Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(1): 953–971.
DOI: 10.3934/mbe.2022044
Received: 30 September 2021
Accepted: 03 November 2021
Published: 22 November 2021

Research article

Federated personalized random forest for human activity recognition

Songfeng Liu[1,2], **Jinyan Wang**[1,2,*] **and Wenliang Zhang**[2]

[1] Guangxi Key Lab of Multi-source Information Mining and Security, Guangxi Normal University, Guilin, China

[2] College of Computer Science and Engineering, Guangxi Normal University, Guilin, China

* **Correspondence:** Email: wangjy612@gxnu.edu.cn.

Abstract: User data usually exists in the organization or own local equipment in the form of data island. It is difficult to collect these data to train better machine learning models because of the General Data Protection Regulation (GDPR) and other laws. The emergence of federated learning enables users to jointly train machine learning models without exposing the original data. Due to the fast training speed and high accuracy of random forest, it has been applied to federated learning among several data institutions. However, for human activity recognition task scenarios, the unified model cannot provide users with personalized services. In this paper, we propose a privacy-protected federated personalized random forest framework, which considers to solve the personalized application of federated random forest in the activity recognition task. According to the characteristics of the activity recognition data, the locality sensitive hashing is used to calculate the similarity of users. Users only train with similar users instead of all users and the model is incrementally selected using the characteristics of ensemble learning, so as to train the model in a personalized way. At the same time, user privacy is protected through differential privacy during the training stage. We conduct experiments on commonly used human activity recognition datasets to analyze the effectiveness of our model.

Keywords: federated learning; random forest; differential privacy; personalization

1. Introduction

Human activity recognition (HAR) is a classification task in machine learning [1]. Its goal is to classify some activities performed by a user within a certain period of time. People's activities can include different types, such as walking, running, going upstairs, going downstairs, sitting, etc. The human activity recognition program can be applied to the fields of medical care, fitness and so on [2, 3]. The machine learning model trains on data collected by smart devices with accelerometer and gyroscope sensors to achieve the purpose of tracking the user's health [4, 5].

In real life, data is scattered among individual users or various institutions. The machine learning model requires a large amount of data to train better model. However, the method of concentrating user data to a central server will cause the privacy of data to be leaked. Moreover, the recent regulation, like GDPR is designed to protect the privacy of user [6]. Users or institutions cannot share their data to the data center, which makes it difficult to use these valuable data to train powerful machine learning models.

Federated learning was proposed by Google, which can train a shared global model collaboratively while keeping user data scattered [7, 8]. A typical method of implementing federated learning is federated averaging (FedAvg) [7], which updates parameters on the server by averaging the local model parameters uploaded from each client. After multiple iterations, a shared global neural network model is generated in the server and distributed to each client. These researches focused on federated learning based on neural networks. Due to the excellent characteristics of other machine learning, many researches have begun to pay more attention to training other machine learning models in federated settings [9, 10]. Considering the faster training speed of the tree model and the high accuracy of the tree-based ensemble model, some studies have applied the tree-based ensemble model to the setting of federated learning, such as federated gradient boosting decision tree [11], federated extreme tree [12].

In the training process of federated learning, the original data is stored locally on the client and is not exposed to the server or any other users for alleviating the privacy leakage of user data. However, some studies have shown that the parameters or intermediate information in the model training process may still leak user privacy [13]. To ensure the privacy and security of user data in federated learning, some works mainly used differential privacy or homomorphic encryption to protect the intermediate parameters in the model training process [14, 15]. Recently, Mo et al. [16] utilized the widely existing trusted execution environment (TEE) in mobile devices to hide model updates from attackers through local TEE training on the client and TEE security aggregation on the server.

Federated learning can effectively alleviate the problem of data islanding and has been widely used in various practical tasks, especially in the field of medical and health care. A system based on blockchain technology elements and threaded federated learning was proposed [17]. An agent with a consortium mechanism was constructed for the classification results of many machine learning solutions. This research provides the new multi-agent model that can be implemented as a real-time medical data processing system. Sozinov et al. [18] used federated learning to train a classifier to solve the challenge of insufficient data for a single user in human activity recognition tasks. However, an important problem in federated learning is that the final global model lacks personalization. Most methods are based on all users to generate a common model. Due to the heterogeneity of user data in actual federated learning, generating a unified model may not be the best solution for all users. We can see that in the human activity recognition task, different users have different physical characteristics and daily activities. Therefore, a unified model cannot meet the needs of all users and cannot achieve personalized medicine. In this case, each user wants to obtain a personalized model instead of a global shared model after participating in federated learning.

Some existing personalization methods are designed for the training of neural networks in federated learning, but there is no relevant research on the personalized methods of tree-based federated models. The lightweight tree-based model is more suitable for training and deployment on wearable devices with limited computing. So we are mainly concerned about how to apply the federated random forest to the task of activity recognition, and generate a personalized federated model for each user. Inspired

by previous work, we propose a new privacy-protected federated personalized random forest model (PP-FPRF) to accurately and securely support real-world activity recognition applications. We have three main contributions:

Personalization. The federated personalized random forest is considered from the two levels. First, from the data point of view, according to the data characteristics of the users in the activity recognition task, the locality sensitive hashing (LSH) [19] is used to measure the data similarity between users, and the user and other users with similar data characteristics are trained in cooperation. Second, from the model of view, the user selects the base classifier by the ensemble learning incremental selection to achieve the purpose of a personalized model.

Privacy protection. In the process of users cooperative training, to protect users' privacy, each user participating in the training communicates the optimal split of candidate attributes in non-leaf nodes and the classes counts in leaf nodes based on exponential mechanism and Laplace mechanism, respectively.

Feasibility. We evaluated the proposed framework based on real human activities recognition datasets and conducted extensive experiments. The experimental results show the effectiveness of our model.

This rest of this paper is organized as follows. Section 2 overviews the related work of our research. The preliminaries on locality sensitive hashing and differential privacy are introduced in Section 3. In Section 4, we describe our approach in detail. The experimental evaluations and results are discussed in Section 5. Finally, Section 6 summarizes the paper.

2. Related work

Since our work is related to tree-based federated learning and personalized federated learning, we discuss some existing methods. In addition, we also analyze the differences between our work and existing methods.

2.1. Personalized federated learning

Some studies have paid attention to the heterogeneity of data in federated learning and have proposed some personalized solutions. Wang et al. [20] fine-tuned the federated model through the local data in each client to realize the personalization of the user model. After training a unified federated model, in the process of personalized learning, all convolutional and pooling layers in the network are frozen, and only the parameters of the fully connected layer are updated by using stochastic gradient descent (SGD). Fedhealth aggregated model parameters through federated learning and then applied in personalized medicine by building a personalized model for each user through transfer learning [21]. In multi-task learning [22], multiple related tasks are solved simultaneously, allowing the model to take advantage of the commonalities and differences of different tasks through cooperative learning. Smith et al. [23] developed the framework for federated multi-task learning (MOCHA) algorithm to solve personalized problem. Yu et al. [24] extended the personalization method and proposed three different schemes to personalize the federated model: fine-tuning, multi-task learning and knowledge distillation.

