electronics

Article

BOppCL: Blockchain-Enabled Opportunistic Federated Learning Applied in Intelligent Transportation Systems

Qiong Li [1,2], Wennan Wang [1,3,*], Yizhao Zhu [1] and Zuobin Ying [1]

[1] Faculty of Data Science, City University of Macau, Macau 999078, China; yifeng8457@163.com (Q.L.); zhuyizhao@gmail.com (Y.Z.); zbying@cityu.mo (Z.Y.)
[2] Department of Science and Technology, Hunan Industry Polytechnic, Changsha 410208, China
[3] Department of Finance, School of Economics, Xiamen University, Xiamen 361005, China
* Correspondence: wwwennan@xmu.edu.cn

Abstract: In this paper, we present a novel blockchain-enabled approach to opportunistic federated learning (OppCL) for intelligent transportation systems (ITS). Our approach integrates blockchain with OppCL to streamline the learning of autonomous vehicle models while addressing data privacy and trust challenges. We deploy resilient countermeasures, incentivized mechanisms, and a secure gradient distribution to combat single-point failure verification attacks. Additionally, we integrate the Byzantine fault-tolerant algorithm (BFT) into the node verification component of the delegated proof of stake (DPoS) to minimize verification delays. We validate our approach through experiments on the MNIST, SVHN, and CIFAR-10 datasets, showing convergence rates and prediction accuracy comparable to traditional OppCL approaches.

Keywords: blockchain; opportunistic federated learning; BOppCL; intelligent transportation system

check for updates

Citation: Li, Q.; Wang, W.; Zhu, Y.; Ying, Z. BOppCL: Blockchain-Enabled Opportunistic Federated Learning Applied in Intelligent Transportation Systems. *Electronics* **2024**, *13*, 136. https://doi.org/10.3390/electronics13010136

Academic Editor: Aryya Gangopadhyay

Received: 2 December 2023
Revised: 22 December 2023
Accepted: 27 December 2023
Published: 28 December 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Rapid urbanization in various countries has caused a growing number of challenges related to traffic congestion and road accidents [1,2]. To address these issues, considerable attention has been paid to smart cities and ITS [3–5]. In the ITS realm, the integration of opportunistic federated learning (OppCL) into autonomous vehicles has gained significant momentum for model learning [6–8]. ITS, particularly through the vehicle-to-everything (V2X) approach, has enabled intelligent traffic management, dynamic information services, and autonomous vehicle control [9–11]. In this context, data sharing and mutual learning among vehicles in vehicular networks have emerged as crucial factors in updating vehicle models and enhancing the driving experience [12,13]. However, in the dynamic ITS environment, where vehicles often encounter opportunities, the challenge lies in incentivizing vehicle participation and efficiently completing learning tasks within their limited encounter time, which remains a prominent research focus.

OppCL [14] provided a distributed decentralized security method for the sharing of data between autonomous vehicles in ITS. By storing local data at vehicle nodes, vehicles exchange gradients with other vehicles based on opportunistic encounters and train local models, privacy issues are solved much more, and data transmission costs are reduced. However, OppCL algorithms lack incentive mechanisms to encourage vehicle nodes involved in learning. Introducing incentive mechanisms into OppCL can quickly improve its ability to collect information. To provide incentive mechanisms for data sharing, encourage nodes to join distributed learning, and improve overall performance, blockchain technology is a good choice. The blockchain [15–17] has been widely used in fields such as cryptocurrency and secure storage. Researchers from both inside and outside the country [18–20] have utilized blockchain technology to replace the central server of federated learning. Its distributed storage capabilities guarantee the consistency of model parameters between the various nodes in federated learning [21] with no restrictions.

However, in the ITS scenario, a combination of blockchain and OppCL in the Internet of Vehicles (IoV) faces new problems [22–24] due to the mobility of vehicles and the opportunity to encounter them. The number of vehicles on the road has increased, and the amount of network bandwidth available in the IoV is limited. This has made communication efficiency a major obstacle to large-scale data exchange in ITS [25–27]. In this case, this article faces mainly three challenges: firstly, the additional computing and communication overhead generated by blockchain puts significant communication pressure on the system; secondly, due to the opportunity encountered by vehicles, they are unfamiliar and lack the motivation to participate in learning; finally, due to the slackness of other devices in intelligent vehicles and ITS, the accuracy is reduced and the overall performance of the system is limited. The purpose of our research is to tackle the difficulties mentioned above, and we suggest a blockchain-based opportunistic federated learning (BOppCL) approach for ITS that guarantees data security while allowing for effective distributed data sharing.

Our contributions. This article presents a new BOppCL approach that focuses on data privacy and facilitates effective distributed data sharing. The key research contributions can be summarized as follows.

(1) Blockchain-enhanced Opportunistic Federated Learning (BOppCL): We propose BOppCL, a novel approach to ITS. BOppCL addresses multiple challenges faced by traditional OppCL approaches. It addresses gradient verification attacks, integrates incentive mechanisms, and ensures a secure gradient distribution. Compared to traditional OppCL methods, BOppCL surpasses their capabilities.
(2) Introduction of Byzantine Fault-Tolerant Algorithm (BFT): In our approach, we introduce the BFT algorithm to improve node verification in the delegated proof-of-stake (DPoS) mechanism. This integration reduces verification delays and introduces penalty mechanisms to assess the quality of node production blocks. As a result, the overall performance of the system is improved.
(3) Experimental validation: We designed and conducted experiments using the MNIST and SVHN datasets. The results demonstrate that our approach achieves a convergence speed and prediction accuracy comparable to those of traditional OppCL methods. Furthermore, the experiments validate that our algorithmic improvements lead to improved efficiency, reducing the time and communication costs required to achieve consensus.

2. Related Work

In ITS, collaborative environmental data detection, calculation, and processing play a crucial role [28]. To overcome computational and storage limitations in the IoV, data sharing between parties in distributed scenarios has emerged as an effective solution. Federated learning [29,30] and OppCL [14] offer robust technical support for IoV data exchange within ITS. Zhao et al. [31] introduced the federated learning framework to the IoV environment, combining it with local differential privacy (LDP) to address privacy concerns and reduce communication costs between vehicles. They proposed the LDP-FedSGD algorithm, which incorporates four differential privacy mechanisms to disrupt the gradient of local model output and a three-output mechanism for privacy budgeting, using two-bit encoding to minimize communication costs. Chen et al. [32] proposed ASTW-FedAVG, a temporally weighted asynchronous federated learning, to reduce communication between nodes and the central server. This approach incorporates a time-weighted aggregation strategy on the central server to enhance the accuracy and convergence of the central model. However, this requires a centralized management server, increasing the risk of a single point of failure of the system.

To address the risk of single point failure, Lee et al. [14] proposed the OppCL method to facilitate autonomous vehicles learning from opportunistic encounters. This method has strong resilience to overfitting and severe fluctuations in the data distribution encountered, but opportunities encountered by vehicles without motivation to

participate in learning. Lu et al. [20] extended blockchain technology to the distributed data sharing architecture of IoV and designed a hybrid blockchain architecture consisting of licensed blockchains and locally directed acyclic graphs to improve the security and reliability of model parameters. Furthermore, Lu et al. [20] suggested an asynchronous federated learning approach that applies deep reinforcement learning for node selection to improve efficiency. Unfortunately, this scheme did not adopt effective methods to improve communication efficiency in asynchronous transmission of federated learning parameters. Similarly, Chai et al. [33] proposed a hierarchical blockchain-enabled federated learning (HBFL) algorithm for data sharing in the IoV. This algorithm allows for the sharing of knowledge in the form of learning parameters during the federated learning process. It also groups vehicles and infrastructure based on their regional characteristics, and maintains exclusive blockchain ledger records in the federated learning model. Meanwhile, Chai et al. [33] also proposed a lightweight consensus mechanism called proof of knowledge (PoK), which models the knowledge-sharing process as a non-cooperative game of multiple leaders and multiple people in the trading market. Compared to traditional blockchain frameworks, although this algorithm fully considers the issue of computational cost, it ignores the communication cost caused by parameter sharing in federated learning. Pokhrel et al. [34] developed a mathematical framework that incorporated blockchain parameters (e.g., retransmission limit, block size, block arrival rate, and frame size) with federated learning-based update reward methods to address communication efficiency issues. Through a thorough analysis and quantification of the end-to-end delay, the optimal block arrival rate was determined to reduce the system delay. This framework ignores the latency issue caused by synchronous aggregation in federated learning, which affects the overall operational latency of the system.

Blockchain-based federated learning (BCFL) [35] is seen as a novel data-sharing model in edge networks of the Internet of Things (IoT), due to its decentralization, collaborative model training, and privacy protection benefits. When comparing existing research schemes in the context of ITS, it becomes clear that many of the data-sharing approaches overlook the critical aspects of user incentives and consensus algorithms and their impact on system efficiency [36]. Additionally, the aggregation methods used in these schemes play an important role in determining the effectiveness and efficiency of the sharing mechanisms. These findings highlight the need for a more comprehensive and holistic approach. To fill this gap, our paper proposes a novel blockchain-based OppCL algorithm tailored specifically for ITS. By integrating blockchain technology [37] and OppCL, our algorithm addresses the limitations mentioned above and provides an innovative solution to share data efficiently and securely in ITS scenarios. The use of blockchain ensures data integrity, transparency, and accountability, while the OppCL framework allows for collaborative learning between distributed vehicles, using their collective intelligence. Taking into account user incentives, consensus algorithms, and aggregation methods, our algorithm aims to optimize the efficiency and effectiveness of the system in data sharing. It offers a robust framework that encourages active participation of vehicles, facilitates consensus among distributed nodes, and employs efficient aggregation methods to minimize computational and communication costs. In summary, our proposed blockchain-based OppCL algorithm fills the existing gaps in data-sharing schemes for ITS. Taking into account user incentives, consensus algorithms, and aggregation methods, we are trying to improve the efficiency and effectiveness of data-sharing systems in ITS scenarios; a comparison is provided in Table 1.

