information

Article

Distributed Learning in Intelligent Transportation Systems: A Survey

Qiong Li [1,2,3], Wanlei Zhou [1] and Xi Zheng [1,4,*]

[1] Faculty of Data Science, City University of Macau, Macau 999078, China; yifeng8457@163.com (Q.L.); wlzhou@cityu.edu.mo (W.Z.)
[2] Hunan Industry Polytechnic, Changsha 410208, China
[3] Hunan Engineering Technology Research Center of Digitalization of CNC Machining Process for Precision Parts, Changsha 410208, China
[4] Intelligent Systems Research Group, School of Computing, Macquarie University, Sydney, NSW 2109, Australia
* Correspondence: james.zheng@mq.edu.au

Abstract: The development of artificial intelligence (AI) and self-driving technology is expected to enhance intelligent transportation systems (ITSs) by improving road safety and mobility, increasing traffic flow, and reducing vehicle emissions in the near future. In an ITS, each autonomous vehicle acts as a node with its own local machine learning models, which can be updated using locally collected data. However, for autonomous vehicles to learn effective models, they must be able to learn from data sources provided by other vehicles and infrastructure, utilizing innovative learning methods to adapt to various autonomous driving scenarios. Distributed learning plays a crucial role in implementing these learning tasks in an ITS. This review provides a systematic overview of distributed learning in the field of ITSs. Within an ITS, vehicles can engage in distributed learning by interacting with peers through opportunistic encounters and clustering. This study examines the challenges associated with distributed learning, focusing on issues related to privacy and security in data intelligence sharing, communication quality and speed, and trust. Through a thorough analysis of these challenges, this study presents potential research avenues to address these issues, including the utilization of incentive mechanisms that rely on reputation, the adoption of rapid convergence techniques, and the integration of opportunistic federated learning with blockchain technology.

Keywords: distributed learning; federated learning; intelligent transportation system; gossip learning

check for updates

Citation: Li, Q.; Zhou, W.; Zheng, X. Distributed Learning in Intelligent Transportation Systems: A Survey. *Information* **2024**, *15*, 550. https://doi.org/10.3390/info15090550

Academic Editor: Michele Ottomanelli

Received: 14 August 2024
Revised: 28 August 2024
Accepted: 6 September 2024
Published: 8 September 2024

1. Introduction

1.1. Motivation

As urbanization deepens and travel demand rapidly expands in various countries, problems such as road congestion, reduced vehicle speed, overcrowding, and frequent traffic accidents have also begun to rapidly increase [1,2]. Intelligent transportation systems (ITSs) integrate cutting-edge technologies, such as electronic sensor technologies, data transmission technologies, and intelligent control technologies, into transportation systems. The objective of an ITS is to improve the quality of services offered to drivers and passengers in transportation systems [3]. These systems are designed to provide a more efficient and effective way of managing transportation in urban areas. Using artificial intelligence, such a system analyzes data and makes decisions that can improve the efficiency of transportation networks. In addition, it can be used to monitor traffic conditions and provide real-time information to drivers [4]. In addition, resource utilization can be improved and the environmental effects of transportation can be minimized through the use of an ITS. These systems are transforming how cities and roads are managed and may be a key part of the smart cities of the future. ITSs have the potential to make traveling more efficient, safer, and comfortable for people all over the world [5]. Intelligent and connected ITSs have been shown to perform well in alleviating traffic congestion and reducing traffic accidents [6,7].

At the same time, the progress of machine learning (ML) due to the emergence of various technologies—particularly cloud and edge computing—has been remarkable. ML techniques can be used for accurate prediction and decision-making [8–10], and are also applicable in the context of vehicle network security. Statistics from scientific publications between 2010 and 2020 show that the number of publications utilizing machine learning approaches to enable and optimize ITS tasks has increased twenty-fold during this period [11]. However, enhancing the accuracy of predictions and making automatic driving models suitable for advanced ITSs requires a substantial amount of training data. As the number of parameters grows, the demand for training data escalates exponentially [12]. Verbraeken, J. et al. [13] pointed out that due to the limitations of current computing devices in processing extensive training data, it is crucial to distribute the learning tasks across multiple units and transition from centralized to distributed learning systems. Investigating and adjusting distributed learning to suit the particular qualities and needs of ITS applications remains a challenging task, offering promising research opportunities. Distributed learning allows mobile devices to protect their privacy by uploading only a limited amount of information to computational access points. Verbraeken, J. [13] and Peteiro-Barral, D. [14] discussed the advantages and disadvantages of distributed learning in comparison with centralized learning, in which privacy and security concerns were thoroughly examined. Vepakomma, P. et al. [15] investigated the privacy implications of distributed learning. This type of learning can be beneficial in reducing the considerable communication overhead associated with centralized learning. Cao, H. et al. [16] developed an algorithm for distributed computing, allowing for offloading to reduce the overall cost of system execution, while Lin, R. et al. used distributed learning to minimize the long-term average latency, which is restricted by computing resources and power consumption [17].

The combination of ITS and ML technologies has led to significant advancements, with ML facilitating precise predictions and decision-making, bolstering the security of vehicle networks. The integration of ML into ITS functions has notably risen in the last ten years, underscoring the necessity for substantial training data and the shift toward distributed learning systems to manage the escalating requirements. Distributed learning not only tackles privacy issues but also lessens communication overhead and system expenses. Explorations in this field present encouraging prospects for refining ITS applications and ensuring the confidentiality and protection of data. The motivation for surveying distributed learning in ITSs includes enhancing traffic supervision and control systems [18], enhancing the efficiency and dependability of transportation networks [19], streamlining real-time decision-making processes [20], and optimizing the allocation and utilization of resources in transportation systems [21].

1.2. Related Work

To the best of our knowledge, this is the first survey to investigate distributed learning in the context of ITSs. While relevant studies have been conducted, they have mainly focused on autonomous vehicle techniques and their implementation in certain scenarios, such as autonomous vehicle coordination [22], Level 5 autonomous driving [23], or traffic efficiency applications for intelligent connected vehicles [24], rather than distributed learning in an ITS.

Recent years have seen a significant increase in ITSs based on autonomous self-driving vehicle technology [22–24]. In fact, the application of intelligent autonomous vehicles often relies on two aspects of technology: intelligence and networking [25]. Many automotive companies have conducted tests while researching integrated sensors for intelligent vehicles [26–28].

Networking is a key element in the development of unmanned driving for ITSs. For example, J. Pereira et al. conducted an interoperability test for connected and autonomous vehicles (CAVs) incorporated into a connected ITS [29]. In a different investigation [30–33], the authors explored a variety of issues related to the coordination of activities between two or more autonomous vehicles and an ITS. D.B. Abeywickrama et al. [24] examined several technology-based approaches to a vehicular network that can be used to improve traffic flow in urban areas. At present, the development of self-driving vehicles is highly dependent on the overall structure of the ITS. To achieve autonomous Level 5 driving, it is necessary not only to strengthen research on the hardware and software of autonomous vehicles, but also to advance the development of its vehicle-to-infrastructure (V2I) and vehicle-to-everything (V2X) technologies—that is, technologies that allow for information to be exchanged between the vehicle and the outside world [34,35].

This section differentiates our survey from previous studies in the literature by presenting related research that has been conducted by others and highlighting the associated shortcomings, as summarized in Table 1. Research in this field is still in its early stages, so a systematic review of the literature (SLR) [36] was carried out to conduct a taxonomic survey to fill this research gap. This study not only offers a thorough examination of what lies ahead but also provides insights into the recognition of the research topics/tendencies in this area.

Table 1. Comparison of review articles.

	Reference	Contributions	Limitations
Traffic efficiency applications	Younes et al. [24]	Improving traffic flow. Predicting traffic conditions.	Challenges in vehicle network. Various requirements and factors.
	Liu et al. [37]	Based on the real world. Collaboration with V2I.	No introduction to collaboration. No introduction to autonomous driving.
Autonomous driving	Khan et al. [23]	Advantages of communication and distributed learning.	No exploration of distributed learning. Not delving into privacy protection.
Coordination of autonomous vehicles	Mariani et al. [22]	Identify categories of coordination. Strategies for autonomous vehicles.	Focused on an autonomous vehicle. Focused on traffic efficiency. Solutions for autonomous vehicle scenarios.

1.3. Contributions

This study provides an overview of the key issues related to distributed learning in the context of ITSs. The key contributions of this study are as follows:

- The requirements for distributed learning are analyzed, and the application scenarios of distributed learning in ITSs are introduced.
- Through an investigation of various distributed learning methods used in the context of ITSs, the constraints of existing ITS distributed learning applications are recognized.
- Potential research directions that can bridge the identified gaps are identified, and some additional research challenges for the application of distributed learning in ITSs are outlined.

The remainder of this article is structured as follows. Section 1 examines topics related to distributed learning. Section 2 provides a categorization, summary, and analysis of the application scenarios of distributed learning in ITSs. Section 3 offers an overview and discussion of distributed learning in ITSs. Section 4 introduces some gaps in the existing literature and additional open research directions. Section 5 concludes the article.

1.4. Methodology

1.4.1. Research Questions

This study investigates the possibilities of distributed learning in the realm of ITSs and to answer the following questions.

RQ1: How can distributed learning improve intelligent transportation systems?

Rationale: We must identify specific ITS applications where the emphasis is placed on distributed learning.

> **RQ2:** What are the limitations to applying existing distributed learning in intelligent transportation systems?

Rationale: It is essential to analyze the advantages and disadvantages of state-of-the-art distributed learning approaches, in order to pinpoint research gaps in the field.

> **RQ3:** What are the possible research directions to bridge the identified gaps?

Rationale: We aim to highlight several potential research avenues through which the entire community can collaborate to close the existing gaps.

The reason for choosing these three research questions is mainly to determine which scenarios in the specific applications of ITSs are focused on distributed learning. Based on this, the advantages and disadvantages of distributed learning that are currently applied in ITSs are analyzed, in order to identify research gaps. Finally, the research directions for bridging these research gaps are identified, along with other potential research avenues.

1.4.2. Search and Selection Strategy for Primary Studies

We used a systematic literature review (SLR) technique, as suggested by Kitchenham et al. [36], to address our research questions. This method is objective, clear, and reproducible, and allows us to review the existing literature without bias. The review protocol is illustrated in Figure 1.

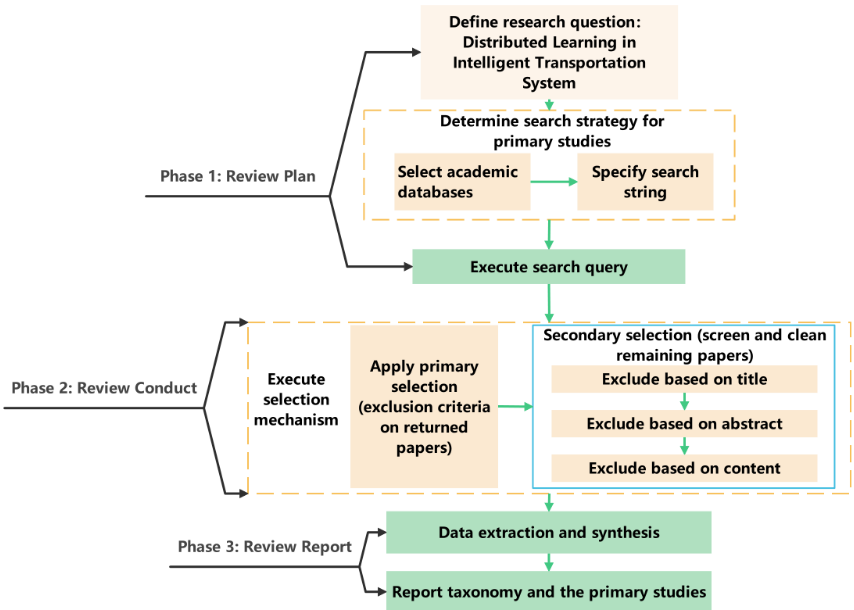

Figure 1. Flowchart for comprehensive analysis of the existing literature on distributed learning in intelligent transportation systems (ITSs), conducted through a systematic review.

This study involved three distinct phases to achieve its goal. Initially, the research questions were established, and a search strategy was determined for the primary studies. The second phase involved the application of inclusion and exclusion criteria to define the study's selection process. Finally, a thematic taxonomy of the retrieved approaches was devised, and the results were reported. To find research papers in the area of computer science, popular databases were used as the main sources [36]. These databases provide sophisticated search capabilities with a set of Boolean functions to carry out precise searches based on certain fields, such as the abstract, title, and keywords, which return more relevant results than searching all fields.

We constructed a search query for use in these databases to identify relevant publications. To ensure a complete overview of the applicable research in the literature and current studies, we needed to be careful in the selection of keywords. The keywords used are shown in Figure 2; in the figure, the larger the font size, the more studies related to that keyword were retrieved.

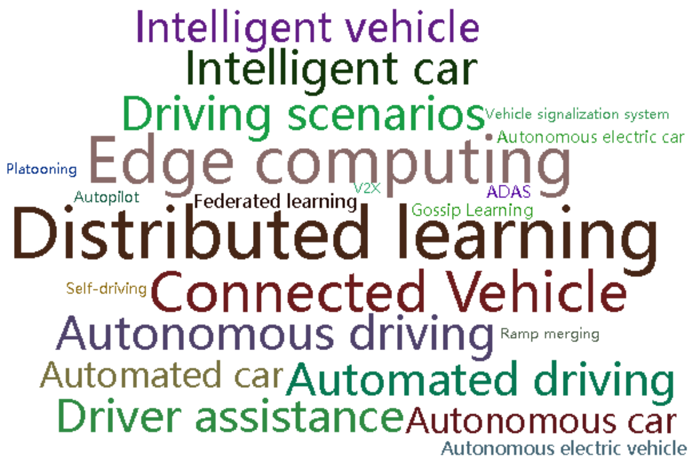

Figure 2. Word cloud diagram of keywords. In the figure, V2X denotes the vehicle-to-everything and ADAS denotes the advanced driving assistance system.

We needed to decide on the criteria for evaluating the articles that would be obtained after using the search string in the following step. Tables 2 and 3 list the inclusion and exclusion criteria used in our systematic review of the literature.

We executed a search string and applied the inclusion and exclusion criteria to relevant publications related to distributed learning, resulting in an initial sample of 188 articles. Then, we performed a stringent selection (secondary selection) of articles based on their titles, abstracts, and contents, only selecting publications that considered distributed learning in an ITS and excluding those that discussed the general application of machine learning. This was because our focus in this study was advancing the development of autonomous vehicles, rather than the technical aspects of machine learning. After this phase, we obtained 44 articles to be systematically reviewed in this study. The trend chart of the literature publication is shown in Figure 3. The distribution of published literature by country is shown in Figure 4.

Table 2. List of inclusion criteria.

Criteria	Description
Reviewed by peers.	Opted for peer-reviewed articles, including conference, workshop, and journal papers, as well as book chapters.
Recommended by important organizations.	Highly recommended by ACM, IEEE, CCF, and so on.
Conducted between 2018 and 2024.	Investigated all publications between 2018 and 2024.

Table 3. List of exclusion criteria.

Criterion	Description
The study was not on ITSs.	We only analyzed the literature related to ITSs.
The study did not go through a peer-review process.	Exclusion of those that were only available as an abstract, not published in a full paper, or not peer-reviewed.
Duplicate publication.	Eliminated duplicates that appeared in different databases.

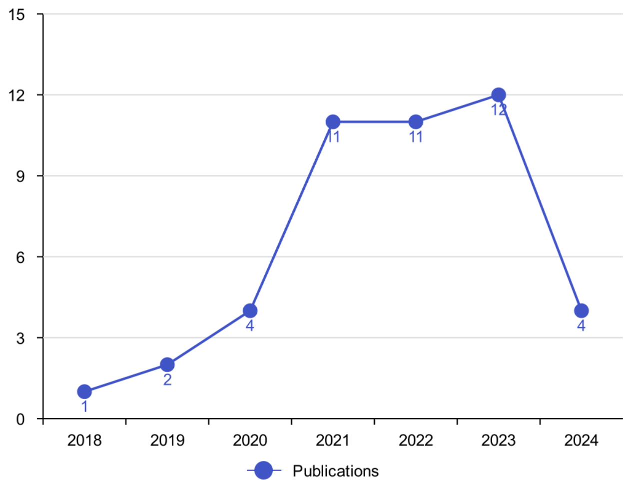

Figure 3. Trend chart of literature publications (2018–2024).

