Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

A differentially private distributed data mining scheme with high efficiency for edge computing

Xianwen Sun[1], Ruzhi Xu[1], Longfei Wu[2] and Zhitao Guan[1*]

Abstract

A wide range of data mining applications benefit from the low latency offered by edge computing. However, edge computing suffers from limited computing resources, which inhibits the applications of the computationally expensive data mining methods. In the edge-cloud environment, usually, the participants turn to collaboratively train machine-learning models that yield more accurate prediction results. However, data owners may not be willing to sharing the own data for the privacy concerns. To handle such disparate goals, we focus on tree-based distributed data mining scheme with differential privacy, which is computationally friendly. The basic idea of our approach is based on a distributed ensemble strategy. Each participant builds an elegant decision model based on their own data, which has a good tradeoff between the computation and the accuracy of the data distribution, and shares it with other participants after being injected with the elaborate noise. Then the useful knowledge transferred from the decision models is acquired by other participants in an adaptive ensemble strategy. Both the theoretical analysis and the experiments show that our scheme provides an efficient data mining manner that can achieve a good prediction accuracy while providing rigorous privacy guarantee over the distributed data.

Keywords: Distributed data mining, Differential privacy, AdaBoost, Edge computing

Introduction

During the past few years, with the advent of the edge computing, numerous smart devices have been popularized and subsequently massive data has been produced [1]. Data mining has been serving as an epoch-making technique to extract the hidden information and valuable knowledge automatically and intelligently. Thus, a wide range of data mining applications in edge computing have been deployed to benefit our daily lives, e.g. smart healthcare [2], smart homing [3], and intelligent transportation [4, 5], etc.

A wide range of data mining applications benefit from the low latency offered by edge computing. However, edge computing suffers from limited computing resources, which inhibits the applications of the computationally expensive data mining methods like artificial neural networks. In the edge computing environments, usually, the participants turn to jointly learn an accurate machine-learning model over multiple data owners. Moreover, the collection and preprocessing of source data are implemented on the edge devices side, which involves some personal privacy data. For example, in the smart healthcare situation, where the massive embedded and wearable devices are used to monitor the physical state of patients, the monitoring data can be used not only for medical diagnosis or treatments, but also for malicious institutions such as insurance companies [6]. Thus, it is pressing to resolve the privacy issue to ensure the data mining over multiple data sets in resource-limited edge computing.

Existing researches on privacy-preserving distributed data mining mainly focus on cryptography [7–9]. A

* Correspondence: guan@ncepu.edu.cn
[1]School of Control and Computer Engineering, North China Electric Power University, Beijing, China
Full list of author information is available at the end of the article

naive solution is to only share the encrypted data and all operations are made directly to the ciphertext. For example, the fully homomorphic encryption is used to protect the private information in Federated Learning [10]. However, these cryptographic solutions usually come with huge additional computing overhead, which hinders their applications in the edge-cloud environment. Differential privacy is a new definition of privacy, whose implementation is based on the perturbation technology [11]. So far, there have been many researches on designing differentially private algorithms for mining tasks such as clustering [12], decision trees [13] and neural networks [14], because of the efficient implementation of differential privacy.

Recent years, we have witnessed the wide use of tree-like model in mining tasks. The current researches on privacy-preserving decision tree mainly concentrate on how to ensure the single data owner publish their data without revealing any private information, but have ignored the distributed scenario where the data is held by multiple owners [15]. A recent work [16] proposed a tree-based data mining scheme for the distributed scenario. However, it cannot be used in edge computing directly for its inflexible training process.

To address the privacy leakage issue in the resource-limited edge-cloud computing, we propose an efficient and privacy-preserving tree-based data mining method over the distributed dataset. Specifically, based on the AdaBoost framework, we design a differentially private scheme which can build an elegant model while achieving a tradeoff between the accuracy and privacy. In addition, we develop an adaptive ensemble strategy to learn the data distribution more accurately when building the basic learner under the AdaBoost framework. Our experiments prove that our scheme is effective in both model construction and privacy protection.

To sum up, we make the following key contributions:

(1) We design a distributed data mining scheme with privacy preservation for edge-cloud computing, which can not only achieve a good tradeoff between the accuracy of the built model and privacy preservation, but also take the limitations of computation resources into account.

(2) We propose an adaptive ensemble strategy for the construction process, which allows the participants to improve the prediction accuracy of the basic learner by combining the models with similar data distribution without access to these private data.

(3) We theoretically analyze the privacy and computational complexity of the proposed scheme. Then a series of simulation experiments are conducted on two real-world datasets, through which we prove the feasibility of our scheme.

The rest of the paper is organized as follows. The related work is presented in Section 2. In Section 3, we introduce the relevant preliminaries. Section 4 gives an overview of the proposed scheme and the design goals. In Section 5, we describe our privacy-preserving distributed data mining scheme in detail. Section 6 demonstrates the feasibility of our scheme by giving a theoretical analysis. In Section 7, we evaluate the performance on two real-world datasets. Finally, we conclude the paper in Section 8.

Related work

Recently, privacy-preserving data mining has attracted much attention especially in edge computing where data mining often involves multiple parties. And several techniques have been proposed to protect privacy of data, such as *k*-anonymity, randomization and cryptographic tools [17]. Differential privacy is the first privacy protection model with rigorous and provable mathematical definition [18]. It has been widely used to design privacy-preserving schemes for data mining such as decision tree [19], principal component analysis [20], and artificial neural network [21].

As a common data classification model, decision tree has been applied in different application scenarios. It is widely used because of its transparency and intelligibility. In order to solve the problem of privacy disclosure, several decision tree construction schemes satisfying differential privacy have been proposed. Blum et al. first proposed the SuLQ framework to achieve differential privacy, and proposed SuLQ-based ID3 algorithm by combining with ID3 [22]. But it at the expense of model accuracy for the excessive noise required to protect the private information. The work [13] proposed a DiffP-ID3 algorithm based on the exponential mechanism to effectively reduce the waste of privacy budget. Additionally, Friedman and Schuster further proposed the DiffP-C4.5 algorithm to overcome the limitation of DiffP-ID3 that it can only solve discrete attributes [13]. Rana et al. [23] and Fletcher [24] proposed the methods to build differentially private random forest, which can reduce the impact of noise on model accuracy by integrating several decision trees into an ensemble.

In order to solve the problem of privacy protection in the distributed environment, Lindell et al. proposed to use garbled circuit to select the segmentation attribute [25]. Mohammed et al. proposed an differentially private algorithm under the noninteractive setting to implement two-party vertically partitioned data mining [26]. And it was used in the scenario of the medical data mining [27]. To implement privacy-preserving collaborative data mining without trust third part, Goryczka et al. presented an m-privacy pruning strategy under the hypothesis that participants could infer the data records contributions of other data owners by using their own data [28]. The work [29] proposed to build an ID3 decision tree over distributed data and secure the private information of participants by the secret sharing scheme. However, these solutions can cause higher communication and computational costs. Gambs et al. proposed the MultBoost algorithm, which achieves the privacy-preserving collaborative data mining based on Boosting and differential privacy protection [30], however, it requires a secure third party to accomplish the information interaction and model integration, which brings additional communication overhead.