But these works are aimed at the personalized training of neural networks in federated learning. The random forest model is a lightweight model that is more suitable for computing constrained wearable

devices [21], so we focus on the application of federated random forest in activity recognition tasks and design the corresponding personalized method to improve the model effect.

2.2. Tree-based federated learning

The tree-based ensemble model has been applied to horizontal and vertical federated learning. In vertical federated learning, the client has the same samples but different feature spaces. In this direction, Liu et al. [9] proposed a federated forest framework based on classification and regression trees (CART) and bagging. This framework has a certain degree of privacy protection, and the communication burden is not high when forecasting. In horizontal federated learning, data samples with the same characteristics are distributed in multiple parties. Li et al. [11] studied an actual federated environment with loose privacy constraints. Medical institutions jointly trained the gradient boosting decision tree (GDBT) model and used gradient weighting to improve the performance of the model. Liu et al. [12] extended extra-trees to provide a concise algorithm to limit the computational complexity to a minimum, and greatly increase the training speed to adapt to horizontal federated scenarios, while using differential privacy to protect intermediate data.

These methods based on tree are suitable for collaborative training between several medical, financial and other data institutions. They consider generating a unified model for all users without considering the personalization. The necessary motivation for this collaboration is that federated learning should generate a better learning model than a model generated from the local data of the users alone. However, for human activity recognition tasks, the number of users is larger and the model needs to be personalized, current methods based on tree are not effective enough. So we investigate the the tree-based personalized federated learning for activity recognition tasks.

3. Preliminaries

In this section, we give some basic concepts and review the knowledge about locality sensitive hashing and differential privacy.

3.1. Locality-sensitive hashing

LSH was originally proposed by Gionis et al. [19]. It is a fast nearest neighbor search algorithm for massive high-dimensional data. The main idea of LSH is to select a hash function so that the hash values of two neighboring points are equal with a high probability. On the contrary, the hash values of two non-neighboring points are not equal with a high probability. For a domain S of the points set, an LSH family is defined as:

Definition 1. [19]. A family H of functions from S to U, $H = \{h : S \to U\}$ is called (r_1, r_2, p_1, p_2)-sensitive if for any $v, q \in S$, $d(v, q)$ is the distance between two vectors:

$$\begin{cases} if \quad d(v,q) < r_1, \quad then \quad Pr[h(q) = h(v)] \geq p_1 \\ if \quad d(v,q) > r_2, \quad then \quad Pr[h(q) = h(v)] \leq p_2 \end{cases} \tag{3.1}$$

The characteristic of LSH is that there will be multiple input data corresponding to the same hash value output. Therefore, LSH has been used to protect user privacy in applications such as keyword search [25] and recommendation systems [26]. A widely used p-stable LSH family is proposed by

Datar et al. [27]. The hash functions $F_{a,b}$ are expressed as

$$F_{a,b}(v) = \lfloor \frac{a \cdot v + b}{r} \rfloor \tag{3.2}$$

where v is a d-dimensional vector representing a sample; a is a d-dimensional vector with entries selected independently from p-stable distribution [27]; b is a real number randomly selected from the range $[0, r]$; r is a positive real number representing the size of the window.

3.2. Differential privacy

The differential privacy model proposed by Dwork et al. [14], disturbs the calculation results to ensure that deleting or adding a single item in the database will not affect the output of the database access mechanism. This shows that it is difficult for opponents to judge whether a person is in the database through indistinguishable differences. In this way, personal sensitive information is protected.

Definition 2. (ϵ-Differential privacy [14]). A randomization mechanism M provides ϵ-differential privacy if for databases D_1 and D_2 differing on one element, R is the output range:

$$Pr[M(D_1) \in R] \leq e^{\epsilon} Pr[M(D_2) \in R] \tag{3.3}$$

The privacy budget ϵ controls the privacy protection level of differential privacy, and a smaller privacy budget represents stronger privacy protection.

Definition 3. (Sensitivity [28]). Given a function $f : D \to R^d$ over an arbitrary domain D, the global sensitivity of f is defined as

$$\Delta f = \max_{D_1, D_2} \|f(D_1) - f(D_2)\|_1 \tag{3.4}$$

where D_1 and D_2 differ in one record.

To obtain ϵ-differential privacy, the noise is calibrated according to the sensitivity of the function. The sensitivity of a real-valued function represents the maximum possible change in its value due to the addition or deletion of a single record.

Theorem 1. (Laplace Mechanism [28]). Given a function $f : D \to R^d$ over an arbitrary domain D, the computation:

$$M(D) = f(D) + Lap\left(\frac{\Delta f}{\epsilon}\right) \tag{3.5}$$

provides ϵ-differential privacy.

For example, the count function f over a set S, $f(S) = |S|$, the sensitivity of the function is 1. Therefore, a noisy count that returns $M(S) = |S| + Lap(\frac{1}{\epsilon})$.

Theorem 2. (Exponential Mechanism [29]). Suppose the input of random calculation M is the dataset D, the output is $r \in Range(M)$, $q(D, r)$ is the quality function, and Δq is the sensitivity of the quality function. If the algorithm selects and outputs r from the range with a probability proportional to $\exp\left(\frac{\epsilon q(d,r)}{2\Delta(q)}\right)$, then algorithm M provides ϵ-differential privacy protection.

The following is an example of the exponential mechanism [30]. If a competition is to be held, the available items are from the collection $\{swimming, running, basketball\}$. Participants will vote to determine an item and ensure that the entire decision process meets the ϵ-differential privacy protection. Taking the number of votes as the availability function, so $\Delta q = 1$. Then according to the exponential mechanism, under a given privacy protection budget ϵ, the output probabilities of various projects can

be calculated.

Theorem 3. (Sequential Composition [31]). Given algorithm $M_1, M_2, \ldots, M_n$, their privacy budgets are set to $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$, respectively. Then for the same dataset D, the combined algorithm composed of these algorithms $M(M_1(D), M_2(D), \ldots, M_n(D))$ provides $(\sum_{i=1}^{n} \epsilon_i)$ −differential privacy protection.

This property shows that for a differential privacy sequential composition algorithm, its privacy protection level is the sum of all privacy budgets.

Theorem 4. (Parallel Composition [31]). Given algorithm $M_1, M_2, \ldots, M_n$, their privacy budgets are set to $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$, respectively. Then for disjoint data sets $D_1, D_2, \ldots, D_n$, the combined algorithm composed of these algorithms $M(M_1(D), M_2(D), \ldots, M_n(D))$ provides $(max\ \epsilon_i)$-differential privacy protection.

