mathematics

MDPI

Article

Binary Encoding-Based Federated Learning for Traffic Sign Recognition in Autonomous Driving

Yian Wen [1], Yun Zhou [2,*] and Kai Gao [3]

[1] School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410114, China; 202103130815@stu.csust.edu.cn
[2] School of Information Technology and Management, Hunan University of Finance and Economics, Changsha 410205, China
[3] College of Automotive and Mechanical Engineering, Changsha University of Science and Technology, Changsha 410114, China; kai_g@csust.edu.cn
* Correspondence: yunzhou_hufe@163.com

Abstract: Autonomous driving involves collaborative data sensing and traffic sign recognition. Emerging artificial intelligence technology has brought tremendous advances to vehicular networks. However, it is challenging to guarantee privacy and security when using traditional centralized machine learning methods for traffic sign recognition. It is urgent to introduce a distributed machine learning approach to protect private data of connected vehicles. In this paper, we propose a local differential privacy-based binary encoding federated learning approach. The binary encoding techniques and random perturbation methods are used in distributed learning scenarios to enhance the efficiency and security of data transmission. For the vehicle layer in this approach, the model is trained locally, and the model parameters are uploaded to the central server through encoding and perturbing. The central server designs the corresponding decoding, correction scheme, and regression statistical method for the received binary string. Then, the model parameters are aggregated and updated in the server and transmitted to the vehicle until the learning model is trained. The performance of the proposed approach is verified using the German Traffic Sign Recognition Benchmark data set. The simulation results show that the convergence of the approach is better with the increase in the learning cycle. Compared with baseline methods, such as the convolutional neural network, random forest, and backpropagation, the proposed approach achieves higher accuracy in the process of traffic sign recognition, with an increase of 6%.

Keywords: federated learning; local differential privacy; traffic sign recognition; autonomous driving

MSC: 68P27

check for updates

Citation: Wen, Y.; Zhou, Y.; Gao, K. Binary Encoding-Based Federated Learning for Traffic Sign Recognition in Autonomous Driving. *Mathematics* **2024**, *12*, 2229. https://doi.org/10.3390/math12142229

Academic Editor: Andrea Scozzari

Received: 20 June 2024
Revised: 9 July 2024
Accepted: 16 July 2024
Published: 17 July 2024

1. Introduction

With the development of the internet of things (IoT) and artificial intelligence (AI), autonomous driving is becoming an effective technology to solve traffic congestion and reduce traffic accidents [1]. As a part of autonomous driving, in addition to using radar to detect surrounding vehicles [2,3], vehicles also need to detect traffic signs in real-time through onboard cameras [4]. The realization of these applications and services is dependent on a large amount of data generated by the vehicles, which contains private data, such as vehicle location information [5] or owner privacy information [6].

In autonomous driving, the commonly used methods for recognizing traffic signs include convolutional neural networks (CNNs) [7], support vector machine (SVM) [8], and random forest (RF) [9]. These classification methods belong to the traditional centralized learning method. A central server is needed to collect all the data, including private data, so as to better train the model and realize high-accuracy traffic sign recognition. However, with the increase in personal privacy issues [10], the centralized learning method is no longer applicable to vehicle applications and services. How to achieve efficient traffic sign recognition while ensuring vehicle data privacy and security has become an urgent problem. Traditional centralized machine learning methods face privacy leakage and security risks when processing vehicle network data, making it difficult to meet the high requirements of modern autonomous driving systems for data protection. Therefore, there is an urgent need to explore a distributed machine learning method to improve the accuracy and efficiency of traffic sign recognition while protecting vehicle privacy.

Federated learning, proposed by Google, is a typical distributed machine learning method that can avoid the above privacy problems. For federated learning, the model is trained locally through the data owners, and then the model parameters are uploaded to the server. All parameters are received and aggregated by the server, and then the updated parameters are transmitted to each owner for model updating. This process continues to be iterative until the learning model is trained. Federated learning works similarly to centralized machine learning methods [11], except that it does not require the transfer of local data.

While federated learning has already received great research interest in agriculture [12], medicine [13], and intelligent security [14], its application to the internet of vehicles is challenging. Although federated learning does not directly expose vehicle information, attackers can obtain it by cracking model parameters, which means that there are still security risks in federated learning [15]. Therefore, researchers have studied a series of defense measures, including defense measures based on secure multi-party computation and privacy protection technology based on differential privacy. In secure multi-party computation, participants can use private data to participate in secure computing without disclosing private data. However, encrypted data onto different technologies cannot be intercommunicated, which easily causes new data island problems. Differential privacy protection technology can be divided into centralized differential privacy and local differential privacy. Centralized differential privacy requires an absolutely credible third party. Once the third party is no longer trusted, private data of the users may be leaked at any time [16]. Local differential privacy technology can resist security attacks by adding random perturbations to model parameters or gradients, but it is still possible to leak private data, although inversion will incur higher costs [17].

In order to address the security risks in federated learning and better apply it to the internet of vehicles, we propose a binary encoding federated learning method based on local differential privacy technology (BCFL-LDP). It can effectively prevent privacy leakage during data transmission by implementing local differential privacy protection. It employs binary encoding technology to reduce data transmission volume and improve processing efficiency. Furthermore, leveraging the federated learning framework, BCFL-LDP realizes distributed training and model aggregation, thereby enhancing the model's generalization capabilities. BCFL-LDP not only resolves privacy and security issues in autonomous driving but also improves the system's adaptability and computational efficiency, offering a novel solution for the advancement of autonomous driving technology. From the perspective of security, federated learning algorithms are superior to centralized ML algorithms and can better protect the privacy of vehicle users. Meanwhile, compared with traditional federated learning algorithms, the proposed BCFL-LDP algorithm adds binary encoding and perturbation techniques before uploading model parameters to the central server. Even if the model parameters are leaked, attackers cannot obtain any valid information, further enhancing the security of the model. In autonomous driving scenarios, the shooting angle of each vehicle and the number of traffic signs captured is limited. Therefore, the vision models trained for traffic signs may not be particularly accurate by employing the information of individual vehicles. Benefiting from federated learning with privacy protection, each vehicle can be trained locally; then, the trained model parameters are encrypted and randomly perturbed and uploaded to the central server. All collected vehicle model parameters are decrypted and aggregated by the central server, and then updated model parameters are transmitted to each vehicle for model updating. Specifically, the contributions of this paper are as follows:

