Computer Fraud and Security
ISSN (online): 1873-7056

Privacy Preserving Federated Learning Efficiency Optimization Algorithm based on Differential Privacy

Rui Xie [1], Xuejiao Zhong [2], Xin Chen[1], Shaohui Xu[1], Haiyang Yu[2], Xinyuan Guo[3*]

[1] *Huizhou Power Supply Bureau, Guangdong Power Grid Co., Ltd., Huizhou, China*
[2] *Southern Power Grid Digital Enterprise Technology (Guangdong) Co., Ltd., China*
[3] *School of Computer Science and Technology, Xi'an Jiaotong University, China*

Abstract: With the advancement of information technology, data security and user privacy protection have become paramount. To achieve efficient privacy protection in a federated learning environment, a differential privacy algorithm is designed using the eXtreme Gradient Boosting (XGBoost) algorithm. This algorithm optimizes the privacy protection process by applying differential privacy to the optimal segmentation point in a weak classifier. Additionally, to address the multi-party collaboration challenge in federated learning, a differential privacy construction scheme based on multi-party collaboration is proposed. The results indicate that the running times of differential privacy algorithms based on multi-party collaboration, XGBoost, and the traditional differential privacy algorithm were 16.2s, 22.1s, and 29.5s, respectively. The optimized approach improved efficiency by 45.08% compared to the traditional algorithm. Overall, the differential privacy-based federated learning efficiency optimization algorithm can ensure privacy protection while enhancing accuracy and efficiency, providing significant technical support.

Introduction: This paper proposes a privacy-preserving joint learning efficiency optimization algorithm based on differential privacy, and designs a differential privacy-preserving algorithm based on XGBoost (DP-XGB). This algorithm enhances privacy preservation by introducing differential privacy at the optimal segmentation point in the weak learner, thereby improving both data security and model accuracy. Building on this foundation, the research further proposes a differential privacy construction scheme (FDP-XGB) based on multi-party collaboration, integrating joint learning techniques to address potential privacy leakage during multi-party collaboration.

Objectives: By applying differential privacy to the optimal splitting point among weak learners, DP-XGB optimizes the privacy protection process, thereby enhancing both data security and model accuracy. FDP-XGB is introduced to safeguard privacy in a joint learning environment, effectively addressing the issue of privacy leakage that can occur during multi-party collaboration.

Methods: We first enhance the original data and obtains weak learners using the XGBoost algorithm. These weak learners are then combined to form a strong learner, and a differential privacy protection algorithm is constructed. Building on this foundation, the second section develops a multi-party collaborative privacy protection algorithm within a federated learning environment.

Results: The results indicate that the running times of differential privacy algorithms based on multi-party collaboration, XGBoost, and the traditional differential privacy algorithm were 16.2s, 22.1s, and 29.5s, respectively. The optimized approach improved efficiency by 45.08% compared to the traditional algorithm. Overall, the differential privacy-based federated learning efficiency optimization algorithm can ensure privacy protection while enhancing accuracy and efficiency, providing significant technical support.

Conclusions: This study proposes a privacy protection technology that combines the XGBoost differential privacy protection algorithm with federated learning to address privacy security and data silos in data mining. FDP-XGB demonstrated the highest prediction accuracy when comparing true and predicted data values, outperforming DP-XGB. For a data volume of 18×104, the computation times for XGBoost, DP-XGB, and FDP-XGB were 29.5 seconds, 22.1 seconds, and 16.5 seconds, respectively, with resource consumption rates of 48.5%, 24.9%, and 21.1%.

Keywords: federated learning, XGboost, differential privacy, efficiency.

INTRODUCTION

With the rapid development of IoT, cloud computing, and big data technologies, enterprises and users generate a large amount of complex and heterogeneous data in their activities. How to reasonably analyze and process these data is of great significance to both enterprises and users. In the process of data transmission and processing, without effective privacy protection measures, a large amount of sensitive information may face the risk of illegal theft and privacy leakage [1]. For instance, enterprises can leverage big data analytics to uncover hidden patterns, gain insights into consumer behavior, enhance product development, and improve customer service. Similarly, users benefit from personalized services and products tailored to their preferences and needs. Therefore, data security and user privacy protection have become the focus of current concerns.

Differential privacy is a crucial privacy-preserving data analytics technique that extracts valuable insights from data by controllably adding noise, preventing the identification of individual records. Federated learning, on the other hand, is a distributed machine learning technique that enables efficient data utilization while preserving privacy by collaboratively training models across different data sources. Therefore, differential privacy and federated learning have become prominent directions in the field of privacy protection, with numerous researchers exploring their applications in cybersecurity. In terms of differential privacy, previous studies have proposed methods based on the Laplace and Gaussian mechanisms [2]-[5]. These methods ensure privacy protection by adding noise to the data distribution process, with the Laplace mechanism commonly used for numerical data and the Gaussian mechanism applicable to more complex data types. Additionally, research has explored histogram- and clustering-based differential privacy techniques that enhance privacy protection by adding noise to data distribution. In the realm of federated learning, previous research has focused on model aggregation and improving communication efficiency [9]-[12]. Scholars have proposed model aggregation methods based on weighted averages and random selection to address the problem of uneven data distribution. Other studies have explored methods to enhance the efficiency of federated learning by reducing communication frequency and compressing the amount of transmitted data [13]. These methods not only reduce communication overhead but also protect data privacy to a significant extent. Overall, the integration of differential privacy and federated learning techniques represents a significant advancement in privacy-preserving data analytics. By leveraging these methods, it is possible to achieve secure and efficient data processing, fostering innovation and ensuring compliance with privacy regulations.