In a differential privacy protection algorithm sequence, if all the datasets processed by these algorithms do not intersect each other, then the privacy protection level provided by the algorithm sequence depends on the algorithm with the worst protection level, that is, the algorithm with the largest privacy budget.

4. Methods

In this section, we introduce the federated personalized random forest framework, which allows random forest models to be trained for every user in a horizontal federated setting. A user only trains with some similar users instead of all users. Our motivation is that in the task of activity recognition, people have different physical characteristics and activity patterns, and their data are very different. The models trained by similar users are more suitable for their own characteristics.

Table 1. Major notations.

Notation	Description
M_l	The trained local model
M_f	The federated model (global model)
M_p	The personalized federated model (our goal)
U	A user in federated learning
D	The data in a client
F	The attributes set of the data
N	The number of data in a certain client
S	The similar clients set of a user

Table 1 summarizes the important symbols that will be used frequently in this paper. In the activity recognition task, each user U_i has its own local dataset D_i and can train a local model M_l^i based on D_i. The user obtains a unified model M_f through traditional federated learning. But the unified model M_f does not consider the differences among users. Therefore, it does not achieve good performance on some users, and is even worse than some users' local models. Our work is to let each user U_i gets a personalized random forest model M_p^i through personalized federated learning.

Figure 1 shows the structure of our approach. The user U_i and its similar users are taken as an example to describe the personalized training phase. Step 1 is the stage of finding similar users. Each user first uses the global hash tables to find users who are similar to them. In step 2, the user trains a random

decision tree with similar users. Then in step 3, the user individually makes a personalized selection of the newly generated decision tree. An overview of PP-FPRF algorithm is shown in Algorithm 1. The inputs are the LSH functions $\{F_k\}_{k=1,2...L}$, all users data $\{D_i\}_{i=1,2...I}$, the number of trees T and differential privacy budget B. User participates in training and gets his own personalized model M_p^i. In line 1, we call function PREPROCESS [11] for each user U_i to obtain his similar users S_i. Each user U_i can initiate a training session and coordinate his similar users S_i to train a tree. In this training, U_i is regarded as the master node, and his similar users participating in the training are regarded as cooperative users. A user in federated learning can initiate a training session as a master or participate in a training as a collaborator. In line 3, a master obtained a new tree by Algorithm 2 (TREEBUILD_M), at the same time, a collaborator get a new tree by Algorithm 3 (TREEBUILD_C). In line 9, the user participating in the training uses INCRE_SELECT of the decision tree based on his local data, to determine whether add the new tree to his own personalization model M_p^i.

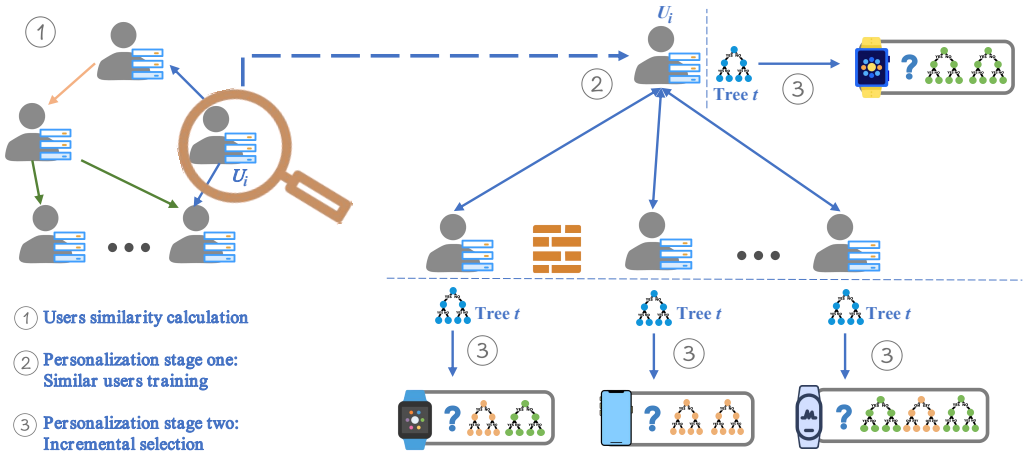


Figure 1. Structure of the PP-FPRF approach.

Algorithm 1 *The learning procedure of PP-FPRF*

Input: The user U_i, LSH functions $\{F_k\}_{k=1,2...L}$, all users data $\{D_i\}_{i=1,2...I}$, hyperparameter k, the number of training sessions T, differential privacy budget $\epsilon = \frac{B}{T}$

Output: The personalized federated model M_p^i of U_i

1: S_i = PREPROCESS($\{F_k\}_{k=1,2...L}$, $\{D_i\}_{i=1,2...I}$, k)
2: **for** $j \leftarrow 1$ to T **do**
3: **if** U_i is master **then**
4: t = TREEBUILD_M(D_i, S_i, ϵ);
5: **else**
6: t = TREEBUILD_C(D_i, ϵ);
7: **end if**
8: **if** INCRE_SELECT(D_i, M_p^i, t) is true **then**
9: $M_p^i \leftarrow$ add(M_p^i , t);
10: **end if**
11: **end for**

4.1. Users similarity calculation

The Preprocess method utilizes the widely used p-stable LSH function to obtain the similarity of any two samples in different users [27], without exposing the original data to other users [11]. According to the characteristics of LSH, if two samples are similar, they are more likely to be hashed to the same value. Therefore, by using multiple LSH functions, the bigger the number of identical hash values of two samples, the greater the likelihood that they are similar.

The description of function Preprocess is as follows. Given L randomly generated $p-$stable hash functions, the users first calculate the hash values corresponding to their samples, and each sample is mapped to L hash values by L hash functions. The AllReduce operation is used to build L global hash tables, and the inputs to AllReduce are the sample IDs and their hash values of all users. The reduction operation is to combine the samples IDs with the same hash value. By adopting the previously proposed bandwidth optimal and contention free approach [32], propagate the aggregated hash tables to each user. After each user gets the global hash tables, calculates the similarity with other users. For example, user U_i calculates the number of identical hash values for each sample in user U_i and each sample in user U_j. If the number of the same hash value of the two samples is bigger than a specific threshold, the two samples are considered similar.

Li et al. [11] used LSH to find similar samples to weight the gradient. The basic idea is that the instance is important if it is similar to many other instances. Different from them, we use LSH to find similar samples and continue to find similar users of users. User U_i counts how many samples in U_j are his similar samples, and calculates the proportion of similar samples to U_j. Because of the randomness of the LSH function, it is not easy for us to define similar users with specific threshold. Therefore, through comparison, each user finds the top-k users with higher similar samples ratio, as their own similar users.