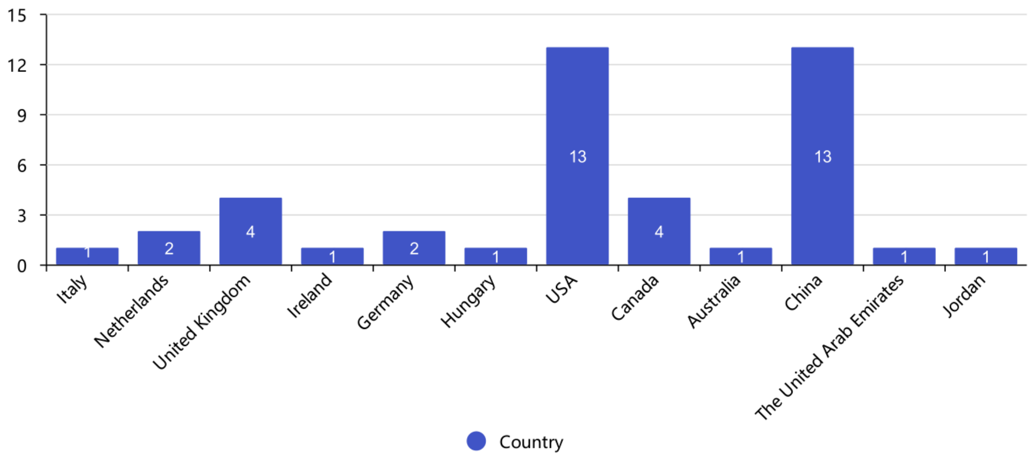

Figure 4. Country distribution of literature publications.

1.4.3. Data Extraction and Synthesis

We analyzed the primary research documents, which resulted in a thematic taxonomy of distributed learning in ITSs (Figure 5). Detailed descriptions of the components of the proposed taxonomy are provided below.

- *Scenarios:* This element examines the behavior of single cars (e.g., when they pass through an intersection) a group of vehicles (e.g., allocating parking spaces to a business's fleet), or the traffic system as a whole (e.g., regulating the flow of traffic in a city). The topic of distributed learning in ITSs is very extensive, encompassing a variety of scenarios from intersection control to car sharing.
- *Approaches:* This element examines approaches based on existing technology. Common machine learning techniques include centralized learning, federated learning, and distributed learning. Distributed learning encompasses opportunistic federated learning, edge computing, gossip learning, and other approaches. Furthermore, autonomous communication, privacy, and security must also be considered in the considered context.
- *Challenges:* In the realm of ITSs, distributed learning applications can be used to ensure that autonomous driving requirements are met in particular or more general scenarios. This may involve one or more challenges, such as privacy and security, data, communication, and trust.

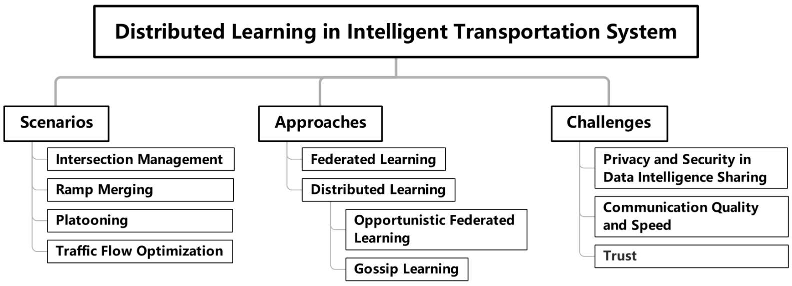

Figure 5. Distributed learning in intelligent transportation systems.

2. Scenarios (in Response to RQ1)

ITSs have a variety of uses, including vehicle networking, intelligent traffic signals, intelligent parking, intersection control, merging ramps, platooning, and optimizing traffic flow. Junyao Guo et al. [38] proposed that an autonomous driving system can adapt to any driving conditions through learning and generalizing from sample scenarios. Therefore, in this study, based on the application of distributed learning in ITSs, typical application scenarios such as intersection management, ramp merging, platooning, and traffic flow optimization were selected for discussion.

2.1. Intersection Management

Various intersections exhibit distinct features, including varying lanes, traffic lights, road conditions, and road signs. The knowledge gained by a car navigating a specific intersection can be transferred to another vehicle, significantly enhancing the efficiency of the latter's traversal through the intersection. An autonomous vehicle that has never been to an intersection can benefit from updating its autonomous driving model to one that is more suitable for the intersection. This can make it easier to manage self-driving cars, enabling the function of ITSs at crossroads.

Hence, the utilization of distributed learning is feasible for sharing experiences among autonomous vehicles, enabling them to enhance their understanding of intersection traffic through model updating. Distributed learning is deemed more appropriate for intricate intersection situations than centralized learning. Notably, in contemporary distributed learning approaches, roadside units (RSUs) are also involved in updating models. Samarakoon et al. [39] examined a network that utilized a 250 m $\times$ 250 m Manhattan mobility model containing nine intersections. The illustration in Figure 6 depicts a structure that integrates federated learning, allowing vehicles to share local and global models with RSUs through this approach. This model utilizes the extreme value theory (EVT) [40] principle of the generalized Pareto distribution (GPD) [41] to gather statistical information on queue lengths that exceed high thresholds. It also imposes local restrictions on extreme events associated with queue lengths that exceed predetermined thresholds for each vehicular user (VUE). The GPD's feature parameters (i.e., the scale and shape parameters) are determined through maximum likelihood estimation (MLE) [42]. In contrast to the conventional MLE method, which requires a central controller (e.g., an RSU) to gather samples of queue lengths surpassing a threshold from all VUEs in the network, federated learning allows each vehicle to create and exchange its own local model (comprising two gradient values) with the RSU. In Figure 6, the communication between VUEs and RSUs involves the following steps: (1) Queue sampling (as described above); (2) estimation of GPD parameters; (3) allocation of transmission power and resource blocks (RBs); and (4) swapping local and global models with RSUs. In step A, the VUE assesses the gradients and GPD parameters locally. Following the local calculation of gradients and GPD parameters, each VUE transmits its model to the RSU in step B. Subsequently, the RSU conducts model averaging across the network, concurrently computing global GPD parameters and gradients in step C. Finally, in step D, the global model is disseminated throughout the network. The results of experiments indicated that VUEs can independently grasp the tail distribution of the entire network queue without the need to exchange specific queue-length data, thus minimizing superfluous overhead. This method demonstrates substantial improvements in mitigating instances of extreme events where the queue length exceeds predetermined thresholds.

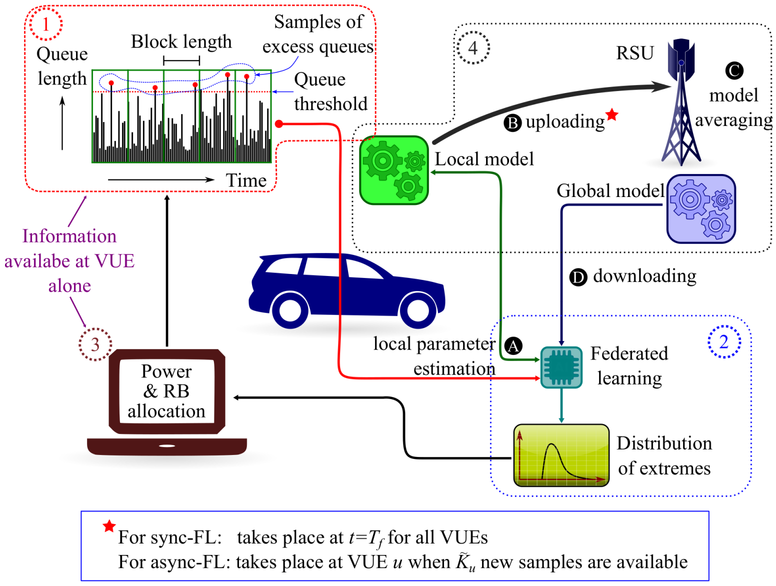

Figure 6. Swapping of the local and global models between vehicles and RSUs [39]. In the figure, RSU denotes the roadside unit, RB denotes the resource block, VUE denotes the vehicular user, and FL denotes federated learning.

2.2. Ramp Merging

Merging from a ramp into a main lane is similar to an intersection where there are no signal lights and vehicles that are moving straight have the priority to pass. On-ramp vehicles must merge into the main lane smoothly and safely to prevent collisions. Ideally, these vehicles should adjust their speed and merge quickly to avoid traffic jams [43,44]. It is clear that understanding the traffic flow and speed in the main lane, as well as the length of the ramp (in order to appropriately adjust the speed of the vehicles on the ramp), provides valuable insights for the model. Learning from this information, autonomous vehicles can quickly adapt their speed and merge efficiently, even without prior experience in ramp merging. This approach can help to prevent traffic congestion and improve the overall flow of traffic.

Each vehicle locally stores its own data and model, which can be enhanced through learning to better handle various merging situations on ramps. Figure 7 illustrates a traffic scenario involving vehicles merging from a ramp onto a highway. Considering this scenario, Rihan et al. designed a distributed learning framework centered around fog computing [45]. This framework enables the sharing of learning experiences among network nodes through the exchange of model updates. As illustrated in Figure 8, every node can store its own data locally, and a common model is created using these data. This process is depicted in Figure 8, where the local parameters μ_i and β_i are uploaded. Subsequently, the node can distribute its acquired knowledge to other nodes in the network through exchanging model updates, which include deep learning model weights and biases. This is illustrated in Figure 8, as a two-fold procedure: downloading global parameters and controlling design parameters.

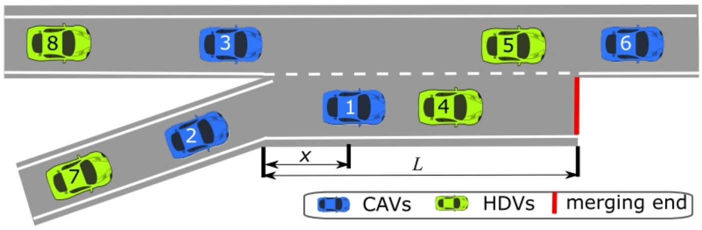

Figure 7. Traffic situation involving vehicles using an entrance ramp to merge onto a highway [46]. In the figure, CAV denotes the connected and autonomous vehicle and HDV denotes the human-driven vehicle.

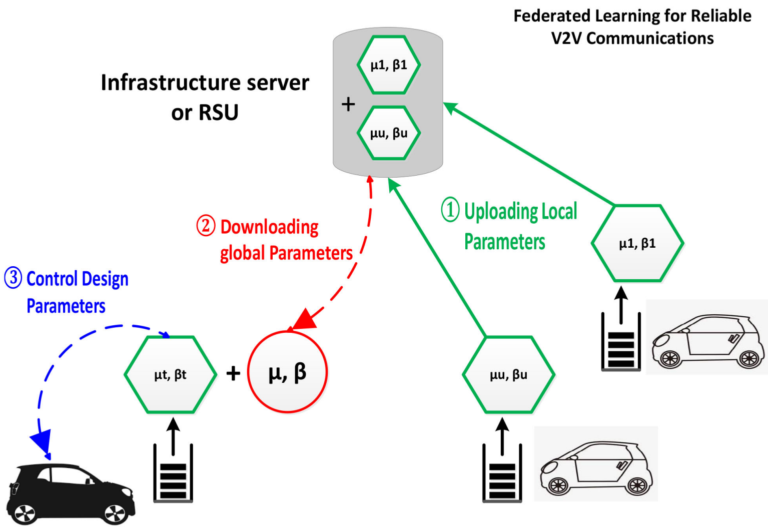

Figure 8. Federated learning for dependable vehicle-to-vehicle (V2V) communication [45].

2.3. Platooning

Platooning involves the formation of a queue of autonomous cars for long-distance driving. For vehicles that must travel long distances, platooning can help to achieve goals such as reducing energy use, reducing CO_2 emissions, and alleviating traffic jams. With the gradual popularization of intelligent network vehicles, platooning is expected to become a common vehicle-driving method. After vehicles form a platoon, they drive in formation for a period of time, during which the local model can be updated through distributed learning among vehicles [47].

It is important to note that methods for the platooning of vehicles are constantly evolving. Various types of platooning arise from factors such as the speed at which the leading vehicle travels, the platoon length, and the entry and exit of vehicles. It is evident that transferring the knowledge gained through vehicle platooning to new vehicles via learning is challenging. The authors of [48] improved the speed of learning convergence through the use of a combined bandwidth allocation and scheduling scheme within a specific training time limit. In contrast, the scheduling policy design reported in [49] considered not only the device's channel conditions but also the significance of its local model update, as indicated by the l_2-norm of the model update. As a result, autonomous vehicle models can adapt and learn new scheduling strategies in vehicle scheduling scenarios, leading to improved traffic efficiency. In the IRS-assisted multi-platoon cooperation scenario depicted in Figure 9, a vehicle queuing network is established, with the queue that requests the learning service being designated as the requester (SP), while the queue that intends to train the shared learning model is considered as the provider (DP). Each row comprises a leading car and a sequence of subsequent cars. As illustrated in Figure 10, a distributed learning system is composed of a demand-driven queue (i.e., the requester), K service-oriented queues (i.e., the provider), and an intelligent reflecting surface (IRS) with N reflection elements. The collection of rows is denoted by $\mathcal{K} = \{0, 1, 2, \ldots, K\}$, where $k = 0$ signifies the requester, and the others are the k providers. All collaborative training approaches involve shared learning models ω in the collection, with all training data stored locally for each end user. The objective of distributed learning is to determine the model parameter ω that minimizes the loss function $L(\omega)$ across the entire dataset. Optimization challenges can be formulated as follows:

$$\min_{\omega} L(\omega) \triangleq \min_{\omega} \{ \frac{1}{K} \sum_{k=1}^{K} L_k(\omega) \}. \quad (1)$$

where $L_k(\omega)$ denotes the local loss function.

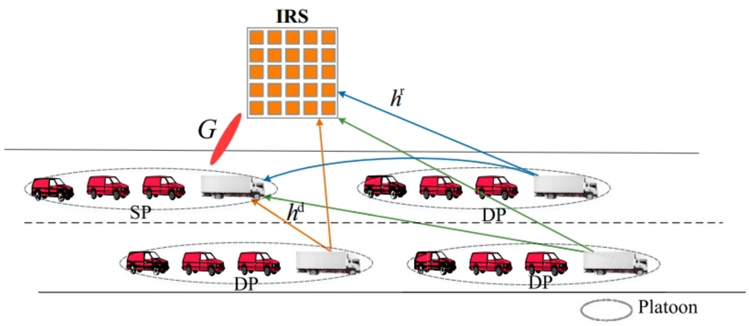

Figure 9. A situation involving IRS-supported collaboration among multiple platoons [47]. In the figure, IRS denotes the intelligent reflecting surface, SP denotes the platoon that requests a learning service, DP denotes the platoons scheduled to train a shared learning model, G denotes the channel responses from the IRS to SP, h^r denotes the channel responses from DP to the IRS, and h^d denotes the channel responses from DP to SP.

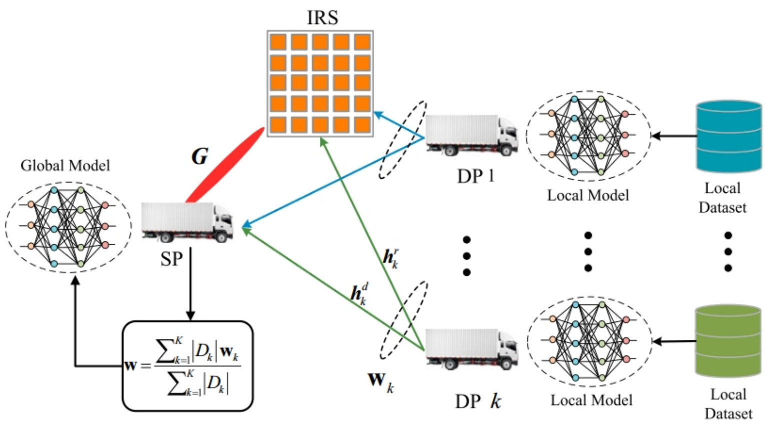

Figure 10. A distributed learning system based on an IRS [47].

Vehicle-to-vehicle (V2V) communication can reduce the gaps between vehicles, thus decreasing their air resistance and increasing road traffic efficiency. The development of platoon control technology has allowed for basic driving functions to be performed, which is essential for industrial applications and brings considerable economic benefits [50].

2.4. Traffic Flow Optimization

Traffic flow optimization is traditionally achieved through adjusting traffic lights and digital signals [22]. However, new solutions have been presented for autonomous vehicles. Generally, congestion can be predicted through a centralized traffic management system, allowing autonomous vehicles to formulate and implement congestion avoidance plans. In urban areas, traffic congestion frequently arises at specific points during set times, such as during rush hours, in tunnels, and on bridges. Having expertise in optimizing traffic flow at these spots can help new vehicles avoid congestion, thus enhancing urban road traffic flows. Consequently, incorporating learning-based techniques into traffic flow optimization scenarios can further enhance vehicle models.