Although differential privacy and federated learning techniques have made significant progress in privacy protection, they still face numerous challenges in practical applications. First, most existing differential privacy techniques are designed for processing and analyzing centralized data and are ineffective for privacy protection in distributed data environments. Second, federated learning can lead to privacy leakage during multi-party collaboration, as participants must send training parameters to the aggregation server, making it easy to reconstruct the original data from these parameters. Finally, differential privacy techniques introduce a certain amount of noise to protect privacy, which can affect the accuracy and efficiency of the model. Therefore, achieving efficient data use and robust privacy protection in federated learning environments remains an urgent challenge.

To address the challenges, this paper proposes a privacy-preserving joint learning efficiency optimization algorithm based on differential privacy. Specifically, the study selects the Extreme Gradient Boosting (XGBoost) algorithm, known for its strong performance in data mining and recommender systems, and designs a differential privacy-preserving algorithm based on XGBoost (DP-XGB). This algorithm enhances privacy preservation by introducing differential privacy at the optimal segmentation point in the weak learner, thereby improving both data security and model accuracy. Building on this foundation, the research further proposes a differential privacy construction scheme (FDP-XGB) based on multi-party collaboration, integrating joint learning techniques to address potential privacy leakage during multi-party collaboration. The contributions of the paper can be summarized as follows.

- Firstly, a differential privacy protection algorithm based on XGBoost, named DP-XGB, is proposed. By applying differential privacy to the optimal splitting point among weak learners, this algorithm optimizes the privacy protection process, thereby enhancing both data security and model accuracy.

- Secondly, the differential privacy construction scheme based on multi-party collaboration (FDP-XGB) is introduced to safeguard privacy in a joint learning environment, effectively addressing the issue of privacy leakage that can occur during multi-party collaboration.
- Thirdly, experimental results demonstrate that the DP-XGB algorithm achieves the lowest fit function value across different datasets, with optimal convergence speed and accuracy compared to other algorithms. Additionally, when dealing with large data volumes, the FDP-XGB algorithm outperforms the traditional XGBoost algorithm in terms of running time, while maintaining an excellent balance between privacy preservation and computational efficiency.

RELATED WORKS

Differential privacy protection technology is crucial for safeguarding sensitive data by adding controlled noise to prevent the extraction of valuable information. Many scholars have conducted in-depth research on this topic. Guo P et al. [2] proposed a differential privacy protection protocol based on location entropy for exposure in location-based services. They designed an optimal auxiliary user selection strategy for constructing anonymous sets and applied smart contracts to assess participant credibility. This protocol effectively resists background knowledge attacks and achieves controllable privacy protection for users. Chen et al. [3] explored the impact of sensor data correlation on differential privacy in mobile crowd perception systems, examining disturbance mechanisms from various perspectives. They used a Bayesian network to model the probability relationships between sensor data, derived scale parameters using the classical definition of differential privacy, and proposed a new perturbation mechanism to minimize noise introduction while aggregating query functions. Zhang J et al. [4] proposed an entropy-driven differential privacy protection scheme based on social graph attributes to protect graphic data in social media. This algorithm converts sensitive graph data into uncertain graphs, balancing privacy and practicality. Zhang C et al. [5] developed a personalized location privacy protection system to address privacy issues in mobile crowdsourcing technology. They introduced an innovative algorithm for calculating the privacy level of worker locations, a personalized differential privacy protection algorithm based on exponential mechanisms, and a personalized localized differential privacy protection algorithm. This system effectively enhances the efficiency and reliability of mobile crowdsourcing systems. Liu et al. [6] proposed models and algorithms for differential privacy metaverse data sharing using Wasserstein Generative Adversarial Networks (WGANs). Hewage et al. [7] conducted a systematic literature review on privacy-preserving data mining (PPDM) and data stream mining (PPDSM) techniques. They categorized PPDM methods into four types and highlighted the accuracy-privacy trade-off, noting a lack of solutions in PPDSM. Yang et al. [8] conducted a comprehensive survey on local differential privacy (LDP), analyzing its techniques and applications. They discussed the challenges and future directions in maintaining model performance while ensuring LDP in machine learning model training.

Federated learning enables the joint development of learning models and provides output results to users, offering an effective solution for data privacy protection. Scholars such as Pillutla K [9] proposed a robust joint aggregation learning method and developed a robust federated learning algorithm for stochastic learning of least squares additive models. Lee Y et al. [10] proposed server-driven and client-driven methods based on greedy algorithms to address the statistical heterogeneity in federated learning. Their experimental results indicated that these methods could improve federated learning technology and reduce wireless communication costs. Chung W et al. [11] introduced a federated feature connection method that accounts for heterogeneous clients. This approach involves model splitting and functional connectivity, which offloads some training load from the client to the aggregation server. Zhang F et al. [12] proposed a federated unsupervised representation learning method to utilize massive unlabeled data on distributed edge devices, aiming to learn a universal representation model without supervision while protecting data privacy. Zong et al. [14] discussed strategies to improve communication efficiency in decentralized federated learning. Ren et al. [15] provided a position paper on Federated Foundation Models (FedFM), outlining the motivations, challenges, and future directions in this emerging field. Shekhar et al. [16] study demonstrates the effectiveness of XGBoost in handling complex data and provides a reference for combining XGBoost with Particle Swarm Optimization for predicting discharge in compound channels with converging and diverging floodplains. Joshi et al. [17] survey the use of synthetic data in human analysis, discussing its benefits, applications, and open challenges, including the generation of synthetic data and its impact on privacy and model performance. Pan et al. [18] introduce Flagger, an efficient and high-performance federated

learning aggregator, which leverages data processing units and computational storage drives to accelerate large-scale cross-silo federated learning aggregation, significantly reducing aggregation time and improving overall training efficiency. Silvi et al. [19] propose FedSeq, a federated learning framework that accelerates training by sequentially training groups of heterogeneous clients, reducing communication overhead, and speeding up convergence.