Existing researches mainly focus on how the data owner can safely publish the tree-based data using differential privacy [30]. Only the work [16] proposed a distributed data mining scheme. In this scheme, the participants share the differentially private model and the corresponding parameters to build the model collaboratively. However, the collaborative training process must follow a certain sequence, making it not suitable for the edge computing.

Preliminaries

Differential privacy

Definition 1. Differential privacy

An algorithm F satisfies ε- differential privacy if for any dataset D and its adjacent dataset $D^{'}$ with symmetric difference $|D\Delta D^{'}|=1$, it satisfies:

$$\frac{\Pr[F(D)\in S]}{\Pr\left[F\left(D^{'}\right)\in S\right]}\leq e^{\varepsilon} \tag{1}$$

where S denotes the subsets of all possible outputs of the algorithm F, ε is privacy budget. The smaller the privacy budget is, the higher the degree of privacy protection is provided by the algorithm F.

Definition 2. Global sensitivity

The global sensitivity of the function f is given below:

$$\Delta f = \max_{D,D^{'}} \|f(D)-f(D')\|_{L_1} \tag{2}$$

Typically, for numerical query functions, the random noise satisfying Laplacian distribution can be added to the query results to satisfy ε- differential privacy.

Theorem 1. Laplace mechanism

For any function $f:D\rightarrow R^{d}$, function F provides ε-differential privacy on condition that:

$$F(D)=f(d)+Lap(\Delta f/\varepsilon) \tag{3}$$

where $Lap(\Delta f/\varepsilon)$ is a random noise drawn from the Laplace distribution whose position parameter is 0 and scale parameter is $\Delta f/\varepsilon$.

However, when it comes to non-numerical query functions, the Exponential mechanism should be used.

Theorem 2. Exponential mechanism

For any quality function M who takes the dataset D as input, the output is $r\in Range(M)$. Its sensitivity is Δu. Then, the function M provides ε- differential privacy when it outputs r from *Range (M)* with a probability proportional to $\exp(\varepsilon u(D^{'},r)/2\Delta u)$.

Theorem 3. Sequential Composition

Suppose that A_1, A_2, $\cdots$, A_n are random algorithms and A_i satisfies ε_i-differential privacy. For arbitrary dataset D, the sequential composition of these n algorithm

$$A(D)=\{t_1=A_1(D),\cdots,t_n=A_n(D,t_1,\cdots,t_{n-1})\} \tag{4}$$

satisfies ε-differential privacy, and $\varepsilon=\sum_{i=1}^{n}\varepsilon_i$.

Theorem 2. Parallel Composition

Suppose that a random algorithm A satisfies ε-differential privacy and a dataset D is divided into m disjoint subsets $\{D_1,D_2,\cdots,D_m\}$. Then the parallel composition of A on these disjoint subsets

$$A(D)=\{t_1=A(D_1),t_2=A(D_2),\cdots,t_m=A(D_m)\} \tag{5}$$

satisfies ε-differential privacy.

AdaBoost algorithm

According to the relationship between the individual learners, ensemble learning can be divided into two categories. One is that basic learners are independent of each other such as Random Forest (RF), where the performance and training process of basic learner are not affected by others. The other is that there is strong dependency between basic learners such as AdaBoost [31].

Let $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$ represent the training dataset. $y_i\in\{-1,+1\}$ is the class label associated with x_i. The distribution $W=(w_{1i},w_{12},\cdots,w_{1N})$ denotes the weight of each instance in D, and the initial value of w is $w_{1i}=1/N$. During the training process, W_{t+1} is readjusted from W_t according to the error rate of the basic learner in each iteration. Each basic learner $H_t(x):x\rightarrow\{-1,+1\}$ is obtained by running learning algorithm in the training data set with weight distribution of W_t. The final classifier is obtained by the linear combination of basic learners.

$$g(x) = \sum_{t=1}^{T} \alpha_t H_t(x) \tag{6}$$

Model and goals

System model

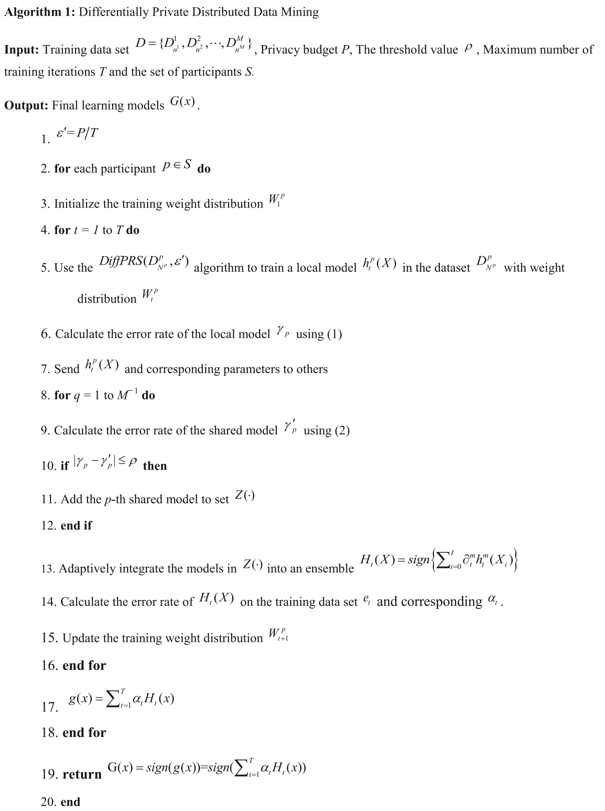
Figure 1 gives an overview of our system model. Assuming that there are M participants in the edge-cloud environment, each participant can exchange the information over networks and each of them occupies a private dataset. Note that the data miner can be one of those participants. During the mining process, each participant builds the learning model based on the AdaBoost algorithm, and the basic learner is composed by the local model and the shared model. Specifically, after training the local model, each participant shares the model and the corresponding parameters to others rather than the original data due to the privacy protection. Furthermore, the *DiffPRS* algorithm is proposed to secure the privacy information revealed by the model and the corresponding parameters. When receiving the shared model, each participant selects the suitable model to be integrated based on the shared model's performance on their respective private datasets, and integrate it in an adaptive manner.

When a new participant joins the system, instead of re-training the entire model, the new participant only needs to execute the proposed learning algorithm to train the local models and adaptively integrate models. For the existing participants, they only need to decide whether the new shared model is worth to integrate or not.

Model goals

In order to solve the privacy issue in distributed data mining for the edge computing, the design goals of our proposed scheme can be summarized as:

1) Privacy protection: sharing the models and its corresponding parameters will not disclose the private information. In other words, the participants in the system do not worry about the potential malicious participants among other participants.
2) Accuracy and efficiency: achieve a good tradeoff between the accuracy of the built model and privacy preservation while taking the limitations of computation resources in edge computing into account.