- A local differential privacy-based federated learning method is proposed. The model is trained locally in vehicles, and then the model parameters are uploaded to the central server. The advantage is that the original data can be saved in the vehicle itself, which can achieve higher recognition accuracy while protecting the privacy of vehicle users.
- A new random perturbation method of binary string bits is proposed. Before the model parameters are uploaded to the central server, binary encoding and disturbance are carried out. Therefore, even if the model parameters are leaked, the attacker cannot obtain any valid information.
- Through the training of the GTSRB dataset, the proposed BCFL-LDP is verified to be superior to the existing traffic sign recognition methods. The proposed BCFL-LDP has a faster convergence speed than the baselines and is more suitable for actual autonomous driving scenes.

The rest of the paper is organized as follows: Section 2 reviews the relevant work. Section 3 presents the basic framework of federated learning for traffic sign recognition in the internet of vehicles. Section 4 introduces federated learning and local differential privacy encryption techniques. The experiment and evaluation results are shown in Section 5. Conclusions are presented in Section 6.

2. Related Work

With the development of the information age, more and more research on privacy protection methods is coming along. The purpose of the privacy protection method in the internet of vehicles is to protect the identity information and location privacy of vehicles during data mining, data analysis, and processing, so as to prevent the illegal tracking down of vehicles by attackers. The aim of our work is to enhance the security and privacy of the traffic sign recognition model, so we mainly introduce privacy protection techniques. This section includes three subsections. In Section 2.1, the related works of federated learning are presented. In Sections 2.2 and 2.3, two traditional privacy protection technologies of federated learning are presented: differential privacy and secure multi-party computation.

2.1. Federated Learning

In the intelligent transportation industry, federated learning [18] is generally used to protect the data privacy of users. In traditional machine learning [19,20], all data are concentrated on a single server for training. In federated learning, each model is first trained locally, and the gradient or parameter of the training model is uploaded to the aggregation server to solve privacy problems. McMahan et al. [21] introduced the federated averaging algorithm, which combined the random gradient descent of each client with the model averaging of the server to protect the original training data. Yuan et al. [10] connected federated learning and broad learning systems and offloaded data sharing into clusters. They improved the aggregation ability of the model and protected the privacy of data sharing.

Traditional federated learning adopts synchronous aggregation, which will undoubtedly produce high communication costs, making it easy for attackers to extract private data information from the model gradient in reverse. Therefore, researchers have made some improvements to existing federated learning methods. Zhu et al. [22] proposed a semi-supervised federated learning framework, which does not require users to provide original trajectory data or rely on prominent data labels. Yu et al. [23] applied federated learning techniques to construct a global model to predict content popularity while protecting the privacy of training data on local vehicles.

2.2. Differential Privacy

2.2.1. Centralized Differential Privacy

In centralized differential privacy protection technology, the privacy of algorithm expansion is defined by the nearest neighbor dataset. Therefore, it requires an absolutely credible third party to aggregate the data together and then add disturbance to the dataset to achieve differential privacy. According to this definition, the data collector has to manage the centralized dataset. In order to obtain private data, the attacker can only infer private information by querying the distribution of statistical data several times. Poddar et al. [24] attempted to predict traffic flow through a real-time traffic state estimator with differential privacy technology to protect the potentially sensitive location information of private users.

2.2.2. Local Differential Privacy

In local differential privacy, each user can process private data independently; that is, the model training process is transferred from the data collector to a single user. Therefore, it can also ensure that users' private information will not be leaked, even if the third party is not reliable. Although local differential privacy technology can resist security attacks by adding random perturbations to model parameters or gradients, private data may still be leaked, even if the cost of backstepping is higher. Wang et al. [25] used local differential privacy technology based on a histogram algorithm to interfere with the context information about the vehicle and then upload it to the base station for offloading decision, which protects the privacy of the resume. Nie et al. [26] introduced Gaussian differential encryption, adding Gaussian noise directly to the Q-network of the output Markov decision process rather than to the model gradient.

2.3. Secure Multi-Party Computation

As a sub-field of cryptography, secure multi-party computation allows multiple users to calculate tasks together and output the calculation results in the case of mutual distrust. Moreover, in the entire user set, two arbitrary users will know nothing about each other except the output results. In terms of the implementation principle, MPC does not rely on a single security algorithm but a comprehensive application of a variety of basic cryptography tools, including homomorphic encryption [27], unintentional transmission, secret sharing [28], symmetric encryption [29], asymmetric encryption [30,31], etc. Through the combination of various algorithms, cipher text data can realize cross-domain flow and security calculation. Peng et al. [29] proposed a security optimization algorithm using symmetric encryption to achieve secure multimedia data transmission of vehicles.

3. Model Definition

3.1. A Basic Model Structure

The vehicular network system consists of a central server and a large number of vehicles, as shown in Figure 1. The distributed vehicles obtain the information of traffic signs through the camera in real-time, and train the model locally. Then, the trained model parameters are uploaded to the central server after processing by local differential privacy technology, which makes all data remain local and protects the privacy and safety of vehicle owners.

In addition, a unified deep-learning model is shared for all vehicles. In this case, vehicles only upload the model parameters to protect data privacy and ensure efficient knowledge sharing. The parameters are received by the central server, and then decoded, corrected, and aggregated. The server returns the updated parameters to each vehicle, until the model is convergent.