Ye et al. [51] investigated the application of federated learning in vehicular edge computing (VEC) to address traffic flow optimization issues. In their study, as depicted in Figure 11, each vehicle client performs local deep neural network (DNN) training using images captured with its onboard camera. Image classification is seen as a common use of artificial intelligence in vehicular edge computing (VEC). The model relies on DNN image classification, which has been extensively applied in autonomous driving and interactive navigation of intelligent connected vehicles (ICVs), along with target tracking and event detection in ITSs. For example, the client within the vehicle utilizes the in-car camera for image capture, followed by classifying and labeling images using automated labeling technology [52]. Subsequently, the central server designates the vehicle client to engage in supervised federated learning, producing global and local updates for DNN models. To address variations in image quality and computational capabilities among vehicle clients during federated learning for image classification, a selective model aggregation approach was suggested. Through experiments conducted on the MNIST and BelgiumTSC datasets, the proposed selective model aggregation approach demonstrated superior performance, in terms of accuracy and efficiency, compared with the original FedAvg method.

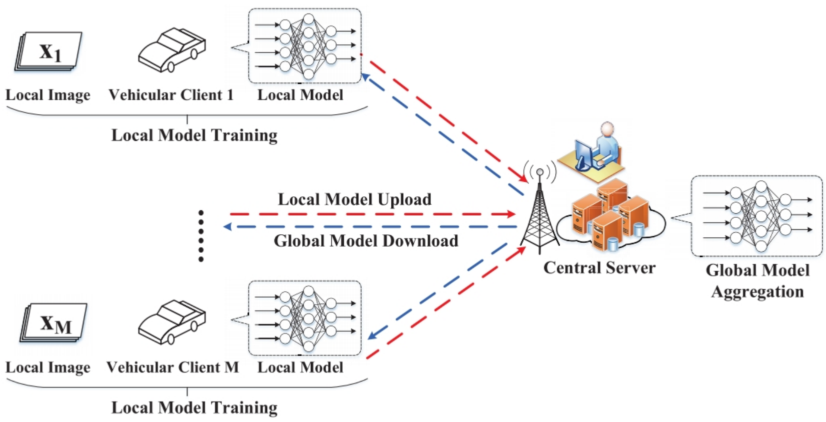

Figure 11. A comprehensive framework for federated learning in the context of edge computing in vehicular environments [51].

2.5. Summarizing Discussion

Selecting the most appropriate approach for a specific situation can be difficult, considering the variety of possible scenarios and options. We outlined various machine learning models and investigated how vehicles acquire model parameters from one another or central servers in different situations.

- Each vehicle is equipped with a unique model tailored to various scenarios. Through incorporating distributed learning among vehicles and implementing specific strategies, a more effective adaptation to varying traffic patterns in ITSs can be achieved.
- Distributed methods enable the sharing of computational resources among various entities, in order to distribute the global computing workload evenly. This approach helps to minimize the risk of individual system failures and ensures that programs are executed on the most appropriate computing nodes. In the context of advanced autonomous driving systems, such as those found in intelligent transportation systems, there is a growing need for complex environmental sensing capabilities. As a result, distributed methods have gained significance in this domain. Consequently, intelligent networked autonomous driving has emerged as a crucial technological pathway,

leading to the development of sophisticated transportation systems that seamlessly integrate vehicles, road infrastructure, and cloud computing resources.

- Federated learning is a decentralized approach to machine learning, in which numerous users work together to train a model without centralizing the original data onto a single server or data center. This method utilizes either raw data or data that have been processed securely based on the raw data for training purposes. In the realm of autonomous driving, federated learning allows vehicles to learn from one another while on the road, facilitating collaborative model training for autonomous vehicles without requiring all data to be stored on a single vehicle.
- Edge computing involves relocating computing tasks from the central server to the network edge devices. Implementing this technology in autonomous vehicles allows them to instantly analyze extensive sensor data, leading to prompt and precise operational decisions. Consequently, this enhances driving safety and efficiency. For instance, in intricate traffic scenarios, autonomous vehicles must swiftly detect and react in diverse situations. Utilizing edge computing, vehicles can analyze camera image data in real-time; recognize pedestrians, vehicles, and obstacles; and take appropriate actions accordingly.
- The peer-to-peer distributed learning algorithm with parameter duplicates for every node enables direct communication among nodes. This approach offers excellent scalability and eliminates any centralized failure point. Peer-to-peer distributed learning shows promising application potential in various scenarios, such as intersection management, ramp merging, platooning, and traffic flow optimization.

3. Approaches (in Response to RQ2)

Based on the discussion in Section 2, we believe that distributed learning using parameter servers in ITSs significantly enables decentralized distributed learning and peer-to-peer distributed learning. In the following sections, we will present a few relevant approaches. They can be seen in Figure 12.

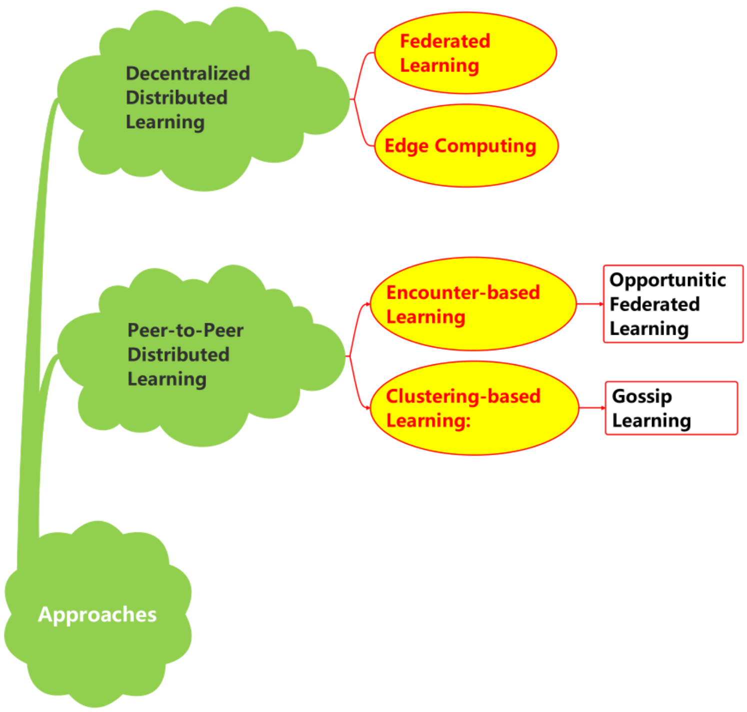

Figure 12. The approaches to distributed learning.

3.1. Decentralized Distributed Learning

Federated learning is a form of global modeling that is achieved by combining the results of training distributed models from multiple data sources without the need to exchange individual local or sample data [53–56]. Such a model is based on the virtual fusion of data, striking a balance between protecting data privacy and enabling data-sharing computation by exchanging only model parameters or intermediate results [57,58]. In an ITS, interactive learning between vehicles and the system environment can help to integrate different sources of information while ensuring privacy protection, further improving the autonomous driving experience through vertical federation using other urban information such as that obtained from city cameras, traffic lights, and smart roads [59]. Kan Xie et al. [60] utilized federated learning to tackle the problem of traffic sign recognition, enabling cooperative training of accurate recognition models without having to exchange the original data of the traffic sign. The spike neural network (SNN), when combined with a traditional federated convolutional neural network, provided greater accuracy, noise resistance, and energy efficiency.

At present, in order to successfully adapt to the application scenarios of ITSs, federated learning must make breakthroughs in terms of efficiency and computational/time costs, as detailed in the references reported in Table 4.

Table 4. Research on federated learning: bi-dimensional taxonomy.

	Efficiency	**Low Cost**
Federated Learning	Liu et al. [61], Mills et al. [62], Wang et al. [63]	Wu et al. [64], Luo et al. [65]

The references in Table 4 have illustrated the advantages of the HybridFL, MFL (momentum federated learning), CFL (cluster-based federated learning), and MTFL (multi-task federated learning) algorithms, in terms of their high efficiency.

HybridFL: Wu et al. [64] proposed a HybridFL federated learning protocol to provide privacy protection and efficiency in a mobile edge computing architecture. This algorithm involves clients $C_r(t)n_r$ setting the region selection ratio $C_r(t)$ in round t, with the expectation that $C \cdot n_r$ will not drop out or opt-out. The desired proportion of clients who submit their local models in a round is denoted as C, which is determined by the cloud. $X_r(t)$ refers to the clients in $X(t)$ that are part of region r. Meanwhile, n_r represents the number of clients that are connected to the edge node r. Consequently, the selected targets in this region-wise manner are as follows:

$$\begin{aligned}\mathbb{E}[|X_r(t)|; C_r(t), n_r] &= \sum_{k=0}^{C_r(t)n_r} k \cdot P(|X_r(t)| = k) \\ &= C_r(t)n_r\theta_r(t).\end{aligned} \tag{2}$$

In HybridFL, a quota-triggered aggregation system is utilized. Once the $C \cdot n$ client models have been sent to the mobile edge computing system, the round is completed in the cloud. $S_r(t)$ refers to the subsets of clients in $S(t)$ that are part of the region r. Consequently, the following steps are carried out:

$$|S_r(t)| = \mathbb{E}[|X_r(t)|; C_r(t), n_r] \cdot \frac{|X_r(t)| \cdot q_r^*(t)}{\mathbb{E}[|X_r(t)|; C_r(t), n_r]}, \tag{3}$$

$$q_r(t) \triangleq \frac{|X_r(t)| \cdot q_r^*(t)}{\mathbb{E}[|X_r(t)|; C_r(t), n_r]}, \tag{4}$$

From Formulas (2)–(4), we have the following:

$$\begin{aligned} |S_r(t)| &= \mathbb{E}[|X_r(t)|; C_r(t), n_r] \cdot q_r(t) \\ &= C_r(t) n_r \theta_r(t) \cdot q_r(t). \end{aligned} \tag{5}$$

In addition,

$$C_r(t) = C \cdot n_r \frac{\sum_{i=1}^{t-1} (C_r(i) q_r(i))^2}{\sum_{i=1}^{t-1} C_r(i) q_r(i) |S_r(i)|}, t > 1. \tag{6}$$

An experimental evaluation revealed that HybridFL outperformed FedAvg and HierFAVG in terms of global model accuracy. In particular, it was able to achieve the best global model with the lowest number of federated rounds, making it suitable for mobile edge computing in ITSs. This can reduce the average length of training, speed up the convergence of the global model, improve the model's accuracy, and reduce energy consumption on the device side. However, the HybridFL algorithm requires different environmental settings, such as $\mathbb{E}[dr]$ and the proportion C of client selection in the Airfoil and MNIST datasets, resulting in a total time consumption of 3143–15,978.4 s to obtain Acc = 0.90, which is too long for deployment in autonomous vehicles in an ITS.

MFL: Liu et al. [61] proposed an alternative to HybridFL called momentum federated learning (MFL), which integrates momentum gradient descent (MGD) into the local update step of the federated learning system. This algorithm takes the momentum term of the previous iteration into account. An experimental evaluation based on the MNIST dataset demonstrated that momentum federated learning is globally convergent. It was confirmed that the use of MFL leads to accelerated convergence. Accelerated federated learning is crucial in ITSs, as vehicles travel at very fast speeds and have a limited time to complete federated learning. Selecting a value of γ of around 0.9 in MFL can allow the optimal convergence speed to be achieved; however, at this point, the number of iterations is still greater than 1000—relatively long compared to The duration required by autonomous vehicles in ITSs.

CFL: Wang et al. [63] proposed an algorithm, referred to as cluster-based federated learning (CFL), in order to address the resource-efficient federated learning problem (RFL-HA) with hierarchical aggregation. Compared with traditional federated learning, CFL can reduce the time needed to complete the task by 34.8–70% and communication resources by 33.8–56.5% while maintaining similar precision. The architecture of the algorithm is shown in Figure 13. In an ITS, clustering analysis and hierarchical aggregation using this algorithm can accelerate the model learning rate. Consider a scenario where we have a total of N edge nodes, which are further categorized into K distinct clusters. The RFL-HA problem can be reformulated as follows:

$$\begin{aligned} L(\mathcal{K}) = {} & [\frac{\mathcal{K}-1}{N} + \alpha(1-\eta\mu)^H]^{\frac{R_m \mathcal{K}}{N(c_m+b_m)}} [F(\omega^0) - F(\omega^*)] \\ & + \frac{(Q_1+Q_2)[1-(\frac{\mathcal{K}-1}{N} + \alpha(1-\eta\mu)^H)^{\frac{R_m \mathcal{K}}{N(c_m+b_m)}}]}{2\eta\mu^2}. \end{aligned} \tag{7}$$

The optimal value of $\mathcal{K}$ can be ascertained as follows:

$$\mathcal{K}^* = arg \min_{\mathcal{K} \in 1,2,\ldots,N} L(\mathcal{K}). \tag{8}$$

This algorithm can achieve excellent performance under resource constraints and provide valuable learning solutions for vehicles in ITSs with limited resources. Tests on the MNIST and CIFAR10 datasets demonstrated that CFL outperformed FedAVG (z = 100) and FedAsync when different $\mathcal{K}$ value time budgets were used (e.g., 600 s for SVM/LR and 1800 s for CNN over MNIST). However, the 600 s and 1800 s time budgets were slightly longer for the ITS.

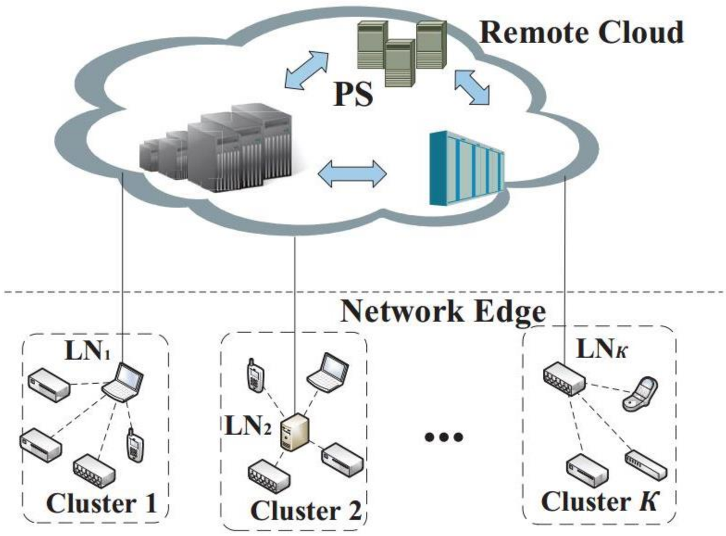

Figure 13. The architecture of RFL-HA [63]. In the figure, PS denotes the parameter server and LN denotes the leader node.

MTFL: Mills et al. [62] proposed a multi-task federated learning algorithm (MTFL) that incorporates a non-federated batch normalization layer (BN) into federated deep neural networks. This allows users to carry out customized model training based on their own data, thus improving the accuracy of the user model and the convergence speed. The illustration in Figure 14 depicts the functioning of the MTFL algorithm in the context of edge computing.

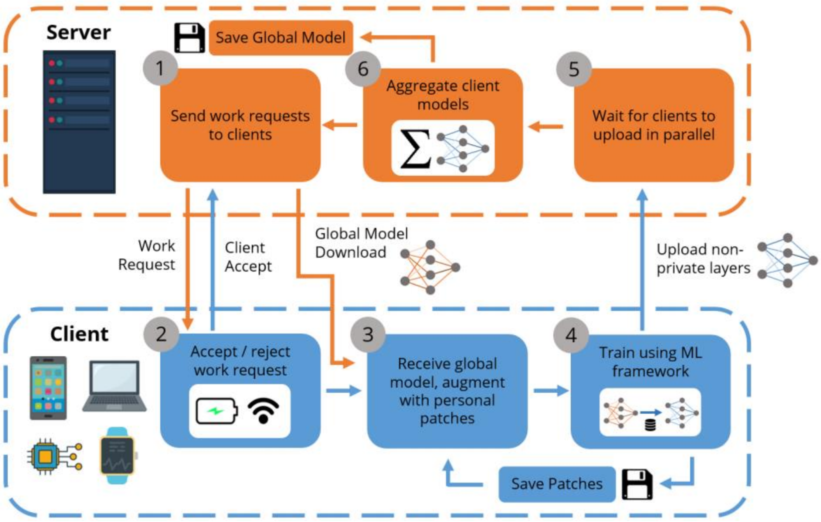

Figure 14. The multi-task federated learning (MTFL) algorithm utilized in edge computing [62]. In the figure, ML is machine learning.

