Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

FLM-ICR: a federated learning model for classification of internet of vehicle terminals using connection records

Kai Yang[1], Jiawei Du[1*], Jingchao Liu[1], Feng Xu[2], Ye Tang[3], Ming Liu[4] and Zhibin Li[4]

Abstract

With the rapid growth of Internet of Vehicles (IoV) technology, the performance and privacy of IoV terminals (IoVT) have become increasingly important. This paper proposes a federated learning model for IoVT classification using connection records (FLM-ICR) to address privacy concerns and poor computational performance in analyzing users' private data in IoV. FLM-ICR, in the horizontally federated learning client-server architecture, utilizes an improved multi-layer perceptron and logistic regression network as the model backbone, employs the federated momentum gradient algorithm as the local model training optimizer, and uses the federated Gaussian differential privacy algorithm to protect the security of the computation process. The experiment evaluates the model's classification performance using the confusion matrix, explores the impact of client collaboration on model performance, demonstrates the model's suitability for imbalanced data distribution, and confirms the effectiveness of federated learning for model training. FLM-ICR achieves the accuracy, precision, recall, specificity, and F1 score of 0.795, 0.735, 0.835, 0.75, and 0.782, respectively, outperforming existing research methods and balancing classification performance and privacy security, making it suitable for IoV computation and analysis of private data.

Keywords Internet of vehicles, Federated learning, Differential privacy, Data security

Introduction

The Internet of Vehicles (IoV) is a network system that connects cars with other objects (such as mobile phones, computers, roads, traffic lights, and pedestrians) using wireless communication and information exchange technologies. At its core is a traffic information network control platform that extracts and utilizes the attributes and static and dynamic information of all vehicles through sensors on each car, enabling effective monitoring of vehicle status and providing comprehensive services based on different needs. IoV has been widely applied in distance assurance and real-time navigation, significantly improving traffic efficiency [1]. However, the development of IoV relies on the big data generated by users and their vehicles, which presents challenges in data collection, transmission, and analysis. Firstly, there is a lack of security, which may involve the risk of privacy breaches [2, 3], and secondly, there is uneven resource allocation, which may lead to service unfairness [4]. To overcome these challenges, IoV needs to strengthen privacy protection measures to ensure the security of user data while also considering differences in data distribution and establishing a fair resource allocation mechanism. When mining sensitive data, it is necessary to extract usable features without revealing privacy and to use privacy-preserving machine learning (ML) algorithms to balance

*Correspondence:
Jiawei Du
d17829393900@163.com
[1] School of Computer Science, Xijing University, Xi'an 710123, China
[2] Guangzhou Institute of Technology, Xidian University, Guangzhou 510555, China
[3] Beijing Special Electromechanical Research Institute, Beijing 100020, China
[4] The 7th Research Institute of China Electronics Technology Group Corporation, Guangzhou 510310, China

learning content and privacy security. For example, extracting helpful information while protecting patient privacy in medical research is necessary. The method to address this issue is to extract general features without disclosing personal privacy, requiring privacy-preserving machine learning algorithms to balance learning objectives and privacy security [5]. In 2016, Google proposed a privacy-preserving learning framework called Federated Learning (FL), which features data providers keeping their data locally, thus suppressing data privacy leakage from the source [6–10]. As a mainstream privacy computing method, FL is a shared ML algorithm with good learning performance. Additionally, FL uses differential privacy (DP) [11] to protect the privacy of the computing process, preventing privacy information leakage and utilizing a large amount of user data for model training.

In recent years, various research methods have emerged. In the study of data distributions, Nilsson et al. [12] conducted a benchmark study on the MNIST dataset, comparing the performance of three FL algorithms using both IID and non-IID data partitions against centralized methods. Li et al. [13] proposed a comprehensive data partition strategy to address non-IID data cases in FL and better understand non-IID data settings. In the study of privacy protection technologies, Abadi et al. [14] proposed the GDP-FL learning algorithm, which combines DP to train models, protect gradient information, and conduct refined privacy cost analysis within the DP framework. Mahawaga Arachchige et al. [15] introduced the LATENT local DP algorithm, providing privacy protection when interacting with untrusted ML services. It enables data owners to add a randomization layer before data leaves their devices. In the study of FL algorithms, Choudhury et al. [16] presented an FL framework where a global model is trained on distributed health data protected by a DP mechanism against potential privacy attacks. Yang et al. [17] proposed the PLU-FedOA algorithm, optimizing horizontal FL deep neural networks with individualized local DP. Lu et al. [18] proposed a new FL-based architecture, comprising a hybrid blockchain architecture composed of permissioned blockchain and locally directed acyclic graph, and suggested an asynchronous FL scheme. Yang et al. [19] proposed an efficient asynchronous FL algorithm and a dynamic hierarchical aggregation mechanism utilizing gradient sparsification and asynchronous aggregation techniques. In the study of IoV applications, Zhao et al. [20] designed an FL collaborative authentication protocol to prevent private data leakage and reduce data transmission delay for vehicle clients sharing data. Luo et al. [21] addressed the issue of private data leakage in smart cars within the IoV network by introducing a local DP algorithm and designing a data privacy protection scheme tailored to IoV characteristics. Bakopoulou et al. [22] applied FL to mobile packet classification, enabling collaboration among mobile devices to train a global model without sharing raw training data.

From the current research work, it can be concluded that there are a series of problems to be solved at present: the existing deep learning algorithm has the risk of leakage when training a large amount of private data, even if the classification performance is good, it cannot consider data privacy. The existing privacy-preserving FL algorithm provides low security, slow training speed, and cannot balance performance and security. Accordingly, this paper aims to build an FL classification model (FLM-ICR) that balances privacy protection and performance to analyze the Internet of Vehicles terminals (IoVT)' connection records, verify the terminal device's function, and dynamically monitor users' normal usage. The FL and ML methods combination in FLM-ICR brings unique advantages to IoVT applications, including protecting user privacy, improving model accuracy and performance, and providing real-time responsiveness and adaptability. This combination promotes the development and innovation of IoVT, providing users with a better driving experience and services. This paper's innovation lies in using skew classes to divide the dataset into four clients by simulating the non-IID data distribution [23–25] in practical application scenarios. Based on the client-server architecture of horizontal FL, the federated Gaussian differential privacy (federated GDP) algorithm is used in the client and server to double protect the security of FL training. The Federated Momentum Gradient Descent (MGD) algorithm [26] is used in local model training to speed up convergence, and an improved multilayer perceptron (MLP) [27] and logistic regression (LR) [28] network is used as the model backbone to improve classification performance. These measures solve privacy leakage problems, low-security protection levels, low efficiency, and poor classification performance in the current research. FLM-ICR securely analyzes shared data in IoVT application scenarios, providing a new direction for the research of privacy-preserving FL. The main contributions of this paper are as follows:

- Using the improved MLP and LR networks as the backbone of FLM-ICR enables better handling of classification problems. It is simple to implement, facilitating the integration of FL for update training.
- Adopting the federated MGD algorithm as the training optimizer accelerates convergence in the local model updates of FLM-ICR and avoids local optima, making it convenient to use and achieving efficient computation.

- Under the client-server architecture of horizontal FL, adopting the federated GDP algorithm safeguards the security of the FL calculation process. It can balance the classification performance and security of FLM-ICR.

The organization structure of the article is as follows: the first part is the introduction, which introduces the background and significance of the research in this field and the related work; the second part is the preliminary knowledge, which introduces the theoretical knowledge of FL and DP; the third part is the proposed methodology, which details the data collection module, federated learning control module, differential privacy training module and classification prediction module in the FLM-ICR model framework; the fourth part is the simulation experiment, which introduces the preliminary preparation, model evaluation, training result, and comparative experiment; the fifth part is the conclusion, which summarizes the full text and looks forward to the subsequent work.

Preliminary

Federated learning

FL, a distributed ML framework, enables data sharing and joint modeling while ensuring data privacy and legal compliance. It includes horizontal FL [29, 30], vertical FL, and federated transfer learning. Horizontal FL involves more overlap in sample features and less overlap in sample sources across multiple sets. In comparison, vertical FL involves less overlap in sample features and more overlap in sample sources. Federated transfer learning applies models learned in one field to another based on data, task, or model similarities. The schematic diagram of the FL classification is shown in Fig. 1.

Since the dataset used for model training in this paper is the connection records of different users under the same type of IoVT, the samples conform to the characteristics of more feature overlap and less source overlap and belong to the horizontal FL model. The main methods to protect privacy and security in FL are homomorphic encryption, secure multi-party computation, and DP. Considering the communication overhead, accuracy, and privacy protection degree comprehensively, the DP method is selected to protect privacy in the FL calculation process.

Differential privacy

DP, as a widely used privacy protection algorithm, uses the technology of adding noise to distort sensitive data, ensuring that deleting or adding a piece of data in the dataset will not affect the query results. DP protects data availability while significantly reducing the risk of privacy leakage. The original definition is: for two datasets D and D' on the independent variable space X that differ only by one data, if there is a random algorithm $A(x), x \in X$, such that any output set S is obtained. There is:

$$\Pr\left[A(D) \in S\right] \leq e^{\epsilon} \Pr\left[A\left(D'\right) \in S\right] + \delta \tag{1}$$

Where Pr is the probability function, ϵ is the privacy budget with $\epsilon > 0$, δ is the disturbance with $0 < \delta < 1$, and the random algorithm $A(x)$ satisfies relaxed $(\epsilon, \delta)-$DP on dataset X. The smaller ϵ and δ are, the closer $\Pr\left[A(D) \in S\right]$ and $\Pr\left[A\left(D'\right) \in S\right]$ are, the smaller the perturbation difference, the better the effect of DP. At this time, the difference between D and D' cannot be inferred from the outputs of $A(D)$ and $A\left(D'\right)$, thus protecting the privacy information of the dataset.