Algorithm 1: Differentially Private Distributed Data Mining

Input: Training data set $D=\{D_{n^1}^1, D_{n^2}^2, \cdots, D_{n^M}^M\}$, Privacy budget *P*, The threshold value ρ, Maximum number of training iterations *T* and the set of participants *S*.

Output: Final learning models $G(x)$.

```
$\varepsilon' = P/T$
for each participant $p \in S$ do
    Initialize the training weight distribution $W_1^p$
    for $t = 1$ to $T$ do
        Use the $DiffPRS(D_{N^p}^p, \varepsilon')$ algorithm to train a local model $h_t^p(X)$ in the dataset $D_{N^p}^p$ with weight distribution $W_t^p$
        Calculate the error rate of the local model $\gamma_p$ using (1)
        Send $h_t^p(X)$ and corresponding parameters to others
        for $q = 1$ to $M^{-1}$ do
            Calculate the error rate of the shared model $\gamma'_p$ using (2)
            if $|\gamma_p - \gamma'_p| \le \rho$ then
                Add the $p$-th shared model to set $Z(\cdot)$
            end if
        Adaptively integrate the models in $Z(\cdot)$ into an ensemble $H_t(X) = sign\left\{\sum_{i=0}^{I} \partial_t^m h_t^m(X_i)\right\}$
        Calculate the error rate of $H_t(X)$ on the training data set $e_t$ and corresponding $\alpha_t$.
        Update the training weight distribution $W_{t+1}^p$
        end for
    $g(x) = \sum_{t=1}^{T} \alpha_t H_t(x)$
end for
return $G(x) = sign(g(x)) = sign(\sum_{t=1}^{T} \alpha_t H_t(x))$
end
```

Description of proposed scheme

Problem definition

Suppose that there are M participants and each of them has a private dataset $D = \{D_{N^1}^1, D_{N^2}^2, \cdots, D_{N^M}^M\}$. Among them, the dataset owned by the m-th participants is $D_{N^m}^m = \{(X_1^m, y_1^m), (X_2^m, y_2^m), \cdots, (X_{N^m}^m, y_{N^m}^m)\}$, which contains N^m tuples and each tuple $X_n^m = (x_{n,1}^m, \cdots, x_{n,K^m}^m)$ has K^m attributes. Each label $y_i \in \{-1, +1\}$ for the binary classification is considered in the proposed scheme.

Our proposed scheme

The scheme proposed in this paper is designed to solve the problem of data privacy protection for distributed data mining in the edge-cloud environment. Considering that the participates may be limited in computation resources, it is hard to train a model by using a learning algorithm that requires high computing resources like artificial neural network (ANN) and so on. Thus, based on the AdaBoost framework, we propose a mining method over the distributed data to reduce the dependency on the computational resources and data collection.

In the AdaBoost framework, firstly, it initializes the weight by average method $W_1 = (w_{1,\ 1}, w_{1,\ 2}, \cdots, w_{1,\ N})$, where the $w_{1,\ i}$ is equal to $1/N$. Secondly, it trains the basic learner $h(x)$ based on the dataset with W_i weight distribution. Then, it calculates its error rate

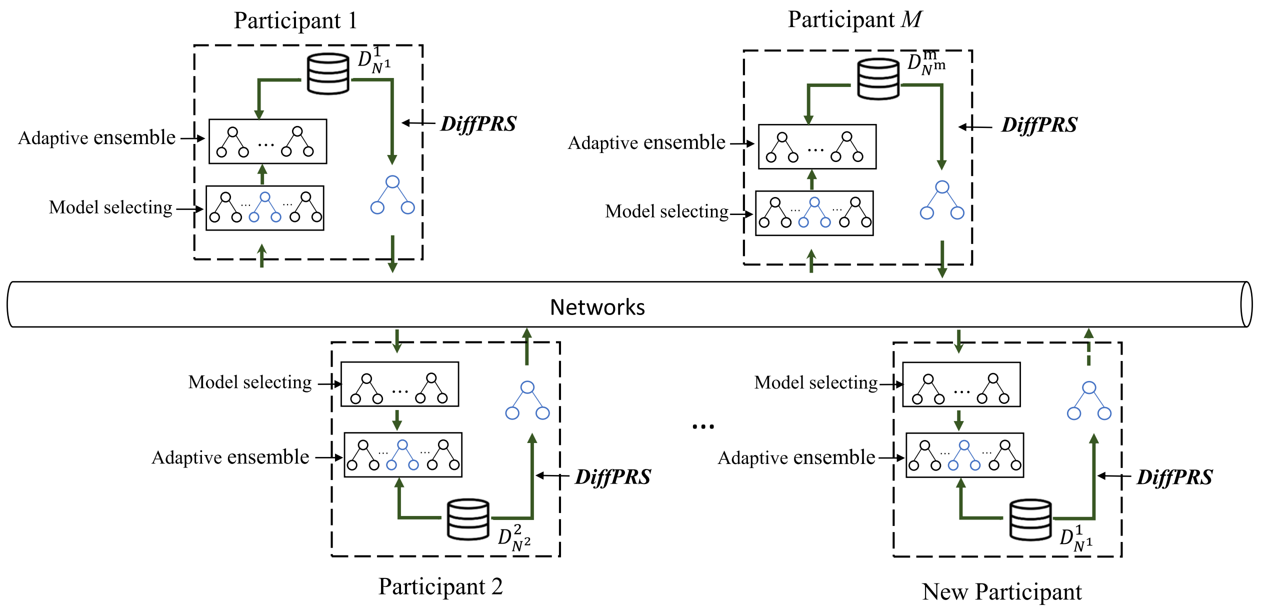


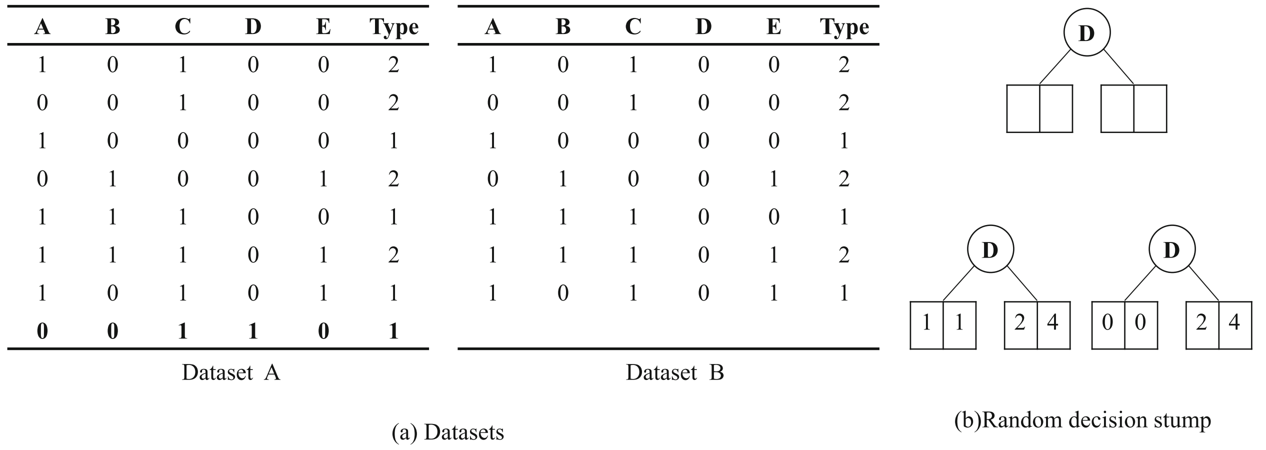
Fig. 1 An Overview of Differentially Private Distributed Data Mining

$e_t = \sum_{i=1}^{N} w_{t,i} I(H_t(X_i) \neq y_i)$ and $\alpha_t = \log((1 - e_t)/e_t)/2$ where the α_t is to measure the importance of weak learner $h(\cdot)$ among the final classifier. After several iterations, the weak models are boosted to be a strong ensemble model.