The dataset generated by a single vehicle shares the same learning model with other vehicles while training. Compared with a single vehicle, the vehicular network system can obtain more different types of data. For example, one vehicle traveling in the city may obtain data on vehicle speed limits, pedestrians, and traffic lights, while another vehicle traveling in mountainous areas may obtain data drawing attention to falling rocks, continuous bends, uphills, etc. Therefore, using the same learning model, when vehicles driving in mountainous areas enter the urban area, they can identify traffic signs faster and better to ensure the safety and effectiveness of autonomous driving. Therefore, the model can obtain better recognition accuracy and universality.

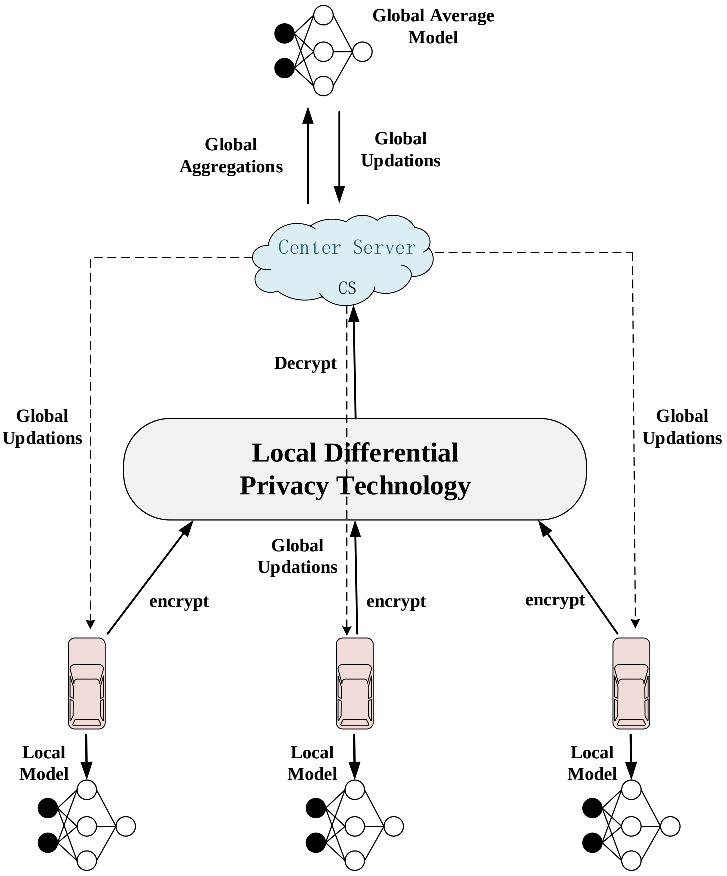

Figure 1. Architecture diagram of the proposed BCFL-LDP for traffic sign recognition in autonomous driving. The framework consists of two parts: the vehicle layer of the local training model and the central server for aggregating model parameters.

3.2. Problem Definition

This paper constructs a vehicle-mounted network system composed of K vehicles and a central server, in which the vehicle is represented by $V = \{v_k\}_{k=1}^{K}$ and the central server is represented by CS. The vehicle takes photos with traffic signs as the original data in real-time and then labels the data. We assume that the original data are $image_k$ and the corresponding label is $label_k$. For each group of data, $Date_k$ can be expressed as $Date_k = (image_k, label_k)$. In traditional centralized learning, vehicles send all data, including original data, location information, etc., to centralized processors (such as central servers). In order to protect the private data of vehicles from being captured by attackers, a federated learning framework is proposed for real-time traffic sign detection across vehicles. In this framework, the vehicle can retain all data, train the deep learning model $M = h(x, w)$ locally, and then send the trained model parameter w to the central server, where x is the dataset $Date$ collected by the vehicles.

As a typical federated average learning framework, our purpose is to make the central server protect the private information of all vehicles well. In order to achieve this, binary encoding encryption and interference are needed for the model parameters trained locally by the vehicle v_k. Then, the encrypted model parameters are transmitted to the central server for aggregation to ensure that the private data of all vehicles are not leaked.

4. BCFL-LDP

4.1. Convolutional Neural Network

In order to better recognize traffic signs, a convolutional neural network (CNN) is used to detect images captured by vehicle cameras. The input sample data of the vehicle v_k are set as x_k, and the deep learning model can be expressed as follows:

$$h(x_k, w) = FC(\text{Pool}(\text{Conv}(x_k, w_{Conv}), w_{\text{Pool}}), w_{FC}), \tag{1}$$

where Conv$(*)$, Pool$(*)$, and FC$(*)$ represent the convolution layer, pool layer, and full connection layer, respectively, and w represents parameters corresponding to different layers, which can be expressed as follows:

$$w = \{w_{\text{Conv}}, w_{\text{Pool}}, w_{FC}\}. \tag{2}$$

The prediction results of model $h(x_k, w)$ are expressed by the SoftMax function:

$$y^{\text{pred}} = \text{SoftMax}(h(x_k, w)). \tag{3}$$

4.2. Federated Learning Based on Local Differential Privacy

4.2.1. Vehicle Layer

In order to prevent the model parameters from being leaked in the process of being uploaded to the central server, a local differential privacy technology of federated learning is introduced to deal with the privacy information contained in the parameters. A new random perturbation method based on binary bits is then proposed. Thus, even if leaks occur, the attacker will not obtain any useful information.

Suppose that the number of vehicles is K. For each iteration, every vehicle divides its dataset into several parts of size M according to parameter M. Each piece of data is updated locally by training round E, and the gradient of data loss is calculated. Then, the gradient descent update is carried out to obtain the new local parameter w_k.

$$w_k \leftarrow w_k - \eta \overline{g_t}, \tag{4}$$

where η is the learning rate, and $\overline{g_t}$ is the mean model gradient.

$$\overline{g_t} = \sum_{k=1}^{K} \frac{n_k}{n} g_i. \tag{5}$$

Before parameter w_k is transmitted to the central server, vehicles perform a binary conversion on w_k and add a disturbance. The binary strings consist of three parts: the symbol of the input, the integer part, and the decimal part. The positive and negative bits of the input symbol are represented by 1 and 0, respectively. For ease of calculation, the integer part is n bits, and the decimal part is d bits. So, the length of the binary string is $l = 1 + n + d$.