Federated learning aims to minimize the objective functions to the greatest extent possible:

$$F_{FL} = \sum_{k=1}^{K} \frac{n_k}{n} \ell_k(\Omega). \tag{9}$$

The federated learning model can be modified by incorporating a different client patch, which will alter the objective function of MTFL, as follows:

$$F_{MTFL} = \sum_{k=1}^{K} \frac{n_k}{n} \ell_k(\mathcal{M}_k), \tag{10}$$

$$\mathcal{M}_k = (\Omega_1 \dots \Omega_{i_1}, P_{k_1}, \Omega_{i_1+1} \dots \Omega_{i_m}, P_{k_m}, \Omega_{i_m+1} \dots \Omega_j). \tag{11}$$

Batch normalization (BN) layers can be used as a patch for multi-task learning (MTL) in a centralized setting. The BN layers are expressed using the following equation:

$$\begin{aligned} \hat{x}_i &= \frac{z_i - \mathbb{E}(z_i)}{\sqrt{Var(z_i) + \epsilon}} \\ BN(\hat{x}_i) &= \gamma_i \hat{x}_i + \beta_i. \end{aligned} \tag{12}$$

The average and variance of neuronal activation (z_i, post-nonlinearity) in the mini-batch, $\mathbb{E}(z_i)$ and $Var(z_i)$, are used to determine the learning parameters γ_i and β_i during the training process. For ITS, the MTFL algorithm can achieve the highest average user model accuracy with a limited number of vehicle communication cycles. Through conducting experiments on the MNIST and CIFAR10 datasets, it was determined that the average times for one MTFL cycle were 38 and 136 s, respectively. This time consumption was much less than that of the HybridFL and MFL algorithms and, so, it should be applicable in ITSs.

Self-adjusting federated learning: Luo et al. [65] proposed the self-adjusting federated learning approach in mobile edge networks to optimize the selection of essential control variables, thus decreasing total costs while ensuring convergence. This can provide cost-effective solutions for large-scale applications in ITSs. We aim to reduce the overall cost as much as possible while ensuring that the process converges. This can be expressed as the following problem:

$$\begin{aligned} min_{E,K,R} \quad & \mathbb{E}[C_{tot}(E,K,R)] \\ s.t. \quad & \mathbb{E}[F(W_R)] - F^* \leq \epsilon; K, E, R \in \mathbb{Z}^+; \text{ and } 1 \leq K \leq N. \end{aligned} \tag{13}$$

Satisfying the convergence constraint on the upper bound, Equation (13) can be approximated as follows:

$$\begin{aligned} min_{E,K,R} \quad & \left(\frac{(1-\gamma)\sum_{i=K}^{N} C_{i-1}^{K-1} t_i}{C_N^K} + \gamma K(e_p E + e_m)\right)R \\ s.t. \quad & \frac{1}{ER}\left(A_0 + B_0\left(1 + \frac{N-K}{K(N-1)}\right)E^2\right) \leq \epsilon, \\ & K, E, R \in \mathbb{Z}^+ \text{ and } 1 \leq K \leq N. \end{aligned} \tag{14}$$

Substituting $\widetilde{\mathbb{E}}[C_{tot}]$ for $\mathbb{E}[C_{tot}]$, we obtain the following approximation:

$$\begin{aligned} min_{E,K} \quad & \frac{((1-\gamma)(t_p E + t_m) + \gamma K(e_p E + e_m)) \cdot (A_0 + B_0(1 + \frac{N-K}{K(N-1)})E^2)}{\epsilon E} \\ s.t. \quad & E \geq 1 \text{ and } 1 \leq K \leq N. \end{aligned} \tag{15}$$

The performance and efficacy of the algorithm were confirmed through a federated learning test using a laptop as the central server, 20 Raspberry Pis, and approximately 10 Jetson Nanos as edge devices. This design differed from most resource allocation tasks in federated learning systems and can help reduce expenses.

Similarly, Sha Liu et al. [66] used federated learning to achieve collaboration among connected autonomous vehicles (CAVs) without revealing local data. A local decision-making model was proposed for typical rear-end collision scenarios, and the accuracy of the global model was enhanced using federated learning.

In general, federated learning only shares gradients and does not share local data and models, providing it with wide application prospects in many scenarios [57]. However, to apply federated learning in ITSs, a balance must be achieved between efficiency and cost [67]. Many scholars have performed research in this regard, but there is still no sufficiently good approach. At the same time, the use of federated learning gradient aggregation servers still poses risks related to privacy and security.

3.1.1. Edge Computing

The correlation between edge computing and distributed learning within the realm of ITSs is pivotal in attaining effective and flexible solutions. In this context, edge computing denotes the practice of conducting computations in proximity to a data source; which, in this instance, would be the vehicles within the ITS. Through the utilization of edge computing, each vehicle can be furnished with a customized model suited for diverse situations, facilitating localized decision-making and diminishing the necessity for continuous communication with a central server. This configuration fosters distributed learning among vehicles, enabling them to exchange insights and cooperate to enhance their individual models. Through incorporating specific methodologies for distributed learning, such as federated learning or collaborative filtering, vehicles can collectively assimilate knowledge from each other's encounters and adjust more efficiently to fluctuating traffic conditions. Overall, the amalgamation of edge computing and distributed learning elevates the intelligence and responsiveness of ITS systems, resulting in enhanced traffic control and safety.

Edge computing is an alternative to traditional cloud computing, which involves providing services on the network edge and performing computing and data generation [68]. It is distinct from the conventional cloud computing approach. For the ITS, it is necessary to improve vehicular services by distributing computing tasks between the remote cloud and the local vehicular terminals [69]. To further reduce the latency of computing off-load and transmission costs, edge computing is a potential solution approach. There are many discussions about the application of edge computing to ITS. Some discuss edge computing architecture [70], some discuss performance [71], some discuss edge server layout [72,73], and some discuss resource allocation [74].

Baidya et al. [70] studied the potential of edge computing architectures for vehicular computing and showed the pros and cons of different offloading strategies. They discussed lightweight, high-performance, and low-power computing paradigms, architecture, and design space exploration tools to meet the changing needs of connected and autonomous vehicles. Experimental results showed that the computing power of vehicular local computing (VLC) is only half that of vehicular edge computing (VEC). When the communication speed was low (5 Mbps), the most efficient approach was not to offload, as it had the least end-to-end delay. As the communication speed increased (10–20 Mbps), partitioning and offloading became the most suitable choices, as full offloading still had a relatively high communication latency. Of course, when the quality of the network is further improved (>30 Mbps), full offload is the best option.

Galanopoulos et al. [71] proposed an automated machine learning (AutoML) framework to tackle the difficulty of optimizing the accuracy of the analysis while still adhering to the minimum frame rate requirement in a situation where a large number of configuration parameters are unknown and can vary over time, which affects the performance of multi-access edge computing (MEC). The aim is to maximize users' cumulative confidence (CC) while meeting their frame rate requirements.

3.1.2. Summarizing Discussion

Due to its particularity in protecting privacy, federated learning has gradually received more applied research in the ITS field [75]. In particular, edge computing provides an effective approach to reducing latency and transmission costs in ITSs.

Scholars have conducted research on the distributed application of federated learning and edge computing in vehicles, with Takayuki Nishio et al. [76] proposing a mobile edge computing framework for federated learning (FedCS) that utilizes distributed learning to create high-performance models while preserving client privacy. Other studies in this area include those of Ye et al. [51], Chi et al. [77], Mo et al. [78], Liu et al. [79], and Lim et al. [80]. These studies demonstrated the possibility of effective training in resource-constrained environments. Simultaneously, Wei Yang Bryan Lim et al. [80] examined federated learning in mobile edge networks. In particular, topics that require further exploration are related to communication expenses, resource distribution, data confidentiality, and data safety.

3.2. Peer-to-Peer Distributed Learning

There are several types of distributed learning: one type is encounter-based learning (e.g., opportunistic federated learning), while another type is clustering-based learning (e.g., gossip learning). In the context of an ITS, a multitude of self-driving vehicles can be expected to join the system in a sequential manner. Learning new data models without revealing local data models is within the application scope of distributed learning.

3.2.1. Encounter-Based Learning: Opportunistic Federated Learning

Encounter-based learning approaches are algorithms that carry out learning from models when they meet each other. A typical algorithm in this category is opportunistic federated learning [81]. In an ITS, all autonomous vehicles have the opportunity to encounter one another. Through the exchange and learning of data from a vehicle model during an encounter, the updated autonomous vehicle model can be adapted to more application scenarios. Therefore, ITSs are a typical field in which encounter-based learning applications can be applied.

Sangsu Lee et al. [81] proposed an approach to federated learning known as opportunistic federated learning (OppCL). This method does not require the aggregation of gradients through a server, as is the case in traditional federated learning. Instead, each client exchanges gradients and updates their models during encounters. An opportunistic momentum was introduced to integrate the experience of others. This approach has two obvious characteristics: one is learning from encounters, and the other is its decentralized nature. Opportunistic momentum is a technique that utilizes the summary of data from neighboring devices to determine whether it is advantageous and possible to request learning support. If advantageous and practicable, the device can share model gradients and generate new local models for learners. The system architecture of opportunistic federated learning is depicted in Figure 15. Devices use locally obtained data to continuously refine the personalized model. Each device applies mobility and contact prediction to the device, in order to determine when to participate in opportunistic federated learning with the encountered device. This technique is highly suitable for the learning situation of autonomous vehicles in ITSs.

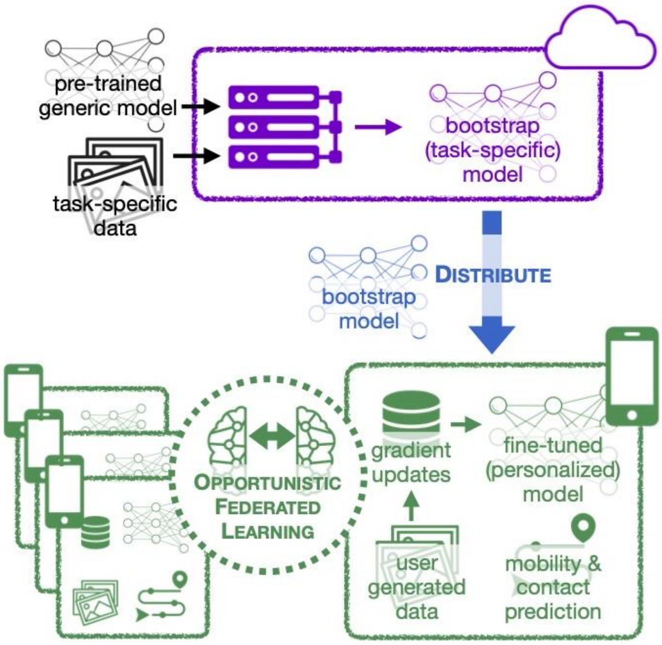

Figure 15. System architecture of opportunistic federated learning [81].

Consider a set of N devices, $C = \{C_1, \ldots, C_i, \ldots, C_N\}$, each with its own dataset D_i and with each sample having a label. We can measure the distribution of the data label $\mathcal{L}_i$ for device C_i as the relative frequency of each label in the local dataset. Each device has a desired distribution of labels G_i that it hopes its model can accurately classify. Each device C_i builds its own local model ω_i, which is associated with a local clock t_i related to the model $\omega_i^{t_i}$. Every time the local model of C_i is updated, t_i increases. The initial model for the device C_i is ω_i^0. Opportunistic federated learning updates the local model based on the combination of the device's local data and the gradient obtained from the encounter to optimize the local model toward the target distribution $\mathcal{G}_i$.

$$\min_{E_i} \sum_{t_i=0}^{|E_i|} \ell(\omega_i^{t_i}; \mathcal{D}_{\mathcal{G}_i}). \tag{16}$$

This algorithm first evaluates the similarity between the target distribution of the learners and the data label distribution of the neighbors to decide whether the encounter will be beneficial for learning. To do this, it takes into account the set of encounters experienced by the device C_i, denoted as E_i, and a hypothetical dataset $\mathcal{D}_{\mathcal{G}_i}$ whose data label distribution matches the target distribution $\mathcal{G}_i$.

$$sim(\mathcal{P}_1, \mathcal{P}_2) = \sum_{l \in L} min(\mathcal{P}_1(l), \mathcal{P}_2(l)). \tag{17}$$

Greedy aggregation directly averages the gradient of neighbor learning after adding a round of local learning, as follows:

$$\omega' = \frac{\omega_{(\mathcal{L}_i, \mathcal{G}_i)}(\nabla \ell(\omega'; \mathcal{D}_i)) + \omega_{(\mathcal{L}_j, \mathcal{G}_j)}(\nabla \ell(\omega'; \mathcal{D}_j))}{\omega_{(\mathcal{L}_i, \mathcal{G}_i)} + \omega_{(\mathcal{L}_j, \mathcal{G}_j)}}. \tag{18}$$

We consider the similarity between the data label distribution and the target distribution when weighing both the local and adjacent gradients. The weights are determined as follows:

$$\omega_{(\mathcal{L}, \mathcal{G})} = exp(-\lambda \times (1 - sim(\mathcal{G}, \mathcal{L}))). \tag{19}$$

The parameter λ reflects the tendency of the model to overfit datasets with a limited number of labels, which can also be seen as the model's preference for datasets that are well-balanced. Every time the model interacts with its neighbors, Γ is updated, connecting the neighbor's data label set to the gradient that was learned from that set. When a new encounter occurs, the technique averages all stored gradients before making changes to ω'.

In each device C_i, the learning rate is determined independently. The learning rate of C_i at time t_i is $\eta_i^{t_i} = \eta\alpha_i^{t_i}$, where η denotes the initial learning rate and $\alpha_i^{t_i}$ denotes the decay factor.

$$\alpha_{t_i} = \frac{exp(k \times (\phi - \|\omega_i^0 - \omega_i^{t_i}\|_2))}{exp(k \times (\phi - \|\omega_i^0 - \omega_i^{t_i}\|_2)) + 1}, \tag{20}$$

where

$$\alpha_{t_i} < \min\{\alpha_0, \ldots, \alpha_{t_{i-1}}\}, 0 < \phi, 0 < k, \tag{21}$$

where ϕ and k are constants, and the decay factor α is a sigmoid function that takes the L2 distance from the initial weight and transforms it to be used as input. This design encourages the model of C_i to look for solutions that are close to the bootstrap model.

Based on the MNIST and CIFAR-10 datasets, learning to navigate opportunistic WiFi and Bluetooth connections took a total time of 19.14 s for $MNIST_{WIFI}$, 55.50 s for $MNIST_{Bluetooth}$, 73.40 s for $CIFAR\text{-}10_{WIFI}$, and 300.77 s for $CIFAR\text{-}10_{Bluetooth}$ when using opportunistic momentum to achieve better accuracy. The use of WiFi can be considered more suitable for various scenarios of autonomous vehicles in an ITS. These results can be positively related to the intended application.

Lee et al. [82] investigated the promotion of decentralized and opportunistic learning in distributed computing. OppCL expands from homogeneous device networks to heterogeneous device networks, and we believe that it can improve negative impacts when dealing with unpredictable mobility patterns. Generally, it is advantageous to conduct extensive research on encounter-based learning algorithms, such as opportunistic federated learning, to teach vehicle models encountered in ITSs. Furthermore, Parijat Dube et al. [83] proposed a technique known as AI Gauge, which can be used to calculate the running time and cost of training deep learning models under various cloud configuration options (with an average relative error of 7–8%). This method can also be used to estimate the time and cost of autonomous vehicle learning models, which helps analyze whether an autonomous vehicle in an ITS can complete distributed learning within a specified encounter time.

3.2.2. Clustering-Based Learning: Gossip Learning

Clustering-based learning refers to clustering the clients to more effectively learn the model data. The most typical algorithm in this category is gossip learning [84]. Clustering-based learning is similar to the behavior of autonomous vehicles in ITSs, as it is fully distributed and decentralized. Therefore, ITSs are a common application area for this type of learning approach.

Gossip learning was first proposed by Ormandi et al. [84], and is known for its flexibility, privacy-preserving nature, and scalability [85]. The key protocol underlying gossip learning is called peer sampling, which provides each node with a set of peers (called views) for gossiping. This protocol is essential for enhancing the speed of information dissemination, and can even affect the performance of the learning model [86]. In fact, the peer sampling protocol can create an unstructured network, following a well-defined expression logic to accelerate the convergence time, reduce communication overhead, and/or improve overall system performance [87].

The gossip learning architecture is illustrated in Figure 16. In this algorithm, the nodes periodically transmit their local models to their views P. When receiving a model m from one of its neighbors, the node combines the model using a predefined model aggregation function. Subsequently, the resulting model is updated by training it on the local dataset D. This updated model is then taken as the new current model. The model aggregation function is a key element of gossip learning, as it allows nodes to learn from data from other nodes without actually having to train the model. This function usually involves a weighted average of the parameters of the model or a subset of these parameters.

Gossip learning assumes that data are stored on the edge device, without the need for a central server or component. Research has shown that gossip learning is superior to federated learning in scenarios where the training data are evenly distributed among nodes, and its overall performance is comparable to that of federated learning [88]. However, existing gossip learning approaches only apply to static nodes or fully connected graphs, making them vulnerable to node churn and any changes in the network configuration. Mina Aghaei Dinani et al. [89] proposed a novel decentralized architecture for gossip learning suitable for dynamic nodes. This algorithm was applied to estimate the short-term trajectory of vehicles in real-world urban settings. There are two key differences between gossip learning and opportunistic federated learning: the first is that gossip learning aggregates models rather than gradients; and the second is that gossip learning is a diffusion method, whereas opportunistic federated learning involves learning through encounters. These two techniques are likely to be used to stimulate the education of self-driving cars in ITSs.