4.2. Personalized training stage

Through similarity calculation, each user gets his own set of similar users. To make the model generated by the federated learning suitable for the user's local data, the user only trains the federated learning model with similar users. The generated personalized model can combine the generalization characteristics of the global model and the data matching characteristics of the local model. In this subsection, we will introduce our personalized methods considered from the data level and model level: similar users training and incremental selection methods.

4.2.1. Similar users training

Algorithms 2 and 3 describe the cooperating training process of the master node and similar users. The key steps of building a tree are as follows.

In the process of training a random tree, the master is responsible to coordinate cooperating users, and according to the total splitting information to determine the splitting attributes of a node or choose to stop splitting. We call function RandomPick provided by Liu et al. [12], allows the master to exchange information with cooperating users and return candidate attributes $F^{'}$ and splitting values $\left\{v_k, k = 1, ..., |F^{'}|\right\}$ to cooperating users. The cooperative user U_j temporarily splits the local data into left and right parts according to each value in $\left\{v_k, k = 1, ..., |F^{'}|\right\}$.We use the information gain (IG) quality function to evaluate the scores of nodes divided by different attributes and corresponding values.

Algorithm 2 *Privacy-protected federated personalized random forest - Master*

Input: Training set D_i of master U_i, the similar users set S_i, differential privacy budget ϵ
Output: A new decision tree t

function TreeBuild_M(D_i, S_i, ϵ)
  if *Stopping Condition* is true **then**
    Receive noisy classes counts from user U_j, $U_j \in S_i$;
    Send leaf labels to collaborative users S_i;
    return leaf node;
  end if
  $\left\{v_k, k = 1, ..., |F^{'}|\right\} \leftarrow$ RandomPick(F, D_i, S_i);
  for every attribute $f_k \in F^{'}$ **do**
    Calculate the information gain $q = IG(f_k, D_i)$;
  end for
  $f_i^{'} \leftarrow$ Select the local optimal splitting attribute based on Exponential Mechanism in Eq. (4.1);
  Receive $f_j^{'}$ from the U_j , $j = 1, ..., |S_i|$;
  $f^{'} \leftarrow$ Weighted voting on the local optimal attributes, send the global best splitting $f^{'}$ to S_i;
  Lsubtree $\leftarrow$ TreeBuild_M(D_i^l, S_i, ϵ);
  Rsubtree $\leftarrow$ TreeBuild_M(D_i^r, S_i, ϵ);
  return tree node;
end function

Each user finds the attribute with the largest information gain as the local best attribute in the candidate set $F^{'}$. To protect user privacy, we use exponential mechanism to select the local best attribute f_k^j, the details are described in Section 4.3. The master also has its own local optimal attribute f_k^i by perturbing, and receives the local optimal splitting attribute sent from the cooperating user. Then, he determines the global optimal splitting attribute $f^{'}$ by means of weighted voting. The larger the sample size N of the user, the greater the weight in the federated learning. The attribute $f^{'}$ with the largest weight is selected as the best splitting method for the current node. Then the master sends the determined node splitting method to the partners.

In RandomPick [12], the master U_i first randomly selects a subset $F^{'} \subset F$ as candidate splitting attributes. Then sends $F^{'}$ to cooperative user $U_j \in S_i$ who participated in the training. For attribute $f_k \in F^{'}$, each cooperative user U_j randomly selects a value v_k^j within the range of minimum and maximum values of attribute f_k, and sends v_k^j to the master. The master U_i combines the local random splitting value v_k^i and the received splitting values $\left\{v_k^j, j = 1, ..., |S_i|\right\}$ from other users, and takes the minimum and maximum value of v_k^i and $\left\{v_k^j, j = 1, ..., |S_i|\right\}$. Then, randomly selects a value v_k from the range of minimum and maximum value, as the splitting value of the candidate attribute f_k. The master calculates corresponding splitting values of candidate attribute in $F^{'}$, and sends these values $\left\{v_k, k = 1, ..., |F^{'}|\right\}$ to cooperative users.

Recurse this step to split the random tree until the stopping condition is met. Before creating a new tree node, the master will check whether the stop condition is met. To prevent the generated decision tree from overfitting, the stopping condition we adopt is to limit the maximum depth of the tree and the number of remaining samples in the node [12, 33]. When the node reaches the stop condition, the

cooperating users perturb the classes counts in the node by Laplace mechanism, as is described in the Section 4.3. They send perturbing counts to the master node to calculate the global classes counts and get the class with the largest total count as the label of the leaf node. In the training stage, by aggregating the random attribute values corresponding to the candidate attributes, the best attribute partition selected by exponential mechanism, and the count of leaf labels after perturbing to complete the training of the model.

Algorithm 3 *Privacy-protected federated personalized random forest - Collaborative user*

Input: Training set D_j of user $U_j \in S_i$, master U_i, differential privacy budget ϵ
Output: A new decision tree t

function TreeBuild_C(D_j, ϵ)
  if *Stopping Condition* is true **then**
    Perturb classes counts by Laplace Mechanism in Eq. (4.2), send to master;
    Receive leaf labels from master;
    return leaf node;
  end if
  $\left\{v_k, k = 1, ..., |F^{'}|\right\} \leftarrow$ RandomPick(F, D_j);
  for every attribute $f_k \in F^{'}$ **do**
    $q = IG(f_k, D_j)$;
  end for
  $f_j^{'} \leftarrow$ Select the local optimal splitting attribute based on Exponential Mechanism in Eq. (4.1);
  Send $f_j^{'}$ to master, receive the global optimal split attribute $f^{'}$;
  Lsubtree $\leftarrow$ TreeBuild_C(D_j^l, ϵ);
  Rsubtree $\leftarrow$ TreeBuild_C(D_j^r, ϵ);
  return tree node;
end function

4.2.2. Incremental selection

This is a personalized approach that we consider from the model level. Ensemble pruning is to select a subset of the ensemble model to form a new ensemble model. Zhou et al. [34] believed that the smaller scale ensemble model after pruning performed better than the original ensemble. Most of the existing ensemble pruning methods are divided into selection or adjusting the weight of the base classifier [35]. We adopt the selected method for ensemble pruning.

Aiming at the random decision tree generated in the similarity learning stage described above, users use the Incre_Select method to adapt the model locally. A user evaluates the importance of the newly generated decision tree to the current ensemble model by testing the performance of the model on the validation set. If a new random decision tree t is added to the user's local personalization model M_p, the accuracy of the model $M_p + t$ on the user's verification set is higher than before. Then the user adds the newly generated random decision tree t to the current ensemble model. The local selection of the newly generated random decision tree just as the previous personalization study makes a local fine-tuning of the neural network obtained in the federated learning. By using incremental selection, users can not only control the number of trees in the model, and reduce the cost of storage and calculation

during prediction, but also further personalize improving the local performance of the model.