The binary encoding processes are as follows:

- The initial value of a binary string of length l is set to 0.
- Model parameters are converted to binary strings and represented as b_k.
- Each bit i on b_k is added with random perturbation to obtain a new string b_k'.

4.2.2. Central Server Layer

As we know from the above, the model parameter w_k is binary encoded, so there are only two results for each bit: 0 and 1. Assuming that the result of a bit in a binary string is 1, and its true proportion is π, but we do not know which one it is. Therefore, the model parameters of all vehicles are counted to obtain the real proportion of bit 1. But, because of privacy reasons, the vehicles answer the bit with probability p and answer the opposite with probability $1 - p$.

Perturbation statistics are the first proceeding. Suppose that in the statistical result, the number of bits 1 is n_1 and the number of bits 0 is $n - n_1$, and J_i represents the answer to the i bit. The probability of the two answers is as follows:

$$P(J_i = 1) = \pi p + (1 - \pi)(1 - p), \tag{6}$$

$$P(J_i = 0) = (1 - \pi)p + \pi(1 - p). \tag{7}$$

By correcting the above results, the likelihood function can be constructed as follow:

$$L = [\pi p + (1 - \pi)(1 - p)]^{n_1}[(1 - \pi)p + \pi(1 - p)]^{n - n_1}. \tag{8}$$

Maximum likelihood estimation of π :

$$\hat{\pi} = \frac{p - 1}{2p - 1} + \frac{n_1}{(2p - 1)n}. \tag{9}$$

Mathematic expectations for $\hat{\pi}$:

$$E(\hat{\pi}) = \frac{1}{2p - 1}[p - 1 + \pi p + (1 - \pi)(1 - p)] = \pi. \tag{10}$$

It follows that $\hat{\pi}$ is an unbiased estimate of π. Therefore, a corrected estimate of the number of vehicle ends for which the bit is 1 can be obtained as follows:

$$N = \hat{\pi} n = \frac{p - 1}{2p - 1}n + \frac{n_1}{2p - 1}. \tag{11}$$

Thus, the binary decoding process is as follows:

- The binary string b'_k is received by the central server from the vehicles.
- The value 1 corresponding to each bit i of b'_k in the central server is summed to obtain B'_k.
- Each B'_k is corrected to obtain the statistical value $\mathrm{T_k}$:

$$T_k = \frac{p - 1}{2p - 1}n + \frac{B'_k}{2p - 1}. \tag{12}$$

- The model parameter w_k is obtained by aggregating the binary string T according to its corresponding weights.

4.3. Model Aggregation Update

The vehicle model parameters w_k in the central server are collected and aggregated on average:

$$\bar{w} = \frac{\sum_{k=1}^{K} w_k}{K}. \tag{13}$$

4.4. BCFL-LDP Method

Federated learning is proposed on the basis that participants retain local data. However, the attackers can obtain private data of vehicles by deducing model gradient or each updated model parameter, which is still privacy leakage in essence. Therefore, if leaks occur in the data transmission of participants (such as vehicles) and the central server, the privacy and security of users cannot be protected.

To solve this problem, the BCFL-LDP method proposed in this paper applies local differential privacy technology based on a random response mechanism. The BCFL-LDP method is mainly divided into three steps to achieve the protection for private data. First of all, binary conversion coding and random disturbance processes are performed on the model parameters that need to be protected in the vehicle. Then, the received binary string is decoded, corrected, and averaged in the central server according to different disturbance methods. Finally, the results are summarized in the server, and the model parameters are aggregated (Algorithm 1).

Algorithm 1 Traffic sign recognitin based on BCFL-LDP.

```
Input: A set of raw provided by data owners $X = \{x_k\}_{k=1}^K$
    Participated data owners $V = \{v_k\}_{k=1}^K$
    Deep learning model $h(x_k, w)$
Output: A trained global object detection model M
Initialize model parameter $w_0$, maximum number of learning round R
for int r=1 to R do
    for each vehicle $v_k \in V$ do
        for each local epoch from 1 to $E$ do
            Update model parameters by Equation (4)
        end for
        Convert model parameter $w_k$ to binary string $b_k$
        Add random perturbations from $b_k$ to $b'_k$
        Send $b'_k$ to server CS
    end for
    CS sums the number 1 of each corresponding bit in all $b'_k$ to $B'_k$
    Correct $B'_k$ to $T_k$ by Equation (12)
    Convert $T_k$ to $w_k$ by proportion value
    Aggregat $w_i$ to obtain $\overline{w_t}$ by Equation (13)
    if the model does not converge then
        goto 14
    else
        send the updated $\overline{w_t}$ to all vehicles $v_i$
    end if    The model parameters of all vehicles and the CS are updated once
end for    The trained model is designated as the global object detection model M
return M
```

5. Experiment and Analysis

5.1. Dataset and Experiment Setup

We consider a dataset named the German Traffic Sign Detection Benchmark (GTSDB). It contains 51,840 traffic sign images with 43 categories, among which 39,210 images constitute the training set and 12,630 images constitute the test set. In this experiment, each vehicle randomly samples the entire dataset.

A central server and 20 candidate vehicles are considered to constitute a vehicular network system. The central server has a huge computing capacity, which can aggregate the obtained model parameters globally and finally transmit them to the vehicles for updating. Candidate vehicles have different categories of data. The size of the dataset owned by either of them is the same. When the number of vehicles involved in training is [1,5,10,15,20], the convergence times and classification accuracy of the model are analyzed, respectively.

In order to evaluate the advantages and disadvantages of the proposed method, the experimental results of three commonly used methods are compared. This experiment is completed on a Windows 10 system, NVIDIA GeForce RTX 3050 Laptop GPU.

