Smart Energy, and Smart Factory. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1175–1211. [CrossRef]
19. Laroui, M.; Nour, B.; Moungla, H.; Cherif, M.A.; Afifi, H.; Guizani, M. Edge and fog computing for IoT: A survey on current research activities & future directions. *Comput. Commun.* **2021**, *180*, 210–231. [CrossRef]
20. Elbamby, M.S.; Perfecto, C.; Liu, C.F.; Park, J.; Samarakoon, S.; Chen, X.; Bennis, M. Wireless Edge Computing with Latency and Reliability Guarantees. *Proc. IEEE* **2019**, *107*, 1717–1737. [CrossRef]
21. Deng, Y.; Chen, X.; Zhu, G.; Fang, Y.; Chen, Z.; Deng, X. Actions at the Edge: Jointly Optimizing the Resources in Multi-Access Edge Computing. *IEEE Wirel. Commun.* **2022**, *29*, 192–198. [CrossRef]
22. Zeng, D.; Ansari, N.; Montpetit, M.J.; Schooler, E.M.; Tarchi, D. Guest Editorial: In-Network Computing: Emerging Trends for the Edge-Cloud Continuum. *IEEE Netw.* **2021**, *35*, 12–13. [CrossRef]
23. Mei, J.; Wang, X.; Zheng, K. Intelligent Network Slicing for V2X Services Toward 5G. *IEEE Netw.* **2019**, *33*, 196–204. [CrossRef]
24. Muscinelli, E.; Shinde, S.S.; Tarchi, D. Overview of Distributed Machine Learning Techniques for 6G Networks. *Algorithms* **2022**, *15*, 210. [CrossRef]
25. Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; Rellermeyer, J.S. A Survey on Distributed Machine Learning. *ACM Comput. Surv.* **2020**, *53*, 30. [CrossRef]
26. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Blockchain Empowered Asynchronous Federated Learning for Secure Data Sharing in Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 4298–4311. [CrossRef]
27. Ridolfi, L.; Naseh, D.; Shinde, S.S.; Tarchi, D. Implementation and Evaluation of a Federated Learning Framework on Raspberry PI Platforms for IoT 6G Applications. *Future Internet* **2023**, *15*, 358. [CrossRef]
28. Cui, Y.; Cao, K.; Zhou, J.; Wei, T. Optimizing Training Efficiency and Cost of Hierarchical Federated Learning in Heterogeneous Mobile-Edge Cloud Computing. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2023**, *42*, 1518–1531. [CrossRef]

29. Zhang, Y.; Zhang, W.; Pu, L.; Lin, T.; Yan, J. To Distill or Not To Distill: Towards Fast, Accurate and Communication Efficient Federated Distillation Learning. *IEEE Internet Things J.* 2023, *early access*. [CrossRef]
30. Naseh, D.; Shinde, S.S.; Tarchi, D. Enabling Intelligent Vehicular Networks Through Distributed Learning in the Non-Terrestrial Networks 6G Vision. In Proceedings of the 28th European Wireless Conference (EW2023), Rome, Italy, 2–4 October 2023; pp. 129–134. To be published. [CrossRef]
31. Chen, M.; Poor, H.V.; Saad, W.; Cui, S. Wireless Communications for Collaborative Federated Learning. *IEEE Commun. Mag.* **2020**, *58*, 48–54. [CrossRef]
32. Shao, C.; Cheng, F.; Xiao, J.; Zhang, K. Vehicular intelligent collaborative intersection driving decision algorithm in Internet of Vehicles. *Future Gener. Comput. Syst.* **2023**, *145*, 384–395. [CrossRef]
33. Song, R.; Liu, D.; Chen, D.Z.; Festag, A.; Trinitis, C.; Schulz, M.; Knoll, A. Federated Learning via Decentralized Dataset Distillation in Resource-Constrained Edge Environments. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–10. [CrossRef]
34. Elgabli, A.; Park, J.; Bedi, A.S.; Issaid, C.B.; Bennis, M.; Aggarwal, V. Q-GADMM: Quantized Group ADMM for Communication Efficient Decentralized Machine Learning. *IEEE Trans. Commun.* **2021**, *69*, 164–181. [CrossRef]
35. Han, S.; Chen, Y.; Du, L.; Lv, J. ADMM-based Energy-Efficient Resource Allocation Method for Internet of Vehicles. In Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 8–12 October 2022; pp. 3768–3773. [CrossRef]
36. Oroojlooy, A.; Hajinezhad, D. A review of cooperative multi-agent deep reinforcement learning. *Appl. Intell.* **2023**, *53*, 13677–13722. [CrossRef]
37. Jiang, K.; Zhou, H.; Zeng, D.; Wu, J. Multi-Agent Reinforcement Learning for Cooperative Edge Caching in Internet of Vehicles. In Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Delhi, India, 10–13 December 2020; pp. 455–463. [CrossRef]
38. Zhou, W.; Chen, D.; Yan, J.; Li, Z.; Yin, H.; Ge, W. Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic. *Auton. Intell. Syst.* **2022**, *2*, 5. [CrossRef]
39. Wang, T.; Hussain, A.; Zhang, L.; Zhao, C. Collaborative Edge Computing for Social Internet of Vehicles to Alleviate Traffic Congestion. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 184–196. [CrossRef]
40. Girelli Consolaro, N.; Shinde, S.S.; Naseh, D.; Tarchi, D. Analysis and Performance Evaluation of Transfer Learning Algorithms for 6G Wireless Networks. *Electronics* **2023**, *12*, 3327. [CrossRef]
41. Wang, M.; Lin, Y.; Tian, Q.; Si, G. Transfer Learning Promotes 6G Wireless Communications: Recent Advances and Future Challenges. *IEEE Trans. Reliab.* **2021**, *70*, 790–807. [CrossRef]
42. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A Survey of Recent Advances in Edge-Computing-Powered Artificial Intelligence of Things. *IEEE Internet Things J.* **2021**, *8*, 13849–13875. [CrossRef]
43. Liu, L.; Yao, Y.; Wang, R.; Wu, B.; Shi, W. Equinox: A Road-Side Edge Computing Experimental Platform for CAVs. In Proceedings of the 2020 International Conference on Connected and Autonomous Driving (MetroCAD), Detroit, MI, USA, 27–28 February 2020; pp. 41–42. [CrossRef]
44. Wang, Y.; Liu, L.; Zhang, X.; Shi, W. HydraOne: An Indoor Experimental Research and Education Platform for CAVs. In Proceedings of the 2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19), Renton, WA, USA, 9 July 2019.