4.3. Privacy protection

In [36, 37], they concerned about how differential privacy interacts with each component of the decision tree algorithm and the conflict that arises when trying to balance the need for privacy and the accuracy of the model. Patil and Singh [38] introduced the concept of differential privacy in the classic random forest algorithm. On the basis of these studies, we analyze the privacy issues in our federated personalized random forest model, and furthermore use differential privacy to protect users privacy during the model training process.

The calculation of information gain and the counts of classes are directly based on the user's data. According to the differential privacy statement, publishing such information may be a leak of privacy, so these potential privacy leaks are exactly what the differential privacy algorithm wants to prevent [37]. Next, we elaborate user privacy protection on the two key steps in model construction: using exponential mechanism to perturb the local optimal attribute in non-leaf nodes, and adding Laplace noise perturbation to the classes counts in the leaf node.

The sensitivity of the information gain calculation function q is $\Delta(q) = \log_2 |C|$, where $|C|$ is the domain size of the class attribute C [38]. Allocate a privacy budget $\epsilon_1 = \frac{\epsilon}{d}$ for the perturbation of local optimal candidate attributes on non-leaf nodes, where ϵ is the privacy budget allocated to a tree of the user, and d is the depth of the tree (including non-leaf nodes and leaf nodes). The exponential mechanism selects a local optimal candidate attribute f_k with the following probability:

$$\frac{\exp\left(\frac{\epsilon_1}{2\Delta q} q(f_k)\right)}{\sum_{f_k \in F'} \exp\left(\frac{\epsilon_1}{2\Delta q} q(f_k)\right)} \tag{4.1}$$

When determining the leaf label, the classes counts in the user's leaf node are required. The sensitivity of the classes counts is 1. Allocate privacy budget $\epsilon_2 = \frac{\epsilon}{d}$ for leaf node classes counts:

$$Noisy\_n_c = n_c + Lap\left(\frac{1}{\epsilon_2}\right), \forall c \in C \tag{4.2}$$

where n_c is the number of c type label elements in the node, and the user adds Laplace noise perturbation on the count of each class.

Given privacy budget ϵ for a private random decision tree, we demonstrate the tree building process preserves ϵ-differential privacy. For the perturbation of local best candidate attributes on non-leaf nodes, the allocated privacy budget is $\epsilon_1 = \frac{\epsilon}{d}$. The privacy budget consumption of each non-leaf node layer of the tree is still ϵ_1. The total privacy budget consumed by the $d-1$ layers of non-leaf nodes is $\epsilon_{in} = \epsilon_1 * (d-1) = \frac{\epsilon*(d-1)}{d}$. For each class of count in user's leaf node, allocate privacy budget $\epsilon_2 = \frac{\epsilon}{d}$. the overall privacy budget allocated on the user leaf nodes is $\epsilon_l = \epsilon_2 = \frac{\epsilon}{d}$. The total privacy budget for the user to select split attributes and leaf labels in a tree is $\epsilon_{in} + \epsilon_l = \epsilon$. As a conclusion, ϵ-differential privacy is provided for each tree of the user. All trees are obtained based on the user's training set, the privacy budget budget is accumulated among T trees. The privacy budget consumed by the user to participate in federated learning is $B = T * \epsilon$.

5. Experimental results

In this section, we describe in detail our extensive experiments to evaluate the effectiveness of personalized federated random forest. We show the public datasets considered in the experiment and discuss the results obtained on the target datasets.

5.1. Experimental setup

We use the public human activity recognition dataset UCI SmartPhone [39]. This dataset collected 6 activities of 30 users. These 6 activities are walking, going upstairs, going downstairs, sitting, standing, and lying down. The 30 users are between 19–48 years old. Each user wears a smartphone (Samsung Galaxy S II) on his waist and uses its built-in accelerometer and gyroscope to collect data generated by activities. We also consider the well-known WISDM dataset [40], which has been widely adopted as a benchmark for human activity recognition tasks. WISDM contains accelerometer data collected from the smartphone in each subject's pocket during the execution of the activity. The activities included in the dataset are as follows: walking, jogging, climbing stairs, brushing teeth, folding clothes and so on. We use the data collected by the mobile phone acceleration sensor in the WISDM dataset. The Table 2 shows the detailed information of the datasets used after preprocessing.

We treat each subject in the activity recognition dataset as an independent user in the actual federated environment. To simulate the heterogeneity of data distribution (non-IID) among users, before training, we randomly perform three different states on each user's data. 1) The user's data is sufficient: the user's original data is retained. 2) Insufficient user data: the user's data is randomly sampled. 3) User data label distribution is unbalanced: a part of the classes is randomly selected. The processed data is used as the actual data in each user's federated training. Then the dataset of each user is randomly divided into three groups, training set, validation set, and test set. Among them, 70% of user data is selected to generate training data, 20% of data is selected to generate test data, and the remaining data is used as user verification data. Through the above settings, we have established an actual complex federated learning environment.

Table 2. HAR datasets in our experiments.

Dataset	Subject	Activity	Sampling rate	Sensor	Features	Instrance
Smartphone	30	6	50 Hz	Gyroscope/Accelerometer	561	10,299
WISDM	51	18	20 Hz	Accelerometer	93	20,650

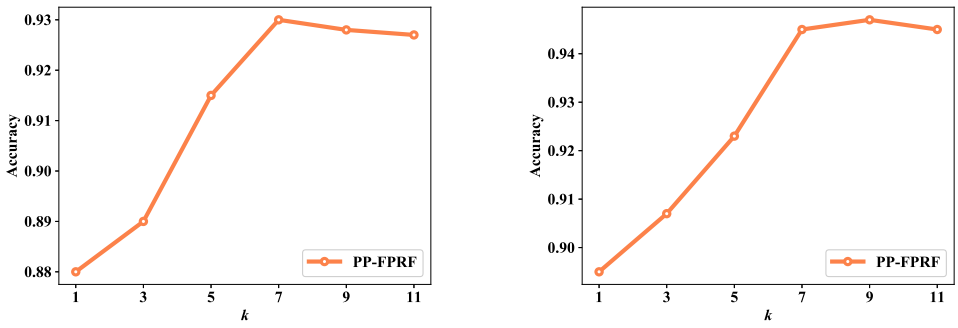
To prove the effectiveness of personalized federated random forest, we compared our privacy-protected federated personalized random forest model (PP-FPRF) with two methods: (1) Local random forest (LRF): The user only trains the random forest model locally. There is no communication among user and others, so the random forest model is only trained based on their own local data, that is, the local model for the user. Users only train locally and do not need to worry about privacy issues. (2) Privacy-protected federated global random forest (PP-FGRF) [12]: All users train a global federated learning model together, without personalized operations. Using differential privacy to protect the intermediate data in the training process, the method of adding noise is the same as that of PP-FPRF.