This paper adopts a noise-based DP algorithm, which is divided into global DP and local DP. The model finally constructed in this paper is used in the FL scenario, so the federated GDP method is adopted. The Gaussian method [31–33] is designed as a random algorithm $A(x)$ to protect the gradient in the model training process, and the privacy of the FL calculation process is protected by adding Gaussian noise to perturb the model. Set the gradient clipping boundary value C and the standard deviation σ of Gaussian distribution in DP, and the privacy budget ϵ is negatively correlated with noise. Define:

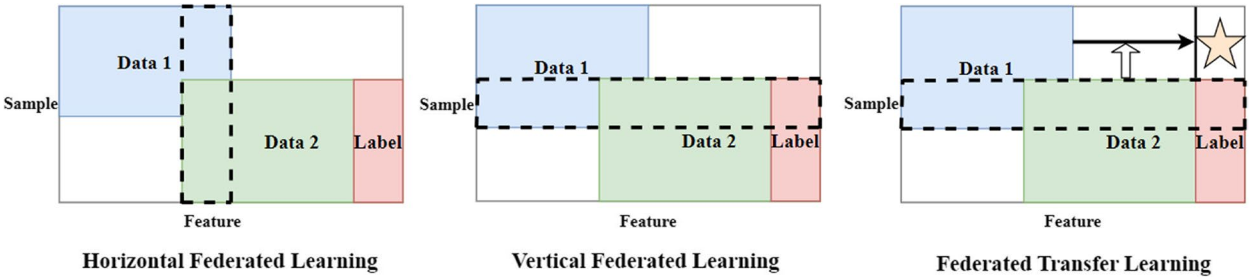


Fig. 1 Schematic diagram of FL classification

$$A(x) = f(x) + Y \tag{2}$$

Where $f(x)$ is the mapping function, and Y is the noise that satisfies $Y \sim N\left(0, \sigma^2\right)$ Gaussian distribution sampling. Local DP based on Gaussian noise solves the upper bound problem of learnable content and loss function in DP methods. It has the characteristics of easy implementation and is lightweight.

Proposed methodology

FLM-ICR, as the FL model used for classifying IoVT using connection records, consists of four main components in its model framework: the data collection module, federated learning control module, differential privacy training module, and classification prediction module, as illustrated in Fig. 2. The key to the model implementation lies in the differential privacy training module, which clips the parameter gradients computed by the local model and adds Gaussian noise. Gradients reflect the training dataset because their values are computed based on the dataset, containing information from it. Compared to using DP at the input stage or the objective function, it is easier to analyze the privacy protection of the gradients. By perturbing the gradients, subsequent parameter update operations can be ensured not to leak data information, thus protecting data privacy and training locally DP-protected models. The federated learning control module randomly selects clients to obtain the current global model, updates the local client model, and aggregates the models on the server using the MGD algorithm for optimization.

This comprehensive framework is visually depicted in Fig. 2, clearly illustrating the interplay between these essential components.

The data collection module is designed to systematically gather a wide array of connection record data from IoVT by leveraging the diverse range of Internet-connected devices integrated into vehicles. These devices encompass in-vehicle communication systems and an assortment of sensors, enabling the collection of crucial information such as communication times, intricate vehicle behaviors, and data transmission. This comprehensive data collection process ensures that a rich and detailed dataset is obtained, facilitating in-depth analysis and insights into the functioning and interactions of IoVT.

The federated learning control module is responsible for the seamless execution of FL algorithms on IoVT, ensuring the efficient coordination of model training and robust data privacy considerations. This is achieved through a series of intricate processes, including local computation, where individual IoVT devices perform computations on their local data, model aggregation, which involves the consolidation of locally trained models from multiple devices, model broadcasting, where the updated global model is distributed to the individual devices, and parameter updates, which involve refining the model parameters based on the aggregated information. This meticulous orchestration ensures that the FL algorithms operate effectively and that data privacy is rigorously maintained throughout the model training process.

The differential privacy training module harnesses the power of DP, an advanced data privacy protection technique, to safeguard the integrity and confidentiality of connection records and other critical vehicle information. This module incorporates sophisticated functionalities such as a noise generation mechanism, which introduces controlled randomness to the data to prevent the extraction of sensitive details, noise addition, which involves the deliberate injection of noise to obscure individual data points, and the calculation of the DP budget, ensuring that the level of privacy protection is carefully calibrated and maintained throughout the training process. By integrating these robust mechanisms, the module ensures that the privacy of sensitive information is upheld, thereby fortifying the security of the entire system.

The classification prediction module, by using the MLP and LR networks as the model backbone, can accurately classify normal and abnormal IoVT based on their

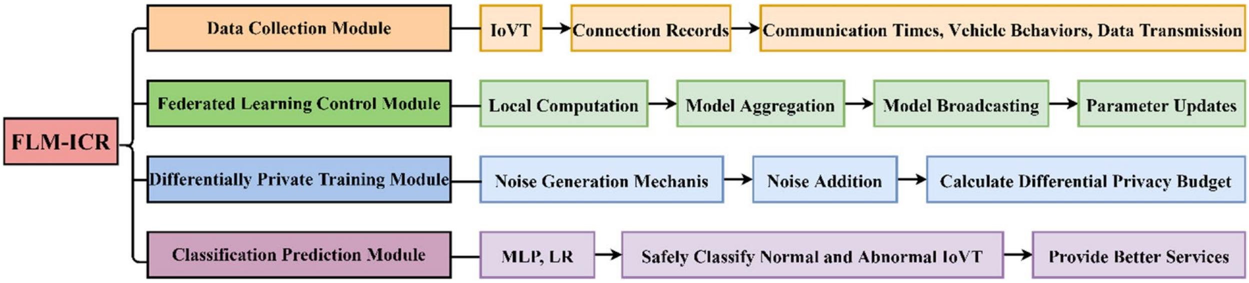

FLM-ICR
Data Collection Module
IoVT
Connection Records
Communication Times, Vehicle Behaviors, Data Transmission
Federated Learning Control Module
Local Computation
Model Aggregation
Model Broadcasting
Parameter Updates
Differentially Private Training Module
Noise Generation Mechanis
Noise Addition
Calculate Differential Privacy Budget
Classification Prediction Module
MLP, LR
Safely Classify Normal and Abnormal IoVT
Provide Better Services

Fig. 2 FLM-ICR model framework schematic diagram

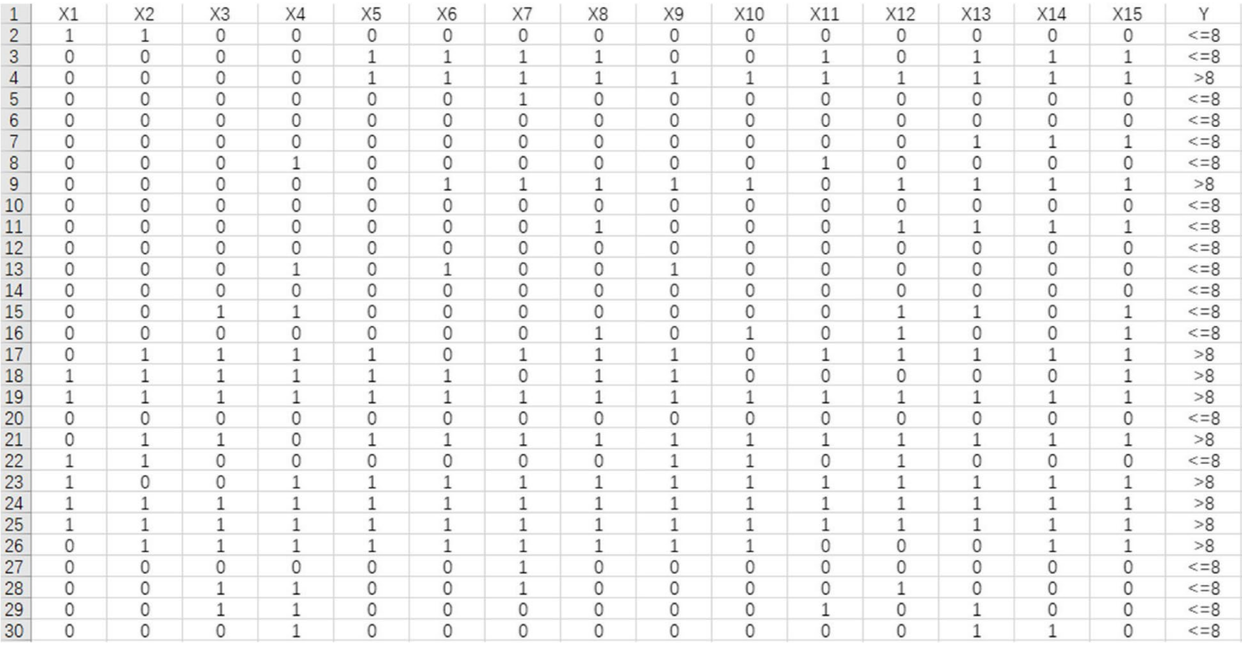
connection records after FL training. This advanced classification capability not only enhances the accuracy and efficiency of the classification process but also plays a crucial role in maintaining the security and privacy of personal data. By accurately identifying normal and abnormal IoVT based on their connection records, this module contributes to an elevated data security and integrity level, ultimately fostering an enhanced driving experience and service for users.

The subsequent sections will provide a comprehensive and detailed examination of the four modules within the model framework. This thorough analysis aims to elucidate each module's specific functions, interactions, and significance, offering a comprehensive understanding of their roles in the context of the FLM-ICR model.

Data collection module

The dataset studied in this paper comes from the user connection records generated by the GT101 terminal equipment of a car networking company. The IoVT user connection record is the interconnection information established between the terminal and the big data platform when the user utilizes the terminal. It has become a vital indicator for assessing the functionality of the terminal device. These records contain extensive information, including vehicle travel, driving behavior, and vehicle health status. The data within these records is typically generated in real-time, providing real-time updates on the vehicle's status and behavior. Regular maintenance by checking the number of days that the terminal equipment is usually connected is a crucial link in the work of the IoV. The dataset used in the experiment is the 15-day user connection records of 1500 IoV terminals (GT101). In order to protect the terminal device information and the privacy of the users, only the terminal number and connection status are kept, named the connection record dataset (101. csv). 101.csv has a total of 1,500 observations. The first 15 columns are used as independent variables (input x), and the extracted features are used to quantify the daily connection status, with no connection records recorded as 0 and those with connection records recorded as 1. The last column is used as the category value (output y), and the category values are <=8 (abnormal connection) and >8 (normal connection). The intercepted part of the dataset is shown in Fig. 3.