The basic idea of proposed scheme is that each participant uses the proposed *DiffPRS* algorithm to train the local model that satisfies the differential privacy and shares it with others, then participants select the proper shared model based on the local model and integrate the local model into an ensemble as the basic model in each iteration. Considering the private information contained in the datasets of the participants, we share the model and the corresponding parameters that satisfy the differential privacy instead of the original data among the participants. In each iteration, the privacy preservation is achieved by adding noise when calculating the supports of the leaf nodes where the supports refers to the number of records in the nodes. The details of our proposed scheme are outlined in Algorithm 1.

Building differentially private random decision stump

When it comes to choosing the suitable model as the basic learner, it is crucial to take the limited computational capability of edge computing into account. It means that the basic learner must be friendly to computation. Random decision stump is a typical representative of the tree-like classification model. In



A	B	C	D	E	Type
1	0	1	0	0	2
0	0	1	0	0	2
1	0	0	0	0	1
0	1	0	0	1	2
1	1	1	0	0	1
1	1	1	0	1	2
1	0	1	0	1	1
0	**0**	**1**	**1**	**0**	**1**

Dataset A

A	B	C	D	E	Type
1	0	1	0	0	2
0	0	1	0	0	2
1	0	0	0	0	1
0	1	0	0	1	2
1	1	1	0	0	1
1	1	1	0	1	2
1	0	1	0	1	1

Dataset B

(a) Datasets

(b)Random decision stump

Fig. 2 An Example of Random decision Stump

general, its performance is not as good as the traditional decision trees, but its construction process is more efficient in computation as it does not completely depend on the specific database. Therefore, the random decision stump is used as the basic learner being boosted to the “strong” classifier in the AdaBoost framework.

From the previous researches, it can be known that the direct publishing of the decision tree model and its parameters has the potential of privacy disclosure [32, 33]. Compared with the traditional decision tree construction methods, each split attribute is determined by randomly selection in the random stump construction process and the queries are executed only when determining the classification in leaf nodes. However, direct publishing of the random stump and the corresponding parameters does not meet the requirements for protecting data privacy. Assuming that the structure of the random stump based on dataset *A* is shown in Fig. 2b and that the attackers have the ability to build an adjacent dataset *B* which has only one different record compared to dataset *A* (shown in Fig. 2a), the private information of dataset *A* can be analyzed by comparing the supports of leaf nodes or the label of each leaf nodes in the random stump after building several random stumps based on dataset *B*.

When building the structure of the random stump, there may be a situation where the support of certain leaf node is zero. Generally, this type of nodes will be omitted when publishing the random stump. However, in order to prevent the attackers from observing a specific random stump structure to obtain the private information of the dataset, the nodes with zero support cannot be omitted when publishing the models. Hence, it is essential to introduce an appropriate noise into the leaf nodes to achieve privacy protection when publishing random decision stump. When it comes to choosing the split attributes, it is unnecessary to introduce the extra randomness, since random selection is used to determine the attributes of split nodes, which does not depend on the dataset queries and introduce randomness at the same time. The details of constructing a differentially private random decision stump is presented in Algorithm 2.

Noting that all the differentially private decision stumps have the same structure, for simplicity, the model and its corresponding parameters can be unified into a vector ***V*** when exchanging the information among the multiple participants. Usually, vector ***V*** contains three main elements: root node, the classification of leaf nodes and its corresponding supports.

Selecting suitable models

Each participant will receive M^{-1} models shared by others during each iteration. Considering that the performance of random decision stump is limited, for each specific participator, integrating their own local model and the shared models into an ensemble model can reduce the error rate of the base model of each iteration and decrease the iteration time of the training process. In other words, model integration can cut down the amount of noise introduced for privacy protection, thereby strengthen the availability of the shared models. Therefore, how to integrate the models into an ensemble model in an appropriate manner so that the ensemble model has a higher prediction accuracy than the local models is significant in the proposed scheme. In addition, simple and indiscriminate integration of various shared models may decrease the prediction accuracy of the local models. To prevent this from happening, we need to decide which ones of the shared models are suitable to integrate.

Algorithm 2: Differentially Private Random Decision Stump (**DiffPRS**)

```
Input: Training data set $D$, privacy budget $\varepsilon$, attribute sets $A=\{a_1,a_2,\cdots,a_n\}$ and classification label
$lab\in\{+1,-1\}$
Output: Random decision stump that satisfies $\varepsilon$-differential privacy

1. Randomly select an attribute $a$ from attribute set $A$, and randomly select an attribute value $a_i$
    2. if $a$ is discrete attribute then
    3. Divide the dataset into $D_1=D_{a=a_i}$ and $D_2=D_{a\neq a_i}$
    4. else
    5. Divide the dataset into $D_1=D_{a<a_i}$ and $D_2=D_{a>a_i}$
    6. end if
7. Add the Laplace noise $Lap(1/\varepsilon)$ to $|D_i^j|$ of each divided subspace, where $|D_i^j|$ represent the number of instances
  with classification label $j$ in the $i$-th subspace
    8. Determine subspace classification labels:
  $lab(D_i)=\arg\max_j |D_i^j=\{x_k\in D_i \ and \ lab(x_k)=j\}|$
    9. end
```

The basic idea is to choose those shared models whose data distribution is similar to the *p*-th participant. Meanwhile, it can effectively reduce the negative effects caused by the sharing model constructed based on the malicious data sets. Due to the privacy protection constraint, checking up the private data sets of other participants is infeasible. Thus, by comparing all the error rates of shared models on the training dataset of the *p*-th participant with the training error of the local model, we can approximately decide which corresponding dataset is similar to the *p*-th participant. Note that the shared model might include certain decision stumps for some unique attributes. So, it is essential to select a suitable subset of shared models based on the attributes. For the *p*-th participant, the error rate of the *q*-th participant is expressed as:

$$\gamma_p^q=\frac{1}{n^p}\left[\sum\nolimits_{i=1}^{n^p} I\left(sign\left\{\tilde{H}^q\left(X_i^p\right)\neq y_i^p\right\}\right)\right] \tag{7}$$

where the $I(\cdot)$ is the indicator function and the $\tilde{H}^q(\cdot)$ is the subset of the random decision stumps trained by the

q-th participant. The error rate of the p-th participant is expressed as:

$$\gamma_p = \frac{1}{n^p}\left[\sum\nolimits_{i=1}^{n^p} I\left(sign\left\{H^p\left(X_i^p\right) \neq y_i^p\right\}\right)\right] \tag{8}$$

For every participant, we calculate the absolute value of the difference between γ_p^q and γ_p. If $\left|\gamma_p - \gamma_p^q\right|$ is less than a certain threshold ρ, we can deem that the data distribution of the q-th participant is the same as the data distribution of the p-th participant. By doing this, each participant promotes the prediction performance in the greatest way and strengthens the data distribution indirectly.