In the experiments, the CNN model was chosen for its powerful ability to extract image features. The learning rate was set to 0.0001 to stably and efficiently approach the optimal solution. The batch size was set to 200 to balance computational efficiency and memory usage while enhancing the model's generalization capabilities. The maximum number of training epochs was set to 500 to ensure that the model learns sufficiently and converges.

5.2. Experiment Results

In order to validate the significance of the results, the proposed BCFL-LDP method was compared with the traditional baselines of the convolutional neural network (CNN), random forest (RF), and backpropagation (BP) methods. The superiority of the BCFL-LDP method was validated from three perspectives: convergence time, loss, and accuracy.

We first evaluated the learning efficiency of this method. The learning rate was set to 0.0001, the batch was 200, and the maximum number of learning rounds was 500.

Figure 2 shows the differences in model accuracy of the four methods with the increase in the number of training vehicles. As can be seen from the figure, the proposed BCFL-LDP had the best accuracy in all cases. When only a small number of vehicles were involved, the model maintained good accuracy, while the accuracy of the baseline methods decreased significantly. Meanwhile, the error rate of the proposed BCFL-LDP was only 4.01%, which is 56% lower than that of BP and RF and 68% lower than that of CNN.

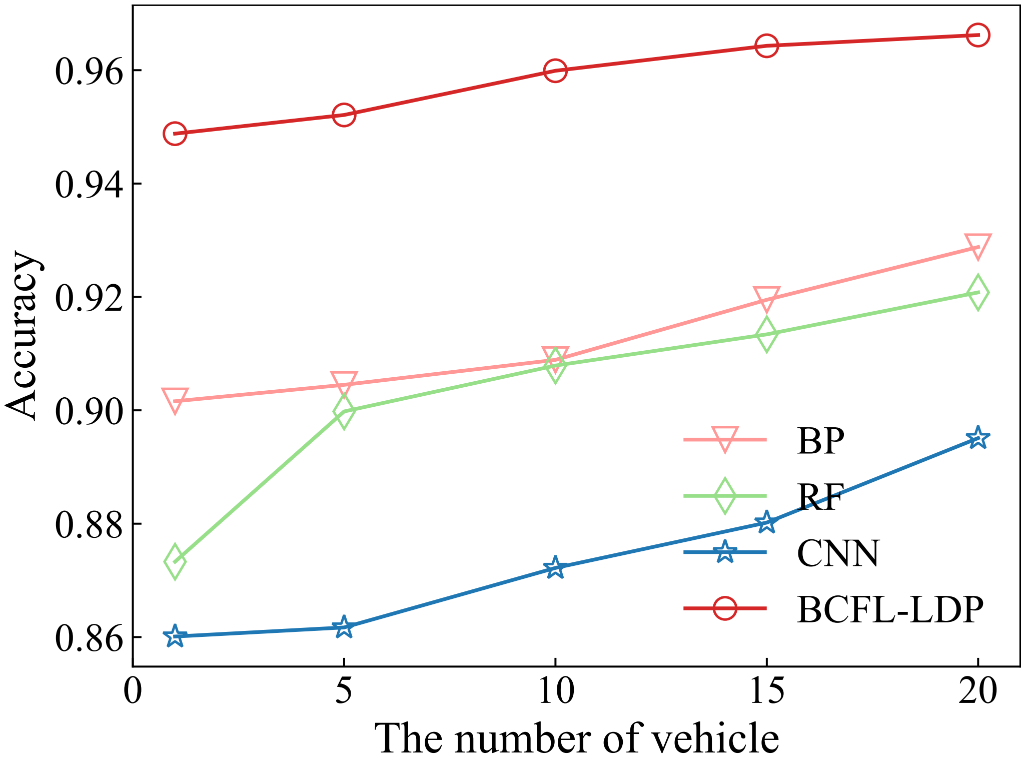

Figure 2. The variation of global model training accuracy of four different methods with a different number of training vehicles is illustrated. As the number of vehicles participating in the training increases, the training accuracy of each method increases. At the same time, the proposed method maintains the highest accuracy regardless of the number of training vehicles.

Figure 3 compares the model convergence time of the proposed method with the other three baseline methods under different numbers of training vehicles. The four methods generally reach a state of convergence around 100 epochs, with the proposed BCFL-LDP method exhibiting the lowest loss value. It can be seen from Figure 3 that BCFL-LDP has a long convergence time when there are fewer participating vehicles, which is roughly the same as BP. When only five vehicles are involved, the convergence time of BCFL-LDP is increased by 7% compared with CNN, which has the fastest convergence. The convergence time of BCFL-LDP and the three baseline methods is approximately the same when there are more vehicles involved. When the number of participating vehicles is 15, the convergence time of BCFL-LDP only increases by 0.4% longer than that of CNN, with the fastest convergence.