Table 1. Comparison between Our BOppCL with related works.

Algorithms	Single Point of Failure	Incentive Mechanism	Sharing Efficiency	Communication Cost	System Runtime
LDP-FedSGD [31]	✗	✗	✗	✓	✓
ASTW-FedAVG [32]	✗	✗	✓	✓	✓
OppCL [14]	✓	✗	✗	✓	✗
PermiDAG [20]	✓	✓	✗	✗	✗
ADMM [33]	✓	✓	✓	✗	✗
BFL [34]	✓	✓	✗	✓	✓
Ours	✓	✓	✓	✓	✓

Note: ✓ indicates that the method supports solving the problem, while ✗ indicates that the method needs further exploration in this area.

3. System Model

3.1. Notations

Table 2 illustrates the notation used in this paper.

Table 2. Notations.

Notations	Description
v_i	ith vehicle
$\omega_i^{t_i}$	Model of the ith vehicle
t_i	Local clock of the ith vehicle
$v_{i,j}$	The ith vehicle within the range of the jth RSU
$r_{j,n}$	The jth RSU within the range of the nth base station
$\mathsf{rate}_{v_{i,j}}$	Communication transmission rate of vehicle $v_{i,j}$
$T_{v_{i,j}}^{\mathsf{cmp}}$	Local calculation time of vehicle $v_{i,j}$
$T_{v_{i,j}}^{\mathsf{com}}$	Model parameter transmission time of vehicle $v_{i,j}$
b_n	The nth base station in the blockchain

3.2. Overview of System Model

Figure 1 illustrates our BOppCL framework. This system consists of three main parts: the vehicle encounter group, the RSU and base station, and the blockchain, which are elaborated below.

- Vehicle encounter group (VEG): A VEG refers to a group of vehicles $\{v_1, \ldots, v_n\}$ that opportunistically meet on the road. Each v_i in VEG is equipped with an intelligent onboard system responsible for the real-time processing of vehicle data and data fusion by multisensors. This system ensures that the vehicle can maintain stable and safe driving in various complex situations.
- RSU and base station (RSU): An RSU refers to a mobile edge computing server that possesses specific edge computing and communication capabilities. The RSU serves as a base station with high computing and communication capabilities, forming an alliance chain.
- Blockchain: The blockchain serves as the platform for data storage and exchange in our system. RSU is responsible for recording the collected data in the blockchain ledger, constructing the block, and subsequently uploading them to the chain. This decentralized process based on blockchain ensures data integrity and opportunistically facilitates collaborative learning.

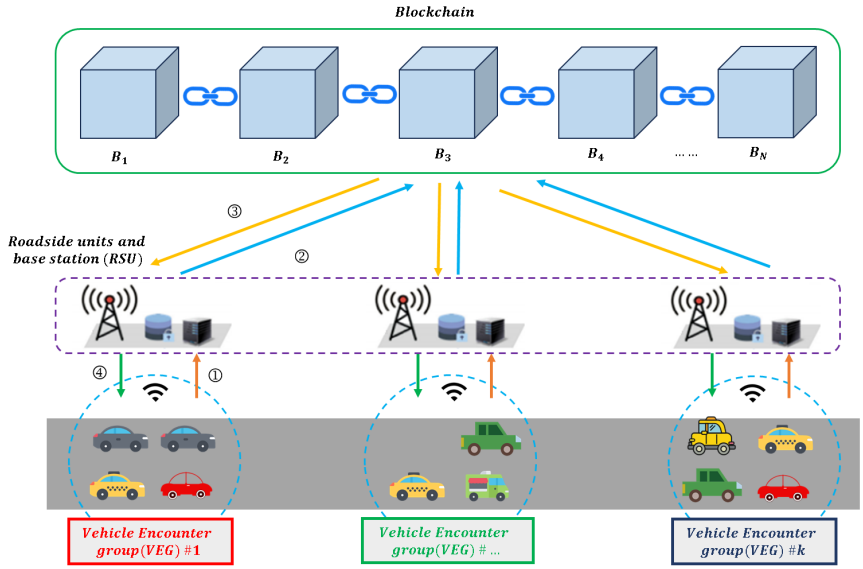

Figure 1. The system model of our BOppCL in ITS.

3.3. System Workflow

The system architecture proposed in this article divides the system workflow into four steps: the local calculation process of a vehicle encounter group, the RSU and base station learning process, the alliance chain learning process, and the update sending process. The system workflow is illustrated in the following.

① Vehicles are dynamically grouped into encounter groups (such as the encounter group #1, #2, #..., #*k*, etc.), using the automatic composition of opportunities. The selection of vehicles (v_i) within an encounter group is based on blockchain and communication resources. The selected vehicle exchanges gradients with other vehicles ($v_{i,j}$) and sends them to the RSU ($r_{j,n}$). The model parameters ($\omega_i^{t_i}$) after each round of training are uploaded to nearby RSUs ($r_{j,n}$) through wireless networks.

② The RSU ($r_{j,n}$) receives model parameters ($\omega_i^{t_i}$) from all participating vehicles and performs global aggregation. The aggregated new model parameters are initiated by consensus by the main accounting node in a transactional manner. The results of consensus among multiple RSUs ($r_{j,n}$) are recorded on the blockchain and sent by the main accounting node to adjacent base stations (b_n).

③ After receiving the model parameters and the calculation results sent by RSUs ($r_{j,n}$), the base station stores (b_n) them locally and aggregates all received parameters.

④ The base station (b_n) that has obtained accounting rights initiates a consensus and sends the agreed result as a global model parameter to the master nodes of each RSU ($r_{j,n}$). The RSU ($r_{j,n}$) master node updates the global model parameters and sends them to the vehicle ($v_{i,j}$) nodes in their respective regions.

3.4. Opportunistic Federated Learning

OppCL distinguishes itself from traditional federated learning by eliminating the need to collect participants' data (gradients) on a central server for training the global model. Instead, each participant trains their local model independently using the gradients exchanged during opportunistic encounters. This approach ensures better data privacy, as data remain localized, and minimizes the risks associated with centralized data storage.

OppCL optimizes the local model, denoted as $\mathcal{G}_i$, by combining the local device data (e.g., autonomous vehicle local data) with the gradient encountered from nearby opportune encounters (ie, other autonomous vehicles present in the surrounding area). This approach

updates the local model to better align with the target distribution. The process can be formally described as follows.

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

The set of encounters E_i owned by the device v_i (which is equivalent to an autonomous vehicle in ITS) is denoted by E_i. $\mathcal{D}_{\mathcal{G}_i}$ is a hypothetical dataset whose data label distribution meets the desired distribution $\mathcal{G}_i$. The model of v_i is represented by $\omega_i^{t_i}$, where t_i is the local clock of v_i.

Greedy aggregation directly averages the gradient of neighboring learning (autonomous vehicles that encounter opportunities around them) after adding a round of local learning.

$$\omega^{'} = \frac{\omega_{(\mathcal{L}_i,\mathcal{G}_i)}(\bigtriangledown\ell(\omega^{'};\mathcal{D}_i)) + \omega_{(\mathcal{L}_j,\mathcal{G}_j)}(\bigtriangledown\ell(\omega^{'};\mathcal{D}_j))}{\omega_{(\mathcal{L}_i,\mathcal{G}_i)} + \omega_{(\mathcal{L}_j,\mathcal{G}_j)}}$$

Both gradients are weighted on the basis of the similarity between the label distribution and the target distribution, so the weight can be used for processing. With weights as follows:

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

The value of λ indicates that the model is likely to be overfitted when the total number of labels in the dataset is small. The similarity is determined by the following equation:

$$sim(P_1,P_2) = \sum_{l\in L}\min(P_1(l),P_2(l))$$

At each time t_i, the learning rate of each device is determined independently. The rate, denoted as v_i, is calculated as $\eta_i^{t_i} = \eta\alpha_i^{t_i}$, where η is the initial learning rate and $\alpha_i^{t_i}$ is the attenuation factor.

$$\alpha_{t_i} = \frac{\exp(k \times (\phi - \parallel \omega_i^0 - \omega_i^{t_i} \parallel_2))}{\exp(k \times (\phi - \parallel \omega_i^0 - \omega_i^{t_i} \parallel_2)) + 1}$$

where

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

The constants ϕ and k are used to define the attenuation factor α, which is a sigmoid function that takes the L2 distance from the initial weight as its input. This design encourages the v_i's model to search for solutions close to the guided model.

The OppCL framework enables distributed learning in ITS, allowing vehicles to maintain their optimal learning models and make intelligent decisions in various scenarios. However, it faces challenges in selecting trustworthy vehicles from within the encounter group and motivating them to participate in gradient sharing. We suggest a blockchain-based solution to address these problems, which would allow for the selection of vehicles and the distribution of communication resources within the encounter group. This blockchain-based method improves trustworthiness and incentivizes active participation in gradient sharing, improving the overall effectiveness of the original OppCL framework.