In summary, existing network privacy protection technologies face significant challenges, particularly when processing and analyzing distributed and uncleaned datasets. Ensuring data security in these contexts remains an unresolved issue. Current methods primarily focus on traditional federated learning and differential privacy for computation, without adequately addressing multi-party collaboration. This study takes a novel approach by developing an information network privacy protection algorithm based on the XGBoost algorithm, with an emphasis on multi-party collaboration.

EFFICIENCY OPTIMIZATIONS FOR FEDERATED LEARNING ALGORITHM WITH DIFFERENTIAL PRIVACY

To address privacy and security issues in distributed datasets, we present a privacy-preserving joint learning efficiency optimization algorithm based on differential privacy. We first enhance the original data and obtains weak learners using the XGBoost algorithm. These weak learners are then combined to form a strong learner, and a differential privacy protection algorithm is constructed. We further develop a multi-party collaborative privacy protection algorithm within a federated learning environment.

Compared with traditional random forest algorithms, the XGBoost algorithm is an efficient gradient boosting decision method that uses the ensemble technique Boosting to combine multiple weak learners into a strong learner. By employing multiple decision trees for joint decision-making, each tree contributes to correcting the difference between the target and the predicted results of all previous trees, thereby improving overall performance. The operating principle of the XGBoost algorithm is illustrated in Figure 1.

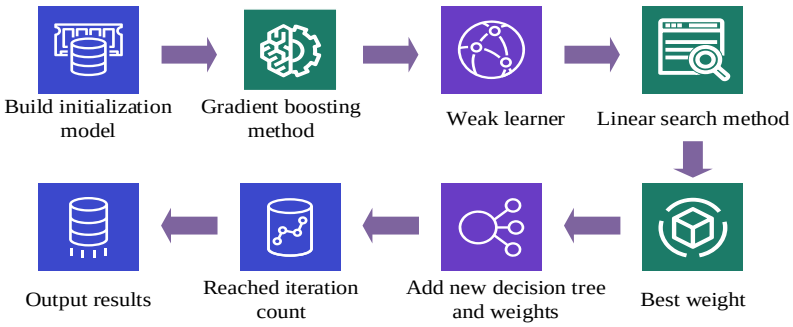

Figure 1 The running process of XGBoost algorithm

As shown in Figure 1, the XGBoost algorithm first constructs an initialization model. Secondly, the gradient boosting method iterates to construct a weak learner, and then uses linear search methods to find the optimal weights. Subsequently, a new decision tree and weights are added to update the model. Finally, when the algorithm has the maximum iteration, the iteration stops and the result is obtained. The target algorithm of XGBoost is shown in Equation (1).

$$Obj(\Theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{j=1}^{t} \Omega(f_j), f_j \in F \tag{1}$$

where y_i represents the real data in dataset D, $\hat{y}_i$ represents the target data, t represents a weak learner, f_j represents the j-th weak learner, $\Omega(f_j)$ represents the regularization term, and $l(y_i, \hat{y}_i)$ represents the training error function of the model. The objective function can effectively constrain the decision tree's complexity, as shown in Equation (2).

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{2}$$

where T signifies the total leaf nodes in the decision tree, ω signifies the output score of the leaf nodes in the decision tree, λ and γ represent constants. The leaf nodes and node scores are unified, as displayed in Equation (3).

$$f_t(x) = \omega_{q(x)}, \omega \in R^T, q: R^d \to \{1,2,\ldots,T\} \tag{3}$$

where x signifies the sample, q signifies the structure of leaf nodes, $q(x)$ represents the mapping leaf node of sample x, and $\omega_{q(x)}$ represents the leaf node's score. The target algorithm $Obj(\Theta)$ is subjected to Taylor expansion, as illustrated in Equation (4).

$$\begin{cases} Obj^{(t)} = -\frac{1}{2}\sum_{i=1}^{T}\left(\frac{D_j^2}{H_j+\lambda}\right) + \gamma T \\ G_j = \sum_{i \in i_j} g_i \\ H_j = \sum_{i \in I_j} h_j \end{cases} \tag{4}$$

where G_j signifies the sum of the first derivative of all input data mapped to the j-th leaf node, H_j represents the sum of its second derivative. The sample set of each leaf node j is $I_j = \{i|q(x\_i) = j\}$, which can convert the traversal form to the traversal based on leaf nodes. There is a dataset D with attribute set $E = \{E_1, E_2, \ldots, E_d, E_{fin}\}$. $E_1, E_2, \ldots, E_d$ represents all attributes. E_{fin} represents the classification label. There are n different types of numerical values for the value of any attribute. The chaos or purity of information entropy can be selected to evaluate the currently selected attribute E_d without dividing points, and the evaluation effect is excellent. However, due to privacy protection conditions, it is necessary to select a reasonable utility function for partitioning. According to the division of different decision trees, there are mainly information gain utility functions based on information entropy, maximum frequency, and utility functions on the ground of information gain ratio replacement, and utility functions based on Gini coefficient. The Gini coefficient utility function is selected for the XGBoost algorithm, and the Gini coefficient is shown in Equation (5).

$$Gini(p) = \sum_{k=1}^{k} pk(1 - p_k) \tag{5}$$

where p_k represents the probability that a sample belongs to class K in a dataset with k categories. The Gini coefficient for binary classification problems is shown in Equation (6).

$$\begin{cases} Gini(p) = 2p(1-p) \\ Gini(D) = 1 - \sum_{k=1}^{k} \frac{|C_k|^2}{|D|} \end{cases} \tag{6}$$

where p represents the probability of the first class in binary classification, $1 - p$ represents the probability of the second type, and D represents the dataset. Based on Equation (4), the utility function of XGBoost algorithm is shown in Equation (7).