The CSV Dataset class serves as a loading class specifically designed for handling CSV datasets. It processes the data into a format suitable for the model. Then, it randomly divides it into a training set and a test set at a ratio of 8:2. This results in 1200 data in the training set and 300 in the test set. Subsequently, the training dataset is partitioned among four clients. In the case of an IID data distribution, 300 pieces are allocated to each client, ensuring that all clients possess an identical number of training data with the same category proportion. Conversely, a skewed class distribution is implemented for non-IID data distributions, resulting in each client receiving a distinct proportion of data from each class while



1	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	Y
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	<=8
3	0	0	0	0	1	1	1	1	0	0	1	0	1	1	1	<=8
4	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	>8
5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	<=8
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=8
7	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	<=8
8	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	<=8
9	0	0	0	0	0	1	1	1	1	1	0	1	1	1	1	>8
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=8
11	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	<=8
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=8
13	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	<=8
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=8
15	0	0	1	1	0	0	0	0	0	0	0	1	1	0	1	<=8
16	0	0	0	0	0	0	0	1	0	1	0	1	0	0	1	<=8
17	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	>8
18	1	1	1	1	1	1	0	1	1	0	0	0	0	0	1	>8
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	>8
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	<=8
21	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	>8
22	1	1	0	0	0	0	0	0	1	1	0	1	0	0	0	<=8
23	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	>8
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	>8
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	>8
26	0	1	1	1	1	1	1	1	1	1	0	0	0	1	1	>8
27	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	<=8
28	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	<=8
29	0	0	1	1	0	0	0	0	0	0	1	0	1	0	0	<=8
30	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	<=8

Fig. 3 IoVT connection record dataset

maintaining an equal total data allocation across all four clients. In scenarios involving unbalanced data distribution, the entire training sample is distributed among the four clients, with each client randomly receiving varying numbers and proportions of training data. It is important to note that the distribution of IID data represents an idealized scenario with no practical significance, as its usage is tantamount to centralized learning. This paper conducts model training on non-IID data distribution to ensure the model's applicability to real-world scenarios.

Federated learning control module

In each round of FL training of FLM-ICR, the client trains the initial model provided by the server and subsequently updates the training model based on its local dataset. Following this local training, the client perturbs the gradient by applying cropping and adding noise before uploading the updated gradient to the server. Upon receiving the updated gradients from the participating clients, the server aggregates these gradients to construct a new global model. Subsequently, the server broadcasts the new global model to each client, ensuring all clients have access to the most recent version of the global model. This iterative process of local model updates, gradient perturbation, global model aggregation, and model broadcasting continues until the conclusion of the training round. The detailed FL training process is visually depicted in Fig. 4.

The schematic diagram of the FL training process, as illustrated in Fig. 4, can be distinctly divided into five fundamental steps, each playing a pivotal role in the collaborative model training across distributed devices. These steps encompass the initial phase of local calculation, where individual devices perform computations on their local data to train the model; the subsequent phase of model perturbation, involving the deliberate introduction of noise or perturbation to the locally trained models to preserve privacy and prevent overfitting; the critical phase of model aggregation, where the locally trained models are consolidated to form an updated global model; the subsequent phase of model broadcast, where the refined global model is distributed to the individual

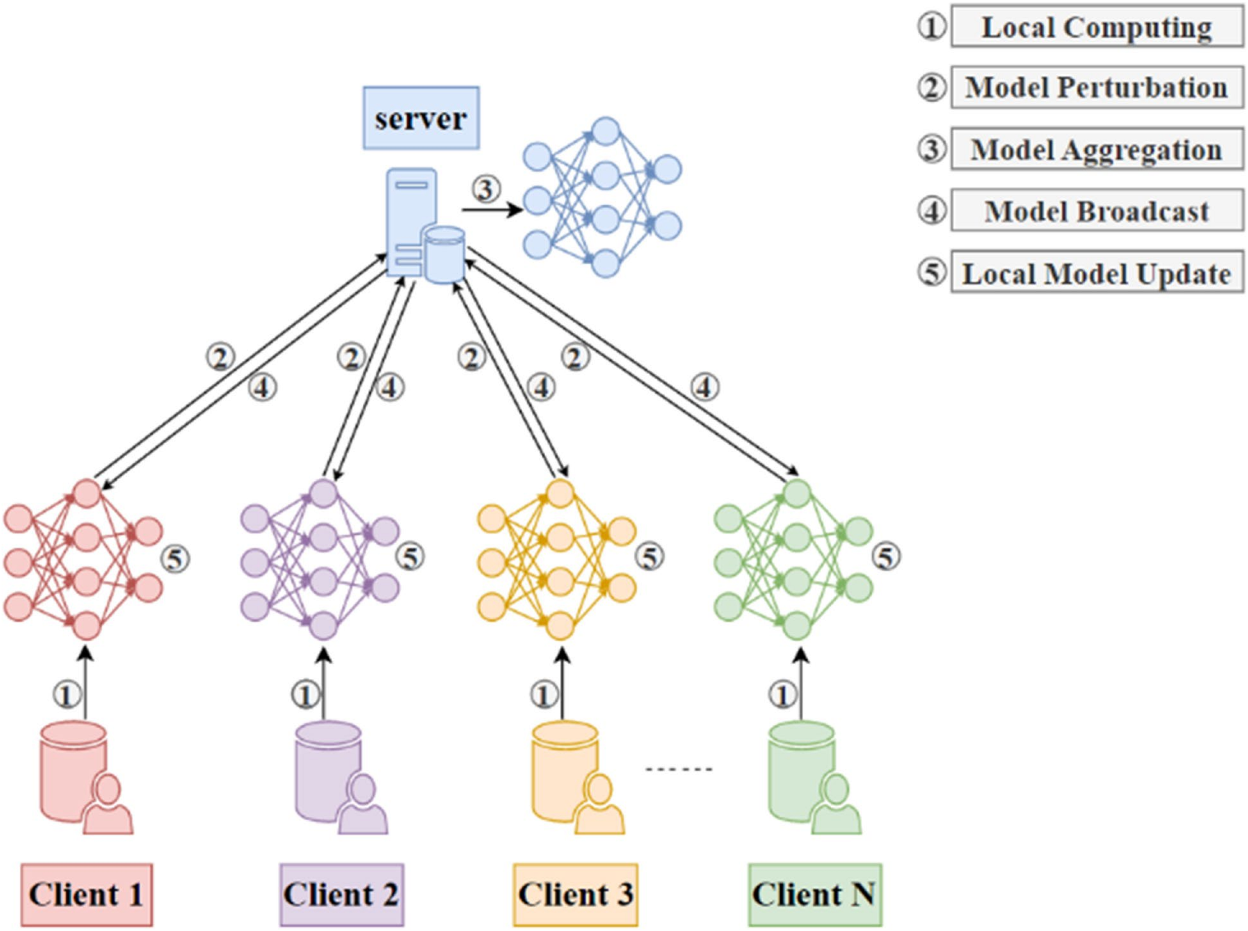


Fig. 4 FL training process schematic diagram

devices; and finally, the phase of the local model update, where the devices integrate the updated global model with their local insights, thereby refining the model based on the collective knowledge. This systematic breakdown provides a comprehensive overview of the intricate process underpinning FL model training.

> ① Local calculation: Client i utilizes its local database D_i and the accepted global model ω_G^t on the server side as a local parameter, denoted as $\omega_i^t = \omega_G^t$, where t represents the current boundary value. The client employs the federated MGD algorithm to iteratively update the local model parameters based on the gradients of the loss function to the model's parameters. This process aims to minimize the loss function and refine the local model, ultimately deriving an updated local model parameter ω_i^{t+1}. The client iteratively adjusts the local model's parameters by leveraging the federated MGD algorithm to enhance its performance and contribute to the collaborative FL process.
> ② Model perturbation: In the model perturbation phase, each client introduces a random noise component, denoted as n, which adheres to a Gaussian distribution. This noise is then incorporated into the local model by adding it to the updated local model parameter obtained in the previous step. Specifically, the perturbed local model parameter, represented as $\overline{\omega}_i^{t+1}$, is computed as the sum of the updated local model parameter ω_i^{t+1} and the noise component n. This perturbation process introduces controlled randomness to the local models, thereby enhancing privacy protection and preventing overfitting in the FL framework.
> ③ Model aggregation: The server utilizes the FedAvg algorithm to aggregate the perturbed model parameters, $\overline{\omega}_i^{t+1}$, received from the clients. This process involves computing the average of the perturbed model parameters from all participating clients to obtain a new global model parameter, denoted as $\overline{\omega}_G^{t+1}$. The FedAvg algorithm ensures that the updated global model parameter reflects the collective knowledge contributed by the individual clients while preserving data privacy and mitigating the impact of potential noisy updates. Subsequently, the average value of the model parameters is computed to update the model on the server side, ensuring that the global model reflects the collaborative insights derived from the FL process.
> ④ Model broadcast: The server takes the lead in disseminating the newly aggregated model parameters to each client participating in the FL process. This broadcast mechanism ensures that all clients receive the updated global model parameters, enabling them to synchronize their local models with the collective insights and refinements derived from the collaborative FL process. This synchronization process plays a pivotal role in fostering a cohesive and updated understanding of the model across all participating clients, ultimately contributing to the continual improvement and convergence of the global model.
> ⑤ Local model update: Each client initiates the process by updating its model parameters and recalculating locally. In the FL framework, the system iterates through randomly selected clients, enabling them to download the parameters of the trainable model from the server. Subsequently, the current global model is passed to the client, empowering them to update their local model based on locally available data. The client then performs local training to refine the model, ultimately returning the updated local model. Following this, the client uploads the new model parameters to the server, prompting the server to aggregate updates from multiple clients. This collaborative process continually improves the global model by integrating insights and refinements from the diverse network of participating clients.