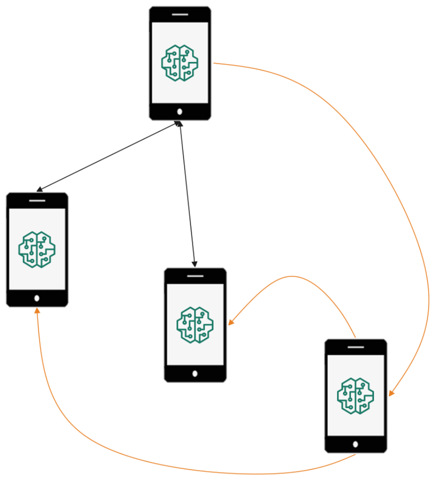

Figure 16. Overview of the gossip learning architecture. In the figure, each device represents a node. A bidirectional arrow indicates bidirectional communication, while a unidirectional arrow indicates unidirectional communication. In gossip learning, a node becomes active upon receiving new data and then periodically disseminates the data to other nodes. These nodes, in turn, propagate the data until all nodes have the updated information.

3.2.3. Summarizing Discussion

Encounter- and cluster-based learning approaches can be considered analogous to the behaviors of autonomous vehicles in ITSs and, so, can be applied to the corresponding autonomous driving scenarios, allowing for vehicle model learning. When autonomous vehicles encounter one another, they can update their autonomous driving models through gradient exchange and opportunistic federated learning to adapt to more autonomous driving scenarios. Some autonomous vehicles may also exchange applicable gradients to vehicles, as needed, through small-scale gradient exchange and model updates, which is the application scope of gossip learning.

4. Challenges and Research Directions (in Response to RQ3)

4.1. Challenges

In view of some of the difficulties and challenges of distributed learning in ITSs, scholars have performed analyses focusing on privacy and security, data, communication, and trust. The relevant literature is listed in Table 5.

Table 5. Relevant literature on the challenges of distributed learning in ITSs.

Challenges	Approach	Scenarios
Privacy and Security	Yin et al. [90], Wang et al. [91]	Cheng et al. [92], Mohseni et al. [93], Gautam et al. [94], Zhu et al. [95], Ren et al. [96]
Data intelligence sharing	Prakash et al. [97]	Paardekooper et al. [98], Prakash et al. [97]
Communication quality & speed	Naghsh et al. [99], Zhang et al. [100], Ding et al. [101], Situ et al. [102]	Zhou et al. [103], Wright et al. [104], Tang et al. [105], Zhou et al. [106]
Trust	Situ et al. [102]	Paardekooper et al. [98], Wang et al. [107], Hu et al. [108], Prakash et al. [97], Cao et al. [109], Liu et al. [37], Quinonez et al. [110]

4.1.1. Privacy and Security in Data Intelligence Sharing

Investigating data intelligence sharing for distributed learning in ITSs is a critical issue. Jan Pieter Paardekooper et al. [98] proposed that as data-driven artificial intelligence (AI) systems become increasingly common in self-driving cars, it is essential to improve the safety of autonomous driving functions by combining data-driven AI systems with knowledge-based AI into a hybrid AI system. Aditya Prakash et al. [97] studied data aggregation in learning visual-based urban autonomous driving policies. Although they studied the means of data aggregation, it is still a challenge to show what kind of data are shared in an ITS. Facilitating the exchange of data between vehicles and infrastructure to enable vehicles to act as cooperative learners, exchange knowledge from their sensor data, and promote smart data sharing poses notable challenges. Through data sharing, vehicles that have not experienced certain conditions can gain understanding and become more knowledgeable about those environments, ultimately improving their driving capabilities. However, enabling vehicles to participate in data sharing presents another barrier that requires attention.

The exploration of privacy and security in data intelligence sharing has been a popular topic. Jiahui Geng et al. [111] developed a versatile gradient reversal attack framework, which is capable of targeting both FedSGD and FedAVG simultaneously. Furthermore, they introduced a defense mechanism through numerous experiments that effectively mitigates attacks without compromising utility. Jingyang Zhang et al. [112] developed a privacy breach technique that enables attackers to precisely reconstruct a user's private training image using the adversarial training (AT) model of federated learning, even when dealing with a substantial training batch. Junbo Wang et al. [91] conducted a thorough investigation of the various privacy, security, and robustness issues in distributed learning. They identified three main types of attack in federated learning: membership inference, unintentional feature inference, and representative sample reconstruction. To protect against privacy leakage [113], techniques that reduce the overall model or its convergence (e.g., adding noise to the data [96]) are usually used, and computationally intensive operations—such as homomorphic encryption or secure multiparty computation—may be necessary. Finding a way to preserve privacy while detecting and avoiding malicious training or updates in an economical manner is a major challenge that requires further research. Xuefei Yin et al. [90] proposed a generic privacy preservation mechanism that encompasses three types of privacy-preserving techniques: cryptographic techniques, perturbative techniques, and anonymization techniques. Their classification, based on the 5W-scenario taxonomy, examines the potential privacy risks in federated learning from five perspectives: "who" (internal and external participants), "what" (active and passive participants), "when" (training phase and invitation phase), "where" (weight update, gradient update, and the final model), and "why" (four types of investment participants). The privacy-preserving federated learning (PPFL) method was studied and summarized based on four privacy-preserving schemes: encryption-based, perturbation-based, anonymization-based, and hybrid PPFL methods. Despite the recent surge in PPFL, this field of research faces many challenges, including the need to improve existing frameworks and develop new methods to enhance both the privacy and utility of data. In particular, in the field of intelligent transportation, low cost, and high efficiency have always been two directions worth exploring for PPFL. Since its introduction in 2016, federated learning has been widely acknowledged as a crucial approach to addressing privacy and security issues in machine learning [114,115], including in the context of ITSs. However, numerous investigations have highlighted that the transmission of shared gradients can also inadvertently disclose local data [113], underscoring the persistent complexity of the PPFL technique as a research avenue [116].

In any intelligent driving application scenario, distributed learning has significant and extensive potential applications. The efficient sharing of data is a crucial aspect of distributed learning that impacts its effectiveness. A key challenge lies in determining the type of data to be exchanged among vehicles and incentivizing participation in data sharing. Moreover, ensuring privacy and security is essential during data and intelligence sharing. Numerous studies have highlighted various attack techniques and the risk of privacy breaches in data sharing for distributed learning. Several scholars have introduced defense mechanisms and approaches, with the PPFL technique currently being among the most favored methods for privacy and security in federated learning. However, the integration of distributed learning techniques into ITSs continues to present notable privacy and security challenges in the realm of data intelligence sharing.

4.1.2. Communication Quality and Speed

Vehicle encounters in an ITS are limited in time and, therefore, must be taken advantage of. The successful completion of distributed learning depends on the quality and speed of communication. Therefore, research on communication methods will seriously affect the implementation of distributed learning.

Generally, two main factors restrict ITS communication technologies. The first is the advancement of V2X technology, and the second is the achievement of cooperative communication. Improving reliability and reducing latency have always been the main goals of communication problems. Yuntao Zhu et al. [117] proposed a Stackelberg game of IoV and a federated-learning-assisted spectrum-sharing framework to address the spectrum allocation issue caused by the rapid growth of IoV traffic. This framework is illustrated in Figure 17, and the simulation results demonstrated that this method can provide considerable performance enhancements. Furthermore, Sumudu Samarakoon et al. [39] proposed a distributed approach based on federated learning to address the joint problem of power allocation and resource allocation (JPRA) for ultra-reliable low-latency communication (URLLC) in-vehicle networks. Yang Chen et al. [118] proposed an asynchronous federated learning technology with temporally weighted aggregation to reduce communication overheads in federated learning. This is illustrated in Figure 18, which depicts federated learning using an asynchronous layer-wise model update and temporally weighted aggregation. A comparison of the MNIST and human action recognition datasets revealed that the algorithm was more effective than traditional federated learning, in terms of performance and communication cost.

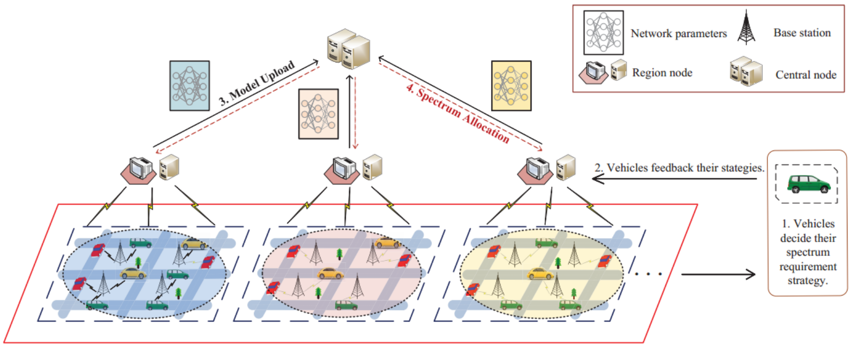

Figure 17. The Stackelberg game and federated learning-assisted spectrum-sharing framework for the Internet of Vehicles (IoV) proposed in [117].

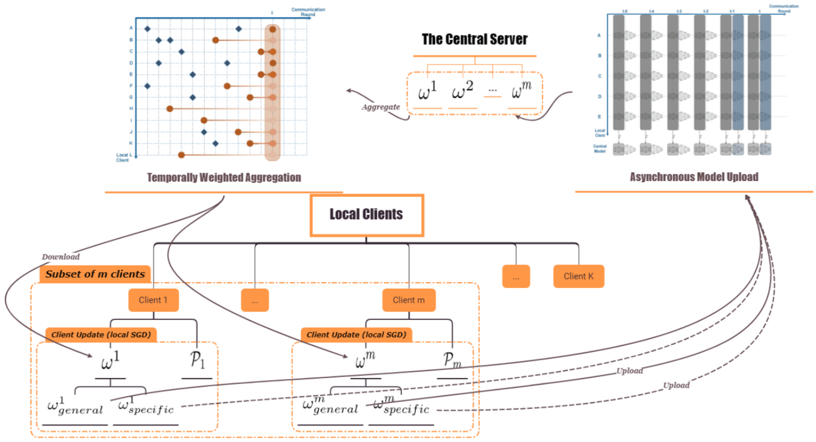

Figure 18. The method for federated learning, which utilizes a layer-wise asynchronous model update and a temporally weighted aggregation approach, proposed in [118].

Some studies have focused on researching communication technology [99,100,103]. Zahra Naghsh et al. [99] proposed a novel polynomial-time heuristic algorithm multichannel scheduler (MUCS) through their research on LTE V2X and 5G networks. This algorithm was found to be 30% more effective than existing methods, although with a moderate (but manageable) level of complexity. Haibo Zhou et al. [103] explored the potential of V2X technology and proposed the Internet of Vehicles (IoV) to meet the increasing demand for ITSs and autonomous vehicles. Figures 19 and 20 illustrate the IoV concept, its relationship with big data, and cloud-based IoV (CIoV), respectively. The two main development trends of CIoV are interoperability and trustworthiness. Minglong Zhang et al. [100] proposed a self-adaptive strategy based on fuzzy logic to address the issue of resource allocation in 5G cellular networks and V2X communications, which require ultra-low latency and ultra-high reliability. Yi Liu et al. [119] discussed the integration of 6G and federated learning, as well as the challenges, methods, and future directions associated with it. Figure 21 provides an overview of federated learning processes in the 6G context.

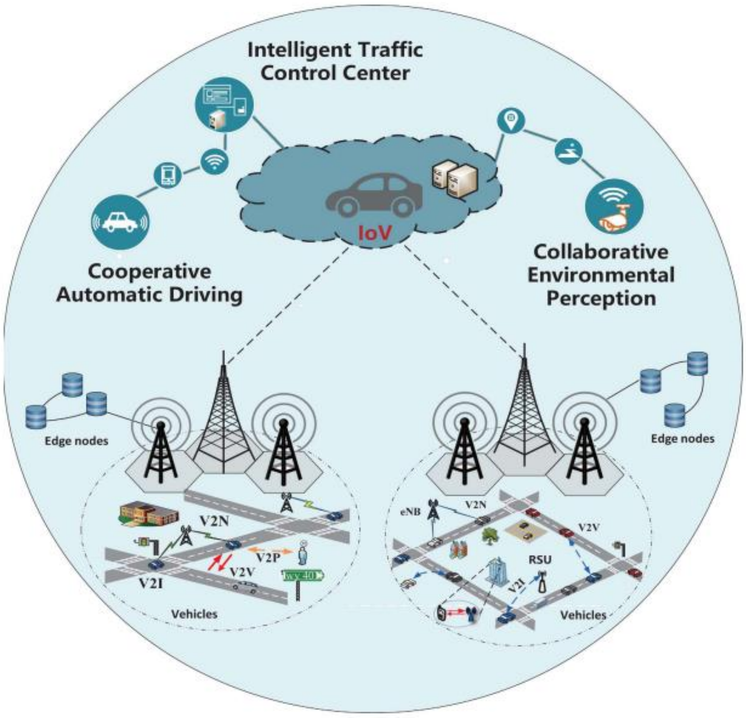

Figure 19. Illustration of IoV concept [103]. In the figure, IoV denotes the internet of vehicles, V2I denotes the vehicle to infrastructure, V2V denotes the vehicle to vehicle, V2P denotes the vehicle to pedestrian, V2N denotes the vehicle to network and RSU denotes the roadside unit.

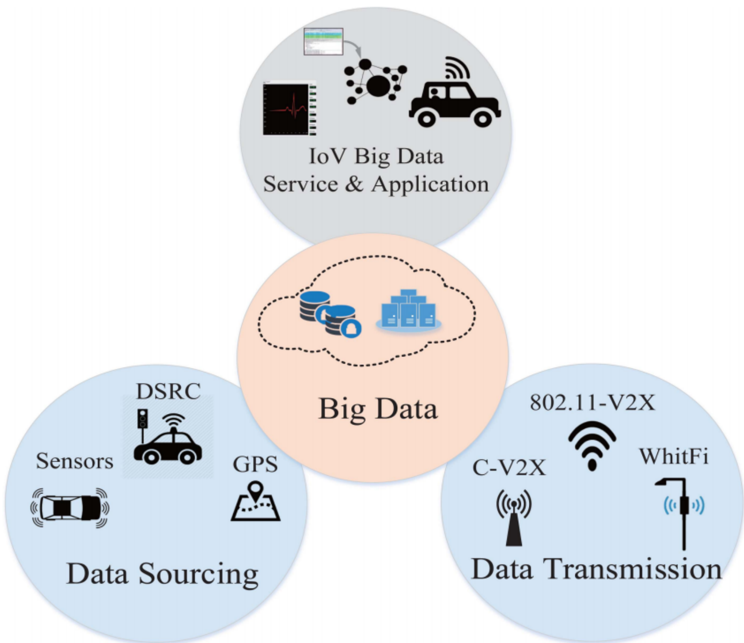

Figure 20. Mutual relationship between IoV and big data [103]. In the figure, DSRC denotes the dedicated short-range communications, C-V2X denotes the cellular vehicle to everything and GPS denotes the global positioning system.

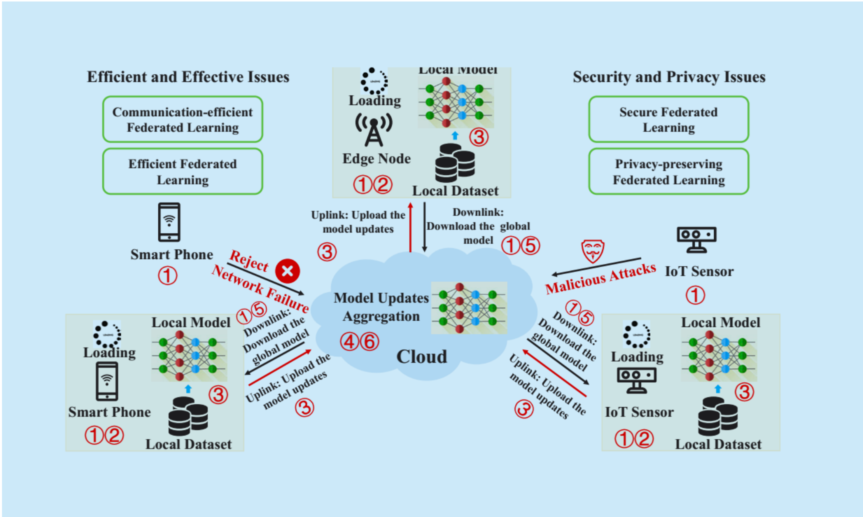

Figure 21. A summary of federated learning in 6G is presented in [119]. This study outlines the process of federated learning in the 6G era.

