Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Privacy protection federated learning system based on blockchain and edge computing in mobile crowdsourcing

Weilong Wang [a,b], Yingjie Wang [a,b,*], Yan Huang [c], Chunxiao Mu [a,b], Zice Sun [a,b], Xiangrong Tong [a,b], Zhipeng Cai [d]

[a] School of Computer and Control Engineering, Yantai University, 264005, China
[b] The Yantai Key Laborary of High-end Ocean Engineering Equipment and Intelligent Technology, Yantai, 264005, China
[c] Department of Software Enportraitgineering & Game Development at Kennesaw State University (KSU), 1100 South Marietta Pkwy, Marietta, GA 30060, USA
[d] Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA

ARTICLE INFO

Keywords:
Mobile crowdsourcing
Privacy protection
Blockchain
Edge computing
Federated learning
Localized Differential Privacy

ABSTRACT

With the rapid popularization and development of the Internet of Things (IoT) and 5G networks, mobile crowdsourcing (MCS) has become an indispensable part in today's society. However, when task participants submit tasks, they are likely to expose their data privacy and location privacy. These privacy will be maliciously attacked and exploited by attackers (external attackers and untrusted third party). With the rapid increase of MCS data throughput, traditional cloud platforms can no longer meet the huge data processing needs. To solve these problems, this paper proposes an MCS federated learning system based on Blockchain and edge computing. This paper uses federated learning as the framework of the MCS system. The system protects data privacy and location privacy by using the Double local disturbance Localized Differential Privacy (DLD-LDP) proposed in this paper. Because the sensed data exists in multiple modalities (text, video, audio, etc.), this paper uses the Multi-modal Transformer (MulT) method to merge the multi-modal data before subsequent operations. To solve the problem that the third party is untrusted, we utilize Blockchain to distribute tasks and collect models in a distributed way. A reputation calculation method (Sig-RCU) is proposed to calculate the real-time reputation of task participants. Through conducting experiments on real data sets, the effectiveness and adaptation of the proposed DLD-LDP algorithm and Sig-RCU algorithm are verified.

1. Introduction

In recent years, with the rapid development of mobile Internet, 5G networks and smart mobile device technologies [1,2], the mobile crowdsourcing (MCS) technology has become the research hot spot in Internet of Things [3,4]. MCS provides a new model for traditional data management-solving problems by gathering group wisdom, which greatly facilitates people's lives. Especially with the rapid rise and development of the sharing economy model, MCS technology has been widely used in real-world scenarios [5], such as Meituan and Baidu map. However, there are also privacy leakage issues that cannot be ignored in MCS, such as the data privacy and location privacy of task participants [6]. These task data and location information, which usually contain the private information of task participants (such as identity information, home address, physical condition, etc.), are likely to be attacked and used by malicious attackers [7]. Therefore, the privacy protection problem has become one of the important research problems that need to be solved urgently in today's society [8].

For the problem of privacy protection, scholars have proposed a variety of solutions [9,10]. For example, Tao et al. [11] proposed a mobile sensing incentive mechanism based on privacy protection. This mechanism combined a trusted third party with partially blind signatures to protect the privacy of task participants by reducing the correlation between task participants and sensed data. Lyu et al. [12] proposed an efficient and privacy-preserving aggregation system with the aid of Fog computing architecture (PPFA). They reduced the risk of privacy leakage by adding noise and double-layer aggregation to the data. However, the above articles are based on the completely trusted third party, but there is no completely trusted third party in reality.

Arachchige et al. [13] proposed a system framework (PriModChain) based on differential privacy, federated learning, Ethereum Blockchain and smart contracts. It achieves privacy protection by adding noise to the original data. An et al. [14] proposed a decentralized privacy-preserving model based on twice verification and

[*] Corresponding author at: School of Computer and Control Engineering, Yantai University, 264005, China.
E-mail addresses: wangyingjie@ytu.edu.cn (Y. Wang), yhuang24@kennesaw.edu (Y. Huang), zcai@gsu.edu (Z. Cai).

https://doi.org/10.1016/j.comnet.2022.109206
Received 14 February 2022; Received in revised form 3 July 2022; Accepted 17 July 2022
Available online 21 July 2022

consensuses of Blockchain (TCNS), which used a lightweight homomorphic encryption algorithm to encrypt and protect the sensed data. However, they all cause relatively high data quality loss or resource consumption, and they do not consider the storage pressure of blocks in the Blockchain, so they are not suitable for MCS systems.