Differential privacy training module

The federated GDP algorithm is employed in both the client and server to enhance the security of the FL calculation process, providing dual protection for both entities during their involvement in FL training. This protection encompasses two key aspects: the client-side federated GDP algorithm training and the server-side federated GDP algorithm aggregation. On the client side, the federated GDP algorithm ensures the privacy and security of the client's data during the FL training process. It employs advanced privacy-preserving mechanisms to safeguard sensitive information. This allows clients to securely contribute their local model updates without compromising the confidentiality of their data. On the server side, the federated GDP algorithm aggregates the client updates. It leverages secure aggregation protocols to combine the model updates from multiple clients while preserving privacy. This ensures that the server can effectively learn from the collective knowledge of the clients without accessing their data. FL training achieves a robust and privacy-preserving framework by employing the client-side federated GDP algorithm training and the server-side federated GDP algorithm aggregation. This approach protects the privacy of the client's data and enables collaborative learning across distributed devices, fostering advancements in ML while maintaining data security.

Training of the Client-Side Federated GDP Algorithm: Model training is executed on the client side, with each federated client possessing a fixed dataset and computing power to engage in federated MGD. Employing Algorithm 1 to process clients with identical network architecture and loss function, each local model is initialized by a global model from the server side. The number of iterations for the federated MGD algorithm aligns with the number of training epochs. Following each step of the local iterative update, the parameters are pruned, and the client computes the gradient update, generates the updated model, and shares it with the aggregation server. However, it is essential to note that local data is private to each client and is not shared. The client-based federated GDP algorithm is detailed in Algorithm 1.

Algorithm 1 Client-Based Federated GDP

```
Input: client $m$, initialization global and local model parameter $\theta_0$ and $\theta$, local model learning rate $\eta$, loss function
 $L$, training sample size $X$ for each round, Gaussian noise standard deviation $\sigma$.
For iterations for each round $t = 1, 2, \ldots, T$ do
  For each batch $x \in X$ do
    Perform gradient descent: $\theta \leftarrow \theta_0 - \eta \nabla L(\theta; x)$
    Parameter clipping: $\theta \leftarrow \theta_0 + clip(\theta - \theta_0)$
    Add Gaussian noise: $N(0, I\sigma^2)$
  End
Local model update: $\Delta^m = \theta - \theta_0$
End
Output: Return the local model to the server.
```

Algorithm 1 incorporates the introduction of noise into the local model training process to safeguard the confidentiality of the client's raw data. More specifically, each client leverages the federated GDP algorithm to handle the gradients when computing the gradient updates, guaranteeing that sensitive client information remains secure and is not divulged during the model training process.

Aggregation of the Server-Side Federated GDP Algorithm: The server side is responsible for housing the global model, overseeing the entire model training process, and disseminating the initial model to all participating clients. It utilizes Algorithm 2 to receive and aggregate updates from all participating clients in each FL iteration, culminating in constructing a new model with updated parameters. The server-based federated GDP algorithm is detailed in Algorithm 2.

Algorithm 2 Server-Based Federated GDP

```
Input: training sample data $\{X_1, X_2, \ldots\ldots, X_N\}$, loss function $L = \frac{1}{N}\sum_{t=1}^{N} L(\theta; X_t)$, the weight corresponding to
client $m$ is $\omega_m = \min\left(\frac{n_m}{\hat{\omega}}, 1\right)$, $W = \sum_m \omega_m$, gradient clipping boundary value $C$.
For iterations for each round $t = 1, 2, \ldots, T$ do
  Randomly select the client set $M^t$ participating in the training with probability $q$
  For each user $m \in M^t$ do
    Perform local training: $\Delta_m^t = ClientUpdate(m, \theta_{t-1})$
  End
  Aggregated client parameters: $\Delta^t = \frac{\sum_{m \in M^t} \omega_m \Delta_m^t}{qW}$
  Clipping $\Delta^t$ value: $\Delta^t \leftarrow \Delta^t / \max\left(1, \frac{\|\Delta^t\|}{C}\right)$
  Update global model parameters: $\theta_t \leftarrow \theta_{t-1} + \Delta^t + N(0, I\sigma^2)$
End
Output: Broadcast the global model to clients.
```

In Algorithm 2, the server employs the federated GDP algorithm to handle the updates during the aggregation of client updates. This ensures that each client's contribution is effectively integrated into the final model, facilitating comprehensive global model updates and refinement.

By combining this client-server architecture with the federated GDP algorithm, FLM-ICR can achieve global model updates and optimization while protecting user privacy. This approach enhances classification performance and guarantees the security and confidentiality of private data, thereby fostering a robust and privacy-conscious framework for model refinement and collaborative learning.

Classification prediction module

Based on the client-server architecture of horizontal FL, FLM-ICR uses the improved MLP and LR network as the backbone of the local model to improve the classification performance. This approach is straightforward to implement and seamlessly integrates FL for update training. The federated MGD algorithm optimizes local model training, accelerating convergence and facilitating efficient calculations for classification tasks.

FLM-ICR uses the improved MLP and LR network as the backbone of the locally trained model. This choice proves more suitable than other algorithms for addressing the IoVT-based connected record classification problem outlined in this paper. The relatively small number of neural network layers utilized by LR and MLP, coupled with a modest parameter count, results in minimal computing resource requirements and swift computational speed during training. Furthermore, LR and MLP model structures are relatively simple, facilitating comprehension and implementation, and are less susceptible to issues such as overfitting. The output results of LR and MLP are calculated based on mathematical formulas, which are highly interpretable. MLP and LR are good at handling classification problems and are easy to use, which can better integrate FL models for update training.

The backbone network structure of FLM-ICR enhances non-linear modeling capability, model expressiveness, feature extraction capability, and generalization ability compared to traditional MLP and LR networks. This augmentation significantly improves the model's classification performance. It enables it to adapt to diverse data distributions, giving FLM-CR a competitive edge over alternative methods across various tasks. Establishing the MLP and LR network involves taking each IoVT connection record as input and outputting the classification accuracy of the two types of connections. PyTorch is employed to classify text data, and the two network architectures are presented below.

```
MLP ((model): Sequential ((0): Linear (in_features=3, out_features=200, bias=True)
  (1): Dropout (p=0.2, inplace=False)
  (2): ReLU ()
  (3): Linear (in_features=200, out_features=2, bias=True)))
```

The improved MLP comprises linear layers, Dropout, and the ReLU activation function. This architecture is established using the Sequential class to construct a feedforward neural network for sample classification. Initially, the linear layer conducts linear transformations to augment the feature information of the samples, with an input dimension of 2 and an output dimension of 200. Dropout is then implemented with a probability of 0.2 for random Dropout, mitigating overfitting. Subsequently, the ReLU non-linear activation function is employed to enhance the network's non-linear expressive capability. Finally, the linear layer is utilized for dimension reduction and classification purposes.

LR ((linear): Linear (in_features=3, out_features=2, bias=True)

(sigmoid): Sigmoid ()

(model): Sequential ((0): Linear (in_features=3, out_features=2, bias=True)

(1): Sigmoid ()))

The improved LR model comprises linear layers and utilizes the Sigmoid activation function. This configuration enables the model to calculate the probability of a given sample belonging to a specific class. Initially, the linear layer conducts linear transformations and then applies the Sigmoid activation function for binary classification. The Sequential class is employed to build a feedforward neural network, with the linear layer executing linear transformations. Ultimately, the Sigmoid activation function produces the probability of a sample being associated with a particular class.

Simulation experiment

This chapter is divided into four subsections: Preliminary Preparation, Model Evaluation, Training Result, and Comparative Experiment. Preliminary Preparation includes experimental environment and parameter settings. Model Evaluation utilizes a confusion matrix as the evaluation metric. Training Result encompasses validating model performance, exploring model performance under different levels of client collaboration and imbalanced data distribution, and verifying the effectiveness of FL. Comparative Experiment involves comparing the model performance with existing representative research methods.

Preliminary preparation

In order to ensure the repeatability of the experimental results, all the experiments in this paper were carried out on the same laptop. Experimental environment configuration: The central processor is an Intel (R) Core (TM) i7-7700K CPU @ 4.20GHz, with 16GB of memory, utilizing the deep learning frameworks Python 3.8 and PyTorch 1.8.1, and running on the Windows 10 operating system.

The improved MLP and LR networks categorize IoVT into corresponding types based on input connections when the dependent variable takes on different categorical values. To achieve the best training result, it is necessary to select appropriate optimizers and training step sizes in the model setup to minimize the value of the loss function. The MLP and LR networks utilize the Federated MGD algorithm to update the optimized network weights with a momentum setting 0.9. In terms of loss function selection, the improved MLP network employs the cross-entropy loss function, while the improved LR network uses the logarithmic loss function. The relevant model parameters are set as follows: output size is 2, the number of clients is 4, the learning rate is 0.01, batch size is 128, training epochs are 60, the number of local update rounds for clients is 1, the gradient clipping boundary value C is 0.5, and the standard deviation σ of Gaussian noise is 0.5.

Model evaluation

To assess the effectiveness and feasibility of FLM-ICR, it is crucial to simultaneously consider multiple indicators for evaluating the model's performance. In binary classification, the confusion matrix is the primary evaluation index during the model evaluation stage. This matrix, obtained from the experiment, is fundamental for measuring classifier accuracy and deriving most evaluation indicators. It categorizes two-category samples into positive (P) and negative (N) samples and predictions into true (T) and false (F), as depicted in Table 1.

In Table 1, TP is the number of predicted positive classes in the actual positive class, TN is the number of predicted negative classes in the actual negative class, FP is the number of predicted positive classes in the actual negative class, FN is the number of predicted negative classes among the actual positive classes. It can be seen that the accuracy $ACC = \frac{TP+TN}{TP+TN+FP+FN}$ is the proportion of the actual positive class in the prediction result. The precision $P = \frac{TP}{TP+FP}$ is the proportion of the actual positive class in the predicted positive class. The recall $R = \frac{TP}{TP+FN}$ is the proportion of the actual positive class correctly classified, also known as the sensitivity. The specificity $TNR = \frac{TN}{TN+FP}$ is the proportion of the actual negative cases that are correctly classified. The F1 score is $F1 = \frac{2P\times R}{P+R}$, which combines the precision and recall scores.