3.5. Vehicle Selection and Communication Resource Allocation Algorithm

The Figure 1 shows the connection between the components of our blockchain-based federated learning system. Our system replaces conventional global servers in federated learning with a blockchain network. The transmission rate of vehicle data $v_{i,j}$ refers to the transmission rate at which this specific vehicle transmits the data. It represents the speed

or efficiency with which data are sent from vehicle $v_{i,j}$ to other components of the system. The vehicle data transmission rate $v_{i,j}$ is:

$$R_{i,j} = \sum_{j=1}^{c_0} \theta_{m,i} \mathsf{rate}_{v_{i,j}}$$

where j is the jth roadside unit. c_0 is the number of channels available and $\mathsf{rate}_{v_{i,j}}$ is the data transmission rate achievable on the vehicle uplink $v_{i,j}$. $\theta_{m,i} \in \{0,1\}$ represents whether the current subchannel is assigned to vehicle $v_{i,j}$, $\theta_{m,i} = 1$ represents whether the current subchannel is assigned to the vehicle, and $\theta_{m,i} = 0$ represents whether the current subchannel is not assigned to the vehicle. For vehicle $v_{i,j}$, the expected average execution time of the algorithm in the t-th iteration is:

$$T_{\mathsf{exec}}(t) = \frac{1}{N} \sum_{i=1}^{N} (T_{v_{i,j}}^{\mathsf{cmp}}(t) + T_{v_{i,j}}^{\mathsf{com}}(t))$$

The optimization problem can be expressed as follows, taking into account the local calculation time of the vehicle $v_{i,j}$ ($T_{v_{i,j}}^{\mathsf{cmp}}$) and the transmission time of the vehicle model parameter $v_{i,j}$ ($T_{v_{i,j}}^{\mathsf{com}}$):

$$\min_{\mu,\theta} \sum_{i=1}^{N} \lambda_i (T_i^{\mathsf{com}}(\theta, t) + T_i^{\mathsf{cmp}} - T_{exec})^2$$

$$\text{s.t. } \mu_i, \theta_{m,i} \in \{0,1\}, \forall i \in N$$

$$\sum_{i \in N, m \in c_0} \theta_{m,i} \leq c_0$$

where μ_i represents whether the vehicle participated in this OppCL, $\mu_i = 1$ represents yes, and $\mu_i = 0$ represents no. The algorithm for vehicle selection and allocation of communication resources is shown in Algorithm 1.

Algorithm 1: The vehicle selection and communication resource allocation

Input : Candidate vehicle set $V = \{v_{0,j}, v_{1,j}, \ldots, v_{i,j}\}$, subchannel set $\Theta = \{\theta_{0,j}, \theta_{1,j}, \ldots, \theta_{m,j}\}$
Output: Solution set

```
for Vehicle set C do
    Random selection i vehicles
    for each vehicles do
        Allocate subchannels
    end
end
Obtain a solution size of S_I^i
Set the binary digit l to meet 2^l ≥ S_I^i
Set the selection rate P_i, the hybridization rate P_c, the variation rate P_h, and the
  number of cycles T
Initialize the original solution set P_0(V_0, Θ_0)
for each solution set do
    Calculate the fitness function of the set of solutions according to Equation (1)
    Generate new subsets based on cross-binary encoding of hybridization rate
    Generate a new subset based on the variation rate and partial binary encoding
    Update solution set
end
```

3.6. Blockchain-Enabled OppCL

For OppCL, autonomous vehicles have opportunities to exchange gradients with each other and update local models. Vehicles will form groups of vehicles on the road, which we refer to as encounter groups. The vehicles in the encounter group are updated with the local model through label recognition and the exchange of gradients based on OppCL. Gradients and their associated labels are then uploaded to the blockchain through roadside units (RSU) [38]. Due to the limited access of the alliance chain, a user must be preapproved before they can join. The PPFL chain utilizes OppCL instead of a parameterized server to store both local and global data, and the system is managed by all members. If one member leaves after a transaction, it will not affect the other members, thus improving the system's disaster recovery capability. Furthermore, blockchain transparency, traceability, and non-repudiation are used to record the reputation of each participant, making the prepayment-based voting mechanism more open, transparent, and reliable [39].

The use of blockchain and OppCL not only solved the problem of crowd-sourced spatial learning in ITS, but also increased the protection of data privacy security. OppCL is used to select reliable task recipients (such as workers) based on opportunistic encounters and dynamically select consensus algorithms, block sizes, block generation times, and block generation nodes for blockchain to ensure data privacy and security. It can reduce the number of nodes in the block directly managed by access control and alleviate the burden of access control [40,41].

Constructing a reliable network of partners to share data on a blockchain will necessitate overcoming a few obstacles. One is the need for a governance system to decide the rules of the system, such as who can be invited to join the network, what data are shared, how they are encrypted, who has access, how conflicts will be settled, and what the scope is for the use of IoT and smart contracts. Another challenge is to figure out how to address the effect that blockchain could have on pricing and inventory allocation decisions by making information about the quantity or age of products in the supply chain more transparent. It is difficult to predict where in the supply chain the costs and benefits of this transparency will be felt. Our solution is simpler consensus protocols. If a blockchain is permissioned and private, the proof-of-work method is not necessary to establish consensus. Simpler methods can be employed to determine who has the right to add the next block to the blockchain. One such method is a round-robin protocol, where the right to add a block rotates among the participants in a fixed order. Since all participants are known, a malicious actor would be detected if it used its turn to modify the chain in a harmful or illegitimate way. And disputes can be easily resolved by verifying the previous blocks of the participants.

Since its introduction in 2016, federated learning has been widely regarded as a significant approach to addressing privacy and security issues in machine learning and intelligent transportation systems [42]. Nevertheless, numerous studies have revealed that shared gradients can also expose local data, making the PPFL method a highly demanding research area. To provide a guaranteed collaborative solution for efficient sharing, we use a new architecture blockchain-based OppCL to protect the privacy and security of ITS, and greatly improve the efficiency of OppCL by choosing to participate at nodes to minimize overall costs. OppCL can be used to update vehicle models locally through the exchange of gradients when the opportunity arises, reducing the risk of data being exposed or mishandled, and ensuring the security and accuracy of the information. It can also solve some sensitive problems by dynamically selecting and controlling block nodes. Leakage of confidential data included in tasks during their assignment and allocation can improve system performance and efficiency [41]. In the future, the combination of OppCL and Blockchain in ITS will be a significant research focus in terms of incentive structures, access protocols, and scalability [40,42].

Exploring the scalability, cost, security, performance, and robustness of the blockchain-enabled OppCL approach is essential in order to understand its potential limitations and

challenges. This includes assessing the scalability of the consensus algorithm, the cost of transactions, and the security of the system. Additionally, the performance of the system can be evaluated by measuring the latency of transactions and the throughput of the system. Finally, the robustness of the system can be tested by simulating different types of attacks and evaluating the system's ability to recover from them. The blockchain-enabled OppCL approach is advantageous in terms of operational efficiency and confirmation delays, as it allows for faster and more secure transactions. The blockchain technology provides a distributed ledger system that is immutable and secure, meaning that transactions are recorded and stored in a secure and transparent manner. Furthermore, the confirmation delays are reduced due to the distributed nature of the blockchain, as transactions are verified and confirmed by multiple nodes in the network, rather than relying on a single centralized authority.

3.7. Incentive Mechanism

The transaction process of the mechanism proposed in this article is shown in Figure 2.

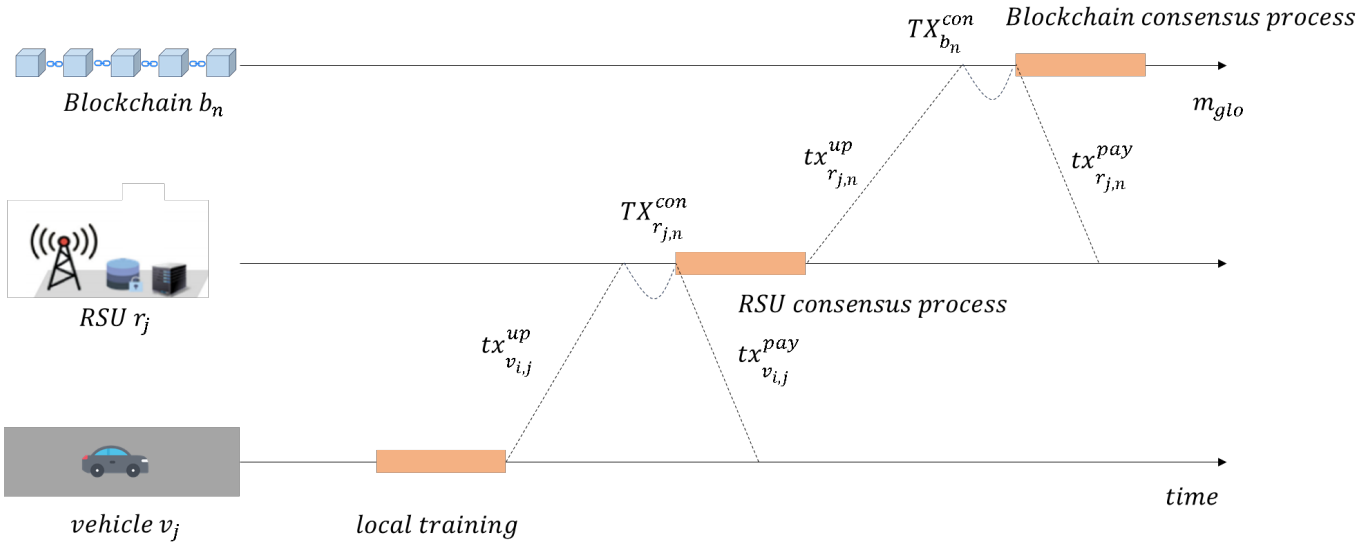

Figure 2. The transaction process of the mechanism proposed in this article.