45. Wu, T.; Wang, Y.; Shi, W.; Lu, J. HydraMini: An FPGA-based Affordable Research and Education Platform for Autonomous Driving. In Proceedings of the 2020 International Conference on Connected and Autonomous Driving (MetroCAD), Detroit, MI, USA, 27–28 February 2020; pp. 45–52. [CrossRef]
46. Maheshwari, S.; Zhang, W.; Seskar, I.; Zhang, Y.; Raychaudhuri, D. EdgeDrive: Supporting Advanced Driver Assistance Systems using Mobile Edge Clouds Networks. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, 29 April–2 May 2019; pp. 1–6. [CrossRef]
47. Villanueva, A.; Benemerito, R.L.L.; Cabug-Os, M.J.M.; Chua, R.B.; Rebeca, C.K.D.; Miranda, M. Somnolence Detection System Utilizing Deep Neural Network. In Proceedings of the 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 24–25 July 2019; pp. 602–607. [CrossRef]
48. Tian, D.; Zhang, C.; Duan, X.; Wang, X. An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems. *IEEE Access* **2019**, *7*, 127453–127463. [CrossRef]
49. Fu, Y.; Li, C.; Yu, F.R.; Luan, T.H.; Zhang, Y. A Survey of Driving Safety with Sensing, Vehicular Communications, and Artificial Intelligence-Based Collision Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6142–6163. [CrossRef]
50. Tufail, A.; Namoun, A.; Abi Sen, A.A.; Kim, K.H.; Alrehaili, A.; Ali, A. Moisture Computing-Based Internet of Vehicles (IoV) Architecture for Smart Cities. *Sensors* **2021**, *21*, 3785. [CrossRef] [PubMed]
51. Wang, J.; Zhu, K.; Hossain, E. Green Internet of Vehicles (IoV) in the 6G Era: Toward Sustainable Vehicular Communications and Networking. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 391–423. [CrossRef]
52. Zhang, J.; Letaief, K.B. Mobile Edge Intelligence and Computing for the Internet of Vehicles. *Proc. IEEE* **2020**, *108*, 246–261. [CrossRef]
53. Rafique, W.; Qi, L.; Yaqoob, I.; Imran, M.; Rasool, R.U.; Dou, W. Complementing IoT Services Through Software Defined Networking and Edge Computing: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1761–1804. [CrossRef]

54. Duan, W.; Gu, J.; Wen, M.; Zhang, G.; Ji, Y.; Mumtaz, S. Emerging Technologies for 5G-IoV Networks: Applications, Trends and Opportunities. *IEEE Netw.* **2020**, *34*, 283–289. [CrossRef]
55. Balkus, S.V.; Wang, H.; Cornet, B.D.; Mahabal, C.; Ngo, H.; Fang, H. A Survey of Collaborative Machine Learning Using 5G Vehicular Communications. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1280–1303. [CrossRef]
56. Dai, P.; Song, F.; Liu, K.; Dai, Y.; Zhou, P.; Guo, S. Edge Intelligence for Adaptive Multimedia Streaming in Heterogeneous Internet of Vehicles. *IEEE Trans. Mob. Comput.* **2023**, *22*, 1464–1478. [CrossRef]
57. Khan Tayyaba, S.; Khattak, H.A.; Almogren, A.; Shah, M.A.; Ud Din, I.; Alkhalifa, I.; Guizani, M. 5G Vehicular Network Resource Management for Improving Radio Access through Machine Learning. *IEEE Access* **2020**, *8*, 6792–6800. [CrossRef]
58. Shinde, S.S.; Bozorgchenani, A.; Tarchi, D.; Ni, Q. On the Design of Federated Learning in Latency and Energy Constrained Computation Offloading Operations in Vehicular Edge Computing Systems. *IEEE Trans. Veh. Technol.* **2022**, *71*, 2041–2057. [CrossRef]