The above five evaluation indicators can reflect the performance of the classification model, and ACC can objectively reflect the overall quality of the model, and the value range is [0, 1]. The closer the ACC is to 1, the better the model performance. However, in the case of unbalanced positive and negative samples, the correct rate can

Table 1 Confusion matrix

Confusion Matrix		**Prediction Category**	
		Positive example	**Negative example**
Real Category	Positive example	TP	FN
	Negative example	FP	TN

only partially reflect the quality of the model. The higher the P, the better the model performance. The higher the R, the better the model performance. The larger TNR is, the smaller the misjudgment rate is and the better the model performance is. Since both P and R only describe the quality of the model from a single aspect, it makes little sense to simply pursue the improvement of a single indicator. Increasing these two indicators simultaneously is necessary to obtain the optimal model. As the harmonic mean of P and R, the $F1$ is a balance point between P and R, which can consider both P and R of the classification model. When P and R increase simultaneously, the larger the $F1$, the better the model. In this paper, the confusion matrix is used to evaluate the classification performance of FLM-ICR. The 300 sample results discussed in the experimental confusion matrix are the classification results of the test data.

Training result

FLM-ICR is trained on the IoVT connection record non-IID data distribution, using MLP and LR networks as the model backbone called FL-MLP and FL-LR, respectively. FLM-ICR trains FL-MLP and FL-LR models with data privacy-preserving capabilities. The confusion matrices of FL-MLP and FL-LR are obtained as shown in Fig. 5(a). After 60 training rounds, the fitting curves of FL-MLP and FL-LR classification accuracy are shown in Fig. 5(b) below. The values of the five evaluation indicators ACC, P, R, TNR, and $F1$ can be obtained from the confusion matrix and related calculation formulas. The training result can be seen more intuitively in Fig. 6.

The confusion matrices in Fig. 5(a) show that FL-MLP and FL-LR trained by FLM-ICR have better classification performance. The fitting curves in Fig. 5(b) show that the classification accuracies of FL-MLP and FL-LR trained by FLM-ICR are 0.81 and 0.78, respectively. Accuracy grows slowly early in training because non-uniform sampling causes each client to have data from one class and very little data from another. As the number of training epochs increases, the accuracy gradually stabilizes to a higher level. The model performance indicators in Fig. 6 show that the FL-MLP model trained by FLM-ICR has ACC of 0.81, P of 0.74, R of 0.88, TNR of 0.74, and $F1$ of 0.8. The FL-LR model trained by FLM-ICR has ACC of 0.78, P of 0.73, R of 0.79, TNR of 0.76, and $F1$ of 0.76. FL-MLP and FL-LR, which FLM-ICR can train, have achieved good model performance.

By exploring various levels of client collaboration for model training, it is demonstrated that the number of clients participating in each round of collaborative training in FLM-ICR impacts model performance. The model's performance in FLM-ICR in the real decentralized data scenario is proved effective by exploring the unbalanced data distribution for model training.

- Client collaboration level

The client collaboration level is called the C value, which controls the number of multi-client parallelisms. In order to explore the influence of the level of collaboration between clients on model performance, FLM-ICR trained FL-MLP and FL-LR models on non-IID data distribution for C=1, C=0.75, C=0.5, and C= 0.25 for experiments. FLM-ICR training FL-MLP model obtained ACC of 0.81,

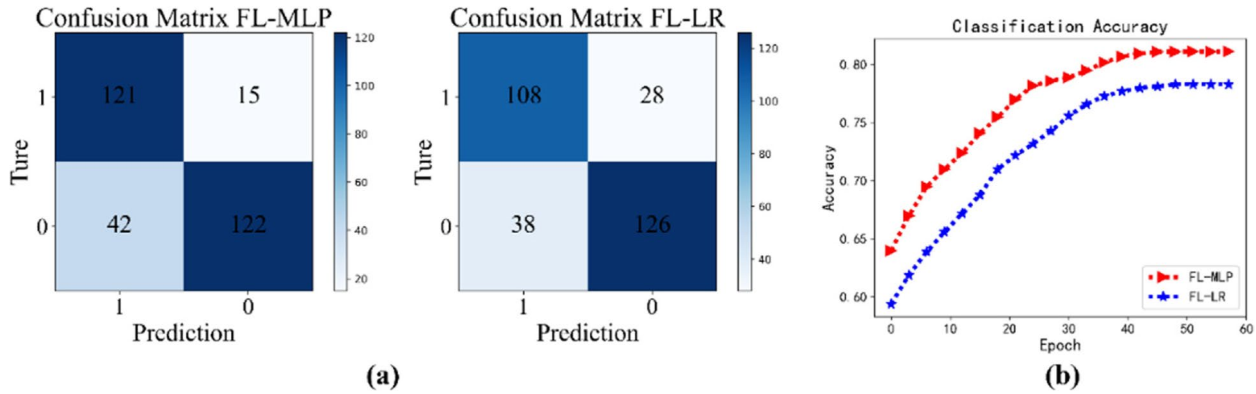
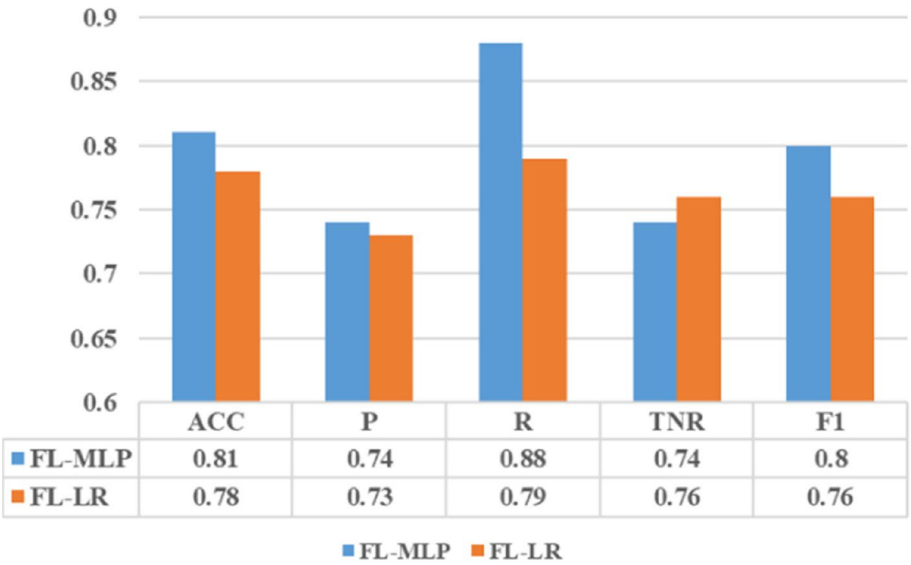


Fig. 5 **(a)** Confusion matrices for FL-MLP and FL-LR (**b**) Fitting curves of FL-MLP and FL-LR classification accuracy



	ACC	P	R	TNR	F1
FL-MLP	0.81	0.74	0.88	0.74	0.8
FL-LR	0.78	0.73	0.79	0.76	0.76

Fig. 6 Performance indicators of FL-MLP and FL-LR

0.79, 0.77, and 0.75, *P* of 0.74, 0.74, 0.73, and 0.71, *R* of 0.88, 0.8, 0.77, and 0.74, *TNR* of 0.74, 0.77, 0.76, and 0.75, *F*1 of 0.8, 0.77, 0.75, and 0.72, respectively. FLM-ICR training FL-LR model obtained *ACC* of 0.78, 0.76, 0.74, and 0.71, *P* of 0.73, 0.71, 0.69, and 0.66, *R* of 0.79, 0.77, 0.75, and 0.73, *TNR* of 0.76, 0.75, 0.73, and 0.68, *F*1 of 0.76, 0.74, 0.72, and 0.69, respectively. The confusion matrices of FL-MLP and FL-LR under different C values are shown in Fig. 7.

Figure 7 displays the confusion matrices of FL-MLP and FL-LR across various C values. These matrices provide a comprehensive overview of the performance of both FL-MLP and FL-LR models in different scenarios. They can gain insights into these models' *ACC*, *P*, *R*, *TNR*, and *F*1 under different C values to compare and evaluate the FL-MLP and FL-LR classification performance. The effects of different C values on the performance of FL-MLP and FL-LR models are shown in Fig. 8.