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{7}$$

where G_L represents left leaf node splitting, G_R represents right leaf node splitting, $\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}$ represents the loss value when not splitting, $\frac{G_L^2}{H_L+\lambda}$ and $\frac{G_R^2}{H_R+\lambda}$ represent the loss of left and right splitting nodes, respectively. γ represents the threshold, which controls the complexity of the tree. When the loss exceeds the threshold γ, splitting can be achieved, and the splitting node can be found by traversing the features. If the splitting can reduce the objective function more, the value of the utility function will be larger, which can be used for the characteristic function of the exponential mechanism. Based on the above calculations, a Differential Privacy based on XGBoost (DP-XGB) is proposed. The Gini function is selected to determine node purity, using the Classification and Regression Tree (CART) to build a decision tree. Based on the above calculations, the decision tree construction of the DP-XGB algorithm is presented in Figure 2.

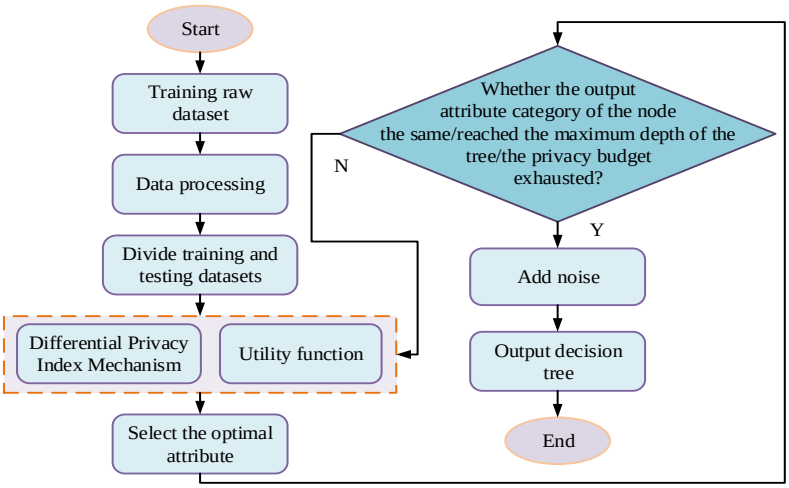

Figure 2 The decision tree construction process of DP-XGB algorithm

From Figure 2, the original training dataset, maximum tree depth, and privacy protection budget are first trained. Secondly, the input data is processed uniformly. Subsequently, the dataset is separated into training and testing datasets in a certain proportion. The optimal attributes of the leaf nodes are selected according to the calculation results of the differential privacy index mechanism and the utility function. When the output attribute category of a leaf node is the same, or reaches the maximum depth, or the privacy budget is exhausted, noise is added to the leaf node to output a single decision tree. Otherwise, the differential privacy index mechanism and utility function are recalculated to select the optimal attribute.

After constructing the differential privacy algorithm DP-XGB based on the XGBoost in the previous section, the study further constructs a multi-party collaborative privacy protection model in a federated learning environment to effectively address privacy protection during transmission in distributed communication processes. Federated learning is a distributed machine learning technology, as shown in Figure 3.

As shown in Figure 3, data sources, federated learning systems, and users are the three major components of federated learning. In a federated learning system, each data source preprocesses user features based on privacy protection, and then multiple parties jointly establish a learning model, and feedback the output results to the user. In differential privacy protection technology, Laplace Mechanism (LM) and exponential mechanism can effectively achieve differential privacy protection. The probability density for Laplace is shown in Equation (8).

$$p(x) = \frac{1}{2\left(\frac{\Delta f}{\varepsilon}\right)} exp\left[-\frac{|x|}{\left(\frac{\Delta f}{\varepsilon}\right)}\right] \tag{8}$$

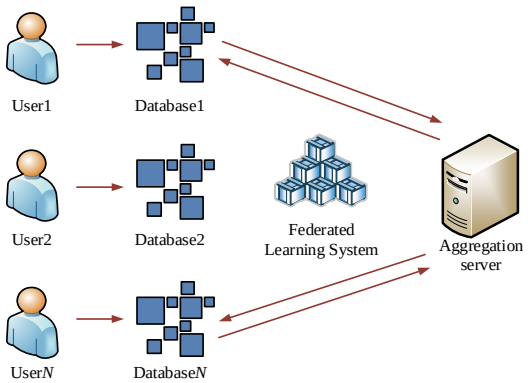

Figure 3 Federated learning architecture

where Δf represents the function sensitivity and $\frac{\Delta f}{\varepsilon}$ represents the scale parameter. In the LM, noise is integrated to the function result and the noise satisfies $Noise{\sim}Lap(\frac{\Delta f}{\varepsilon})$. The inverse function of the probability value of each result is the noise value. The LM is often used for numerical query data. For non-numerical data, the exponential mechanism extracts feature attributes. When the random algorithm satisfies ε - difference privacy, its calculation is shown in Equation (9).

$$F(D,q) = \{r|P(r \in R)\} \propto exp(\frac{\varepsilon q(D,r)}{2\Delta q}) \tag{9}$$

where F represents the random algorithm, D represents the dataset, R represents the output range of the query function, and r signifies any entity object within the output range R, satisfying $r \in R$. $q(D,r)$ represents the availability function to evaluate the quality of the output value r. Δq represents the sensitivity of the function. Under this Equation, the random algorithm F satisfies ε-differential privacy. The study combines exponential mechanism in differential privacy to improve the accuracy of training results, and its enhanced model is presented in Figure 4.