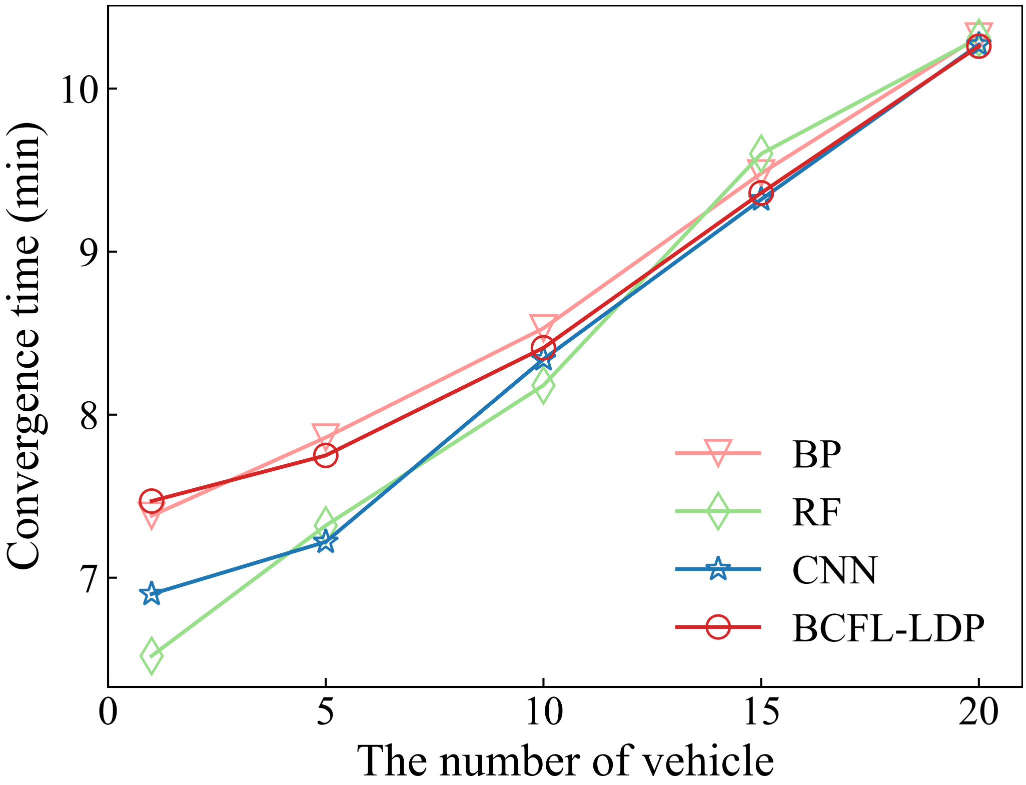

Figure 3. The convergence time variation of four different methods with different number of training vehicles is illustrated. When the number of training vehicles is 10, the convergence time gap between the proposed BCFL-LDP and the three baseline methods starts to become small.

Figures 4 and 5 show the comparison results of the influence of BCFL-LDP and three baseline methods on accuracy and loss when the training vehicle is 10. As can be seen from the figure, the number of learning rounds required for BCFL-LDP convergence is the least. This is not inconsistent with the above experimental results because each learning round of BCFL-LDP consumes more time. At the same time, the accuracy of BCFL-LDF is the best (up to 96.6%) when convergence is achieved. From Figure 5, it can be seen that as the training epochs increase, the accuracy of the BCFL-LDP method always remains the highest, and the convergence is very smooth. Finally, a comparison table was drawn for the convergence time and model accuracy of four methods with a vehicle quantity of 10, as shown in Table 1. Based on the benchmark results of the most classic CNN method, the BCFL-LDP method has 10% higher accuracy, while the convergence times of the four methods are relatively similar.

Table 1. Comparisons on traffic sign recognition performance.

Model	**Methods**			
Performance	***CNN***	***RF***	***BP***	***BCFL-LDP***
Accuracy	1	104%	104%	110%
Convergence time	1	98%	102%	101%

Finally, we compared four traffic sign recognition methods when the training vehicle is 10. Based on Table 1, the proposed BCFL-LDP had the highest accuracy, which was 110% of the CNN. The convergence time of BCFL-LDP was 101% of the CNN. This shows that compared with other baseline methods, the proposed BCFL-LDP has obvious advantages in traffic sign recognition, with a slight sacrifice in time, but the cost is acceptable.

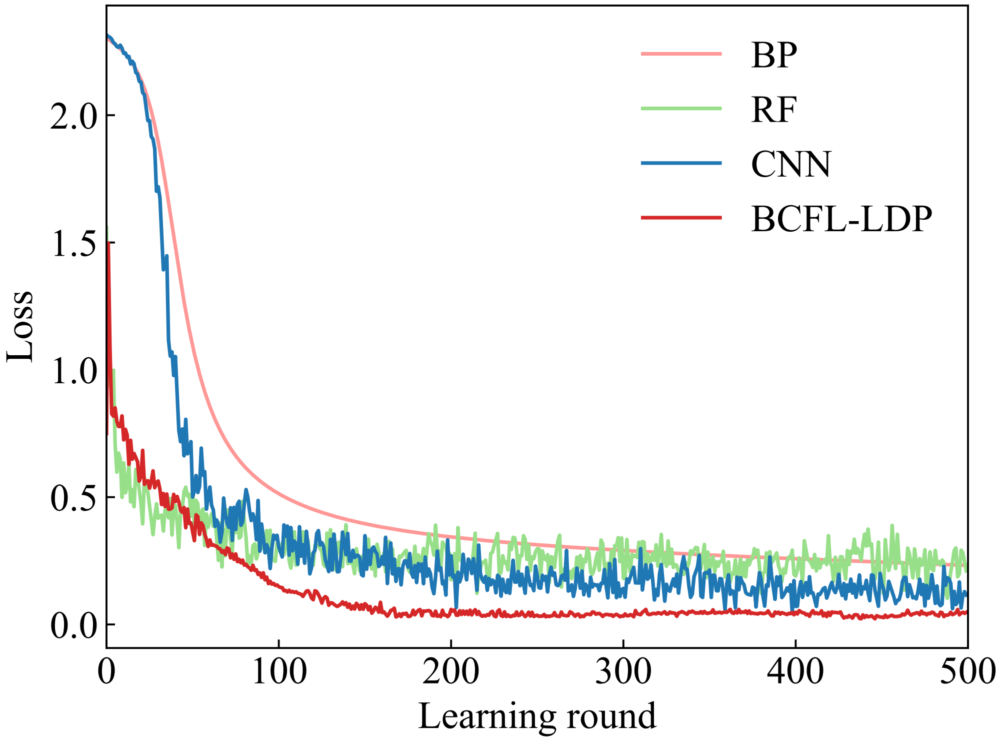

Figure 4. The change trend of model loss with an increasing number of learning rounds was compared between the proposed method and the three baseline methods. When the number of learning rounds was 100, the loss of the four methods was below 0.5 and tends to converge, and the loss of the proposed method was the lowest.