Adaptive ensemble method

After selecting the suitable subset of shared models, we resort to aggregation to achieve a better performance, where the aggregation method plays a crucial role. In the case of classification, plurality voting is the most popular aggregation methods to draw the final classification. That is, the final classification is determined by labeling whose votes is the most:

$$L(x) = \arg\max_c \sum_{i=1}^{T} h_i(x) \tag{9}$$

Although this aggregation method does promote the stability across the stumps, it may still have a low prediction accuracy since each decision stump is themselves poor [34]. Thus, we should not only take advantage of the statistics on the private data of other participants, but also take the individual situations into consideration, when determining the weight of the models selected by the method mentioned above.

To achieve those goals, when assigning the weight to each model, we should abide by following principles: (1) The weights assigned to the shared models should be positively related to the amount of the corresponding data sets. This is because that the larger amount of data set can lead to the lower error rate of the decision stump due to insufficient sampling. (2) More weights should be assigned to the local models compared with the shared models, but it should have an upper bound. This is because we assume that participates can enhance their data distribution by integrating other models but the models from other participants might have a little difference on data distribution. The purpose of setting the upper bound is to avoid the situation that the final ensemble model is the same as the local model due to the excessive weight of the local model.

Considering those principles, for p-participant, let $\partial_{p(p=q)}^{(q)}$ denote the weight assigned to the local model and $\partial_{p(p\neq q)}^{(q)}$ denote the weight assigned to the shared model. When we determine the weight of the corresponding models, first to calculate the proportion of data sets owned by all participants

$$\eta_p = n^p / \sum\nolimits_{z\in Z} n^z, p\in Z^p(X).$$

Then the weight is updated as following:

$$\partial_q^{(p)} = \begin{cases} \eta_q & \mathrm{p\neq q} \\ \dfrac{\eta_q}{\eta_{\max}^2}\cdot\left\lceil\dfrac{\eta_{\max}^2}{\eta_{\min}}\right\rceil & \mathrm{p = q} \end{cases} \tag{10}$$

where the $\lceil\cdot\rceil$ is the ceiling function, $\eta_{\max}$ is the maximum proportion over all participants and $\eta_{\min}$ is the minimum proportion.

When updating the weight of models, for the p-th participant, if $p \neq q$ then $\partial_q^{(p)} = \eta_q$ which means that the weight assigned to the shared models of other participants is positively related to the amount of corresponding data sets; when $p = q$ then the weight of the p-th participant is $\partial_q^{(p)} = \left(\eta_q/\eta_{\max}^2\right)\left\lceil\left(\eta_{\max}^2/\eta_{\min}\right)\right\rceil$. The value of η_p belongs to $[\eta_{\min}, \eta_{\max}]$. So when $\eta_q = \eta_{\min}$, $\partial_q^{(p)}$ gets its minimum value. And

$$\left\lceil\left(\eta_{\max}^2/\eta_{\min}\right)\right\rceil/\eta_{\max}^2 \geq 1 \Rightarrow \partial_q^{(p)} \geq \eta_q.$$

We observe that the maximum value for $\partial_{p(p\neq q)}^{(q)}$ is obtained when

$$\begin{cases} \eta_q = \eta_{\max} \\ \underset{\max}{\partial_q^{(p)}} = \left(1/\eta_{\max}\right)\left\lceil\left(\eta_{\max}^2/\eta_{\min}\right)\right\rceil \end{cases}$$

Based on the above analysis, the proposed renewal rules abide by the principles mentioned above.

Theoretical analysis of proposed scheme

Privacy analysis

Often, a sophisticated differentially private scheme requires multiple uses of algorithms that satisfy differential privacy. In our proposal, for a specific participant, the total privacy budget is P and T denotes the number of iterations throughout the collaborative building process. Thus, the privacy budget that each single random decision stump can get is $\varepsilon' = P/T$. In addition, during the building process, since a series of random decision stumps of each participant are trained on the same data set, the total privacy budget consumed by a specific participant is the sum of the privacy budget used up in each iteration according to the sequential composition property of differential privacy. For the proposed distributed data mining scheme, it satisfies the ε-differential privacy on condition that the constructing process of each participant satisfies the ε-differential privacy according to the parallel composition property of differential privacy. Therefore, how to prove that the proposed *DiffPRS* algorithm satisfies the $\varepsilon^{'}$-differential privacy is the key point.

Lemma 1

DiffPRS algorithm satisfies $\varepsilon^{'}$-differential privacy.

Proof

Let R represent the *DiffPRS* algorithm, S denotes a decision stump constructed by the random decision stump method and $\lambda(S)$ denotes the decision stump S adding noise to the leaf nodes according to the *DiffPRS* algorithm. D and $D^{'}$ are two adjacent data sets that differ by at most one piece of data. Supposing that the V_1 and V_2 are two leaf vectors based on the data sets D and $D^{'}$, we can know that the V_1 and V_2 have at most one different element. Thus, the global sensitivity of leaf vector is 1.

If the eq. 3 holds for any decision stump S,

$$\frac{P(R(D)=S)}{P\left(R\left(D^{'}\right)=S\right)} \leq e^{\varepsilon^{'}} \tag{11}$$

then the *DiffPRS* algorithm satisfies the $\varepsilon^{'}$-differential privacy.

When publishing the models and the corresponding parameters, it can be divided into two parts: the structure of the decision stump and the leaf vector. We can conclude from the previous discussion that the structure of the random decision stump introduces randomness into the construction process and does not rely on the database queries, so it will not cause privacy disclosure. For leaf vector, its global sensitivity is 1, so the eq. 6 holds after adding the noise that satisfies the $Lap(1/\varepsilon)$ distribution.

$$\frac{P(\lambda(R(D))=V)}{P\left(\lambda\left(R\left(D^{'}\right)\right)=V\right)} \leq e^{\varepsilon^{'}} \tag{12}$$

In a word, the *DiffPRS* algorithm satisfies $\varepsilon^{'}$-differential privacy.

Complexity analysis

Based on the AdaBoost framework, this paper proposed a differentially private distributed data mining scheme. Its computational complexity depends on the AdaBoost algorithm. When the weak learner is the decision stump, the overall cost of AdaBoost in T iterations is.