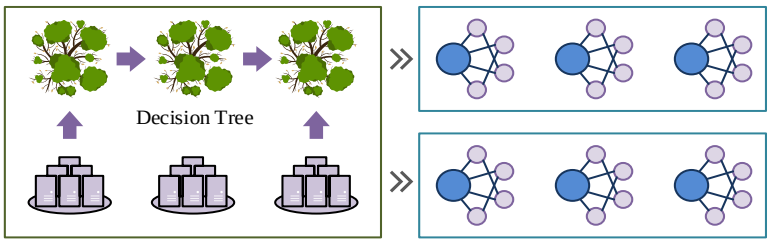

Figure 4 Enhanced model architecture

In Figure 4, the main body of federated learning is the main program aggregator and the participating builder. The federated learning construction has two stages. Firstly, overlapping user features from different participant databases are aligned, and privacy protection conditions must be met during the alignment process. In the second stage, multiple parties collaborate to achieve parameter learning, and then construct a collaborative model. XGBoost, when combined with exponential mechanism, not only satisfies differential privacy, but also reduces the number of exponential mechanism calls and privacy budget allocation, prolongs privacy budget depletion time, protects data privacy while decentralization, and obtains high accuracy training results.

According to Equation (7), since γ represents a threshold, it controls the complexity of the tree. To achieve pre-pruning of the decision tree, when $Gain$ is greater than γ, selecting the optimal splitting node can be achieved. Therefore, it is necessary to first traverse several features of each node, then arrange each feature value in order, and perform linear scanning on the feature values to obtain the optimal splitting feature values. Finally, the optimal splitting point is selected from the feature values to maximize the gain after splitting. When the greedy algorithm scans a large amount of data globally, an approximation algorithm can be used to determine candidate segmentation points. Then the corresponding samples can be placed in the corresponding boxes based on the candidate segmentation points, and the boxes can be accumulated. In the implicit budget allocation stage, equal distribution, uniform distribution, and proportional distribution are selected for privacy budget allocation. After the allocation is completed, the cumulative probability can be calculated using the inverse function based on the Laplace probability density in Equation (8). Therefore, the current noise calculation is shown in Equation (10).

$$Y = \begin{cases} \mu + bln(1+2x), x > 0 \\ \mu - bln(1+2x), x \le 0 \end{cases} \tag{10}$$

where Y represents noise, μ represents the positional parameter, usually set to 0. $x \in [-0.5,0.5]$, and b represents the scale parameter, and $b = \frac{\Delta f}{\varepsilon}$. When sensitivity Δf is 1, $b = \frac{1}{\varepsilon}$. ε represents the privacy budget of the current layer. In the construction of decision trees, the available privacy budget for the current leaf node is ε, the Laplacian noise result is calculated. The privacy budget value is calculated based on the privacy allocation method. At this time, the noise to be added is calculated according to Equation (10). Based on the above calculations of the main program aggregator and participating builders, the model prediction analysis process trained by the multi-party

collaborative Federated Differential Privacy based on XGBoost (FDP-XGB) is shown in Figure 5 when new data appears.

As shown in Figure 5, the aggregator first queries the data that the current data wants to match, then sends this information to the participants and asks for the search direction of the leaf nodes in the next stage. Secondly, when the participant receives the information, it is compared with the threshold γ in the participant's local record. After determining the search direction of the leaf node in the next stage, the information is returned to the aggregator. Subsequently, when the aggregator receives the information, it selects the determined leaf node. The process is repeated until the final node is obtained, and the classification label and weight of the last leaf node are obtained. After searching for a single decision tree, all decision trees are searched accordingly based on the above method. Finally, the results of all leaf nodes obtained are accumulated with their weights to obtain the final class label.

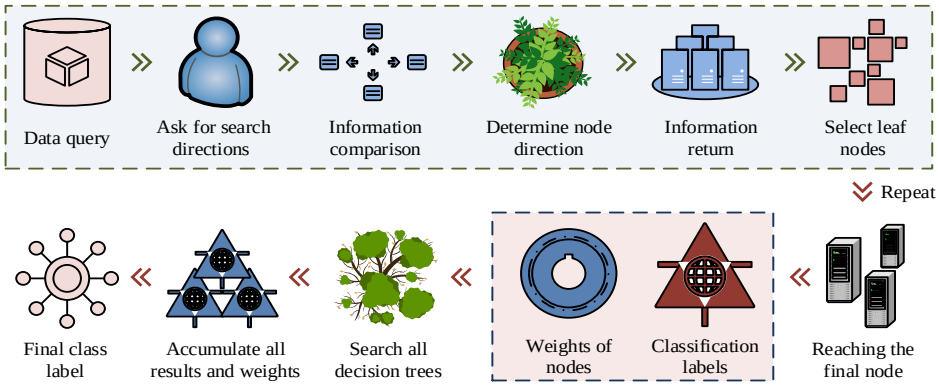

Figure 5 Model prediction analysis process

RESULTS

To verify the feasibility of the proposed DP-XGB based on XGBoost and the multi-party collaborative factor protection algorithm FDP-XGB in data transmission privacy protection, the first section conducts experimental testing on the DP-XGB algorithm. The second section verifies the FDP-XGB. All comparative algorithms in the experiment use the same parameters and dataset to ensure the accuracy.

The research is carried out in a suitable experimental environment, using an 8-core 2.4Hz Intel Core i7 CPU with 16GB of memory, as well as Python 3.7 programming, and operating system CentOS. The UCI dataset in the machine learning benchmark database is adopted, including zoo, glass, haberman, and wdb. The selected dataset contains training and testing sets, with a 7:3 partition ratio. The GBDT, RF, Adaptive Boosting algorithm (AdaBoost), and the proposed DP-XGB algorithm are compared. Due to the impact of the maximum depth of the decision tree, the minimum weight of leaf nodes, and the learning rate on classification accuracy, relevant tests are first conducted to determine the above parameter settings, as displayed in Table 1.