It can be seen from Fig. 8 that when C=1, the four clients have performed cooperative training in each round, and the overall performance of the FL-MLP and FL-LR models trained by FLM-ICR is the best. When C=0.75, there are three clients for cooperative training in each round, and the overall performance of FL-MLP and FL-LR models trained by FLM-ICR is worse than that of C=1. When C=0.5, there are two clients for cooperative training in each round. The overall performance of FL-MLP and

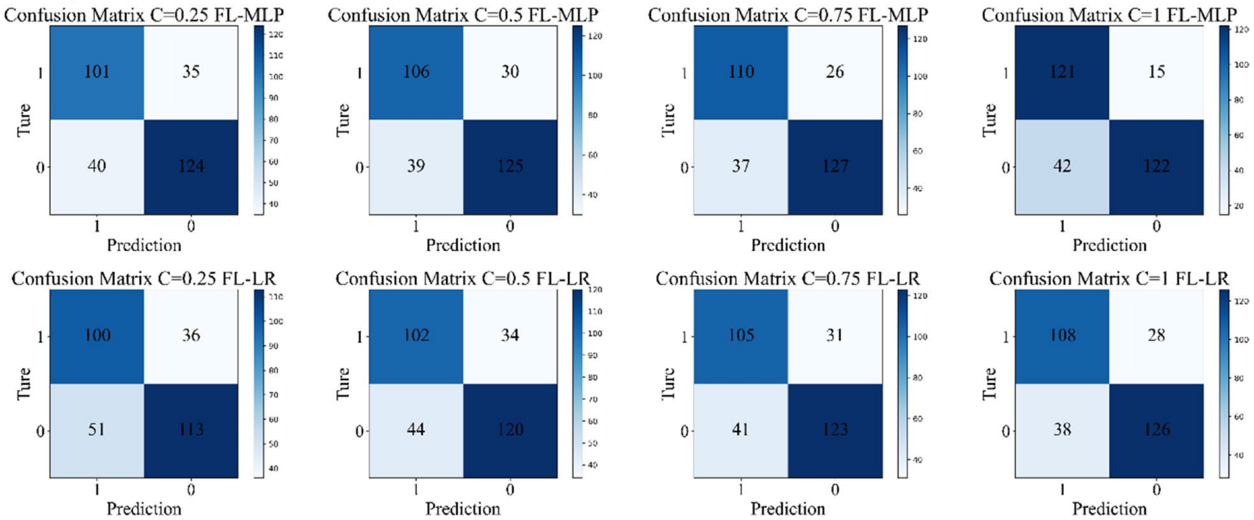
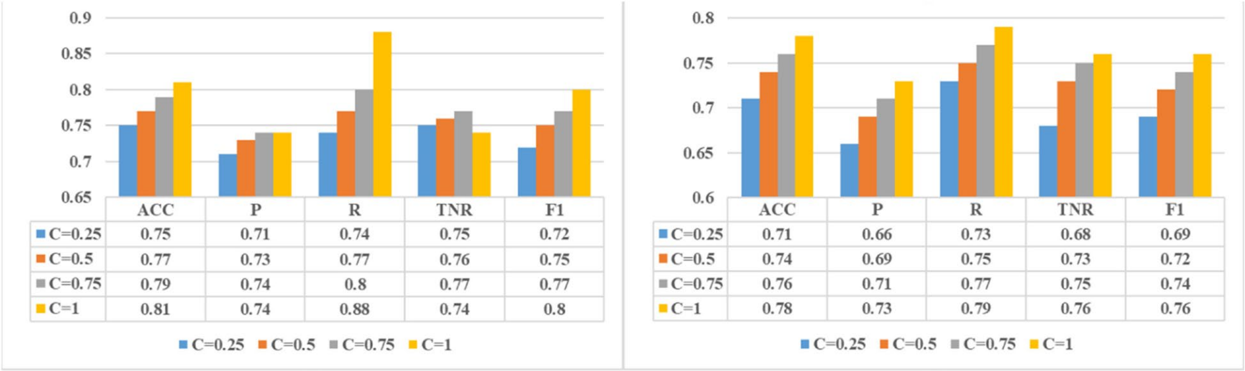


Fig. 7 Confusion matrices for FL-MLP and FL-LR with different C values



	ACC	P	R	TNR	F1
C=0.25	0.75	0.71	0.74	0.75	0.72
C=0.5	0.77	0.73	0.77	0.76	0.75
C=0.75	0.79	0.74	0.8	0.77	0.77
C=1	0.81	0.74	0.88	0.74	0.8

(a)

	ACC	P	R	TNR	F1
C=0.25	0.71	0.66	0.73	0.68	0.69
C=0.5	0.74	0.69	0.75	0.73	0.72
C=0.75	0.76	0.71	0.77	0.75	0.74
C=1	0.78	0.73	0.79	0.76	0.76

(b)

Fig. 8 **(a)** Effects of different C values on the performance of FL-MLP (**b**) Effects of different C values on the performance of FL-LR

FL-LR models trained by FLM-ICR is worse than that of C=0.75. When C=0.25, only one client is trained in each round, and the overall performance of the FL-MLP and FL-LR models trained by FLM-ICR is the worst among the four C-value experiments. The experimental results are that as the number of clients participating in each round of FL training decreases, the quality of the model obtained by the client through collaborative training becomes worse. The accuracy of the global model obtained after each round of server-side aggregation of model updates sent by the client is lower, leading to the model's poorer overall performance. Therefore, it is essential to reasonably set the number of clients participating in training. For non-IID data distribution, increasing the number of cooperative clients in each round of FL training, that is, improving the level of cooperation between clients, positively impacts the model's overall performance.

- Unbalanced data distribution training

The unbalanced data distribution represents the distribution in practical application scenarios like the IoV. In order to verify the actual feasibility of FLM-ICR, an experiment is carried out under the condition of unbalanced data distribution in the client. The confusion matrices of FL-MLP and FL-LR on non-IID and unbalanced data distribution are shown in Fig. 9.

Figure 9 presents the confusion matrices of FL-MLP and FL-LR when applied to non-IID and unbalanced data distributions. These matrices can assess the models' ability to handle data heterogeneity and class imbalances and facilitate a comprehensive comparison and evaluation of the FL-MLP and FL-LR classification performance. The performance indicators of FLM-ICR trained FL-MLP and FL-LR models on non-IID and unbalanced data distribution, respectively, are shown in Fig. 10.

It can be seen from Fig. 10 that the FL-MLP model trained by FLM-ICR on the unbalanced data distribution obtains ACC, P, R, TNR, and $F1$ as 0.76, 0.72, 0.75, 0.76, and 0.73, respectively. The FL-LR model trained by FLM-ICR obtained ACC, P, R, TNR, and $F1$ as 0.72, 0.67, 0.74, 0.7, and 0.7, respectively. The performance of the FL-MLP model trained by FLM-ICR on the non-IID data distribution is very similar to that of the FL-MLP model trained on the imbalanced data distribution. The performance of the FL-LR model trained on the non-IID data distribution by FLM-ICR is slightly lower than that of the FL-LR model trained on the unbalanced data distribution. The reason is that the client in the unbalanced data distribution differs in the amount of data. Approaching the model performance under the non-IID data distribution takes more training rounds. However, the final results are similar. The experimental results prove that the unbalanced data distribution has little effect on the model performance of FLM-ICR, and the practical feasibility of FLM-ICR has been fully verified.

- Verify the validity of FL

To verify the effectiveness of FL in FLM-ICR, the MLP and LR network models are trained separately. To compare the model performance of FL-MLP and MLP and the model performance of FL-LR and LR to illustrate that using FL in FLM-ICR can protect data privacy while still having better model performance. The confusion matrices of FL-MLP and MLP and FL-LR and LR are shown in Fig. 11.

Figure 11 showcases the confusion matrices of FL-MLP, MLP, and FL-LR and LR. These matrices provide a comprehensive visual representation of the classification performance of these models. They can gain insights into the ACC, P, R, TNR, and $F1$ of FL-MLP, MLP, FL-LR, and

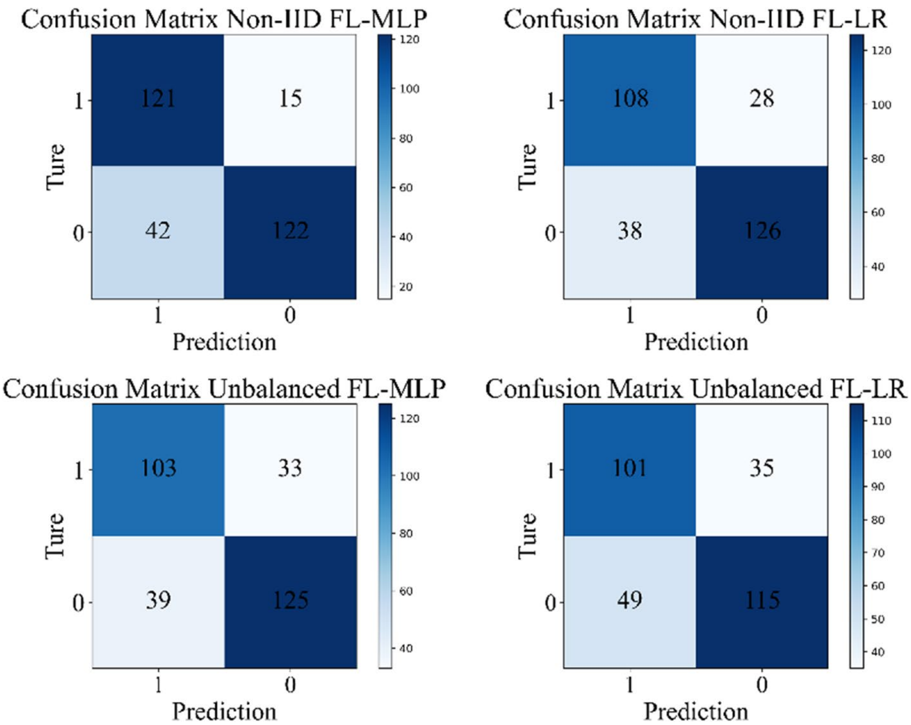


Fig. 9 Confusion matrices for FL-MLP and FL-LR on non-IID and unbalanced data distributions

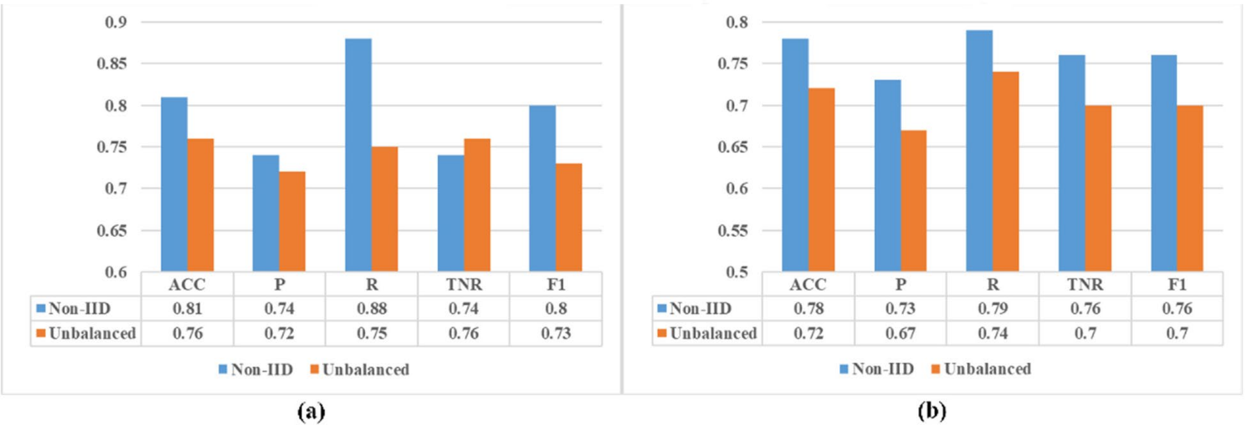


Fig. 10 (**a**) Performance indicators of FL-MLP on non-IID and unbalanced data distributions (**b**) Performance indicators of FL-LR on non-IID and unbalanced data distributions

LR in handling the given dataset for a direct comparison between the FL approaches (FL-MLP and FL-LR) and their non-FL approaches (MLP and LR) to evaluate the impact of FL on model performance. The model performances of FL-MLP and MLP and FL-LR and LR are shown in Fig. 12.