$\Theta(K(T+\log n^p)+MT+Mn^p)$, where n^p is the number of samples of the p-th participant and K represents the number of attributes in the training data set. The error rate of model can be computed with $\Theta(n^p)$. In the ensemble procedure, the computation is consumed by testing in the selecting step so the computational cost is $\Theta(Mn^p)$, where M denotes the number of participants. In the worst case that all M models are chosen in the selecting step, the computational cost of determining the weight is $\Theta(M)$.Therefore, the total computational cost is $\Theta(K(T+\log n^p)+MT(1+n^p))$ for the p-th participant. In the proposed scheme, the training procedure and the ensemble procedure are performed by each participant independently. In addition, the ensemble procedure is started after completing its own local model for participants. Hence, the computational complexity of the proposed scheme depends on the participant with the largest number of samples, in other words, the total computational complexity of the proposed scheme is $\Theta(K(T+\log n^{\max})+MT+Mn^{\max})$.

Performance evaluation

In this section, we conduct a series of experiments to measure the prediction accuracy of our distributed data mining scheme with two real-world data sets for classification. We implement our scheme on a machine with Inter Core i5-8265 U CPU 1.6GHz and 8GB RAM running Windows 10. The proposed scheme is developed in Python 3.7.

Experiments setting

The experiments are conducted on two real-world datasets. The first one is Adult for UCI Machine Learning Repository [35], which contains 48,442 census records from the 1994 Census database. After removing the records with missing values, there are 45,222 records left. Each record has 14 attributes. The two-class classification task is to predict whether an individual's income exceeds 50 K or not. The other dataset is General Social Survey (GSS). There are 11 personal information related to the happiness, along with 51,020 records [36]. The final classification task is to infer the response to the question "Did you watch X-rated movies in the last year?". In addition, in order to simulate the scenario of multiple participants, we horizontally divide Adult and GSS data sets into 10 sub-datasets with different sizes in a random way.

In order to evaluate the effectiveness of the proposed scheme, we compare the prediction accuracy of the privacy-preserving methods with their counterparts under different conditions. To be specific, we compare the performance of our distributed data mining scheme under various privacy budgets from 0.1 to 4. Also, we test with different amount of iterations to find out the suitable value of T. Furthermore, we compare our proposed scheme with the scheme in [16], denoted as InPriv. Without loss of generality, we set the proposed scheme without the privacy protection as the base-line. In our scheme, we set the threshold value $\rho=2$ for the fact that the error rate of AdaBoost is rather small, but the situation of random decision stump is otherwise. In addition, all the results are the average of 5 runs.

Experiment result and analysis

Figure 3 and Fig. 4 show the prediction accuracy of the classification model established by the proposed scheme under different privacy budget ε and number of iterations T on the data sets Adult and GSS. From Fig. 3, we can see that the classification accuracy shows an increasing trend with the privacy budget and the number of iterations. Once the number of iterations is determined, the prediction accuracy gradually increases with the privacy budget. This is because that the amount of noise that needs to be added to the leaf nodes to protect the privacy reduces, so that the availability of data increases and the shared model can more accurately describe the data distribution of the corresponding participants. Usually, increasing the value of T to have more iterations contributes to more precise models. However, when the privacy budget is rather small, the more iterations may have negative effect on the classification performance. The larger the number of iterations is, the less privacy budget the random decision stump can get. The final ensemble models still have a low performance since each component is themselves poor for privacy protection. According to the Fig. 4, the performance of the proposed scheme is in line with the expectations on the GSS dataset.

We also compare our scheme with *Inpriv* that is based on the Gradient Boost Descent Tree (GBDT) under the various privacy budgets: 0.1, 0.3, 0.5, 1.0, 2.0 and 4.0. In addition, we set T = 30 for Adult dataset and T = 20 for GSS dataset based on the above observation. The results are shown on Fig. 5 and Fig. 6.

Through comparison, we can find that sacrificing a certain level of classification accuracy is required to protect the privacy over the distributed data. Overall, our proposed scheme performs better than *Inpriv* especially when the privacy budget is small. The reasons are two-folds: (1) the proposed differentially private decision stump only needs a small amount of noise to meet the requirements of differential privacy. (2) In the building process, the accuracy of the basic learner in each iteration is improved by integrating the shared models adaptively, reducing the number of iterations, and avoiding the extra noise caused by multiple iterations. But in some situations, the performance of *InPriv* is a little better than that our proposed scheme. The reason is that the GBDT classifier can achieve better performance than that of AdaBoost model in some data set. And when privacy budget is larger, the privacy protection is no longer an important factor limiting model accuracy.

Conclusion and future work

In this paper, we proposed an efficient and privacy-preserving data mining scheme for distributed collaborative data mining in the edge-cloud environment. Focusing on the adaptive boosting, we analyzed the boosting process, and tailor the elaborate noise to realize differential privacy for all participants. During the boosting process, the random decision stump is chosen as the