Table 1 Maximum depth of tree, minimum weight of leaf nodes, and learning rate testing

Parameter	Index				
Tree depth	2	4	6	8	10
Classification accuracy/%	82.53	84.95	89.16	86.57	85.24
Minimum weight	2	4	6	8	10
Classification accuracy/%	88.95	86.54	84.97	85.67	84.53
Learning rate	0.2	0.4	0.6	0.8	1
Classification accuracy/%	84.66	88.67	87.52	85.47	85.31

According to Table 1, when the maximum depth was 6, the minimum weight of leaf nodes was 4, and the learning rate was 0.4, the DP-XGB algorithm had the best classification accuracy. Therefore, the above adjustment parameters are used for performance testing in subsequent experiments. Firstly, the trend test results of the fitness function values of the four algorithms in different datasets are shown in Figure 6.

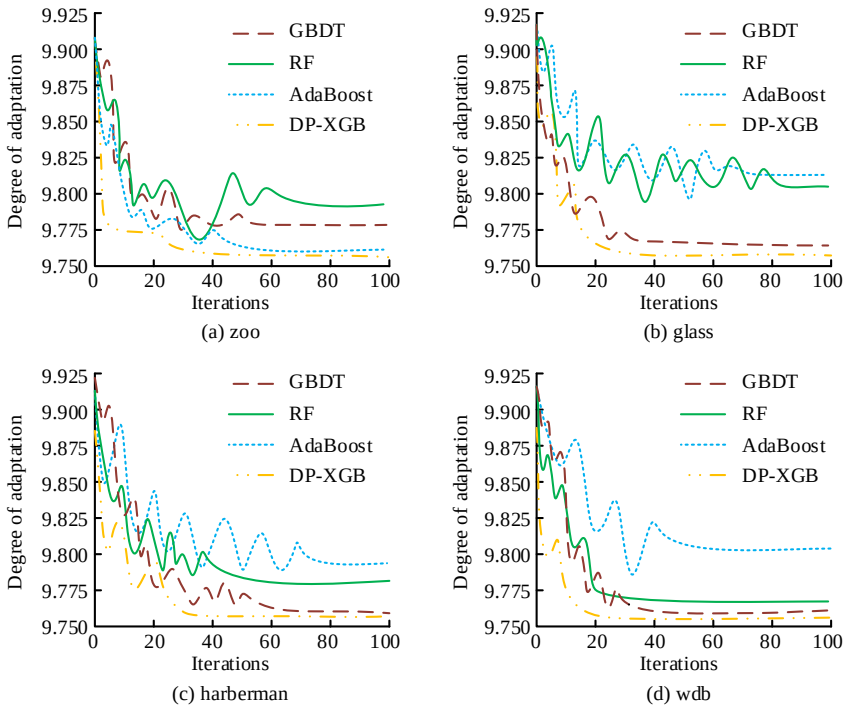

Figure 6 Trend chart of fitness function values

Figure 6 shows the fitness function curves of four algorithms on the zoo, glass, haberman and wdb datasets. In Figure 6 (a), the fitness of the DP-XGB algorithm was 9.760 after 32 iterations, which was the lowest among the comparison methods. In Figure 6 (b), when the DP-XGB algorithm iterated 30 times, its fitness function value was 9.758. In Figure 6 (c), the fitness function value of DP-XGB algorithm was 9.757 after 33 iterations. In Figure 6 (d), the DP-XGB algorithm was 9.755 after 26 iterations. As shown in the Figure, in the four datasets, DP-XGB can be optimized with the least number of iterations, and its convergence speed and accuracy are both optimal in comparison algorithms. Secondly, the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-Square (R2) are compared, as shown in Table 2.

Table 2 RMSE, MAE, MAPE, R2 values of each algorithm

Data set	Index	Algorithm			
		GBDT	RF	AdaBoost	DP-XGB
Training set	RMSE	2.32	2.50	2.61	1.61
	MAE	1.36	1.51	1.97	1.20
	MAPE	0.22	0.31	0.44	0.19
	R^2	0.76	0.62	0.54	0.86
Testing set	RMSE	2.33	2.60	2.55	1.58
	MAE	1.38	1.67	1.83	1.24
	MAPE	0.25	0.36	0.45	0.21
	R^2	0.75	0.59	0.49	0.84

As shown in Table 2, the RMSE refers to the deviation between the predicted and the true values. The MAE signifies the deviation between the observed and the mean. The MAPE is used to measure the relative magnitude of deviation. The R2 value represents the variation part of the dependent variable. In the two sets, the RMSE, MAE and MAPE of the DP-XGB were both the lowest in the comparison algorithms. The R2 value was maximum, which was close to 1. It indicates that the algorithm has excellent predictive ability and fitness, indicating the best model quality. Finally, the classification accuracy and time consumption test results of the four algorithms are shown in Figure 7.

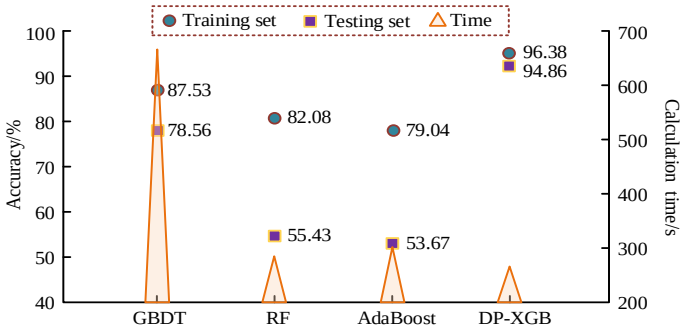

Figure 7 Accuracy and time consumption of wdb dataset

From Figure 7, the classification accuracy of the four algorithms in the wdb differed significantly from the computation time for the entire dataset. In the two sets, the classification accuracy of the DP-XGB algorithm was 96.38% and 94.86%, respectively, which was the best among the comparison algorithms and had the best prediction stability. The accuracy of RF and AdaBoost was similar, but lower than that of GBDT algorithm. In terms of computational time, the total time consumption of GBDT, RF, AdaBoost, and DP-XGB algorithms was 658s, 289s, 309s, and 272s, respectively. The DP-XGB algorithm has the best classification accuracy and computational time.