The works of Zuoyin Tang et al. [105], Huiyi Ding et al. [101], and Zhenhui Situ et al. [102] all focused on non-orthogonal multiple-access (NOMA) technology in V2X. They aimed to reduce channel load and improve the reliability and latency of communication. Relevant research has shown that NOMA technology may have good prospects for the communication aspect of ITSs. Jianshan Zhou et al. [106] studied the fundamental problems of communication in ITSs. The impacts of vehicle communication and networking on the advantages of distributed learning in a cooperative vehicle-infrastructure system (CVIS) were examined, and the optimal reliability collaboration result was determined. An analytical framework for the optimization of reliability-oriented collaborative computing was presented, as illustrated in Figure 22, based on the cooperative vehicle-infrastructure system (as depicted in Figure 23).

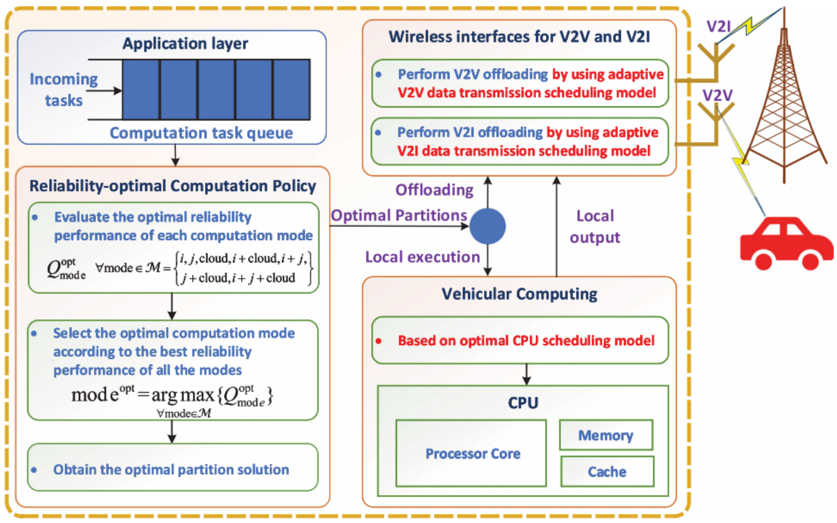

Figure 22. A structured approach for an implementation framework based on the reliability optimization models suggested in [106].

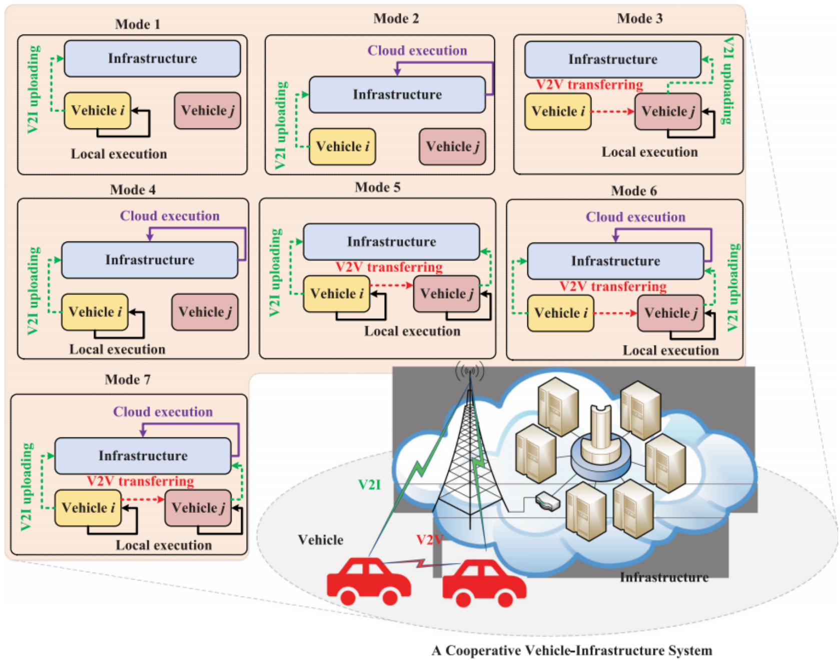

Figure 23. A cooperative vehicle-infrastructure system that can run a distributed mobile application in seven distinct modes, each of which is a blend of vehicle- and infrastructure-based executions [106].

Luping Wang et al. [120] proposed communication-mitigated federated learning (CMFL), which can identify irrelevant updates made by clients and prevent them from uploading to reduce the occupation of the network. This algorithm can greatly reduce communication overhead while ensuring learning convergence.

Wu et al. [121] proposed an efficient federated graph neural network (EmbC-FGNN) based on existing distributed GNN frameworks and research on federated graph neural networks (FedGNNs). This algorithm, as depicted in Figure 24, introduces an embedded server (ES) to store potential edge connections and embeddings between devices, broadcasting additional information between edge connections to expand its local training graph. The edge device trains the GNN model based on the extended graphic data and uploads the updates from boundary nodes required by other clients to the ES. A periodic embedding synchronization test was proposed to enable edge clients to use outdated embeddings for model training, thereby reducing the communication overhead. Fast asynchronous K training is also introduced for the parameter server (PS) and the embedded server, thus accelerating the convergence rate. This algorithm uses an embedded server to maintain edge connections between devices and synchronize the embedding of related nodes shared between training clients. In order to reduce the high communication cost when synchronous nodes are embedded, the edge device can employ outdated embeddings for model training and update them at a lower rate. To speed up the rate of convergence in the heterogeneous federated learning setting, the embedding server and parameter server are allowed to update the embedded node and model parameters asynchronously. This algorithm was found to enhance the overall accuracy in most GNN models and graph datasets. However, when the hidden size increases or the number of NN layers increases, EmbC-FGNN may suffer performance penalties. When the model is close to convergence, the performance of EmbC-FGNN may be inferior to that of FedGCN. The concepts of embedded servers and asynchronous updates can be combined with other federated learning methods in the future, which may lead to some beneficial attempts to reduce communication costs.

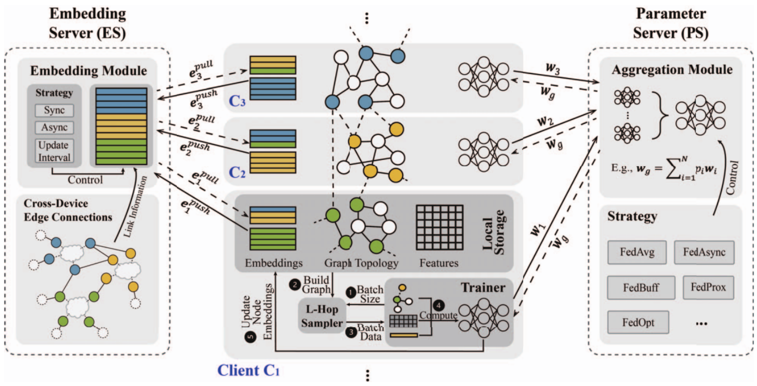

Figure 24. An overview of the EmbC-FGNN framework.

In order to facilitate the rapid implementation of distributed learning among vehicles, effective communication plays a crucial role. Numerous studies have explored advancements in communication technology, with some suggesting more efficient communication methods. Nevertheless, within the realm of ITSs, ensuring the high quality and speed of communication remains a significant challenge that warrants further investigation.

4.1.3. Trust

For ITS, trust is a very important issue [4]. This is because, once a network attack occurs and affects trust, autonomous vehicles may become uncontrollable and incidents that affect safety may occur. Zisheng Wang et al. [107], Raul Quinonez et al. [110], and Yulong Cao et al. [109] all suggested that attacking autonomous vehicle sensors through a network can pose a very serious threat or providing incorrect sensor data to provide untrusted information, which will have a great impact on the ITS. This impact is particularly evident in distributed systems, as it is difficult to allocate a large number of resources at each node for information recognition.

Prior work has aimed to address the issues of trustworthiness and safety through research on autonomous vehicle systems [92–96]. Kun Cheng et al. [92] proposed Guardauto, a decentralized runtime protection system for autonomous driving. As illustrated in Figure 25, the system is a partitioning model, which can separate the auto-driving system and insulate its components through a set of partitions. In the local protection function, the Guardauto function can alleviate the transmission delay caused by attacks or faults. In the global collaborative protection function, the average time from collecting beacon information to completing local recovery and sending clear beacons is 893.0 ms, while the average time for affected units from receiving beacon information to clearing fault status is 296.1 ms. For distributed learning in an ITS, if attack vehicles are mixed in, the learning accuracy will be affected. Referring to Guardauto's design, partition isolation should be implemented, and the mitigation of transmission delays caused by attacks or faults in local protection functions should be analyzed. Analysis of the time consumption and learning accuracy of the global collaborative protection function will also be beneficial.

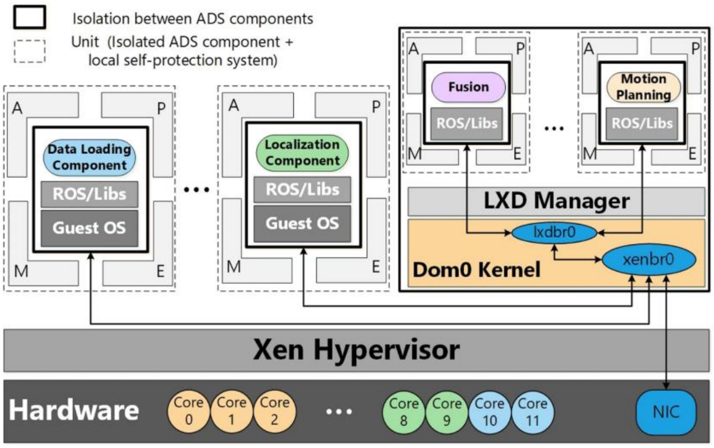

Figure 25. The Guardauto architecture [92].

Vibhu Gautam et al. [94] proposed the need to establish antivirus control software to address uncertainties from multiple sources and ensure the safe operation of autonomous vehicles. Pengfei Zhu et al. [95] studied the potential security vulnerabilities of the LTE-V2X protocol and developed six attack models based on malicious nodes. Their simulation results demonstrated that communication parameters, such as the packet delivery rate (PDR) and end-to-end delay, can affect the platooning status (leading to instability or even crashes), confirming the effectiveness of the attack.

Generally, in order to allow autonomous vehicles to adjust to different scenarios in an ITS, a variety of distributed learning algorithms have been proposed. However, it is still necessary to investigate how to ensure that the data obtained through distributed learning are reliable or to ensure that autonomous driving models are not adversely affected by distributed learning. Qiong Li [122] proposed a blockchain-based edge computing privacy protection method and federated learning to address the issue of efficiency when sharing privacy data on the Internet of Vehicles. Furthermore, this integration offers a partial solution to the issues of data privacy and trust [123,124]. Yunlong Lu et al. [125] attempted to solve the security and reliability problems of data sharing among vehicles by developing a hybrid blockchain and federated learning architecture. As illustrated in Figure 26, the architecture consists of the pre-commissioned blockchain and a locally directed acyclic graph (DAG). Figure 27 shows the process of the federated learning program with blockchain authorization. According to the evaluation results on the MNIST dataset, this scheme is very close to a centralized CNN, in terms of efficiency and accuracy, and far superior to the local CNN method. However, centralized CNNs have high privacy and data security risks. Meanwhile, this scheme outperforms the other two schemes with minimal time cost.

However, these schemes do not yet fully solve the trust problem of distributed learning in ITSs. Therefore, the problem of trust with distributed learning in an ITS remains a huge gap and a challenge that is worth further study and analysis.

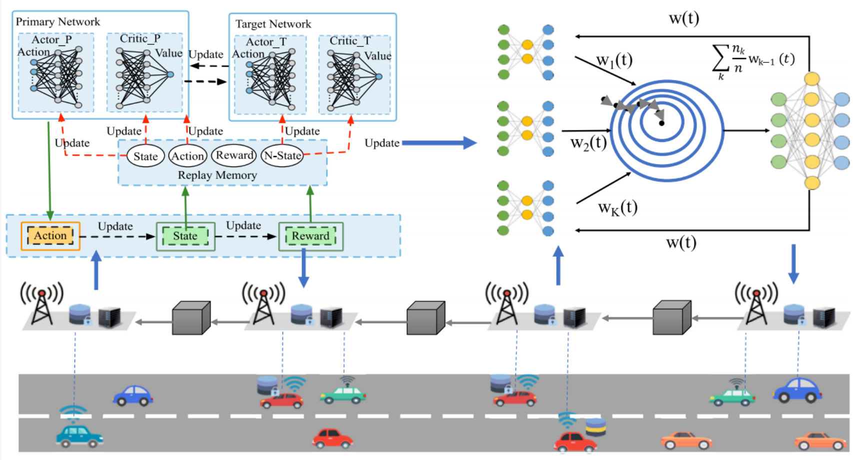

Figure 26. The structure of blockchain technology utilized to enable federated learning for the exchange of data in the Internet of Vehicles (IoV) [125].

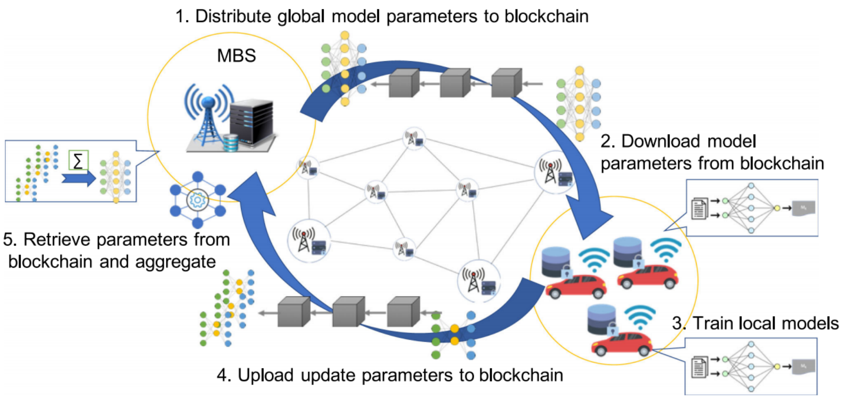

Figure 27. The procedure of a blockchain-enabled federated learning system, as explored in [125].

4.2. Research Directions

After the conducted survey and analysis of related challenges, we found that while distributed learning is indeed suitable for various ITS scenarios, there are still gaps and challenges relating to privacy and security, data, communication, and trust. As a result of our analysis of the relevant literature, we believe that the following five aspects are highly important research directions in this field.

- **Fast convergence of models**. For autonomous vehicles in an ITS, the time in which an encounter takes place, as well as the time to complete distributed learning, is limited. Therefore, developing rapid convergence methods for relevant models will be the key technical path to solving this problem. Recent studies [121,126] have made efforts and attempts in terms of gradient compression, acceleration of model convergence, and reduction of communication costs. Lin Zhang et al. [126] conducted a comparative study of three gradient compression methods (Sign-SGD, Top-k SGD, and Power-SGD) in distributed learning and found that, although they can achieve high compression ratios, they cannot provide performance improvements greater than those of S-SGD. Consequently, they proposed an alternative compressed Power-SGD (ACP-SGD) algorithm, which uses power iteration to compress each large gradient matrix into two low-rank matrices (P and Q). Unlike Power-SGD, it does not calculate and aggregate P and Q in one iteration but, instead, alternately compresses the gradients into P and Q. System optimization was also carried out through wait-free backpropagation (WFBP), tensor fusion (TF), buffer size, and other aspects. The results showed that the algorithm can achieve model accuracy comparable to that of S-SGD and is more efficient than S-SGD and Power-SGD. For distributed learning that requires gradient exchange, this algorithm can be used to enhance communication efficiency and reduce communication burden while achieving comparable model accuracy. Xueyu Wu et al. [121] introduced an effective federated graph neural network framework called EmbC-FGNN. As shown in the framework depicted in Figure 24, they incorporated an embedding server (ES) to manage potential inter-client node connections and embeddings, facilitating the exchange of additional information among edge clients to enhance the local training graph. The GNN models on edge devices are trained using augmented graph data, and updates of boundary node embeddings required by other clients are stored in the ES. They also suggested a periodic embedding synchronization test, enabling edge clients to utilize outdated embeddings during model training to minimize communication overhead. Additionally, they proposed a fast K-asynchronous training approach for parameter servers and embedding servers to speed up convergence. Future research directions could focus on achieving swift model convergence through employing techniques such as low-rank matrix factorization, gradient compression, and other methods to address the challenge of learning speed in distributed learning in ITSs.