All vehicles participating in the learning process $v_{i,j}$ obtain the model parameter $\omega_{i,j}$ and the loss function decrease ratio of the training results $\varepsilon_{i,j}$ by calculating the local dataset $D_{i,j}$; each vehicle packages the above parameters into a transaction format and sends them to adjacent RSU:

$$\mathrm{tx}_{v_{i,j}}^{\mathrm{up}} = \{\mathrm{Addr}_{v_{i,j}} \mid 0 \mid \omega_{i,j} \mid \varepsilon_{i,j} \mid \mathrm{Addr}_{r_{j,n}} \mid \mathrm{SIG}_{v_{i,j}}\}$$

The second item in the transaction is 0, which means that the transaction is a trained model parameter uploaded by the vehicle. After receiving the transaction, the roadside unit $r_{j,n}$ first checks the authenticity of the transaction, then extracts the parameters from the transaction to prepare for the subsequent aggregation process, and returns a reward to the vehicle. This article proposes a corresponding incentive mechanism to address the problem of slack in vehicles. For example: all uploaded vehicle results are sorted in reverse order according to the proportion of reduced loss function values. Vehicles are expected to contribute more computing resources to the learning process to achieve the fastest system convergence. For a sorted queue, assuming that the transactions uploaded by $v_{i,j}$ rank nth, the reward they receive is:

$$p_{i,j} = \mu \frac{p_{j,n}}{2In}$$

where I represents the number of all vehicles participating in the learning process; $p_{j,n}$ is the local reward for RSU, with an initial value set at 1. The transactions in which RSU return rewards to the corresponding vehicles are as follows:

$$\mathsf{tx}_{v_{i,j}}^{\mathsf{pay}} = \{\mathsf{Addr}_{r_{j,n}} \mid p_{i,j} \mid 0 \mid 0 \mid \mathsf{Addr}_{v_{i,j}} \mid \mathsf{SIG}_{r_{j,n}}\}$$

In the next consensus process, the roadside unit is responsible for forging new blocks. This article aims to improve communication efficiency, so hashing is used to replace the parameter content in the original transaction during the consensus process.

Because the gradient of OppCL is huge compared to the median of general block transactions, commonly used datasets such as the MNIST dataset update around 1 MB of parameters each time. Therefore, in this article, the method to record model parameters in block transactions is to record its hash value. When the smart contract verifies the transaction, it needs to query the Inter-Planetary File System (IPFS) to obtain the off-chain value. At this point, the transaction format included in the block initiated by the roadside unit during the consensus stage is the following.

$$\mathsf{TX}_{r_{j,n}}^{\mathsf{con}} = \{\mathsf{Addr}_{r_{j,n}} \mid \mathsf{TXID} \mid H(\omega_{j,n}) \mid \delta_{j,n} \mid p_{j,n} \mid \mathsf{SIG}_{r_{j,n}}\}$$

where $\mathsf{Addr}_{r_{j,n}}$ is the confirmation of the identity of the block initiator, $p_{j,n}$ is the reward for this transaction after reaching consensus, $H(\omega_{j,n})$ is the hash value of the model, and $\delta_{j,n}$ is the accuracy of the model. After consensus is passed, the main accounting node uploads the new transaction to the neighboring base station:

$$\mathsf{tx}_{r_{j,n}}^{\mathsf{up}} = \{\mathsf{Addr}_{r_{j,n}} \mid 0 \mid \omega_{j,n} \mid \mathsf{Addr}_{b_n} \mid \mathsf{SIG}_{r_{j,n}}\}$$

In the setting of this article, both the RSU and base stations will not slack off, so the reward returned by the base station is the following:

$$p_{j,n}(t) = p_{j,n}(t-s) + \frac{p_n}{\mid J \mid}$$

where $p_{j,n}(t-s)$ is the value of $p_{j,n}$ in the previous iteration process, p_n is the local reward for b_n, and $\mid J \mid$ is the total number of RSUs participating in learning in the region b_n. Similarly to the previous process, the base station will return rewards to the RSU:

$$\mathsf{tx}_{r_{j,n}}^{\mathsf{pay}} = \{\mathsf{Addr}_{b_n} \mid p_{j,n} \mid 0 \mid 0 \mid \mathsf{Addr}_{r_{j,n}} \mid \mathsf{SIG}_{b_n}\}$$

After aggregating all parameters, the base station starts forging blocks and initiating consensus. At this time, the transaction is:

$$\mathsf{TX}_{b_n}^{\mathsf{con}} = \{\mathsf{Addr}_{b_n} \mid \mathsf{TXID} \mid H(\omega_n) \mid \delta_n \mid p_n \mid \mathsf{SIG}_{b_n}\}$$

After reaching a consensus, all base stations obtain a new global model ω_{glo} and distribute the global model. The framework proposed in this article combines blockchain and OppCL techniques to address privacy issues. The OppCL method replaces traditional data upload methods with gradient upload mechanisms, effectively protecting the privacy of participants. Additionally, blockchain technology uses asymmetric encryption technology and digital signature technology to replace gradients with hash values, further protecting user privacy.

3.8. BFT-DPoS Consensus Mechanism

In the blockchain framework proposed in this article, the distribution of gradients and labels is achieved through consensus. This article uses multiple RSU to aggregate gradients within encounter groups and uses blockchain to synchronize these gradients, achieving consensus between different RSU and base stations. Due to the need for global models to

be confirmed as blockchain transactions, the operational efficiency of blockchain is crucial for the entire learning process.

The traditional consensus mechanism, proof of work (PoW) [43], uses hash puzzles to determine the publisher of candidate blocks. However, this method has a lower throughput and a longer confirmation delay. To address this, the DPoS mechanism [44] was developed. This algorithm selects certain special nodes to proxy the remaining nodes in the network, reducing the number of nodes participating in block production and verification. This can meet the throughput requirements of the public chain and reduce the confirmation delay. To further improve the system performance, this article introduces the BFT algorithm [45,46] into the node verification part of DPoS. This can reduce verification delay and introduce a penalty mechanism to evaluate the quality of node production blocks. By introducing an additional layer of BFT, the DPoS consensus mechanism can ensure a robust and efficient blockchain with low consensus latency [47].

The potential limitations and challenges of integrating the BFT into the node verification component of the DPoS include: The BFT algorithm requires a large number of nodes to be present in order to be effective, which can be difficult to achieve in a DPoS system. The BFT algorithm is computationally expensive, which can lead to increased transaction costs. The BFT algorithm is vulnerable to Sybil attacks, which can lead to a decrease in the security of the system. The BFT algorithm is vulnerable to network latency, which can lead to a decrease in the performance of the system. The BFT algorithm is vulnerable to malicious actors, which can lead to a decrease in the trustworthiness of the system. The BFT algorithm is vulnerable to forks, which can lead to a decrease in the stability of the system.

3.8.1. Node Type

This article discusses a model of witness election that includes four distinct roles for the nodes: ordinary, candidate, witness, and candidate witness.

The majority of nodes in the system are ordinary nodes, which are granted voting rights and the ability to be elected. From these ordinary nodes, candidate nodes are chosen. These candidate nodes are divided into two categories: witness nodes and candidate witness nodes. Witness nodes are responsible for generating blocks, whereas alternative witness nodes are responsible for verifying the blocks created by the witness nodes and replacing any inefficient witness nodes.

3.8.2. Election Mechanism

In this article's solution, all participating RSU act as blockchain users. Blockchain users vote to choose the preferred roadside unit as the validator based on its computing and communication capabilities. During the voting and election stage, the shareholding nodes will use their shares as the number of votes to vote for the supporting nodes through affirmative voting, and each node is allowed to cast one vote for other nodes. After the vote is completed, the system calculates the number of valid votes for all nodes and selects the top two TN nodes with valid votes as candidate witness nodes (TN is the number of witness nodes that the system considers sufficient for decentralization through at least 50% voting shareholding nodes), and divides them into two groups, with the top TN nodes with the highest number of votes as the set of witness nodes $A_1 = \{x_0, x_1, \ldots, x_{TN-1}\}$ for this round, and the other group as the set of alternative witness nodes $A_2 = \{x_{TN}, x_{TN+1}, \ldots, x_{2TN-1}\}$.

3.8.3. Witness Node Block Out

After each witness node is sorted, the blocks are produced in sequence within the specified time interval. If production is not successful, the witness is skipped, and the next witness continues to forge the blocks. This can effectively avoid system latency issues caused by witness block errors.

3.8.4. Block Validation

In response to the issue of long verification delays after block generation by nodes in DPoS algorithms, the BFT algorithm [48] has been introduced to quickly verify the blocks generated by witness nodes. This new mechanism can finish block verification in a much shorter time, significantly decreasing the confirmation time of transactions. In the original DPoS protocol, witness nodes are randomly selected and arranged in a sequence. The blocks are then generated in this order within a given time frame. The newly created blocks are passed on to the subsequent witness nodes for verification, along with the sequence of witness nodes. For a block to be added to the blockchain, it must receive confirmation from two-thirds of the total number of witness nodes, which can significantly extend the verification time. As it improves the latency of block verification and optimizes the utilization of the chosen set of candidate witness nodes A_2, this article verifies the blocks generated by A_1 through the nodes in the set A_2 immediately.