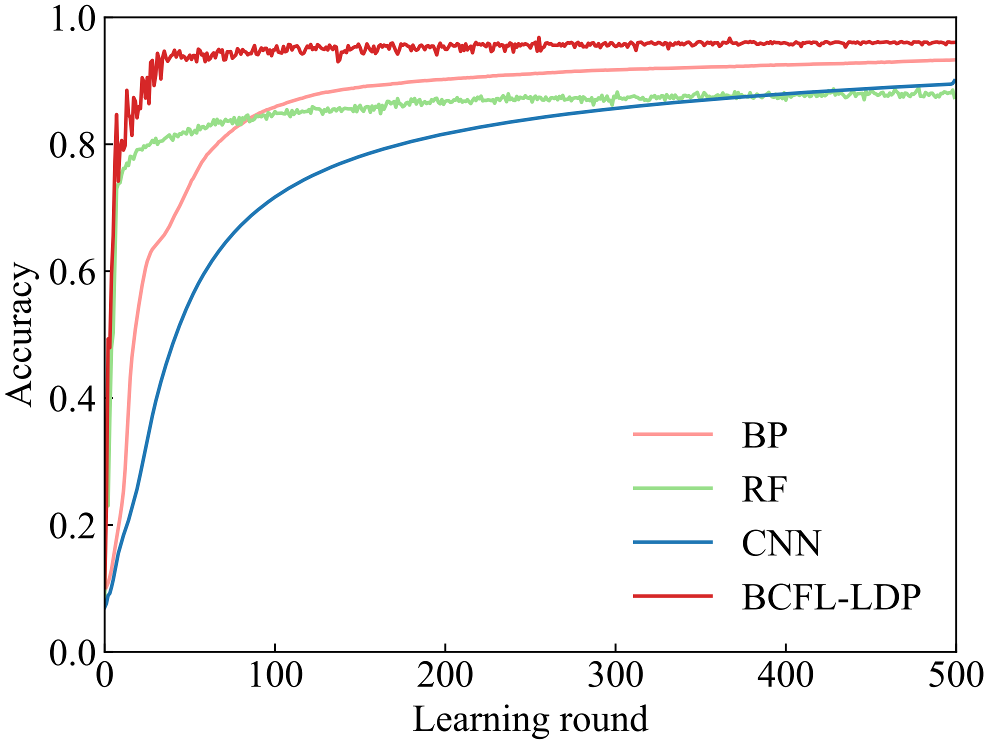

Figure 5. The variation trend of model accuracy with the increase of learning rounds was compared between the proposed method and the three baseline methods. As the number of learning rounds increased, the model accuracy of each method increased and gradually converged, and the proposed BCFL-LDP always achieved the highest accuracy.

6. Conclusions

In this paper, a federated averaging method based on the model parameter averaging was proposed for intelligent traffic sign recognition in autonomous driving. This method has a generalization ability and aims to achieve faster model convergence speed and higher training accuracy.

In particular, a federated learning method based on local differential privacy technology was proposed, which uses a locally trained model and only uploads parameters to the central server. This method can protect the privacy of vehicle users and effectively improve the recognition accuracy of model training. On this basis, a new random perturbation method based on binary bit string was proposed. An integrated BCFL-LDP was also constructed. Before model parameters were uploaded to the central server, binary encoding and disturbance were carried out. Therefore, even if parameters are leaked, no privacy information of vehicle users will be exposed to the attacker. However, in extremely complex or highly variable traffic environments, the recognition performance of BCFL-LDP may be affected.

The experimental results show that the proposed BCFL-LDP has higher training accuracy (up to 96%), approximately 6% higher than the three baseline methods, and has a better convergence effect.

In the future, with the continuous development of autonomous driving technology, it is necessary to further improve the real-time processing capability and computational efficiency of BCFL-LDP and consider the crossing pedestrians. Meanwhile, exploring more advanced privacy protection techniques and optimization algorithms to further enhance the robustness and scalability of BCFL-LDP is also a topic worthy of in-depth research.

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

Funding: This research was funded by the Hunan Provincial Department of Education Scientific Research Project of China: 23B0898.

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 the Associate Editor and the anonymous reviewers for their suggestions which have improved the quality of the work.

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

References

1. Garg, S.; Kaur, K.; Kaddoum, G.; Ahmed, S.H.; Jayakody, D.N.K. SDN-based secure and privacy-preserving scheme for vehicular networks: A 5G perspective. *IEEE Trans. Veh. Technol.* **2019**, *68*, 8421–8434. [CrossRef]
2. Yang, F.; Wang, S.; Li, J.; Liu, Z.; Sun, Q. An overview of internet of vehicles. *China Commun.* **2014**, *11*, 1–15. [CrossRef]
3. Lin, X.; Sun, X.; Ho, P.; Shen, X. GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Trans. Veh. Technol.* **2007**, *56*, 3442–3456.
4. Zhou, X.; Liang, W.; She, J.; Yan, Z.; Wang, K.I.-K. Two-Layer Federated Learning with Heterogeneous Model Aggregation for 6G Supported Internet of Vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 5308–5317. [CrossRef]
5. Zhou, X.; Chen, M.; Peng, L.; Li, H. ACPP: An effective privacy preserving scheme for precise location sharing in internet of vehicles. In Proceedings of the 2017 IEEE International Conference on Information and Automation (ICIA), Macau, China, 18–20 July 2017; pp. 883–887.
6. Wei, F.; Zeadally, S.; Vijayakumar, P.; Kumar, N.; He, D. An Intelligent Terminal Based Privacy-Preserving Multi-Modal Implicit Authentication Protocol for Internet of Connected Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3939–3951. [CrossRef]
7. Sermanet, P.; LeCun, Y. Traffic sign recognition with multi-scale convolutional networks. In Proceedings of the 2011 International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 2809–2813.
8. Yao, C.; Wu, F.; Chen, H.-J.; Hao, X.-L.; Shen, Y. Traffic sign recognition using hog-svm and grid search. In Proceedings of the 2014 12th International Conference on Signal Processing (ICSP), Hangzhou, China, 19–23 October 2014; pp. 962–965.