5.2. Experimental results

We test the performance of the personalized random forest model on the activity recognition datasets, and analyze the impact of the tree settings and the privacy budget on the model.

5.2.1. Classification accuracy

We fix the depth of each tree in LRF (local model), PP-FGRF (global federated model), and PP-FPRF (personalized federated model) to 15, and fix the number of trees in the user random forest model to 20. We assign the same privacy budget $\epsilon = 1$ to each tree of the user in the global federated model and the personalized federated model. The hyperparameter k in the personalized random forest is set to 7, that is, the number of similar users for each user. k is varied from 1 to 11, the experimental results in Figure 2 shows that the performance of the model is better when k reaches 7. It can not only improve the generalization ability of users, but also maintain the personalized characteristics of the models. We train the three models on the SmartPhone and WISDM datasets, and the experimental results are shown in the Tables 3 and 4, where A, B, and C represent three types of users with sufficient data, insufficient data, and unbalanced label distribution, respectively. We can see the average accuracy of different models on these three types of users, as well as the overall accuracy of all users participating in the training.



(a) *SmartPhone* (b) *WISDM*

Figure 2. The effect of varying k.

According to the performance of the model on different datasets, we can see that for all users, the average accuracy of the federated random forest is better than the average accuracy of the random forest trained independently by the users. And our personalization method can further improve the effect of federated random forest. We focus on measuring the average accuracy of several models on the users participating in the training, in addition, we also measured the accuracy of individual users, and then compared whether their participation in federated learning has improved the effect of their models.

We analyze the detailed results of each user in the SmartPhone dataset, as shown in Table 5. For most users, the achieved performance (ie accuracy) of the federated personalized random forest is better than other methods. Clients 1 to 10 are type A users, and their data is relatively sufficient. Clients 11 to 20 are type B users, and their data volume is small. Clients 21 to 30 are type C users, their label distribution is not balanced. Whether it is PP-FGRF or our PP-FPRF, compared with the local models

LRF of type *B* and *C* that don't participate in federated training, there is a big improvement. Although the generalization ability of the global federated random forest PP-FGRF for the overall users has been improved, for type *A* users, such users with sufficient data, the global federated learning has not brought effective improvement. That makes these users lose their motivation to participate in federated training. For them, more data from other irrelevant users is equivalent to noisy data, disturbing the effect of the model. Therefore, we use the personalized training method of similar users and incremental selection to better adapt to the user's personal data and effectively alleviate the above mentioned contradiction.

Table 3. Achieved accuracy of different methods on SmartPhone dataset.

Method	A	B	C	Avg
LRF(Local)	0.932	0.859	0.795	0.862
PP-FGRF(Global)	0.930	0.909	0.912	0.917
PP-FPRF(Personalized)	**0.955**	**0.922**	**0.914**	**0.930**

Table 4. Achieved accuracy of different methods on WISDM dataset.

Method	A	B	C	Avg
LRF(Local)	0.930	0.831	0.871	0.877
PP-FGRF(Global)	0.949	0.936	0.933	0.940
PP-FPRF(Personalized)	**0.956**	**0.939**	**0.940**	**0.945**

Table 5. Accuracy obtained by different methods for each user on the SmartPhone dataset.

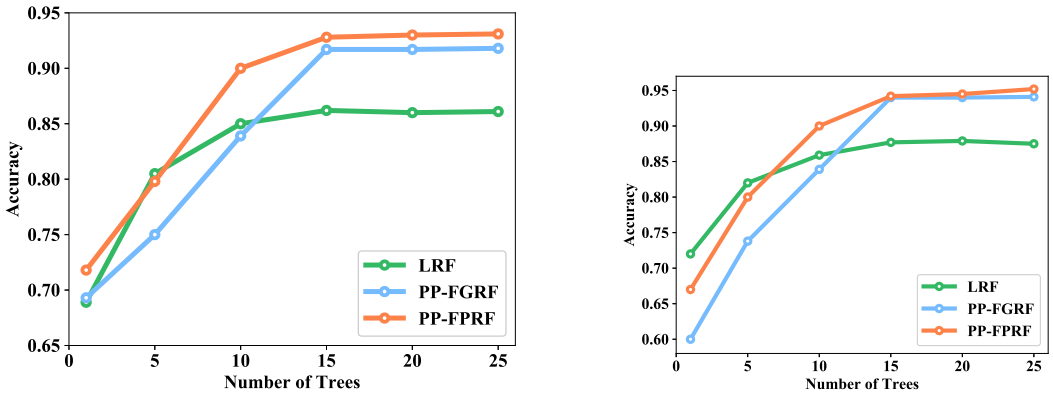
Method	client1	client2	client3	client4	client5	client6	client7	client8	client9	client10
LRF	0.943	**0.976**	0.855	0908	0.963	0.912	0.954	0.882	0.942	0.959
PP-FGRF	0.957	0.967	0.912	**0.968**	0.885	0.938	0.935	0.942	0.913	0.881
PP-FPRF	**0.958**	0.973	**0.935**	0.934	**0.969**	**0.942**	**0.967**	**0.952**	**0.957**	**0.961**
Method	client11	client12	client13	client14	client15	client16	client17	client18	client19	client20
LRF	0.773	0.894	0.900	0.777	0.853	0.809	0.894	0.922	0.896	0.870
PP-FGRF	0.887	**0.943**	0.957	0.827	0.934	0.875	**0.922**	0.902	0.899	**0.942**
PP-FPRF	**0.929**	0.894	**0.967**	**0.881**	**0.959**	**0.914**	0.889	**0.923**	**0.955**	0.913
Method	client21	client22	client23	client24	client25	client26	client27	client28	client29	client30
LRF	0.682	0.831	0.813	0.818	0.768	0.835	0.776	0.805	0.870	0.753
PP-FGRF	**0.885**	0.927	0.919	0.942	**0.945**	**0.938**	0.914	0.870	0.915	0.866
PP-FPRF	0.836	**0.940**	**0.962**	**0.958**	0.865	0.878	**0.951**	**0.895**	**0.939**	**0.909**

5.2.2. Effect of tree settings

We separately tested the influence of the number of trees and the maximum tree depth in the model. When experimenting with the number of trees, fix the remaining hyperparameters and change the number of trees. By observing the changes in the Figure 3, we can find that the accuracy of the three methods from a single tree to multiple trees has been greatly improved, reflecting the advantages of the