The DPoS-based witness election model generates two sets of nodes: the witness node set A_1 and candidate witness node set A_2. The main function of node A_1 is to package the transactions generated in the network and produce blocks, and node A_2 acts as a candidate node to run the BFT algorithm. The blocks produced by node A_1 are immediately transmitted to the nodes in the set A_2 for verification, thus completing the block verification work faster and reducing the latency of intrablock transactions. Additionally, during the block verification process, each base station sends its aggregated model to other verification nodes for verification. In addition to regular validation, the validator also verifies the received model based on whether the model has made a positive contribution to updating the last global model. The main validation node collects the validation results from all validation nodes and confirms transactions. The approved block is added to the blockchain and transmitted to other base stations for storage.

The witness node broadcasts the generated blocks to a group of potential candidates. After receiving the block data, the primary node in the set of alternative witnesses will package the information and affix its signature. The selection of the primary node is based on the following criteria:

$$P = c \ mod \ |A_2|$$

The BFT view number is denoted by c, and the number of potential witness nodes is represented by A_2. The main node sends the signed and encapsulated message to the other nodes in A_2. When other alternative witnesses receive the block message, verification is required, and the verification rules are as follows.

(1) Is the signature correct?
(2) Is the view number in the message consistent with the view number of the node?
(3) Has the block message been received?
(4) Is the block height in the message consistent with the block height of the node?
(5) Has the model made a positive contribution to the last global model?

Only block messages that meet the above conditions will be recognized by the candidate witness node. When the number of confirmed messages reaches $\frac{2TN}{3} + 1$, the message is verified, the block is completed, and the verification result is returned to the witness node of the production block, indicating that the block can be added to the blockchain. Formal methods can be used to verify the correctness of smart contracts and blockchain codes [49], which can help prevent costly errors and security breaches [50]. The optimized block validation algorithm is shown in Algorithm 2.

Algorithm 2: Block validation algorithm

Input : Witness node collection $A_1 = \{x_0, x_1, \ldots, x_{TN-1}\}$, main node of witnesses in this round $N_{R_i,i}$, alternative witness node Set $A_2 = \{x_{TN}, x_{TN+1}, \ldots, x_{2TN-1}\}$

Output: Block confirmation message $\text{block}_{\text{confirm}}$ or block error message $\text{block}_{\text{error}}$

Encapsulate block messages
$N_{R_i,i}$ broadcast (block, blockMessage)
Select the validation master node: $N_{R_2,P} \leftarrow c \ mod \ |N_{R_2}|$
Update the block preparation message to $<$ c, blockHeight, tx, Hash (tx), blockMessage $>$
if *Node validation preparation message is true* **then**
    Node $N_{R_2,P}$ broadcast preparation message
    if *Node validation preparation message is true and validation accumulates to* $\frac{TN}{3} + 1$ **then**
        Node $N_{R_2,i}$ broadcast confirmation message
        if *Node validation confirmation message is true and validation accumulates to* $\frac{2TN}{3} + 1$ **then**
            Node $N_{R_2,P}$ broadcasts a message ($\text{block}_{\text{true}}$) and adds the block to the blockchain
        else
            Node $N_{R_2,P}$ broadcasts a message ($\text{block}_{\text{error}}$) and logs the error message (error, $N_{R_i,i}$)
        end
    else
        Node $N_{R_2,P}$ broadcasts a message ($\text{block}_{\text{error}}$) and logs the error message (error, $N_{R_i,i}$)
    end
end

4. Simulation and Performance Analysis

4.1. Simulation Settings

We tested our proposed method by running experiments on three different datasets: MNIST, SVHN, and CIFAR-10. MNIST is a popular dataset for image recognition tasks, containing 60,000 training samples and 10,000 handwritten character test samples. The second dataset is SVHN, which is specifically designed for autonomous vehicles to recognize house numbers in Google Street View photos. It consists of 73,257 training sets, 26,032 test sets, and 531,131 additional photos for training. The third dataset is CIFAR-10, which contains 10 different types of images, each with 6000 images, for a total of 60,000 training images.

We divided our experimental results into 100 small samples and allocated 100 distinct nodes to evaluate our proposed algorithm. To carry this out, we employed a convolutional neural network (CNN) architecture that included a 5×5 convolutional layer, a fully connected layer, and a softmax output layer. With these datasets and the CNN architecture specified, we sought to evaluate the performance and efficacy of our algorithm.

4.2. Comparison Scheme

We used three main comparison techniques during the simulation testing stage to assess the effectiveness of our proposed approach.

(1) Evaluation of the incentive mechanism: We compared the BOppCL framework proposed in this article with FedAVG and ASTW-FedAVG. This comparison was designed to demonstrate the effectiveness of our method in enhancing system communication efficiency through the incentive mechanism.

(2) We assessed the performance of the BFT-DPoS consensus mechanism by comparing it with the BFT and DPoS consistency approaches. Our evaluation focused on how well BFT-DPoS could improve the efficiency of system communication.
(3) We evaluated the scalability of the BOppCL method by comparing it to the ASTW-FedAVG and local CNN approaches. This comparison looked at how the method behaves when the number of learning nodes in the network is changed.

By conducting these evaluations, our objective was to provide a comprehensive assessment of the proposed approach, highlighting its strengths and advantages compared to existing methods in terms of incentive mechanisms, consensus mechanisms, and overall system performance.

4.3. Analysis of Simulation Results

4.3.1. Incentive Mechanism Evaluation

We employed two methods to compare the efficiency of the algorithm:

(1) The results showed that the model achieved optimal prediction accuracy. We evaluated the accuracy of the central model's predictions after 200 rounds of training and found that it had reached its peak performance. This criterion allowed us to assess the algorithm's ability to achieve high accuracy within a specified number of rounds.
(2) Prediction accuracy of 98% (95% for the SVHN dataset and 90% for the CIFAR-10 dataset) in the central model: We aimed to achieve a prediction precision of 98% (95% for the SVHN dataset and 90% for the CIFAR-10 dataset) in the central model. This criterion served as a benchmark for evaluating the algorithm's performance in achieving high accuracy levels.

We split the MNIST, SVHN, and CIFAR-10 datasets into 100 subsets and assigned them to 100 nodes to ensure a fair comparison. We conducted three different random divisions of the datasets, labeled 1 @ MNIST, and each division yielded different results. The results of the experiments are shown in Table 3.

Table 3. Performance testing results of incentive mechanisms.

Dataset	BOppCL		ASTW-FedAVG		FedAVG	
	Communication Rounds	Accuracy	Communication Rounds	Accuracy	Communication Rounds	Accuracy
1@MNIST	29	98.85%	61	96.69%	75	98.19%
2@MNIST	30	99.50%	70	99.07%	75	98.18%
3@MNIST	29	99.81%	70	98.79%	73	98.29%
1@SVHN	71	95.11%	97	95.57%	127	93.33%
2@SVHN	83	95.98%	106	95.15%	159	91.12%
3@SVHN	69	98.87%	93	98.59%	97	98.19%
1@CIFAR-10	87	91.3%	117	91.5%	147	89.3%
2@CIFAR-10	103	93.1%	126	91.1%	176	87.3%
3@CIFAR-10	95	90.3%	113	90.7%	158	89.5%

Based on the observations in Table 3, it is evident that both ASTW-FedAVG and BOppCL exhibit superior communication rounds and precision compared to FedAVG in all datasets. This can be attributed to the absence of any incentive mechanism or optimization algorithms in FedAVG to improve communication efficiency in Federated Learning. For example, considering the 1@MNIST dataset, BOppCL achieves 98% precision in just 29 communication rounds, outperforming ASTW-FedAVG, which requires 61 rounds, and FedAVG, which requires 75 rounds to achieve the same level of precision. These results highlight that BOppCL demonstrates optimal performance in terms of communication rounds and accuracy in most datasets. The experimental results underscore that the

incentive mechanism proposed in BOppCL significantly accelerates the convergence speed of learning and improves learning performance. This, in turn, leads to a substantial reduction in the communication cost of OppCL.

4.3.2. Evaluation of the Consensus Mechanism

Block confirmation delay is a significant measure to assess the effectiveness of consensus algorithms. In this article, the phrase "block confirmation delay" is used to refer to the time period between the production of a block by a witness node and its eventual inclusion in the blockchain by a candidate witness node. To make a comparison, we studied and analyzed the confirmation latency of BFT-DPoS, DPoS, and PBFT in the same environment, as illustrated in Figure 3.

By assessing the confirmation latencies of these consensus algorithms, we gain insight into their respective performance and efficiency in terms of block confirmation time. The comparison depicted in Figure 3 allows for a comprehensive evaluation and analysis of the delay characteristics exhibited by the BFT-DPoS, DPoS, and PBFT consensus strategies in similar settings.

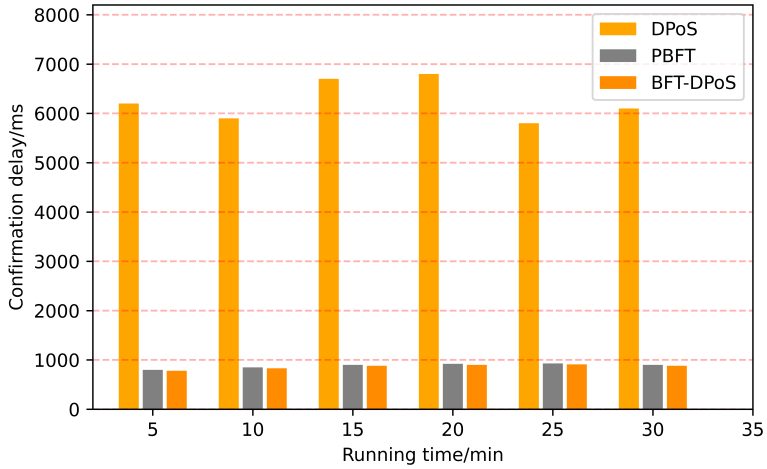

Figure 3. Block confirmation delay comparison.