It can be seen from Fig. 12 that ACC, P, R, TNR, and $F1$ obtained by the FL-MLP model are 0.81, 0.74, 0.88, 0.74, and 0.8, respectively. The MLP model obtained ACC, P, R, TNR, and $F1$ as 0.83, 0.76, 0.9, 0.76, and 0.82, respectively. The FL-LR model obtained ACC, P, R, TNR, and $F1$ as 0.78, 0.73, 0.79, 0.76, and 0.76, respectively. The LR model obtained ACC, P, R, TNR, and $F1$ as 0.8, 0.73, 0.88, 0.73, and 0.79, respectively. It can be seen from the experimental results that although the MLP and LR models provide the best model performance, they have

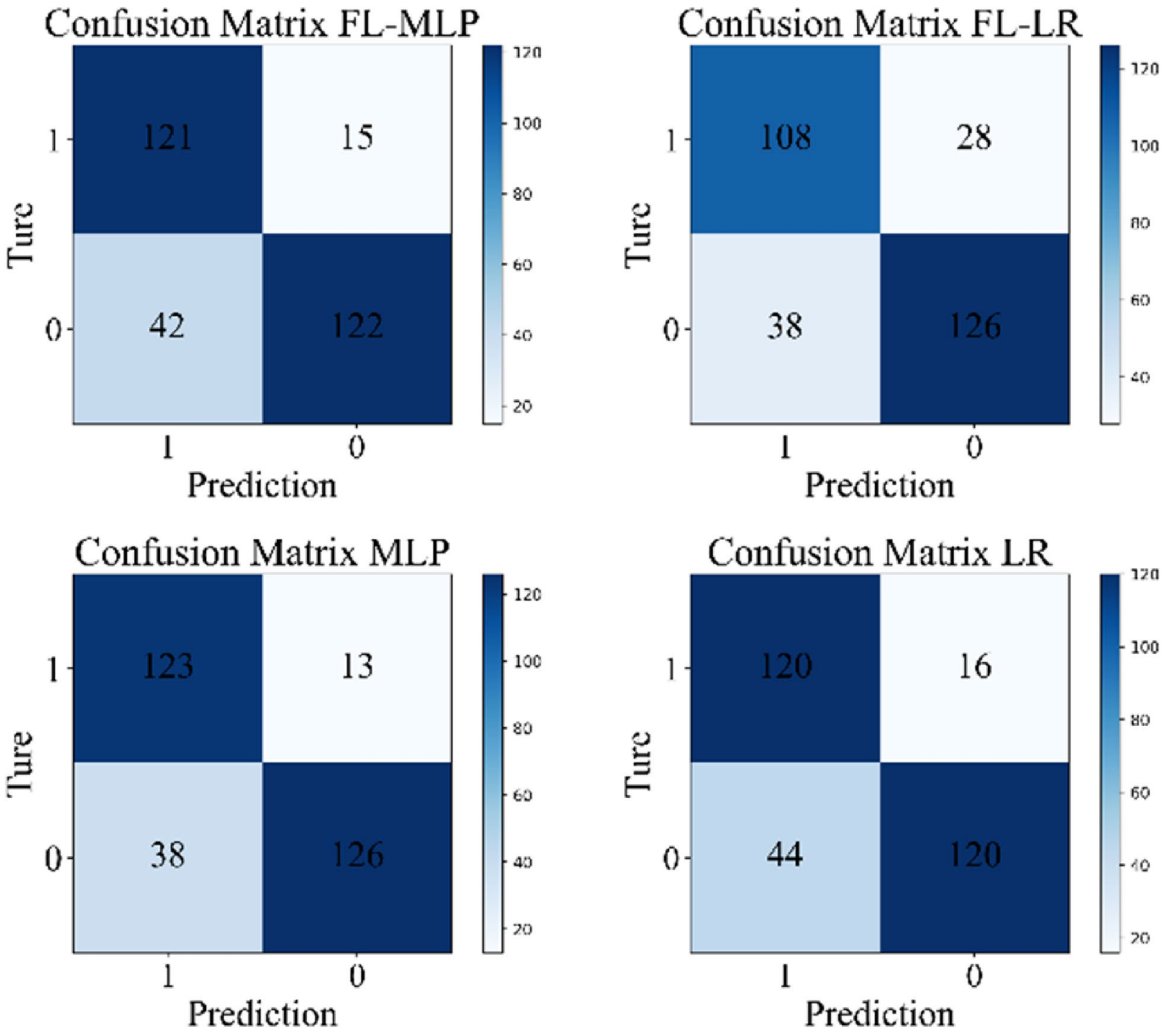


Fig. 11 Confusion matrices for FL-MLP and MLP, and FL-LR and LR

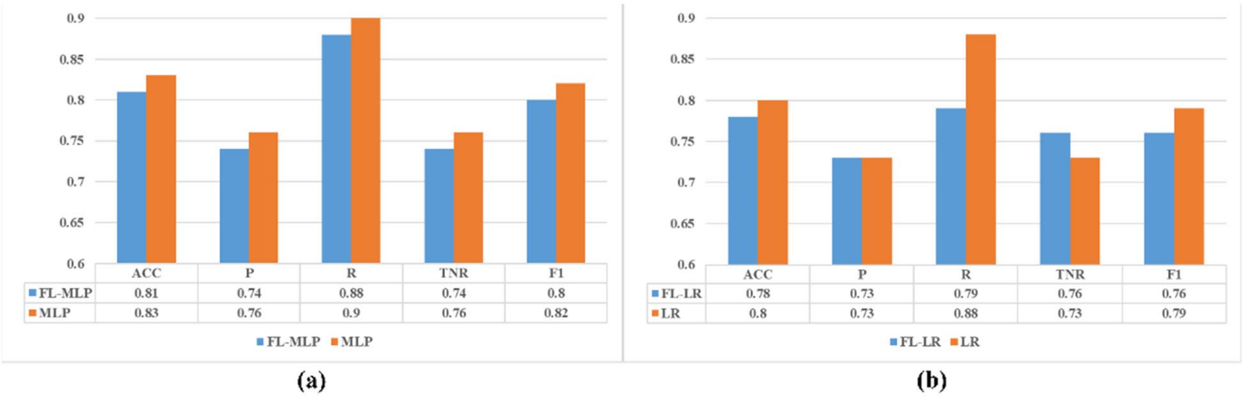


Fig. 12 (**a**) Model performance comparison of FL-MLP and MLP (**b**) Model performance comparison of FL-LR and LR

no privacy protection capabilities and lack FL training. However, FL-MLP and FL-LR trained with FL in FLM-ICR can protect data privacy and are very close to the model performance of MLP and LR. Furthermore, it can be proved that FL in FLM-ICR is effective and excellent, which can maintain the balance of data privacy and model performance.

The above three parts of the experiment: the positive impact of the number of clients participating in each FLM-ICR FL collaborative training round on model performance is explored; the applicability of the model performance of FLM-ICR under unbalanced data distribution in natural scenes is demonstrated; it is verified that FL-MLP and FL-LR trained with FL in FLM-ICR are effective. The experimental results fully demonstrate the positive significance of the client-side cooperative training mode, confirm that FLM-ICR is suitable for practical application scenarios, and illustrate that FL plays a vital role in the model establishment.

Comparative experiment

In order to further verify the validity and feasibility of the model, the comparative experiment was set up under the same dataset and experimental environment, compared with CNN-LSTM [34], GDP-FL [14], LATENT [15], and PLU-FedOA [17] for comparison. To prove that the model performance of FLM-ICR is superior to other methods and has better model performance while protecting data privacy. The model performance comparison table between FLM-ICR and four algorithms is shown in Table 2.

As can be seen from the performance comparison between FLM-ICR and the four algorithms in Table 2, although CNN-LSTM has the highest ACC among them, as a traditional deep learning method, it has no privacy protection capability, so the overall performance is not as good as FLM-ICR. The five performance indexes of FLM-ICR are better than those of GDP-FL and LATENT. Depending on the data and application scenario, FLM-ICR outperforms other methods in terms of ACC, P, R, TNR, and $F1$. The following insights may explain the advantages of FLM-ICR in these aspects: (1) Data diversity: FLM-ICR can fully utilize data on multiple vehicles and terminal devices for model training, improving model accuracy and performance. (2) Privacy protection: FLM-ICR keeps data on the local device for model training, avoids centralized data storage and transmission, and effectively protects user privacy. (3) Real-time and adaptability: FLM-ICR can perform real-time model training on vehicles and terminal devices, allowing the model to respond and adapt to different driving scenarios and needs on time. (4) Distributed computing: FLM-ICR distributes model training tasks across multiple vehicles and terminal devices and integrates model updates from all parties through aggregation algorithms, thereby improving the efficiency of model training. The feasibility of FLM-ICR is analyzed theoretically and verified by experiments on IoVT-connected recording datasets.