After conducting experimental tests on the DP-XGB algorithm, the study further tests the FDP-XGB. The experimental environment and parameters for testing are the same as the DP-XGB algorithm in the previous section. XGBoost and DP-XGB are used as comparison algorithms. Firstly, the Receiver Operating Characteristic curves (ROC) of the three algorithms on the wdb dataset is shown in Figure 8.

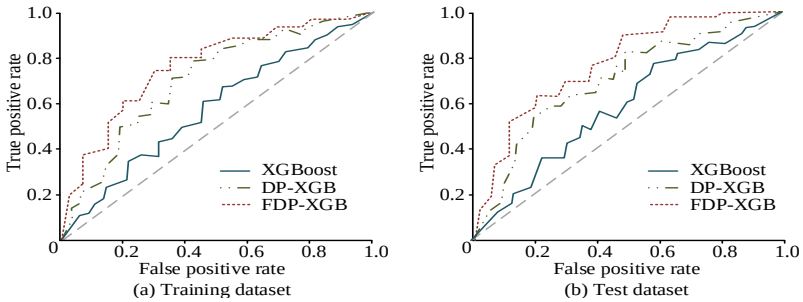

Figure 8 Statistical results of AUC indicators

Figures 8 (a) and 8 (b) show the ROC curves of the three algorithms. The horizontal axis signifies the false positive rate, and the vertical axis signifies the true positive rate. The larger the Area Under the Curve (AUC) enclosed by the ROC curve and the horizontal and vertical coordinates, the better the model performance. The ROC curve of FDP-XGB was higher than other comparison algorithms in both the training and testing sets. The DP-XGB curve was surrounded by the FDP-XGB curve. The XGBoost curve was further surrounded by the curves of DP-XGB and FDP-XGB. In the training set, the AUC values of XGBoost, DP-XGB, and FDP-XGB algorithms were 0.57, 0.75, and 0.79, while the AUC in the testing set was 0.55, 0.73, and 0.78. Secondly, the experimental results of the predicted values and actual values of the three algorithms are shown in Figure 9.

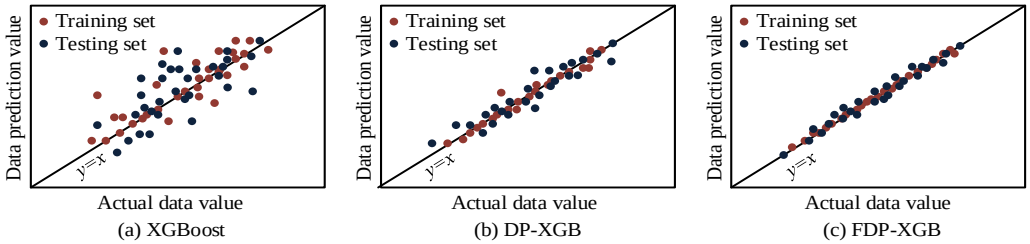

Figure 9 Comparison between predicted and actual values

Figures 9 (a), 9 (b), and 9 (c) show the true and predicted values of XGBoost, DP-XGB and FDP-XGB algorithms. As shown in the Figure, the FDP-XGB algorithm had the smallest scatter distribution and was closest to the $y = x$ function axis. The scatter of XGBoost algorithm had the highest dispersion among the three algorithms, and its prediction accuracy was lower than other comparison functions, resulting in poor data prediction performance. The dispersion degree of the DP-XGB algorithm was between the two, and the prediction accuracy was in the middle. The FDP-XGB algorithm had the highest prediction accuracy and good data prediction performance. Finally, the running time is displayed in Table 3.

Table 3 The running time of three algorithms

Index	Algorithm	Data volume/10^4					
		3	6	9	12	15	18
Running time/s	XGBoost	5.2	13.9	17.6	20.8	25.9	29.5
	DP-XGB	4.9	9.4	10.8	12.9	14.1	22.1
	FDP-XGB	5.1	12.3	15.7	17.6	19.8	16.5
Resource consumption rate/%	XGBoost	15.7	22.8	27.9	35.7	42.3	48.5
	DP-XGB	11.4	14.5	17.4	21.0	22.8	24.9
	FDP-XGB	10.4	12.8	14.9	17.4	18.3	21.1

According to Table 3, when the data volume was 3×104, the calculation time of XGBoost, DP-XGB, and FDP-XGB algorithms was 5.2s, 4.9s, and 5.1s respectively. When the data volume was small, the running time of the three algorithms was similar, with only a slight difference. However, when the data volume was 18×104, the computation time of XGBoost, DP-XGB, and FDP-XGB algorithms was 29.5s, 22.1s, and 16.5s, respectively. The FDP-XGB algorithm allocates privacy budgets to each decision tree, with budget consumption on an exponential scale, resulting in a higher total computational time than the DP-XGB algorithm. In the resource consumption rate test, as the amount of data increased, the resource consumption rates of the three algorithms showed a gradual upward trend, but the growth rate of FDP-XGB was lower than other comparison models. When the data volume was 18×104, the resource consumption rates of XGBoost, DP-XGB, and FDP-XGB algorithms were 48.5%, 24.9%, and 21.1%, respectively.