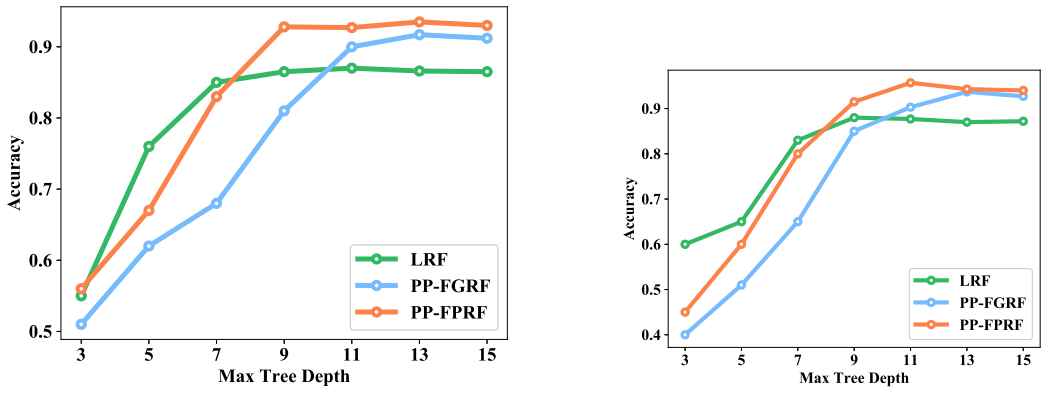
forest structure. However, when the number of trees in the model reaches a certain value, the increase in the number of trees has little effect on the results. This shows that it is necessary for us to adopt an incremental selection method to control the number of trees in the model, which is conducive to reducing the storage and computational overhead of the model.

When testing the influence of maximum tree depth, fix the remaining hyperparameters and change the maximum tree depth of the tree. By observing the changes in the Figure 4, the maximum tree depth has a greater impact on the results, and the accuracy of the model increases as the tree depth threshold increases. The local training model converges when the maximum depth is small. When our personalization model reaches convergence, the maximum depth is smaller than the global model's depth. When the global federated learning trains one tree, there are more users participating in the training, so the overall data is more. The range of data feature values is larger, and deeper nodes are needed to divide the data. Therefore, we use the personalized learning of similarity, users only choose similar users instead of all users to train together, which further reduces the complexity of the model.



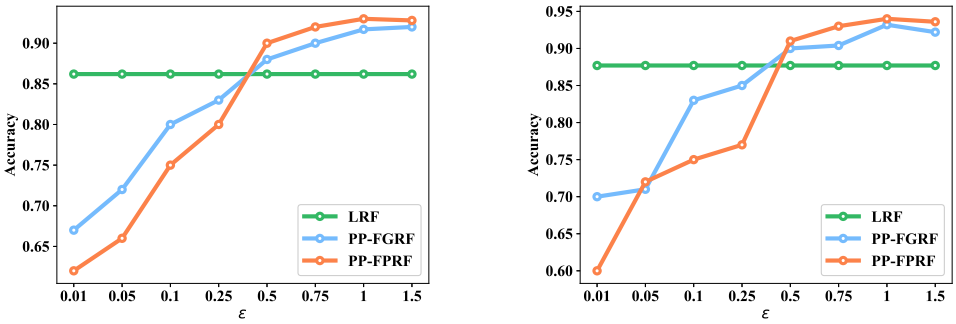
(a) *SmartPhone* (b) *WISDM*

Figure 3. The effect of varying number of trees.



(a) *SmartPhone* (b) *WISDM*

Figure 4. The effect of varying max depth of tree.



(a) *SmartPhone* (b) *WISDM*

Figure 5. The effect of varying privacy budget ϵ.

5.2.3. Effect of ϵ

We observe the change in accuracy by changing the privacy budget ϵ from 0.01 to 1.5. Because the local model does not need to add noise protection, it is not affected by the privacy budget. The results are summarized in Figure 5. The accuracy of the model increases with the increase of the privacy budget. When the privacy budget is small, the accuracy of the model obtained by federated learning is worse than that of the local model. So we should strike a balance between privacy and model utility.

6. Conclusions

In this paper, based on the existing traditional federated random forest model, we propose a personalized federated learning framework. We pay more attention to the improvement of the model accuracy of each user by personalization, so that the federated random forest model is more suitable for human activity recognition task. The personalization method is considered at the two levels of user data and model. Firstly, using effective locality sensitive hashing functions to collect the similarity information without exposing individual data records, users conduct personalized training by selecting similar users. Then, combining with the ensemble learning pruning operation, the generated random tree is personalized selection by the incremental method. At the same time, differential privacy is used in the training phase to protect the private information of users. The experiments show that PP-FPRF improves the classification accuracy of users in activity recognition tasks, and the personalized method also simplifies the complexity of the federated trees model. The personalization method is introduced into the federated random forest model to ensure that more users benefit from federated learning and are more suitable for actual activity recognition tasks. Using differential privacy to protect user data will also result in loss of model accuracy. In the actual application process, the balance between user privacy and model utility must also be considered. In future works, to ensure the matching of user distribution, we will use users with similar distribution to collaboratively train the federated model, and other privacy protection methods to ensure user privacy.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (Nos. 62162005 and 61763003), Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (No. 19-A-02-01), 2021 National Undergraduate Student Innovation Training Program Project (No. 202110602075), Guangxi 1000-Plan of Training Middle-aged/Young Teachers in Higher Education Institutions, Guangxi "Bagui Scholar" Teams for Innovation and Research Project, Guangxi Talent Highland Project of Big Data Intelligence and Application, Guangxi Collaborative Innovation Center of Multisource Information Integration and Intelligent Processing.

Conflict of interest

No potential conflict of interest was reported by the authors.

References

1. O. D. Lara, M. A. Labrador, A survey on human activity recognition using wearable sensors, *IEEE Commun. Surv. Tutorials*, **15** (2013), 1192–1209. doi: 10.1109/SURV.2012.110112.00192.
2. J. P. Queralta, T. N. Gia, H. Tenhunen, T. Westerlund, Edge-AI in LoRa-based health monitoring: fall detection system with fog computing and LSTM cecurrent neural networks, in *Proceedings of the 42nd International Conference on Telecommunications, Signal Processing (TSP)*, Budapest, (2019), 601–604. doi: 10.1109/TSP.2019.8768883.
3. E. Shirin, Z. Ahmed, M. Andreas, S. Severin, A. S. Thomas, E. Tarek, et al., Health management and pattern analysis of daily living activities of people with dementia using in-home sensors and machine learning techniques, *PLoS ONE*, **13** (2018), e0195605. doi: 10.1371/journal.pone.0195605.
4. J. A. Ward, G. Pirkl, P. Hevesi, P. Lukowicz, Towards recognising collaborative activities using multiple on-body sensors, in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Adjunct)*, (2016), 221–224. doi: 10.1145/2968219.2971429.
5. J. Wang, Y. Chen, S. Hao, X. Peng, L. Hu, Deep learning for sensor-based activity recognition: a survey, *Pattern Recognit. Lett.*, **119** (2019), 3–11. doi: 10.1016/j.patrec.2018.02.010.
6. P. Voigt, A. Bussche, The eu general data protection regulation (gdpr), *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, **10** (2017), 3152676.
7. B. Mcmahan, E. Moore, D. Ramage, S. Hampson, B. A. Arcas, Communication-efficient learning of deep networks from decentralized data, in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, **54** (2017), 1273–1282.
8. Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with Non-IID data, preprint, arxiv:1806.00582.
9. Y. Liu, Y. Liu, Z. Liu, J. Zhang, C. Meng, Y. Zheng, Federated forest, *IEEE Trans. Big Data*, 2020. doi: 10.1109/TBDATA.2020.2992755.