Table 2 Performance comparison

Algorithms	ACC	P	R	TNR	F1
CNN-LSTM	**0.807**	0.729	0.832	0.748	0.777
GDP-FL	0.751	0.684	0.795	0.7	0.735
LATENT	0.732	0.66	0.756	0.671	0.705
PLU-FedOA	0.788	0.697	0.78	0.713	0.736
FLM-ICR	0.795	**0.735**	**0.835**	**0.75**	**0.782**

Conclusion

In the IoV application scenario, the FLM-ICR proposed in this paper is based on the connection record data of IoVT and uses FL and ML methods to classify normal and abnormal terminals while ensuring data privacy efficiently. FLM-ICR uses the improved MLP and LR network as the backbone of the model, which can better handle classification problems and is simple and easy to implement, which is convenient for integrating FL for update training. Under the client-server architecture of horizontal FL, FLM-ICR uses the federated GDP algorithm to protect the security of the FL calculation process and uses the federated MGD algorithm as the training optimizer to accelerate the local model convergence and achieve efficient calculation. The FL-MLP model trained by FLM-ICR safely and cooperatively obtained ACC, P, R, TNR, and $F1$ as 0.81, 0.74, 0.88, 0.74, and 0.8, respectively, and the trained FL-LR model obtained ACC, P, R, TNR, and $F1$ are 0.78, 0.73, 0.79, 0.76, and 0.76, respectively. Experiments explore the positive impact of the number of clients participating in FLM-ICR federated collaborative training in each round on model performance. The applicability of the model performance of FLM-ICR under unbalanced data distribution in natural scenes is demonstrated. It is verified that FL-MLP and FL-LR trained with FL in FLM-ICR are effective. The comparative experiment in the same dataset and experimental environment shows that the model performance of FLM-ICR is better than the existing four methods and has higher classification performance and security. FLM-ICR provides a new idea for future big data sharing and collaboration. It can be extended to actual scenarios such as hospitals and banks to protect data privacy and collaborative training and analysis of data while ensuring personal privacy information.

In future work, FLM-ICR needs to be improved in the following areas: (1) Communication and computation costs: Reducing bandwidth and energy consumption through techniques like optimized aggregation or compressed model updates enhances the efficiency of FL. (2) Model personalization and adaptation: Techniques such as user feedback and context-aware learning enable personalized model training and adaptation to individual user preferences and driving behaviors. (3) Scalability and large-scale deployment: Developing scalable algorithms and infrastructure facilitates the widespread deployment of FL in the IoV domain as the number of IoVT and connected vehicles increases. By addressing these limitations and exploring potential improvements, future research can advance FL in IoV applications, leading to more effective models that enhance user-driving experiences and services.

Acknowledgments
We thank Xijing University for the first batch of special funds for high-level talents in 2022 [XJ22B04].

Authors' contributions
K.Y. was responsible for funding acquisition, data curation, and supervision; J.W.D. was responsible for Conceptualization, methodology, and software; J.L. was responsible for formal analysis and resources; F.X. was responsible for investigation; Y.T. was responsible for project administration, M.L. was responsible for visualization, and Z.L. was responsible for validation. All authors reviewed the manuscript.

Funding
This work was supported by the High-level Talents Special Fund Project of Xijing University in 2022, 'Research on Industrial Internet of Things Intrusion Detection Technology Based on Deep Learning' (XJ22B04).

Availability of data and materials
No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
The authors declare no competing interests.

Received: 19 December 2023 Accepted: 1 March 2024
Published: 13 March 2024

References

1. Liu L, Zhao M, Yu M, Jan MA, Lan D, Taherkordi A (2022) Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks. IEEE Trans Intell Transport Syst 24(2):2169–2182
2. Kong Q, Lu R, Ma M, Bao H (2019) A privacy-preserving sensory data sharing scheme in Internet of Vehicles. Future Gener Comput Syst 92:644–655
3. Rathore MS, Poongodi M, Saurabh P, Lilhore UK, Bourouis S, Alhakami W, Osamor J, Hamdi M (2022) A novel trust-based security and privacy model for internet of vehicles using encryption and steganography. Comput Electrical Eng 102:108205
4. Liu L, Feng J, Mu X, Pei Q, Lan D, Xiao M (2023) Asynchronous Deep Reinforcement Learning for Collaborative Task Computing and On-Demand Resource Allocation in Vehicular Edge Computing. IEEE Trans Intell TransportSyst 24:15513–15526
5. Liu Y, Yu W, Ai Z, Xu G, Zhao L, Tian Z (2023) A blockchain-empowered federated learning in healthcare-based cyber physical systems. IEEE Trans Network Sci Eng 10(5):2685–2696
6. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. 20th International Conference on Artificial Intelligence and Statistics. PMLR 54:1273–1282
7. Li Z, Sharma V, Mohanty SP (2020) Preserving data privacy via federated learning: challenges and solutions. IEEE Consum Electron Mag 9(3):8–16
8. Chamikara MA, Bertok P, Khalil I, Liu D, Camtepe S (2021) Privacy preserving distributed machine learning with federated learning. Comput Commun 171:112–125
9. Liu L, Tian Y, Chakraborty C, Feng J, Pei Q, Zhen L, Yu K (2023) Multilevel federated learning based intelligent traffic flow forecasting for transportation network management. IEEE Trans Netw Serv Manag 20:1446–1458
10. Xiang T, Bi Y, Chen X, Liu Y, Wang B, Shen X, Wang X (2023) Federated Learning with Dynamic Epoch Adjustment and Collaborative Training in Mobile Edge Computing. IEEE Transactions on Mobile Computing, vol 01, pp. 1–16. https://doi.org/10.1109/TMC.2023.3288392
11. Zhao P, Zhang G, Wan S, Liu G, Umer T (2020) A survey of local differential privacy for securing internet of vehicles. J Supercomput 76(11):8391–8412
12. Nilsson A, Smith S, Ulm G, Gustavsson E, Jirstrand M (2018) A performance evaluation of federated learning algorithms. Proceedings of the second workshop on distributed infrastructures for deep learning. ACM, New York, pp 1–8
13. Li Q, Diao Y, Chen Q, He B (2022) Federated learning on non-iid data silos: An experimental study. IEEE 38th International Conference on Data Engineering. IEEE, Kuala Lumpur, pp 965–978
14. Abadi M, Chu A, Goodfellow , I, McMahan HB, Mironov I, Talwar K, Zhang L (2016) Deep learning with differential privacy. Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, New York, pp 308–318
15. Arachchige PC, Bertok P, Khalil I, Liu D, Camtepe S, Atiquzzaman M (2019) Local differential privacy for deep learning. IEEE Internet Things J 7(7):5827–5842
16. Choudhury O, Gkoulalas-Divanis A, Salonidis T, Sylla I, Park Y, Hsu G, Das A (2019) Differential privacy-enabled federated learning for sensitive health data. arXiv preprint arXiv:1910.02578
17. Yang G, Wang S, Wang H (2021) Federated learning with personalized local differential privacy. IEEE 6th International Conference on Computer and Communication Systems. IEEE, Chengdu, pp 484–489
18. Lu Y, Huang X, Zhang K, Maharjan S, Zhang Y (2020) Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. IEEE Trans Veh Technol 69(4):4298–4311
19. Yang Z, Zhang X, Wu D, Wang R, Zhang P, Wu Y (2022) Efficient Asynchronous Federated Learning Research in the Internet of Vehicles. IEEE Internet Things J 10(9):7737–7748
20. Zhao P, Huang Y, Gao J, Xing L, Wu H, Ma H (2022) Federated learning-based collaborative authentication protocol for shared data in social IoV. IEEE Sensors J 22(7):7385–7398
21. Luo X, Wang J, Xu J, Shen M (2020) Research on Data Privacy Protection of Internet of Vehicles Based on Differential Privacy. IOP Conference Series: Earth and Environmental Science. IOP Publishing, Guangzhou, p 012007
22. Bakopoulou E, Tillman B, Markopoulou A (2021) Fedpacket: A federated learning approach to mobile packet classification. IEEE Trans Mobile Comput 21(10):3609–3628
23. Zhu H, Xu J, Liu S, Jin Y (2021) Federated learning on non-IID data: a survey. Neurocomputing 465:371–390
24. Zhang W, Wang X, Zhou P, Wu W, Zhang X (2021) Client selection for federated learning with non-iid data in mobile edge computing. IEEE Access 9:24462–24474
25. Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V (2018) Federated learning with non-IID data arXiv:1806.00582. Available: https://arxiv.org/abs/1806.00582
26. Liu W, Chen L, Chen Y, Zhang W (2020) Accelerating federated learning via momentum gradient descent. IEEE Trans Parallel Distrib Syst 31(8):1754–1766

27. Yuan X, Ni W, Ding M, Wei K, Li J, Poor HV (2023) Amplitude-Varying Perturbation for Balancing Privacy and Utility in Federated Learning. IEEE Trans Inform Forensics Secur 18:1884–1897
28. He D, Du R, Zhu S, Zhang M, Liang K, Chan S (2021) Secure logistic regression for vertical federated learning. IEEE Internet Comput 26(2):61–68
29. Wei S, Tong Y, Zhou Z, Song T (2020) Efficient and fair data valuation for horizontal federated learning. Federated Learn 12500:139–152
30. Majeed U, Khan LU, Hong CS (2020) Cross-silo horizontal federated learning for flow-based time-related-features oriented traffic classification. 21st Asia-Pacific Network Operations and Management Symposium. IEEE, Daegu, pp 389–392
31. Xin B, Geng Y, Hu T, Chen S, Yang W, Wang S, Huang L (2022) Federated synthetic data generation with differential privacy. Neurocomputing 468:1–10
32. Liu W, Cheng J, Wang X, Lu X, Yin J (2022) Hybrid differential privacy based federated learning for Internet of Things. J Syst Architect 124:102418
33. El Ouadrhiri A, Abdelhadi A (2022) Differential privacy for deep and federated learning: a survey. IEEE Access 10:22359–22380
34. Du J, Yang K, Hu Y, Jiang L (2023) Nids-cnnlstm: Network intrusion detection classification model based on deep learning. IEEE Access 11:24808–24821

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.