Based on the information presented in Figure 3, it is evident that the BFT-DPoS algorithm achieves prompt block validation confirmation, which requires only around 800 ms. In contrast, the original DPoS algorithm requires at least 6 seconds for verification confirmation from a minimum of 2/3 of the total witness nodes. Furthermore, due to the incorporation of the core principles of the PBFT algorithm, the verification latency of BFT-DPoS and PBFT is relatively similar. One notable advantage of the BFT-DPoS algorithm proposed in this article is that it generates blocks by electing specific witness nodes, with the number of witnesses remaining fixed over an extended period. As a result, the throughput and verification delay of blocks do not change significantly with increasing number of nodes within the network. This characteristic effectively ensures the stability of the blockchain network. In summary, the BFT-DPoS algorithm introduced in this article exhibits efficient and timely block validation confirmation, surpassing the original DPoS algorithm in terms of verification time. Furthermore, its stability is guaranteed by the consistent number of witness nodes, leading to consistent throughput and verification delay, despite network size variations.

4.3.3. Comprehensive Performance Evaluation

The results of the precision of the proposed mechanism in the MNIST, SVHN, and CIFAR-10 datasets, considering different numbers of training nodes, are shown in Figures 4–6. To replicate real scenarios in ITS, three nodes were randomly selected as low-quality participants during the experiment. These nodes possessed limited communi-

cation and computing capabilities and introduced poor model parameter quality through random noise interference with the original parameters during the model aggregation process. The experimental findings demonstrate that the proposed mechanism achieves commendable accuracy. Although the inherent complexity of the SVHN and CIFAR-10 dataset leads to slightly lower global accuracy compared to MNIST, the proposed algorithm still achieves high accuracy in SVHN and CIFAR-10, showcasing its versatility in different datasets. When the number of participating nodes in the training process increases from 20 and 40 to 60, a slight decrease in global accuracy is observed as the number of iterations increases. However, the overall difference is not substantial. This minor fluctuation in the experimental results highlights the robust scalability of the proposed mechanism. The proposed mechanism is effective and scalable, as evidenced by the experimental results. It can reduce the negative effect of low-quality nodes on overall learning outcomes, even when the number of participating nodes increases, can maintain a high level of accuracy despite the presence of low-quality nodes, and can adapt to varying numbers of participating nodes, minimizing their influence on overall learning outcomes.

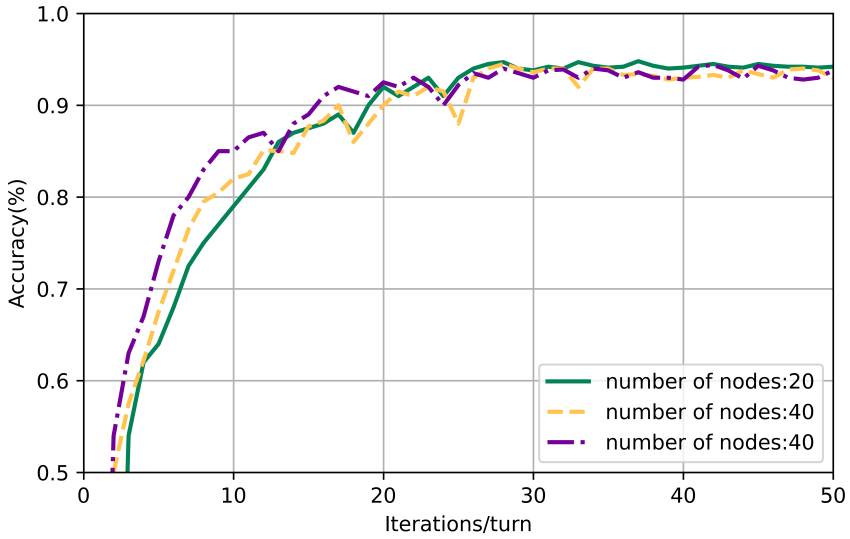

Figure 4. The precision of various amounts of nodes on the MNIST dataset was examined. Results showed that the more nodes used, the higher the accuracy.

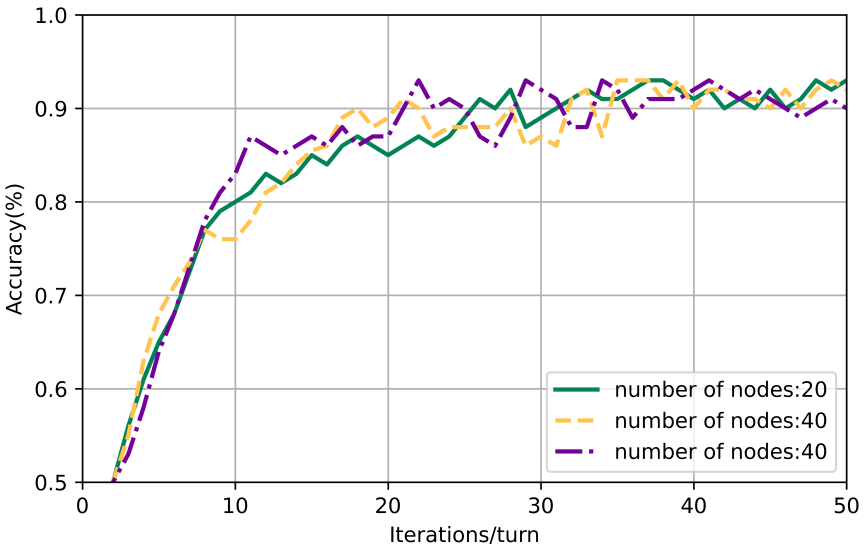

Figure 5. The precision of various amounts of nodes on the SVHN dataset was examined. The results showed that the accuracy increased as the number of nodes increased.

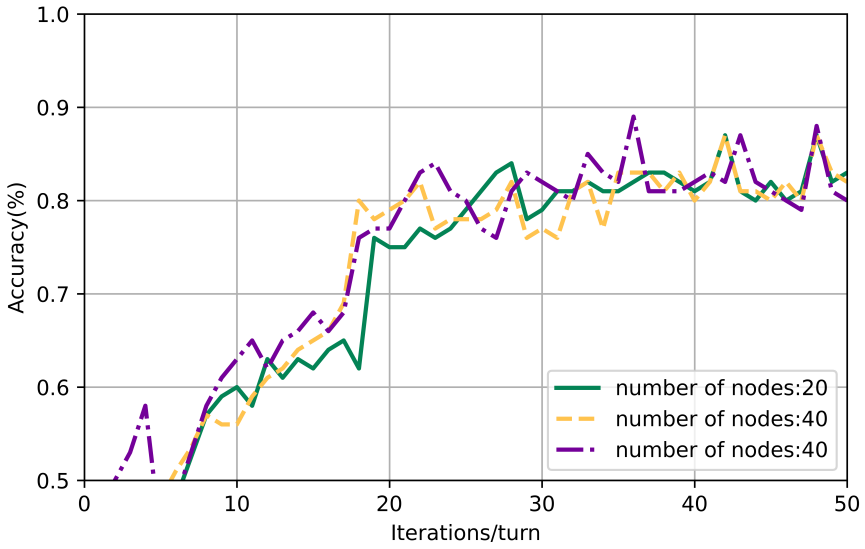

Figure 6. The precision of various amounts of nodes on the CIFAR-10 dataset was examined.

In our comprehensive performance evaluation and algorithm comparison experiment, we compared the mechanism proposed in this article with local CNN and ASTW-FedAVG. The dataset was randomly divided into 100 subsets and assigned to 100 training nodes. In the local CNN approach, each node trained its model using the assigned subset. ASTW-FedAVG, on the other hand, trained local models on individual nodes' subsets and updated the global model using a weighted average aggregation algorithm on a central server. Figures 7–9 present the accuracy results, indicating that the mechanism proposed in this document exhibits slightly better accuracy compared to ASTW-FedAVG. However, the accuracy of the local CNN is significantly lower than that of the other two mechanisms. This discrepancy can be attributed to the local CNN training algorithm, where the primary objective is to minimize the loss in the local dataset. As a result, it may lead to the attainment of a local optimal solution that deviates from the global optimal solution, resulting in lower accuracy. The experiments demonstrate that the mechanism proposed in this article achieves high accuracy while ensuring data security and privacy protection.