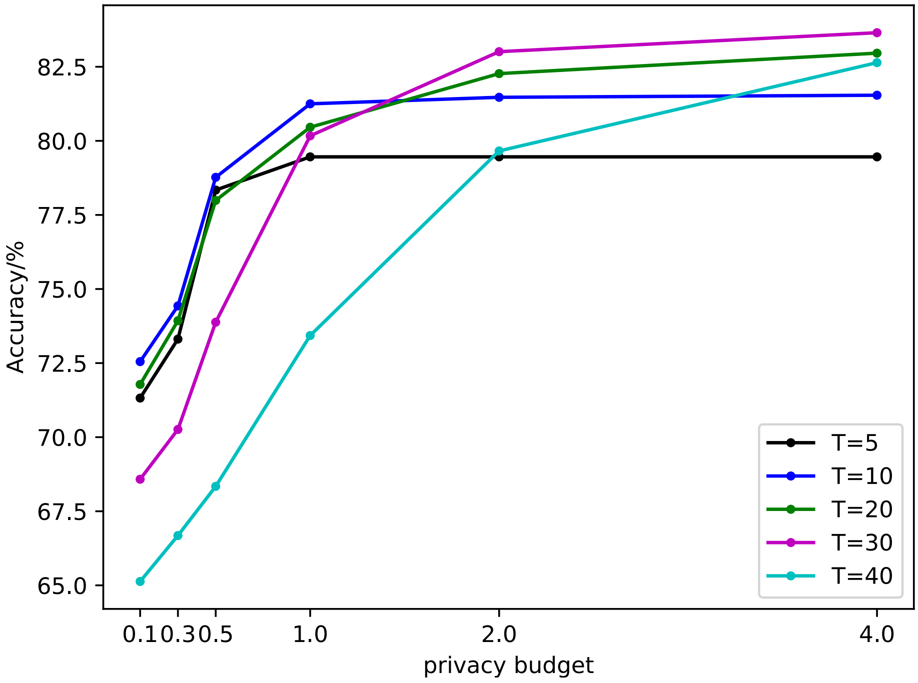


Fig. 3 The results of different value of T and ε on Adult dataset

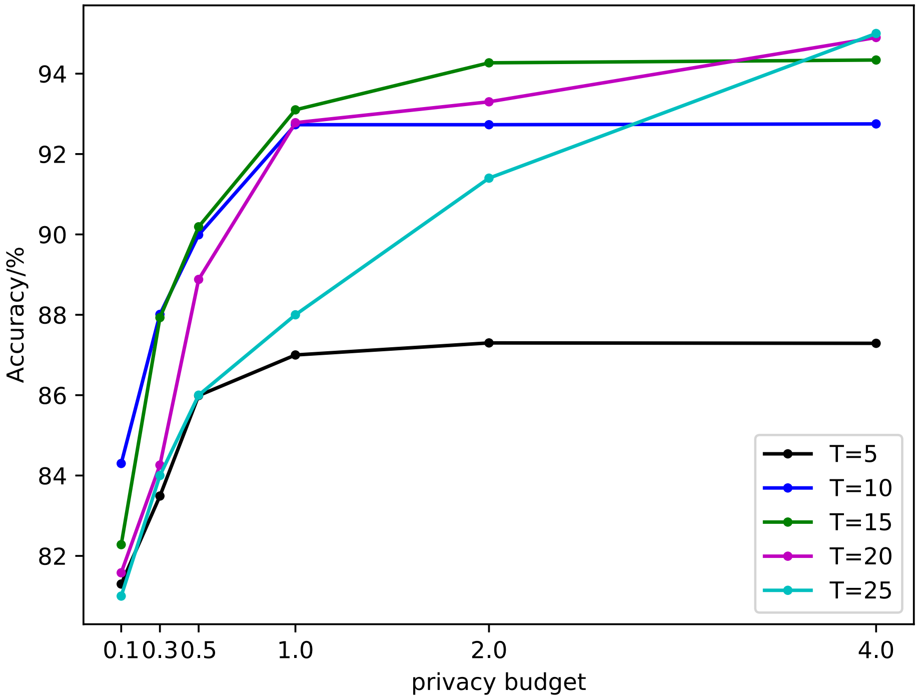


Fig. 4 The results of different value of T and ε on GSS dataset

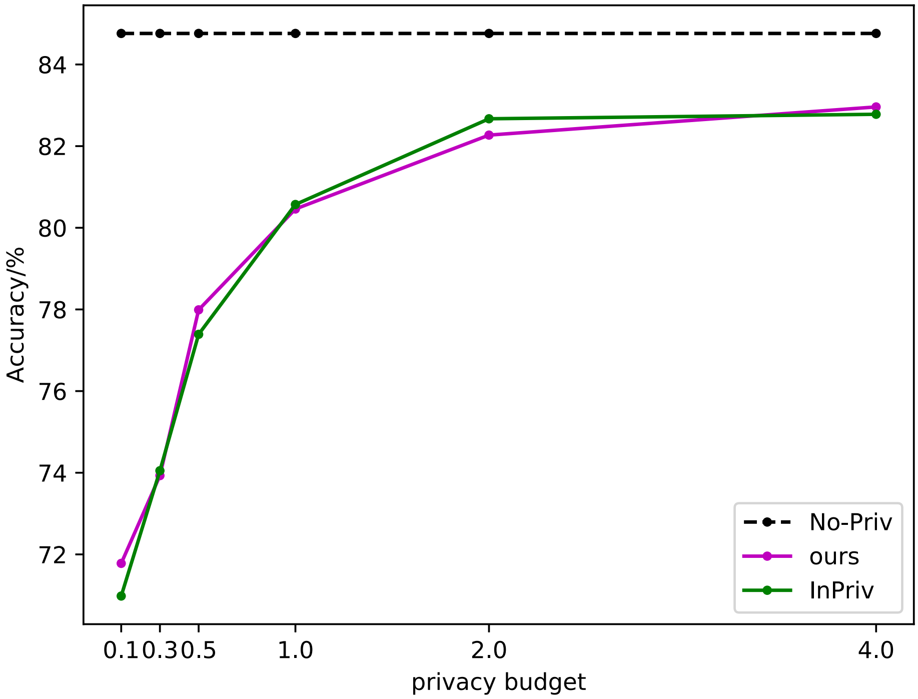


Fig. 5 The comparison of model performance on Adult dataset

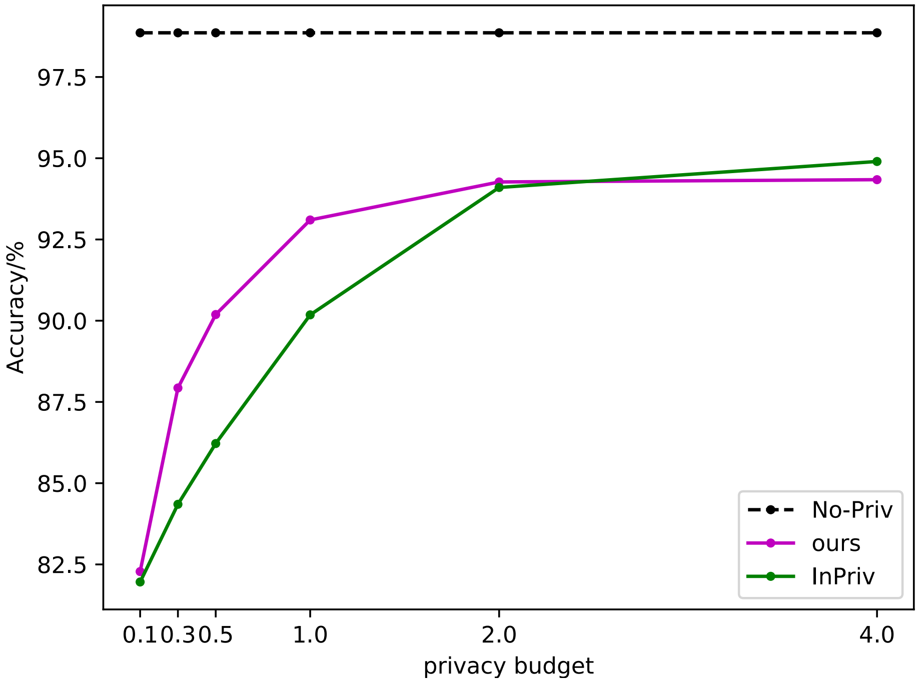


Fig. 6 The comparison of model performance on GSS dataset

basic learner for the reason that the edge computing suffers from the limited computational resources. Then, we proposed an adaptive ensemble method, which can enhance the data distribution of participants and avoid the negative impact of unwanted models. Theoretical analysis and experimental results verify that our scheme can efficiently construct mining model with high performance while providing rigorous privacy guarantee.

This work also poses several future challenges. It is worthwhile to find a more reasonable way to measure the difference of data distributions among multiple data owners without compromising the privacy of data owners. One promising solution is to relax the privacy requirements and allow the participants estimate the data distributions more precisely under privacy constraints. Another interesting direction is to make the better tradeoff between the accuracy and the communication costs.

Acknowledgments
We sincerely thank the Reviewers and the Editor for their valuable suggestions.

Authors' contributions
Conceptualization, Zhitao Guan.; investigation, Xianwen Sun; methodology, Xianwen Sun, Ruzhi Xu, Longfei Wu and Zhitao Guan; writing—original draft, Xianwen Sun, and Ruzhi Xu; writing—review & editing, Lonfei Wu and Zhitao Guan. The authors read approved the final manuscript.