- **Incentive mechanisms**. The use of distributed learning in an ITS requires the participation of autonomous vehicles. To encourage them to participate, a reinforcement-learning-based reward system should be explored. This is a critical step in resolving the issue. For example, a new method utilizes a blockchain to enable edge computing with the ability to resist tampering and single-point-of-failure attacks and incorporate gradient verification and incentive mechanisms into consensus protocols, encouraging more local devices to contribute computing power and data to federated learning in an honest manner [124,127]. Li et al. [128] presented a framework that focuses on preserving privacy while building a reputation based on OppCL. Their research incorporated a reputation metric within opportunistic federated learning, considering elements such as time and model loss, in order to incentivize clients interested in quality data to participate in the training. Additionally, the approach merged OT with gradient sharing to safeguard the privacy of vehicles. Encouraging high-quality data-seeking clients to participate in the training process would prove to be a highly effective approach.
- **Benefits of blockchain**. The reliability of each node in the system depends on privacy and trust. Blockchain provides a great solution to the trust issue [129], as its immutability ensures that the data are secure. In addition, its anonymity can help to protect privacy and security [130,131]. According to recent research, the combination of blockchain and OppCL can streamline the process of acquiring knowledge about autonomous vehicle models [124]. In a recent study, a new architecture called blockchain-based asynchronous signSGD (BASS) [132] was introduced, which combines a blockchain-based semi-asynchronous aggregation method with symbol-based gradient compression to enhance communication and aggregation efficiency, as well as to enhance security against attacks. Xu et al. [133] also suggested a novel dynamic optimized personal deep learning approach that uses blockchain and federated learning, enabling edge devices to collectively agree on the best weights for personalized models. These blockchain-based techniques have enhanced the efficiency of communication and convergence in federated learning. As a result, research on the combination of opportunistic federated learning and blockchain is a key area of study.
- **Methods of continual learning**. In vehicle systems, when new samples are available as new labels (e.g., new vehicle types or new traffic signs) or new features (e.g., the same stop sign but with different features from that in the training dataset and the deployment site), the underlying machine learning approach is continual learning [134]. When conducting continual learning, one may encounter catastrophic forgetting, where old labels or old features for the same label are forgotten [135]. In certain significant situations, continual learning can be employed to update the model by

incorporating new data while also maintaining previously acquired knowledge [136]. This approach obviates the necessity of storing training data from past tasks, thereby addressing issues related to data retention limitations imposed by physical devices (e.g., machine memory) or learning methodologies (e.g., privacy concerns) while conserving memory resources. Simultaneously, the model can retain the insights gained from prior tasks and effectively leverage this knowledge for future task learning, thereby enhancing overall learning effectiveness. Therefore, continual learning is also a very important research direction for distributed learning in ITSs.

- **Simulation analysis**. SUMO (which stands for Simulation of Urban Mobility) is a widely used open-source tool for analyzing traffic. Lee et al. [137] introduced SWARM, a tool designed to perform large-scale simulations for evaluation of the practical performance of distributed learning algorithms. SWARM is a versatile system that removes the need to assess intricate distributed learning choices. It can manage the setup, provisioning, and supervision of numerous working nodes and distribute tasks to working nodes that operate stateless servers. In SWARM, users can write code to specify how devices interact with each other, allowing for the simulation of various algorithms. The framework, known as Flower [138], which is specialized for federated learning, can conduct extensive FL trials through the introduction of novel features to accommodate various heterogeneous FL device scenarios. Flower can conduct FL experiments with a client base of up to 15M using only a set of top-tier GPUs. Additionally, it can effortlessly transfer experiments to actual devices, to explore different aspects of the design landscape. A hybrid testing tool, EdgeTB [139], provides multiple simulation nodes to create comprehensive and flexible testing environments. Additionally, it integrates physical nodes to ensure the high precision of the obtained results. This tool facilitates the efficient development and testing of various distributed machine learning architectures, with a focus on both accuracy and scalability. Researching method validation is a crucial area of study, which involves utilizing simulation methods to achieve validation.

5. Conclusions

In this study, we first introduced the taxonomy of different categories of distributed learning in ITSs and summarized various challenging problems according to these categories. Then, we formulated key research directions that could be determined for these challenging issues.

To build an ITS, it is essential to implement distributed learning in each situation involving autonomous vehicles. This will enable autonomous driving to be carried out more intelligently in various locations through the acquired models, resulting in more secure and efficient ITSs.

We hope that this work will be of assistance to both scholars and professionals, allowing them to gain a comprehensive understanding of the use of distributed learning in ITSs through a single conceptual framework, recognize the similarities and distinctions of these learning techniques, understand the current challenges in this field, and potentially identify new research directions through our framework.

Finally, we realize that there may be many additional problems that we have not discussed regarding the application of distributed learning in ITSs. With the gradual popularization of autonomous vehicles and the construction of ITSs, further research challenges can be expected to emerge in the future. However, we believe that our taxonomy, as well as our discussion of the application of distributed learning in ITSs, is universal enough and provides a clear basis to which future research issues can be adapted.

Author Contributions: Conceptualization, Q.L. and X.Z.; methodology, X.Z. and W.Z.; investigation, Q.L.; resources, X.Z. and W.Z.; data curation, Q.L.; writing—original draft preparation, Q.L.; writing—review and editing, Q.L. and W.Z.; visualization, Q.L.; supervision, W.Z. and X.Z.; project administration, X.Z.; funding acquisition, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Hunan Industry Polytechnic Project for 2024 under grant GYKYWT202403.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank ITSEG (Intelligent Systems Research Group) for helpful discussions on topics related to this work.

Conflicts of Interest: The author(s) declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Mishra, P.; Thakur, P.; Singh, G. Sustainable Smart City to Society 5.0: State-of-the-Art and Research Challenges. *SAIEE Afr. Res. J.* **2022**, *113*, 152–164. [CrossRef]
2. Nguyen, N.P.; Mogaji, E. Information technology for enhancing transportation in developing countries. In *Management and Information Technology in the Digital Era: Challenges and Perspectives*; Emerald Publishing Limited: Bradford, UK, 2022; pp. 81–94.
3. Zhu, L.; Yu, F.R.; Wang, Y.; Ning, B.; Tang, T. Big data analytics in intelligent transportation systems: A survey. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 383–398. [CrossRef]
4. Alladi, T.; Kohli, V.; Chamola, V.; Yu, F.R. A deep learning based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems. *Digit. Commun. Netw.* **2023**, *9*, 1113–1122. [CrossRef]
5. Almeida, J.; Alam, M.; Ferreira, J.; Oliveira, A.S. Mitigating adjacent channel interference in vehicular communication systems. *Digit. Commun. Netw.* **2016**, *2*, 57–64. [CrossRef]
6. Nama, M.; Nath, A.; Bechra, N.; Bhatia, J.; Tanwar, S.; Chaturvedi, M.; Sadoun, B. Machine learning-based traffic scheduling techniques for intelligent transportation system: Opportunities and challenges. *Int. J. Commun. Syst.* **2021**, *34*, e4814. [CrossRef]
7. Hahn, D.; Munir, A.; Behzadan, V. Security and privacy issues in intelligent transportation systems: Classification and challenges. *IEEE Intell. Transp. Syst. Mag.* **2019**, *13*, 181–196. [CrossRef]
8. Mao, Q.; Hu, F.; Hao, Q. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2595–2621. [CrossRef]
9. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [CrossRef]
10. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. [CrossRef]
11. Yuan, T.; Da Rocha Neto, W.; Rothenberg, C.E.; Obraczka, K.; Barakat, C.; Turletti, T. Machine learning for next-generation intelligent transportation systems: A survey. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e4427. [CrossRef]
12. Scardapane, S.; Wang, D. Randomness in neural networks: An overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2017**, *7*, e1200. [CrossRef]
13. Verbraeken, J.; Wolting, M.; Katzy, J.; Kloppenburg, J.; Verbelen, T.; Rellermeyer, J.S. A survey on distributed machine learning. *Acm Comput. Surv. (CSUR)* **2020**, *53*, 30. [CrossRef]
14. Peteiro-Barral, D.; Guijarro-Berdiñas, B. A survey of methods for distributed machine learning. *Prog. Artif. Intell.* **2013**, *2*, 1–11. [CrossRef]
15. Vepakomma, P.; Swedish, T.; Raskar, R.; Gupta, O.; Dubey, A. No peek: A survey of private distributed deep learning. *arXiv* **2018**, arXiv:1812.03288.
16. Cao, H.; Cai, J. Distributed multiuser computation offloading for cloudlet-based mobile cloud computing: A game-theoretic machine learning approach. *IEEE Trans. Veh. Technol.* **2017**, *67*, 752–764. [CrossRef]
17. Lin, R.; Zhou, Z.; Luo, S.; Xiao, Y.; Wang, X.; Wang, S.; Zukerman, M. Distributed optimization for computation offloading in edge computing. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 8179–8194. [CrossRef]
18. Liu, J.; Huang, J.; Zhou, Y.; Li, X.; Ji, S.; Xiong, H.; Dou, D. From distributed machine learning to federated learning: A survey. *Knowl. Inf. Syst.* **2022**, *64*, 885–917. [CrossRef]
19. Liu, Y.; Zhang, X.; Kang, Y.; Li, L.; Chen, T.; Hong, M.; Yang, Q. Fedbcd: A communication-efficient collaborative learning framework for distributed features. *IEEE Trans. Signal Process.* **2022**, *70*, 4277–4290. [CrossRef]
20. 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]
21. Yin, Q.; Yu, T.; Shen, S.; Yang, J.; Zhao, M.; Ni, W.; Huang, K.; Liang, B.; Wang, L. Distributed Deep Reinforcement Learning: A Survey and a Multi-player Multi-agent Learning Toolbox. *Mach. Intell. Res.* **2024**, *21*, 411–430. [CrossRef]
22. Mariani, S.; Cabri, G.; Zambonelli, F. Coordination of autonomous vehicles: Taxonomy and survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 19. [CrossRef]

23. Khan, M.A.; Sayed, H.E.; Malik, S.; Zia, T.; Khan, J.; Alkaabi, N.; Ignatious, H. Level-5 Autonomous Driving—Are We There Yet? A Review of Research Literature. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 27. [CrossRef]
24. Younes, M.B.; Boukerche, A. Traffic efficiency applications over downtown roads: A new challenge for intelligent connected vehicles. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 102. [CrossRef]
25. Yang, B.; Cao, X.; Xiong, K.; Yuen, C.; Guan, Y.L.; Leng, S.; Qian, L.; Han, Z. Edge intelligence for autonomous driving in 6G wireless system: Design challenges and solutions. *IEEE Wirel. Commun.* **2021**, *28*, 40–47. [CrossRef]
26. Tan, L.; Yu, K.; Lin, L.; Cheng, X.; Srivastava, G.; Lin, J.C.W.; Wei, W. Speech emotion recognition enhanced traffic efficiency solution for autonomous vehicles in a 5G-enabled space–air–ground integrated intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 2830–2842. [CrossRef]
27. Bilik, I.; Longman, O.; Villeval, S.; Tabrikian, J. The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE Signal Process. Mag.* **2019**, *36*, 20–31. [CrossRef]
28. Xia, X.; Bhatt, N.P.; Khajepour, A.; Hashemi, E. Integrated Inertial-LiDAR-Based Map Matching Localization for Varying Environments. *IEEE Trans. Intell. Veh.* **2023**, *8*, 4307–4318. [CrossRef]
29. Pereira, J.; Premebida, C.; Asvadi, A.; Cannata, F.; Garrote, L.; Nunes, U. Test and evaluation of connected and autonomous vehicles in real-world scenarios. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 14–19.
30. Abeywickrama, D.B.; Mamei, M.; Zambonelli, F. Engineering collectives of self-driving vehicles: The SOTA approach. In Proceedings of the International Symposium on Leveraging Applications of Formal Methods, Limassol, Cyprus, 5–9 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 79–93.
31. Bicocchi, N.; Mamei, M.; Sassi, A.; Zambonelli, F. On recommending opportunistic rides. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3328–3338. [CrossRef]
32. Cabri, G.; Montangero, M.; Muzzini, F.; Valente, P. Managing human-driven and autonomous vehicles at smart intersections. In Proceedings of the 2020 IEEE International Conference on Human-Machine Systems (ICHMS), Rome, Italy, 7–9 September 2020; pp. 1–4.
33. Lippi, M.; Mamei, M.; Mariani, S.; Zambonelli, F. An argumentation-based perspective over the social IoT. *IEEE Internet Things J.* **2017**, *5*, 2537–2547. [CrossRef]
34. Ouaissa, M.; Ouaissa, M.; Houmer, M.; El Hamdani, S.; Boulouard, Z. A Secure Vehicle to Everything (V2X) Communication Model for Intelligent Transportation System. In *Computational Intelligence in Recent Communication Networks*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 83–102.
35. Hejazi, H.; Bokor, L. A survey on the use-cases and deployment efforts toward converged internet of things (IoT) and vehicle-to-everything (V2X) environments. *Acta Tech. Jaurinensis* **2022**, *15*, 58–73. [CrossRef]
36. Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering–a systematic literature review. *Inf. Softw. Technol.* **2009**, *51*, 7–15. [CrossRef]
37. Liu, S.; Yu, B.; Tang, J.; Zhu, Q. Towards fully intelligent transportation through infrastructure-vehicle cooperative autonomous driving: Challenges and opportunities. In Proceedings of the IEEE 2021 58th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 5–9 December 2021; pp. 1323–1326.
38. Guo, J.; Kurup, U.; Shah, M. Is it safe to drive? An overview of factors, metrics, and datasets for driveability assessment in autonomous driving. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3135–3151. [CrossRef]
39. Samarakoon, S.; Bennis, M.; Saad, W.; Debbah, M. Distributed federated learning for ultra-reliable low-latency vehicular communications. *IEEE Trans. Commun.* **2019**, *68*, 1146–1159. [CrossRef]
40. Liu, C.; Shu, Y.; Liu, J.; Yang, O.W. Application of extreme value theory to the analysis of wireless network traffic. In Proceedings of the 2007 IEEE International Conference on Communications, Glasgow, UK, 24–28 June 2007; pp. 486–491.
41. Finkenstadt, B.; Rootzén, H. *Extreme Values in Finance, Telecommunications, and the Environment*; CRC Press: Boca Raton, FL, USA, 2003.
42. Hoel, P.G.; Craig, A.; McKean, J.W. *Introduction to Mathematical Statistics*; Pearson: London, UK, 1954.
43. Khattak, Z.H.; Smith, B.L.; Park, H.; Fontaine, M.D. Cooperative lane control application for fully connected and automated vehicles at multilane freeways. *Transp. Res. Part C Emerg. Technol.* **2020**, *111*, 294–317. [CrossRef]
44. Bevly, D.; Cao, X.; Gordon, M.; Ozbilgin, G.; Kari, D.; Nelson, B.; Woodruff, J.; Barth, M.; Murray, C.; Kurt, A.; et al. Lane change and merge maneuvers for connected and automated vehicles: A survey. *IEEE Trans. Intell. Veh.* **2016**, *1*, 105–120. [CrossRef]
45. Rihan, M.; Elwekeil, M.; Yang, Y.; Huang, L.; Xu, C.; Selim, M.M. Deep-VFog: When artificial intelligence meets fog computing in V2X. *IEEE Syst. J.* **2020**, *15*, 3492–3505. [CrossRef]
46. Chen, D.; Hajidavalloo, M.R.; Li, Z.; Chen, K.; Wang, Y.; Jiang, L.; Wang, Y. Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11623–11638. [CrossRef]
47. Ma, X.; Zhao, J.; Liao, J.; Zhang, Z. Intelligent reflecting surface-assisted federated learning in multi-platoon collaborative networks. *Digit. Commun. Netw.* **2023**, *9*, 628–637. [CrossRef]
48. Shi, W.; Zhou, S.; Niu, Z.; Jiang, M.; Geng, L. Joint device scheduling and resource allocation for latency constrained wireless federated learning. *IEEE Trans. Wirel. Commun.* **2020**, *20*, 453–467. [CrossRef]
49. Amiri, M.M.; Gündüz, D.; Kulkarni, S.R.; Poor, H.V. Convergence of update aware device scheduling for federated learning at the wireless edge. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 3643–3658. [CrossRef]