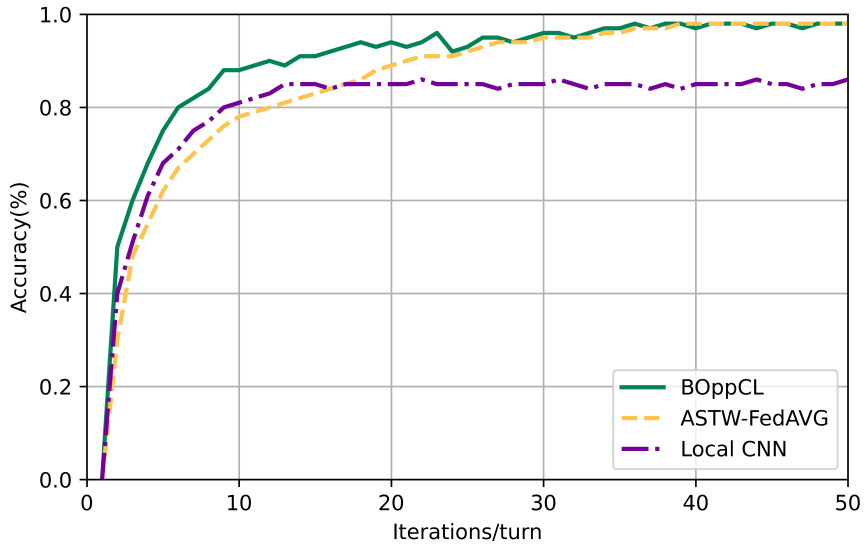

Figure 7. Accuracy of 3 algorithms on the MNIST dataset.

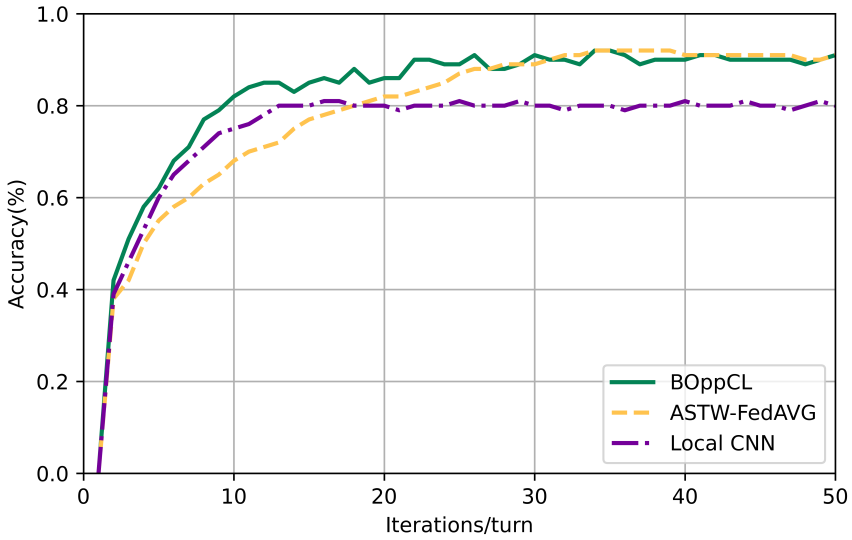

Figure 8. Accuracy of 3 algorithms on the SVHN dataset.

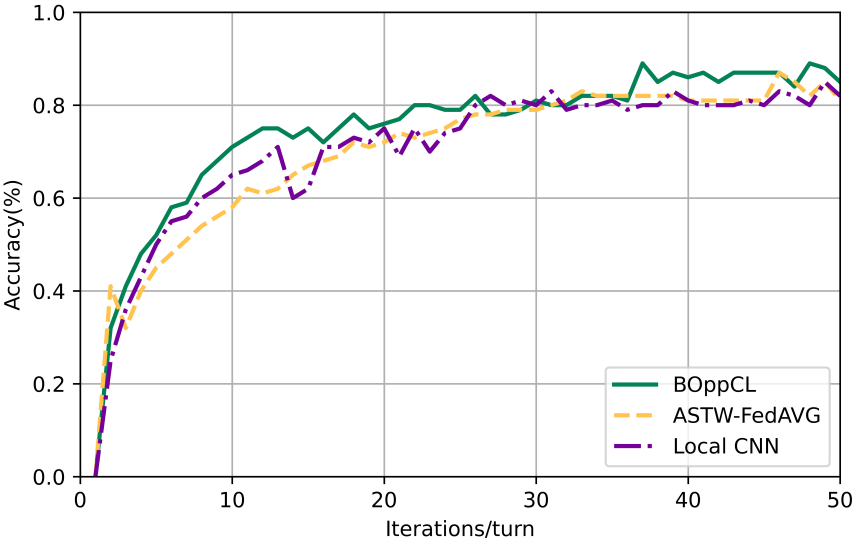

Figure 9. Accuracy of 3 algorithms on the CIFAR-10 dataset.

The evaluation of the overall system run time is shown in Figure 10. It demonstrates that for a fixed number of users, the system's running time increases with the growth of the dataset size and eventually stabilizes. This behavior can be attributed to optimization of computational efficiency in federated learning and consensus efficiency in the proposed mechanism, as discussed in this article. These optimizations effectively improve the overall operational efficiency of the system. Furthermore, when considering different numbers of users, the system running time increases as the number of users increases. This correlation arises because of the collaborative nature of the system. As the number of users increases, more time is required to achieve effective collaboration and synchronization among a larger user base. In summary, the mechanism proposed in this article improves the operating efficiency of the system by optimizing computational efficiency and consensus. The system running time is influenced by factors such as the size of the dataset and the number of participating users. Understanding these dynamics helps in managing and optimizing the system run time for efficient execution.

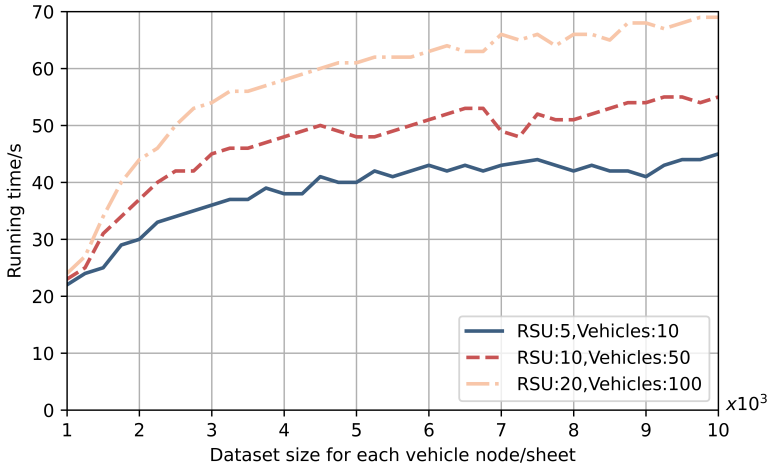

Figure 10. Overall system running time.

On the basis of the experiments carried out, it is observed that the accuracy of the mechanism proposed in this article is minimally affected by an increase in the number of users. However, the running time experiences a significant increase. The stable accuracy is mainly attributed to the blockchain mode implemented in the proposed scheme, which ensures consistent learning accuracy. However, as the number of users increases, the mechanism encounters additional computational and consensus tasks. More local models must be updated and calculated, and a greater number of RSUs are involved in executing the consensus process. Consequently, the time required to train, update, and transmit the data increases. Although this leads to a slight increase in the overall run time, the participation of multiple users expands the dataset size used for calculations. As a result, shared data become more accurate, enhancing the quality and reliability of the collaborative learning process. In summary, while the accuracy of the proposed mechanism remains stable with an increasing number of users, the run time experiences a notable increase due to the additional computational and consensus tasks involved. However, the participation of multiple users contributes to a larger and more accurate dataset, thus improving the precision of data sharing and the overall effectiveness of the mechanism.

In ITS, motivating autonomous vehicles to participate in distributed learning to improve scene adaptability requires addressing accuracy and operational efficiency issues. The BOppCL algorithm updates the local model of vehicles mainly through the gradients encountered through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, which could potentially affect the utilization of ITS communication resources.

5. Conclusions

In this paper, we propose a novel blockchain-enabled OppCL (BOppCL) approach for ITS; it offers benefits such as updating autonomous vehicle models, improving driving decisions, and achieving intelligent efficient driving. Meanwhile, the authors address challenges such as vehicle reliability, limited learning time, and participant motivation. We integrate blockchain into OppCL, propose vehicle selection and resource allocation algorithms, and study the BFT-DPoS consensus mechanism. Simulation experiments show that our approach outperforms existing methods in accuracy and run time, enabling safe, reliable, and efficient distributed learning.

Future work involves optimizing node selection based on factors such as energy consumption and improving consensus algorithms for scalability. These enhancements will advance OppCL's applicability and performance in ITS. In future work, we can focus on optimizing node selection considering factors such as energy consumption, communication cost, and computational cost. Improving consensus algorithms will enhance scalability and robustness, advancing the applicability and performance of OppCL in ITS.