10. K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, Q. Yang, Secureboost: A lossless federated learning framework, *IEEE Intell. Syst.*, 2021. doi: 10.1109/MIS.2021.3082561.

11. Q. Li, Z. Wen, B. He, Practical federated gradient boosting decision trees, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 4642–4649. doi: 10.1609/aaai.v34i04.5895

12. Y. Liu, M. Chen, W. Zhang, J. Zhang, Y. Zheng, Federated extra-trees with privacy preserving, preprint, arxiv:2002.07323.

13. L. T. Phong, Y. Aono, T. Hayashi, L. Wang, S. Moriai, Privacy-preserving deep learning via additively homomorphic encryption, *IEEE Trans. Inf. Forensics Secur.*, **13** (2017), 1333–1345. doi: 10.1109/TIFS.2017.2787987.

14. C. Dwork, Differential privacy, in *Proceedinds of the 33rd International Colloquium on Automata, Languages and Programming (ICALP)*, **4052** (2006), 1–12. doi: 10.1007/11787006_1.

15. C. Gentry, *A Fully Homomorphic Encryption Scheme*, Ph.D thesis, Stanford University, 2009.

16. F. Mo, H. Haddadi, K. Katevas, E. Marin, D. Perino, N. Kourtellis, PPFL: privacy-preserving federated learning with trusted execution environments, in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications and Services (Mobisys)*, (2021), 94–108. doi: 10.1145/3458864.3466628.

17. D. Polap, G. Srivastava, K. Yu, Agent architecture of an intelligent medical system based on federated learning and blockchain technology, *J. Inf. Secur. Appl.*, **58** (2021), 102748. doi: 10.1016/j.jisa.2021.102748.

18. K. Sozinov, V. Vlassov, S. Girdzijauskas, Human activity recognition using federated learning, in *2018 IEEE International Conference on Big Data and Cloud Computing (BDCloud)*, (2018), 1103–1111. doi: 10.1109/BDCloud.2018.00164.

19. A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in *Proceedings of 25th International Conference on Very Large Data Bases*, (1999), 518–529.

20. K. Wang, R. Mathews, C. Kiddon, H. Eichner, F. Beaufays, D. Ramage, Federated evaluation of on-device personalization, preprint, arxiv:1910.10252.

21. Y. Chen, X. Qin, J. Wang, C. Yu, W. Gao, Fedhealth: A federated transfer learning framework for wearable healthcare, *IEEE Intell. Syst.*, **35** (2020), 83–93. doi: 10.1109/MIS.2020.2988604.

22. R. Caruana, Multitask learning, *Mach. learn.*, **28** (1997), 41–75. doi: 10.1023/A:1007379606734.

23. V. Smith, C. Chiang, M. Sanjabi, A. Talwalkar, Federated multi-task learning, in *Annual Conference on Neural Information Processing Systems (NIPS)*, (2017), 4424–4434.

24. T. Yu, E. Bagdasaryan, V. Shmatikov, Salvaging federated learning by local adaptation, preprint, arxiv:2002.04758.

25. B. Wang, S. Yu, W. Lou, Y. T. Hou, Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud, in *Proceedings of the 2014 IEEE Conference on Computer Communications (INFOCOM)*, (2014), 2112–2120. doi: 10.1109/INFOCOM.2014.6848153.

26. L. Qi , X. Zhang, W. Dou, Q. Ni, A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data, *IEEE J. Sel. Areas Commun.*, **35** (2017), 2616–2624. doi: 10.1109/JSAC.2017.2760458.

27. M. Datar, N. Immorlica, P. Indyk, V. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in *Proceedings of the 20th Annual Symposium on Computational Geometry*, **34** (2004), 253–262. doi: 10.1145/997817.997857.
28. C. Dwork, F. Mcsherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in *Proceedings of the Third conference on Theory of Cryptography*, **3876** (2006), 265–284. doi: 10.1007/11681878_14.
29. F. Mcsherry, K. Talwar, Mechanism design via differential privacy, in *48th Annual IEEE Symposium on Foundations of Computer Science*, (2007), 94–103. doi: 10.1109/FOCS.2007.41.
30. P. Xiong, T. Q. Zhu, X. F. Wang, A survey on differential privacy and applications, *Chin. J. Comput.*, **37** (2014), 101–122. doi: 10.3724/SP.J.1016.2014.00101.
31. F. Mcsherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, (2009), 19–30. doi: 10.1145/1559845.1559850.
32. P. Patarasuk, X. Yuan, Bandwidth optimal all-reduce algorithms for clusters of workstations, *J. Parallel Distrib. Comput.*, **69** (2009), 117–124. doi: 10.1016/j.jpdc.2008.09.002.
33. L. Breiman, J. Friedman, C. J. Stone, R. A. Olshen, Classification and regression trees, The Wadsworth and Brooks-Cole statistics-probability series, 1984.
34. Z. H. Zhou, J. Wu, W. Tang, Ensembling neural networks: many could be better than all, *Artif. Intell.*, **137** (2002), 239–263. doi: 10.1016/S0004-3702(02)00190-X.
35. H. Chen, P. Tiňo, X. Yao, Predictive ensemble pruning by expectation propagation, *IEEE Trans. Knowl. Data Eng.*, **21** (2009), 999–1013. doi: 10.1109/TKDE.2009.62.
36. A. Friedman, A. Schuster, Data mining with differential privacy, in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, (2010), 493–502. doi: 10.1145/1835804.1835868.
37. S. Fletcher, M. Z. Islam, Decision tree classification with differential privacy: a survey, *ACM Comput. Surv. (CSUR)*, **52** (2019), 1–33. doi: 10.1145/3337064.
38. A. Patil, S. Singh, Differential private random forest, in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, (2014), 2623–2630. doi: 10.1109/ICACCI.2014.6968348.
39. D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine, *Int. workshop ambient assisted living*, **7657** (2012), 216–223. doi: 10.1007/978-3-642-35395-6_30.
40. R. K. Jennifer, M. W. Gary, M. Samuel, Activity recognition using cell phone accelerometers, *ACM SigKDD Explor. Newsl.*, **12** (2010), 74–82. doi: 10.1145/1964897.1964918.

AIMS Press