50. Hou, J.; Chen, G.; Huang, J.; Qiao, Y.; Xiong, L.; Wen, F.; Knoll, A.; Jiang, C. Large-Scale Vehicle Platooning: Advances and Challenges in Scheduling and Planning Techniques. *Engineering* **2023**, *28*, 26–48. [CrossRef]
51. 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]
52. Dube, P.; Bhattacharjee, B.; Huo, S.; Watson, P.; Belgodere, B.; Kender, J.R. Automatic labeling of data for transfer learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Vancouver, BC, Canada, 17–24 June 2019; Volume 192255.
53. AbdulRahman, S.; Tout, H.; Ould-Slimane, H.; Mourad, A.; Talhi, C.; Guizani, M. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. *IEEE Internet Things J.* **2020**, *8*, 5476–5497. [CrossRef]
54. Khondkar, U. Development of a Federated Learning Software. Ph.D. Thesis, The University of Texas at Austin, Austin, TX, USA, 2022.
55. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
56. Singh, P.; Singh, M.K.; Singh, R.; Singh, N. Federated learning: Challenges, methods, and future directions. In *Federated Learning for IoT Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 199–214.
57. Ding, J.; Tramel, E.; Sahu, A.K.; Wu, S.; Avestimehr, S.; Zhang, T. Federated learning challenges and opportunities: An outlook. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 7–13 May 2022; pp. 8752–8756.
58. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [CrossRef]
59. Zhao, J.; Chang, X.; Feng, Y.; Liu, C.H.; Liu, N. Participant selection for federated learning with heterogeneous data in intelligent transport system. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1106–1115. [CrossRef]
60. Xie, K.; Zhang, Z.; Li, B.; Kang, J.; Niyato, D.; Xie, S.; Wu, Y. Efficient federated learning with spike neural networks for traffic sign recognition. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9980–9992. [CrossRef]
61. Liu, W.; Chen, L.; Chen, Y.; Zhang, W. Accelerating federated learning via momentum gradient descent. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1754–1766. [CrossRef]
62. Mills, J.; Hu, J.; Min, G. Multi-task federated learning for personalised deep neural networks in edge computing. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *33*, 630–641. [CrossRef]
63. Wang, Z.; Xu, H.; Liu, J.; Huang, H.; Qiao, C.; Zhao, Y. Resource-efficient federated learning with hierarchical aggregation in edge computing. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
64. Wu, W.; He, L.; Lin, W.; Mao, R. Accelerating federated learning over reliability-agnostic clients in mobile edge computing systems. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 1539–1551. [CrossRef]
65. Luo, B.; Li, X.; Wang, S.; Huang, J.; Tassiulas, L. Cost-effective federated learning design. In Proceedings of the IEEE INFOCOM 2021-IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
66. Liu, S.; Fu, Y.; Zhao, P.; Li, F.; Li, C. Autonomous Braking Algorithm for Rear-End Collision via Communication-Efficient Federated Learning. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 01–06.
67. Luo, B.; Li, X.; Wang, S.; Huang, J.; Tassiulas, L. Cost-effective federated learning in mobile edge networks. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3606–3621. [CrossRef]
68. Cao, K.; Liu, Y.; Meng, G.; Sun, Q. An overview on edge computing research. *IEEE Access* **2020**, *8*, 85714–85728. [CrossRef]
69. Zhang, K.; Mao, Y.; Leng, S.; He, Y.; Zhang, Y. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading. *IEEE Veh. Technol. Mag.* **2017**, *12*, 36–44. [CrossRef]
70. Baidya, S.; Ku, Y.J.; Zhao, H.; Zhao, J.; Dey, S. Vehicular and edge computing for emerging connected and autonomous vehicle applications. In Proceedings of the 2020 57th ACM/IEEE Design Automation Conference (DAC), San Francisco, CA, USA, 20–24 July 2020; pp. 1–6.
71. Galanopoulos, A.; Ayala-Romero, J.A.; Leith, D.J.; Iosifidis, G. AutoML for video analytics with edge computing. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
72. Moubayed, A.; Shami, A.; Heidari, P.; Larabi, A.; Brunner, R. Edge-enabled V2X service placement for intelligent transportation systems. *IEEE Trans. Mob. Comput.* **2020**, *20*, 1380–1392. [CrossRef]
73. Zhang, J.; Chen, S.; Wang, X.; Zhu, Y. DeepReserve: Dynamic edge server reservation for connected vehicles with deep reinforcement learning. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
74. Keshavamurthy, P.; Pateromichelakis, E.; Dahlhaus, D.; Zhou, C. Edge cloud-enabled radio resource management for co-operative automated driving. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1515–1530. [CrossRef]
75. Kong, X.; Gao, H.; Shen, G.; Duan, G.; Das, S.K. Fedvcp: A federated-learning-based cooperative positioning scheme for social internet of vehicles. *IEEE Trans. Comput. Soc. Syst.* **2021**, *9*, 197–206. [CrossRef]

76. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC), Shanghai, China, 21–23 May 2019; pp. 1–7.
77. Chi, J.; Xu, S.; Guo, S.; Yu, P.; Qiu, X. Federated Learning Empowered Edge Collaborative Content Caching Mechanism for Internet of Vehicles. In Proceedings of the NOMS 2022—2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2022; pp. 1–5.
78. Mo, X.; Xu, J. Energy-efficient federated edge learning with joint communication and computation design. *J. Commun. Inf. Netw.* **2021**, *6*, 110–124. [CrossRef]
79. Liu, J.; Xu, H.; Wang, L.; Xu, Y.; Qian, C.; Huang, J.; Huang, H. Adaptive asynchronous federated learning in resource-constrained edge computing. *IEEE Trans. Mob. Comput.* **2023**, *22*, 674–690. [CrossRef]
80. Lim, W.Y.B.; Luong, N.C.; Hoang, D.T.; Jiao, Y.; Liang, Y.C.; Yang, Q.; Niyato, D.; Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2031–2063. [CrossRef]
81. Lee, S.; Zheng, X.; Hua, J.; Vikalo, H.; Julien, C. Opportunistic federated learning: An exploration of egocentric collaboration for pervasive computing applications. In Proceedings of the 2021 IEEE International Conference on Pervasive Computing and Communications (PerCom), Washington, DC, USA, 17–21 March 2021; pp. 1–8.
82. Lee, S.; Julien, C.; Zheng, X. Facilitating Decentralized and Opportunistic Learning in Pervasive Computing. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Pisa, Italy, 21–25 March 2022; pp. 144–145.
83. Dube, P.; Suk, T.; Wang, C. AI gauge: Runtime estimation for deep learning in the cloud. In Proceedings of the 2019 31st International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Campo Grande, MS, Brazil, 15–18 October 2019; pp. 160–167.
84. Ormándi, R.; Hegedüs, I.; Jelasity, M. Efficient p2p ensemble learning with linear models on fully distributed data. *arXiv* **2011**, arXiv:1109.1396.
85. Belal, Y.; Bellet, A.; Mokhtar, S.B.; Nitu, V. Pepper: Empowering user-centric recommender systems over gossip learning. *Proc. Acm Interactive Mobile Wearable Ubiquitous Technol.* **2022**, *6*, 101. [CrossRef]
86. Hegedűs, I.; Danner, G.; Jelasity, M. Decentralized learning works: An empirical comparison of gossip learning and federated learning. *J. Parallel Distrib. Comput.* **2021**, *148*, 109–124. [CrossRef]
87. Baraglia, R.; Dazzi, P.; Mordacchini, M.; Ricci, L. A peer-to-peer recommender system for self-emerging user communities based on gossip overlays. *J. Comput. Syst. Sci.* **2013**, *79*, 291–308. [CrossRef]
88. Hegedűs, I.; Danner, G.; Jelasity, M. Gossip learning as a decentralized alternative to federated learning. In Proceedings of the Distributed Applications and Interoperable Systems: 19th IFIP WG 6.1 International Conference, DAIS 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, 17–21 June 2019; Proceedings 19; Springer: Berlin/Heidelberg, Germany, 2019; pp. 74–90.
89. Dinani, M.A.; Holzer, A.; Nguyen, H.; Marsan, M.A.; Rizzo, G. Gossip learning of personalized models for vehicle trajectory prediction. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Nanjing, China, 29 March–1 April 2021; pp. 1–7.
90. Yin, X.; Zhu, Y.; Hu, J. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 131. [CrossRef]
91. Wang, J.; Pal, A.; Yang, Q.; Kant, K.; Zhu, K.; Guo, S. Collaborative machine learning: Schemes, robustness, and privacy. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 9625–9642. [CrossRef]
92. Cheng, K.; Zhou, Y.; Chen, B.; Wang, R.; Bai, Y.; Liu, Y. Guardauto: A decentralized runtime protection system for autonomous driving. *IEEE Trans. Comput.* **2020**, *70*, 1569–1581. [CrossRef]
93. Mohseni, S.; Pitale, M.; Singh, V.; Wang, Z. Practical solutions for machine learning safety in autonomous vehicles. *arXiv* **2019**, arXiv:1912.09630.
94. Gautam, V.; Gheraibia, Y.; Alexander, R.; Hawkins, R.D. Runtime decision making under uncertainty in autonomous vehicles. In Proceedings of the Workshop on Artificial Intelligence Safety (SafeAI 2021), CEUR Workshop Proceedings, Virtual, 8–9 February 2021.
95. Zhu, P.; Zhu, K.; Zhang, L. Security analysis of LTE-V2X and a platooning case study. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 532–537.
96. Ren, K.; Wang, Q.; Wang, C.; Qin, Z.; Lin, X. The security of autonomous driving: Threats, defenses, and future directions. *Proc. IEEE* **2019**, *108*, 357–372. [CrossRef]
97. Prakash, A.; Behl, A.; Ohn-Bar, E.; Chitta, K.; Geiger, A. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 11763–11773.
98. Paardekooper, J.P.; Comi, M.; Grappiolo, C.; Snijders, R.; van Vught, W.; Beekelaar, R. A Hybrid-AI Approach for Competence Assessment of Automated Driving functions. In Proceedings of the SafeAI@ AAAI, Virtual, 8 February 2021.
99. Naghsh, Z.; Valaee, S. Conflict-Free Scheduling in Cellular V2X Communications. *IEEE/ACM Trans. Netw.* **2020**, *29*, 106–119. [CrossRef]

100. Zhang, M.; Dou, Y.; Chong, P.H.J.; Chan, H.C.; Seet, B.C. Fuzzy logic-based resource allocation algorithm for V2X communications in 5G cellular networks. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2501–2513. [CrossRef]
101. Ding, H.; Leung, K.C. Resource allocation for low-latency NOMA-V2X networks using reinforcement learning. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Virtual, 9–12 May 2021; pp. 1–6.
102. Situ, Z.; Ho, I.W.H.; Hou, Y.; Li, P. The feasibility of NOMA in C-V2X. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Toronto, ON, Canada, 6–9 July 2020; pp. 562–567.
103. 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]
104. Wright, K.L.; Steenkiste, P. Meeting connected vehicle application requirements: It's not just about bandwidth. In Proceedings of the 16th ACM Workshop on Mobility in the Evolving Internet Architecture, New Orleans, LO, USA, 25 October 2021; pp. 13–18.
105. Tang, Z.; He, J. NOMA enhanced 5G distributed vehicle to vehicle communication for connected autonomous vehicles. In Proceedings of the ACM MobiArch 2020 the 15th Workshop on Mobility in the Evolving Internet Architecture, London, UK, 21 September 2020; pp. 42–47.
106. Zhou, J.; Tian, D.; Wang, Y.; Sheng, Z.; Duan, X.; Leung, V.C. Reliability-optimal cooperative communication and computing in connected vehicle systems. *IEEE Trans. Mob. Comput.* **2019**, *19*, 1216–1232. [CrossRef]
107. Wang, Z.; Blum, R.S. Algorithms and Analysis for Optimizing the Tracking Performance of Cyber Attacked Sensor-Equipped Connected Vehicle Networks. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 5061–5076. [CrossRef]
108. Hu, S.; Chen, Q.A.; Sun, J.; Feng, Y.; Mao, Z.M.; Liu, H.X. Automated Discovery of {Denial-of-Service} Vulnerabilities in Connected Vehicle Protocols. In Proceedings of the 30th USENIX Security Symposium (USENIX Security 21), Virtual, 11–13 August 2021; pp. 3219–3236.
109. Cao, Y.; Wang, N.; Xiao, C.; Yang, D.; Fang, J.; Yang, R.; Chen, Q.A.; Liu, M.; Li, B. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. In Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 24–27 May 2021; pp. 176–194.
110. Quinonez, R.; Giraldo, J.; Salazar, L.; Bauman, E.; Cardenas, A.; Lin, Z. {SAVIOR}: Securing autonomous vehicles with robust physical invariants. In Proceedings of the 29th USENIX Security Symposium (USENIX Security 20), Online, 12–14 August 2020; pp. 895–912.
111. Geng, J.; Mou, Y.; Li, Q.; Li, F.; Beyan, O.; Decker, S.; Rong, C. Improved Gradient Inversion Attacks and Defenses in Federated Learning. *IEEE Trans. Big Data* 2023, *early access*. [CrossRef]
112. Zhang, J.; Chen, Y.; Li, H. Privacy Leakage of Adversarial Training Models in Federated Learning Systems. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 108–114.
113. Zhu, L.; Liu, Z.; Han, S. Deep leakage from gradients. In Proceedings of the NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
114. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, PMLR, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
115. Zhang, J.; Liu, Y.; Wu, D.; Lou, S.; Chen, B.; Yu, S. VPFL: A verifiable privacy-preserving federated learning scheme for edge computing systems. *Digit. Commun. Netw.* **2023**, *9*, 981–989. [CrossRef]
116. Yang, A.; Ma, Z.; Zhang, C.; Han, Y.; Hu, Z.; Zhang, W.; Huang, X.; Wu, Y. Review on application progress of federated learning model and security hazard protection. *Digit. Commun. Netw.* **2023**, *9*, 146–158. [CrossRef]
117. Zhu, Y.; Hu, D.; Qian, B.; Yu, K.; Liu, T.; Zhou, H. A Stackelberg Game and Federated Learning Assisted Spectrum Sharing Framework for IoV. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–6.
118. Chen, Y.; Sun, X.; Jin, Y. Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 4229–4238. [CrossRef]
119. Liu, Y.; Yuan, X.; Xiong, Z.; Kang, J.; Wang, X.; Niyato, D. Federated learning for 6G communications: Challenges, methods, and future directions. *China Commun.* **2020**, *17*, 105–118. [CrossRef]
120. Luping, W.; Wei, W.; Bo, L. CMFL: Mitigating communication overhead for federated learning. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–9 July 2019; pp. 954–964.
121. Wu, X.; Ji, Z.; Wang, C.L. Embedding Communication for Federated Graph Neural Networks with Privacy Guarantees. In Proceedings of the 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS), Hong Kong, China, 18–21 July 2023; pp. 305–315.
122. Li, Q. Research on Application of Federated Learning of Blockchain Technology System under Computer Networking Technology. In Proceedings of the 2022 IEEE 5th International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 23–25 September 2022; pp. 471–474.
123. Zhu, Y.; Liu, Z.; Wang, P.; Du, C. A dynamic incentive and reputation mechanism for energy-efficient federated learning in 6g. *Digit. Commun. Netw.* **2023**, *9*, 817–826. [CrossRef]
124. Li, Q.; Wang, W.; Zhu, Y.; Ying, Z. BOppCL: Blockchain-Enabled Opportunistic Federated Learning Applied in Intelligent Transportation Systems. *Electronics* **2023**, *13*, 136. [CrossRef]

125. 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]
126. Zhang, L.; Zhang, L.; Shi, S.; Chu, X.; Li, B. Evaluation and Optimization of Gradient Compression for Distributed Deep Learning. *arXiv* **2023**, arXiv:2306.08881.
127. Li, Q.; Gong, B.; Zhu, Y.; Cai, R.; Kong, X. Research on Decentralized Federated Learning System for Vehicle Data Privacy Protection Based on Blockchain. In Proceedings of the 2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA), Changchun, China, 11–13 August 2023; pp. 320–324.
128. Li, Q.; Yi, X.; Nie, j.; Zhu, Y. PR-OppCL: Privacy-Preserving Reputation-Based Opportunistic Federated Learning in Intelligent Transportation System. *IEEE Trans. Veh. Technol.* 2024, *early access*. [CrossRef]
129. Shin, D.D. Blockchain: The emerging technology of digital trust. *Telemat. Inform.* **2019**, *45*, 101278. [CrossRef]
130. Zhang, R.; Xue, R.; Liu, L. Security and privacy on blockchain. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 51. [CrossRef]
131. Zhu, J.; Cao, J.; Saxena, D.; Jiang, S.; Ferradi, H. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Comput. Surv.* **2023**, *55*, 240. [CrossRef]
132. Xu, C.; Ge, J.; Gao, L.; Zhang, M.; Li, Y.; Zhou, W.; Zheng, X. BASS: A Blockchain-Based Asynchronous SignSGD Architecture for Efficient and Secure Federated Learning. *IEEE Trans. Dependable Secur. Comput.* 2024, *early access*. [CrossRef]
133. Xu, C.; Ge, J.; Li, Y.; Deng, Y.; Gao, L.; Zhang, M.; Xiang, Y.; Zheng, X. Scei: A smart-contract driven edge intelligence framework for iot systems. *IEEE Trans. Mob. Comput.* **2023**, *23*, 4453–4466. [CrossRef]
134. De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 3366–3385.
135. Wang, Z.; Zhang, Z.; Lee, C.Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; Pfister, T. Learning to prompt for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 139–149.
136. Zenke, F.; Poole, B.; Ganguli, S. Continual learning through synaptic intelligence. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 3987–3995.
137. Lee, S.; Yu, H.; Zheng, X.; Julien, C. Swarm: Playground for Large-scale Decentralized Learning Simulations. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Pisa, Italy, 21–25 March 2022; pp. 115–117.
138. Beutel, D.J.; Topal, T.; Mathur, A.; Qiu, X.; Fernandez-Marques, J.; Gao, Y.; Sani, L.; Li, K.H.; Parcollet, T.; de Gusmão, P.P.B.; et al. Flower: A friendly federated learning research framework. *arXiv* **2020**, arXiv:2007.14390.
139. Yang, L.; Wen, F.; Cao, J.; Wang, Z. Edgetb: A hybrid testbed for distributed machine learning at the edge with high fidelity. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 2540–2553. [CrossRef]