CONCLUSION

This study proposes a privacy protection technology that combines the XGBoost differential privacy protection algorithm with federated learning to address privacy security and data silos in data mining. Performance testing of the DP-XGB algorithm showed that optimal recognition accuracy was achieved with a maximum decision tree depth of 6, a minimum leaf node weight of 4, and a learning rate of 0.4. The DP-XGB algorithm exhibited the lowest iteration count and fitness function values across different datasets. In the training set, the root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and R2 were 1.61, 1.20, 0.19, and 0.86, respectively, demonstrating superior performance compared to other methods. FDP-XGB demonstrated the highest prediction accuracy when comparing true and predicted data values, outperforming DP-XGB. For a data volume of 18×104, the computation times for XGBoost, DP-XGB, and FDP-XGB were 29.5 seconds, 22.1 seconds, and 16.5 seconds, respectively, with resource consumption rates of 48.5%, 24.9%, and 21.1%. However, this study was limited to cleaned data and did not test uncleaned datasets. Future research should aim to improve the model's performance on uncleaned data and reduce its reliance on computational resources.

The FDP-XGB algorithm allocates privacy budgets to each decision tree, with budget consumption on an exponential scale, resulting in a higher total computational time than the DP-XGB algorithm. Future research should aim to improve the model's performance on uncleaned data and reduce its reliance on computational resources.

ACKNOWLEDGEMENTS

This work was supported in part by the Guangdong Power Grid Co., Ltd. Huizhou Power Supply Bureau Science and Technology Project under grant 031300KC23030012.

The authors declare that they have no conflicts of interest.

REFRENCES

[1] Gheisari M, Hamidpour H, Liu Y, Saedi P, Raza A, Jalili A, Rokhsati H, Amin R. Data Mining Techniques for Web Mining: A Survey. Artificial Intelligence and Applications, 2023, 1(1): 3-10.
[2] Guo P, Ye B, Chen Y, Li T, Yang Y, Qian X, Yu X. A differential privacy protection protocol based on location entropy. Tsinghua Science and Technology, 2022, 28(3): 452-463.
[3] Chen J, Ma H, Zhao D, Liu L. Correlated differential privacy protection for mobile crowdsensing. IEEE Transactions on Big Data, 2021, 7(4): 784-795.
[4] Zhang J, Zeng ZY, Si KL, Ye XC. Entropy-driven differential privacy protection scheme based on social graphlet attributes. The Journal of Super computing, 2024, 80(6): 7399-7432.
[5] Zhang C, Wang Y, Wang W, Zhang H, Liu Z, Tong X, Cai Z. A Personalized Location Privacy Protection System in Mobile Crowdsourcing. IEEE Internet of Things Journal, 2023, 11(6): 99995-10006.
[6] H. Liu, D. Xu, Y. Tian, C. Peng, Z. Wu, and Z. Wang, "Wasserstein Generative Adversarial Networks Based Differential Privacy Metaverse Data Sharing," in IEEE Journal of Biomedical and Health Informatics, 2024, 28(11): 6348-6359.
[7] Hewage U, Sinha R, Naeem M A. Privacy-preserving data (stream) mining techniques and their impact on data mining accuracy: a systematic literature review[J]. Artificial Intelligence Review, 2023, 56(9): 10427-10464.
[8] Yang M, Guo T, Zhu T, et al. Local differential privacy and its applications: A comprehensive survey[J]. Computer Standards & Interfaces, 2024, 89: 103827.
[9] Pillutla K, Kakade SM, Harchaoui Z. Robust aggregation for federated learning. IEEE Transactions on Signal Processing, 2022 9(70): 1142-1154.
[10] Lee Y, Park S, Ahn JH, Kang J. Accelerated federated learning via greedy aggregation. IEEE Communications Letters, 2022, 26(12): 2919-2923.
[11] Chung W, Chang Y, Hsu C, Chang C, Hung C. Federated feature concatenate method for heterogeneous computing in Federated Learning. Computers, Materials and Continua, 2023, 75(1): 351-370.
[12] Xu J, Glicksberg BS, Su C, Walker P, Bian J, Wang F. Federated learning for healthcare informatics. Journal of healthcare informatics research. 2021, 5(1): 2-10.
[13] Zhang F, Kuang K, Chen L, You Z, Shen T, Xiao J, Zhang Y, Wu C, Wu F, Zhuang Y, Li X. Federated unsupervised representation learning. Frontiers of Information Technology & Electronic Engineering, 2023, 24(8): 1181-1193.
[14] Zong R, Qin Y, Wu F, et al. Fedcs: Efficient communication scheduling in decentralized federated learning[J]. Information Fusion, 2024, 102: 102028.
[15] Ren C, Yu H, Peng H, et al. Advances and open challenges in federated learning with foundation models[J]. arXiv preprint arXiv:2404.15381, 2024.
[16] Sandilya S S, Das B S, Proust S, et al. Discharge estimation in compound channels with converging and diverging floodplains using an optimised Gradient Boosting Algorithm[J]. Journal of Hydroinformatics, 2024, 26(5): 1122-1149.
[17] I. Joshi, M. Grimmer, C. Rathgeb, C. Busch, F. Bremond, and A. Dantcheva, "Synthetic Data in Human Analysis: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(7): 4957-4976.
[18] X. Pan et al., "Flagger: Cooperative Acceleration for Large-Scale Cross-Silo Federated Learning Aggregation," 2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA), Buenos Aires, Argentina, 2024: 915-930.
[19] A. Silvi, A. Rizzardi, D. Caldarola, B. Caputo and M. Ciccone, "Accelerating Federated Learning via Sequential Training of Grouped Heterogeneous Clients," in IEEE Access, 2024, 12: 57043-57058.