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

Funding: This research was funded by NSFC-FDCT under its Joint Scientific Research Project Fund (0051/2022/AFJ) and Philosophy and Social Science Foundation of Hunan Province (21YBA279).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The anonymized data used in this study are available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

ITS	Intelligent Transportation Systems
OppCL	Opportunistic Federated Learning
BOppCL	Blockchain-enabled Opportunistic Federated Learning
BFT	Byzantine Fault-Tolerant
DPoS	Delegated Proof of Stake
V2X	Vehicle-to-Everything
IoV	Internet of Vehicles

References

1. Veres, M.; Moussa, M. Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3152. [CrossRef]
2. Ei Leen, M.W.; Jafry, N.H.A.; Salleh, N.M.; Hwang, H.; Jalil, N.A. Mitigating Traffic Congestion in Smart and Sustainable Cities Using Machine Learning: A Review. *Int. Conf. Comput. Sci. Its Appl.* **2023**, *13957*, 321–331. [CrossRef]
3. Haydari, A.; Yılmaz, Y. Deep Reinforcement Learning for Intelligent Transportation Systems: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 11. [CrossRef]
4. Saleem, M.; Abbas, S.; Ghazal, T.M.; Khan, M.A.; Sahawneh, N.; Ahmad, M. Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques. *Egypt. Inform. J.* **2022**, *23*, 417. [CrossRef]
5. Peng, T.; Yang, X.; Xu, Z.; Liang, Y. Constructing an Environmental Friendly Low-Carbon-Emission Intelligent Transportation System Based on Big Data and Machine Learning Methods. *Sustainability* **2020**, *12*, 8118. [CrossRef]
6. 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; p. 83. [CrossRef]
7. Hejazi, H.; László, B. 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. [CrossRef]
8. Kaleem, S.; Sohail, A.; Tariq, M.U.; Asim, M. An Improved Big Data Analytics Architecture Using Federated Learning for IoT-Enabled Urban Intelligent Transportation Systems. *Sustainability* **2023**, *15*, 15333. [CrossRef]
9. Zhou, H.; Xu, W.; Chen, J.; Wang, W. Evolutionary V2X Technologies Toward the Internet of Vehicles: Challenges and Opportunities. *Proc. IEEE* **2020**, *108*, 308. [CrossRef]
10. Hildebrand, B.; Tabassum, S.; Konatham, B.; Amsaad, F.; Baza, M.; Salman, T.; Razaque, A. A comprehensive review on blockchains for Internet of Vehicles: Challenges and directions. *Comput. Sci. Rev.* **2023**, *48*, 100547. [CrossRef]
11. Javed, A.R.; Hassan, M.A.; Shahzad, F.; Ahmed, W.; Singh, S.; Baker, T.; Gadekallu, T.R. Integration of Blockchain Technology and Federated Learning in Vehicular (IoT) Networks: A Comprehensive Survey. *Sensors* **2022**, *22*, 4394. [CrossRef]
12. Jeong, B.-G.; Youn, T.-Y.; Jho, N.-S.; Shin, S.U. Blockchain-Based Data Sharing and Trading Model for the Connected Car. *Sensors* **2020**, *20*, 3141. [CrossRef] [PubMed]
13. Ying, Z.; Ma, M.; Zhao, Z.; Liu, X.; Ma, J. A reputation-based leader election scheme for opportunistic autonomous vehicle platoon. *IEEE Trans. Veh. Technol.* **2021**, *71*, 3519. [CrossRef]
14. 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), Kassel, Germany, 22–26 March 2021. [CrossRef]

15. Hahn, D.; Munir, A.; Behzadan, V. Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. *IEEE Intell. Transp. Syst. Mag.* **2021**, *13*, 181. [CrossRef]
16. Du, G.; Cao, Y.; Li, J.; Zhuang, Y. Secure Information Sharing Approach for Internet of Vehicles Based on DAG-Enabled Blockchain. *Electronics* **2023**, *12*, 1780. [CrossRef]
17. Cocîrlea, D.; Dobre, C.; Hîrţan, L.-A.; Purnichescu-Purtan, R. Blockchain in Intelligent Transportation Systems. *Electronics* **2020**, *9*, 1682. [CrossRef]
18. Mondal, A.; Virk, H.; Gupta, D. Beas: Blockchain enabled asynchronous & secure federated machine learning. *arXiv* **2022**, arXiv:2202.02817. https://doi.org/10.48550/arXiv.2202.02817.
19. Saraswat, D.; Verma, A.; Bhattacharya, P.; Tanwar, S.; Sharma, G.; Bokoro, P.N.; Sharma, R. Blockchain-based federated learning in UAVs beyond 5G networks: A solution taxonomy and future directions. *IEEE Access* **2022**, *10*, 33154. [CrossRef]
20. 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. [CrossRef]
21. Zhu, J.; Cao, J.; Saxena, D.; Jiang, S.; Ferradi, H. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Comput. Surv.* **2023**, *55*, 1. [CrossRef]
22. Distefano, S.; Di Giacomo, A.; Mazzara, M. Trustworthiness for transportation ecosystems: The blockchain vehicle information system. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 2013. [CrossRef]
23. Sadaf, M.; Iqbal, Z.; Javed, A.R.; Saba, I.; Krichen, M.; Majeed, S.; Raza, A. Connected and Automated Vehicles: Infrastructure, Applications, Security, Critical Challenges, and Future Aspects. *Technologies* **2023**, 11, 117. [CrossRef]
24. Ying, Z.; Cao, S.; Liu, X.; Ma, Z.; Ma, J.; Deng, R.H. PrivacySignal: Privacy-preserving traffic signal control for intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 16290. [CrossRef]
25. Chai, Z.-Y.; Yang, C.-D.; Li, Y.-L. Communication efficiency optimization in federated learning based on multi-objective evolutionary algorithm. *Evol. Intell.* **2023**, *16*, 1033. [CrossRef]
26. Almanifi, O.R.A.; Chow, C.-O.; Tham, M.-L.; Chuah, J.H.; Kanesan, J. Communication and computation efficiency in Federated Learning: A survey. *Internet Things* **2023**, *22*, 100742. [CrossRef]
27. Zhao, Z.; Mao, Y.; Liu, Y.; Song, L.; Ouyang, Y.; Chen, X.; Ding, W. Towards efficient communications in federated learning: A contemporary survey. *J. Frankl. Inst.* **2023**, *360*, 8669–8703. [CrossRef]
28. Mariani, S.; Cabri, G.; Zambonelli, F. Coordination of autonomous vehicles: Taxonomy and survey. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1. [CrossRef]
29. 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*, 1. [CrossRef]
30. 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), Singapore, 23–27 May 2022; p. 8752. [CrossRef]
31. Zhao, Y.; Zhao, J.; Yang, M.; Wang, T.; Wang, N.; Lyu, L.; Niyato, D.; Lam, K.-Y. Local differential privacy-based federated learning for internet of things. *IEEE Internet Things J.* **2020**, *8*, 8836. [CrossRef]
32. 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. [CrossRef]
33. Chai, H.; Leng, S.; Chen, Y.; Zhang, K. A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 3975. [CrossRef]
34. Pokhrel, S.R.; Choi, J. Federated learning with blockchain for autonomous vehicles: Analysis and design challenges. *IEEE Trans. Commun.* **2020**, *68*, 4734. [CrossRef]
35. Cheng, Z.; Wang, B.; Pan, Y.; Liu, Y. Strategic Analysis of Participants in BCFL-Enabled Decentralized IoT Data Sharing. *Mathematics* **2023**, *11*, 4520. [CrossRef]
36. Si, X.; Li, M.; Yao, Z.; Zhu, W.; Liu, J.; Zhang, Q. An Efficient and Secure Blockchain Consensus Protocol for Internet of Vehicles. *Electronics* **2023**, *12*, 4285. [CrossRef]
37. Thantharate, P.; Thantharate, A. ZeroTrustBlock: Enhancing Security, Privacy, and Interoperability of Sensitive Data through ZeroTrust Permissioned Blockchain. *Big Data Cogn. Comput.* **2023**, *7*, 165. [CrossRef]
38. Xie, Q.; Ding, Z.; Tang, W.; He, D.; Tan, X. Provable secure and lightweight blockchain-based V2I handover authentication and V2V broadcast protocol for VANETs. *IEEE Trans. Veh. Technol.* **2023**, *2022*, 3372489. [CrossRef]
39. Mecheva, T.; Kakanakov, N. Cybersecurity in Intelligent Transportation Systems. *Computers* **2020**, *9*, 83. [CrossRef]
40. 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; p. 471. [CrossRef]
41. 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; p. 320. [CrossRef]
42. Yang, J.; Zhou, Y.; Wen, W.; Zhou, J.; Zhang, Q. Asynchronous Hierarchical Federated Learning Based on Bandwidth Allocation and Client Scheduling. *Appl. Sci.* **2023**, *13*, 11134. [CrossRef]

43. Cao, B.; Zhang, Z.; Feng, D.; Zhang, S.; Zhang, L.; Peng, M.; Li, Y. Performance analysis and comparison of PoW, PoS and DAG based blockchains. *Digit. Commun. Netw.* **2020**, *6*, 480. [CrossRef]
44. Saad, S.M.S.; Radzi, R.Z.R.M. Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos). *Int. J. Innov. Comput.* **2020**, *10*, 27. [CrossRef]
45. Le, X.; Yao, S.; Rafiq, S.; Lina, M.; Zhang, L.; Muhammad Ali, I. Smart and secure CAV networks empowered by AI-enabled blockchain: The next frontier for intelligent safe driving assessment. *IEEE Netw.* **2022**, *36*, 197. [CrossRef]
46. Li, C.; Xu, R.; Li, D. Characterizing Coin-Based Voting Governance in DPoS Blockchains. *Proc. Int. Aaai Conf. Web Soc. Media* **2023**, *17*, 1148. [CrossRef]
47. Yang, X.; Ning, Z.; Wenjing, L.; Thomas, H.Y. A survey of distributed consensus protocols for blockchain networks. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1432. [CrossRef]
48. Song, T.; Zhiqiang, W.; Jian, J.; Suli, G.; GaiFang, T. Improved PBFT algorithm for high-frequency trading scenarios of alliance blockchain. *Sci. Rep.* **2022**, *12*, 4426. [CrossRef]
49. Abdellatif, T.; Brousmiche, K.L. Formal verification of smart contracts based on users and blockchain behaviors models. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018. [CrossRef]
50. Krichen, M.; Lahami, M.; Al-Haija, Q.A. Formal methods for the verification of smart contracts: A review. In Proceedings of the 2022 15th International Conference on Security of Information and Networks (SIN), Sousse, Tunisia, 11–13 November 2022. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.