Funding
The work is partially supported by the National Natural Science Foundation of China under Grant 61972148, Beijing Natural Science Foundation under grant 4182060, and the Fundamental Research Funds for the Central Universities under grant 2019MS020.

Availability of data and materials
The data used to support the findings of this study is available from the corresponding author upon request.

Competing interests
The authors declare that there is no conflict of interests regarding the publication of this paper.

Author details
[1]School of Control and Computer Engineering, North China Electric Power University, Beijing, China. [2]Department of Mathematics and Computer Science, Fayetteville State University, Fayetteville, NC, USA.

Received: 30 August 2020 Accepted: 29 December 2020
Published online: 19 January 2021

References

1. Santos G, Takako Endo P, da Silva Lisboa MF, da Silva LG, Sadok D, Kelner J, Lynn T (2018) Analyzing the availability and performance of an e-health system integrated with edge, fog and cloud infrastructures. J Cloud Comp 7(1):16
2. Chen M, Li W, Hao Y, Qian Y, Humar I (2018) Edge cognitive computing based smart healthcare system. Futur Gener Comput Syst 86:403–411
3. Liu H, Kou H, Yan C, Qi L (2019) Link prediction in paper citation network to construct paper correlation graph. EURASIP J Wirel Commun Netw 2019(1): 1–12
4. Chen C, Liu Z, Wan S, Luan J, Pei Q (2020) Traffic flow prediction based on deep learning in internet of vehicles. In: IEEE transactions on intelligent transportation systems, pp 1–14
5. Wan S, Xu X, Wang T, Gu T (2020) An intelligent video analysis method for abnormal event detection in intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems 1–9. https://doi.org/10.1109/TITS.2020.3017505
6. Centers for Disease Control and Prevention (2003) HIPAA privacy rule and public health. Guidance from CDC and the US Department of Health and Human Services. MMWR Morb Mortal Wkly Rep 52(1):1–17

7. Guan Z, Lu X, Yang W, Wu L, Wang N, Zhang Z (2021) Achieving efficient and privacy-preserving energy trading based on blockchain and ABE in smart grid. J Parallel Distribut Comput 147:34–45
8. Zhong W, Yin X, Zhang X, Li S, Dou W, Wang R, Qi L (2020) Multi-dimensional quality-driven service recommendation with privacy-preservation in Mobile edge environment. Comput Commun 157:116–123
9. Guan Z, Liu X, Wu L, Xu R, Zhang J, Li Y (2020) Cross-lingual multi-keyword rank search with semantic extension over encrypted data. Inf Sci 514:523–540
10. Yang Q, Liu Y, Chen T, Tong Y (2019) Federated machine learning: concept and applications. Transact Intell Syst Technol 10(2):1–19
11. Dwork C (2008) Differential privacy: a survey of results. International conference on theory and applications of models of computation. Springer, Switzerland, pp 1–19
12. Guan Z, Lv Z, Sun X, Wu L, Wu J, Du X, Guizani M (2020) A differentially private big data nonparametric Bayesian clustering algorithm in smart grid. IEEE Trans Netw Sci Eng 7(4):2631–2641
13. Friedman A, Schuster A (2010) Data mining with differential privacy. 16th ACM Sigkdd international conference on knowledge discovery and data mining. ACM, Washington, DC, pp 493–502
14. Phan NH, Wu X, Hu H, Dou D (2017) Adaptive Laplace mechanism: differential privacy preservation in deep learning. International Conference on Data Mining. IEEE, New York, pp 385–394
15. Qi L, He Q, Chen F, Zhang X, Dou W, Ni Q (2020) Data-driven web APIs recommendation for building web applications. In: IEEE Transactions on Big Data, p 1. https://doi.org/10.1109/TBDATA.2020.2975587
16. Zhao L, Ni L, Hu S, Chen Y, Zhou P, Xiao F (2018) Inprivate digging: enabling tree-based distributed data mining with differential privacy. In: IEEE International Conference on Computer Communications. IEEE, pp 2087–2095
17. Aggarwal CC, Yu PS (2008) A general survey of privacy-preserving data mining models and algorithms. J Vasc Surg 8(1):64–70
18. Zhao P, Zhang G, Wan S, Liu G, Umer T (2019) A survey of local differential privacy for securing internet of vehicles. J Supercomput 76:1–22
19. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106
20. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemom Intell Lab Syst 2(1–3):37–52
21. Basheer IA, Hajmeer M (2000) Artificial neural networks: fundamentals, computing, design, and application. J Microbiol Methods 43(1):3–31
22. Blum A, Dwork C, Mcsherry F, Nissim K (2005) Practical Privacy: the SuLQ Framework. In: 24th ACM Sigmod-Sigact-Sigart Symposium on Principles of Database Systems. ACM, pp 128–138
23. Rana S, Gupta SK, Venkatesh S (2015) Differentially private random forest with high utility. In: IEEE International Conference on Data Mining. IEEE, pp 955–960
24. Fletcher S, Islam MZ (2015) A differentially private random decision forest using reliable signal-to-noise ratios. In: Australasian Joint Conference on Artificial Intelligence, pp 192–203
25. Huang Y, Evans D, Katz J, Malka L (2011) Faster secure two-party computation using garbled circuits. In: Usenix Conference on Security. USENIX Association, pp 331–335
26. Mohammed N, Fung BCM, Debbabi M (2011) Anonymity meets game theory: secure data integration with malicious participants. VLDB J 20(4): 567–588
27. Mohammed N, Jiang X, Chen R, Fung B, Ohno-Machado L (2013) Privacy-preserving heterogeneous health data sharing. J Am Med Inform Assoc 20(3):462–469
28. Goryczka S, Xiong L, Fung BCM (2013) *m*-privacy for collaborative data publishing. IEEE Trans Knowl Data Eng 26(10):2520–2533
29. Emekçi F, Sahin OD, Agrawal D, Abbadi E (2007) Privacy preserving decision tree learning over multiple parties. Data Knowl Eng 63(2):348–361
30. Gambs S, Kégl B, Aïmeur E (2007) Privacy-preserving boosting. Data Mining Knowl Discov 14(1):131–170
31. Freund Y, Schapire RE (1997) A decision-theoretic generalization of online learning and an application to boosting. In: European Conference on Computational Learning Theory. Springer, pp 23–37
32. Guan Z, Sun X, Shi L, Wu L, Du X (2020) A differentially private greedy decision forest classification algorithm with high utility. Comput Sec 96: 101930
33. Li J, Cai T, Deng K, Wang X, Sellis T, Xia F (2020) Community-diversified influence maximization in social networks. Inf Syst 92:1–12
34. Sagi O, Rokach L (2018) Ensemble learning: a survey. Wiley Interdisc Rev 8(4):e1249
35. Amarnath B, Balamurugan S, Alias A (2016) Review on feature selection techniques and its impact for effective data classification using UCI machine learning repository dataset. J Eng Sci Technol 1(11):1639–1646
36. J. Prince. Social science research on pornography, http://byuresearch.org/ssrp

Publisher's Note

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