9. Kuang, X.; Fu, W.; Yang, L. Real-Time Detection and Recognition of Road Traffic Signs Using MSER and Random Forests. *Int. J. Online Eng.* **2018**, *14*, 34–51. [CrossRef]
10. Yuan, X.; Chen, J.; Zhang, N.; Fang, X.; Liu, D. A federated bidirectional connection broad learning scheme for secure data sharing in Internet of Vehicles. *China Commun.* **2021**, *18*, 117–133. [CrossRef]
11. Aadhavan, A.; Ahmed, A.; Vellaian, V.M.; Dhanush, K.P.; Rajkumar, S. Prediction and classification of traffic data with KNN and RFR for a smart internet of vehicles system. In Proceedings of the 4th Smart Cities Symposium (SCS 2021), Online, 21–23 November 2021; pp. 146–151.
12. Manoj, T.; Makkithaya, K.; Narendra, V.G. A Federated Learning-Based Crop Yield Prediction for Agricultural Production Risk Management. In Proceedings of the 2022 IEEE Delhi Section Conference (DELCON), New Delhi, India, 11–13 February 2022; pp. 1–7.
13. Nasajpour, M.; Karakaya, M.; Pouriyeh, S.; Parizi, R.M. Federated Transfer Learning For Diabetic Retinopathy Detection Using CNN Architectures. *SoutheastCon* **2022**, *2022*, 655–660.
14. Abdel-Basset, M.; Moustafa, N.; Hawash, H.; Razzak, I.; Sallam, K.M.; Elkomy, O.M. Federated Intrusion Detection in Blockchain-Based Smart Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 2523–2537. [CrossRef]
15. Li, L.; Fan, Y.; Tse, M.; Lin, K.Y. A review of applications in federated learning. *Comput. Ind. Eng.* **2020**, *149*, 106854. [CrossRef]
16. Dwork, C. Differential privacy: A survey of results. In Proceedings of the International Conference on Theory and Applications of Models of Computation, Xi'an, China, 25–29 April 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–19.
17. Tchaye-Kondi, J.; Zhai, Y.; Shen, J.; Zhu, L. Privacy-Preserving Offloading in Edge Intelligence Systems with Inductive Learning and Local Differential Privacy. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 5026–5037. [CrossRef]
18. Mills, J.; Hu, J.; Min, G. Multi-Task Federated Learning for Personalised Deep Neural Networks in Edge Computing. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 630–641. [CrossRef]
19. Lin, K.P.; Chang, Y.W.; Chen, M.S. Secure support vector machines out sourcing with random linear transformation. *Knowl. Inf. Syst.* **2015**, *44*, 147–176. [CrossRef]
20. Ünal, A.B.; Akgün, M.; Pfeifer, N. Escaped: Efficient secure and private dot product framework for kernel-based machine learning algorithms with applications in healthcare. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual Event, 2–9 February 2021; Volume 35, pp. 9988–9996.
21. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. *arXiv* **2016**, arXiv:1602.05629.
22. Zhu, Y.; Liu, Y.; Yu, J.J.Q.; Yuan, X. Semi-Supervised Federated Learning for Travel Mode Identification from GPS Trajectories. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 2380–2391. [CrossRef]
23. Yu, Z.; Hu, J.; Min, G.; Zhao, Z.; Miao, W.; Hossain, M.S. Mobility-Aware Proactive Edge Caching for Connected Vehicles Using Federated Learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 5341–5351. [CrossRef]
24. Poddar, M.; Ganta, S.; Swaraj, K.R.; Das, D. Privacy in the Internet of Vehicles: Models, Algorithms, and Applications. In Proceedings of the 2019 International Conference on Information Networking (ICOIN), Kuala Lumpur, Malaysia, 9–11 January 2019; pp. 78–83.
25. Wang, S.; Li, J.; Wu, G.; Chen, H.; Sun, S. Joint Optimization of Task Offloading and Resource Allocation Based on Differential Privacy in Vehicular Edge Computing. *IEEE Trans. Comput. Soc. Syst.* **2022**, *9*, 109–119. [CrossRef]
26. Nie, Y.; Zhao, J.; Gao, F.; Yu, F.R. Semi-Distributed Resource Management in UAV-Aided MEC Systems: A Multi-Agent Federated Reinforcement Learning Approach. *IEEE Trans. Veh. Technol.* **2021**, *70*, 13162–13173. [CrossRef]
27. Karim, H.; Rawat, D.B. TollsOnly Please—Homomorphic Encryption for Toll Transponder Privacy in Internet of Vehicles. *IEEE Internet Things J.* **2022**, *9*, 2627–2636. [CrossRef]
28. Thant, M.; Zaw, T.M. Authentication Protocols and Authentication on the Base of PKI and ID-Based. In Proceedings of the 2018 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), St. Petersburg, FL, USA, 26–30 November 2018; pp. 1–8.
29. Peng, H.; He, M.; Li, L.; Yang, Y. A New Lightweight Key Exchange Protocol Based on T—Tensor Product. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; pp. 736–741.
30. Boicea, A.; Radulescu, F.; Truica, C.-O.; Costea, C. Database Encryption Using Asymmetric Keys: A Case Study. In Proceedings of the 2017 21st International Conference on Control Systems and Computer Science (CSCS), Bucharest, Romania, 29–31 May 2017; pp. 317–323.
31. Wu, X.; Zhu, X.; Kong, F. Routing and Data Security Scheme Based on Double Encryption in Mobile Ad Hoc Networks. In Proceedings of the 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, China, 18–20 September 2015; pp. 1787–1791.

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.