In addition, none of the above algorithm frameworks consider the attacks of task participants. However, when there are malicious attackers among the task participants, it will seriously affect the quality of service that the task publisher can obtain. In addition, most current mobile crowdsourcing frameworks target data that is uni-modal (text), but the data that sensing devices can perceive is often multi-modal (images, text, audio, etc.).

In order to solve the above-mentioned problems, this paper proposes an MCS federated learning system based on Blockchain and edge computing. We utilize the steps of model distribution, local training and global aggregation [15] of federated learning to fulfill the requirements of task publishing, data collection and data sorting in traditional mobile crowdsourcing. We offload part of the training tasks to the edge computing server, thereby greatly reducing the computing pressure on the mobile terminal. We utilize consortium Blockchain to distribute tasks and collect models in a distributed way to avoid privacy leakage issues caused by untrusted third parties. We use the DLD-LDP algorithm proposed in this paper to perturb the data and location privacy of task participants, which effectively solves the problems of data quality loss and relatively high resource consumption while protecting privacy. We attach a corresponding reputation value to each task candidate through the reputation calculation algorithm (Sig-RCU) proposed in this paper. Selecting task participants by their reputation value can effectively solve the data quality problem caused by malicious task participants. In addition, we also introduce the Multi-modal Transformer (MulT) [16] algorithm to solve the problem of data form multi-modality. We use the Serverless hosting incentivized peer-to-peer storage and content distribution (Swarm) as a distributed storage solution to effectively solve the storage limitation of blocks in the Blockchain problem. The main contributions of this paper are summarized as follows:

- This paper proposes a MCS federated learning system based on Blockchain and edge computing, which could effectively protect the privacy of task participants, ensure the real-time nature of submitted data and improve the service quality.
- This paper proposes a dual local perturbation local differential privacy mechanism (DLD-LDP), which effectively reduces the loss of data quality under the premise of ensuring the privacy security of task participants through dual local perturbation of the original data. Thereby effectively protecting the privacy of task participants and ensuring the quality of service for task participants.
- This paper proposes the Reputation Computing Algorithm (Sig-RCU), which is able to predict the next task quality of service of task participants with extremely high accuracy. In this way, the best task participant set is selected for the task issuer more accurately and the overall service quality is guaranteed.
- Through conducting experiments on real data sets, the effectiveness and adaptation of the proposed DLD-LDP algorithm and Sig-RCU algorithm are verified.

The rest of this paper is as follows. In Section 2, the current related work is introduced. Section 3 introduces the structure of the proposed system in this paper. Section 4 mainly introduces in detail the key algorithmic mechanisms used in our system. Section 5 gives experiments and analysis to show the effectiveness of the system in this paper. The conclusions are discussed in Section 6.

2. Related works

This section introduces the existing privacy protection methods and cutting-edge technologies in MCS. With the advent of the Internet of Everything era [17] and the rapid development of science and technology [18,19], MCS has rapidly developed into an indispensable part of today's society [20]. In the MCS system, the privacy of task participants is extremely important. The task data uploaded by task participant is often private which may be attacked and used by malicious attackers [21]. Therefore, the issue of privacy protection has always been an important research focus in the field of MCS. Distributed technologies such as federated learning, edge computing, and Blockchain provide new solutions to the above problems.

The workflow of the mobile crowdsourcing system (MCSs) mainly includes that the task publisher publishes the task, the task participant completes and uploads the task data, and the crowdsourcing platform processes the data submitted by all the task participants and sends it to the task publisher. However, task data often contains sensitive information of task participants, which may lead to the leakage of the privacy of task participants. This will reduce the motivation of the task publisher, resulting in a decrease in the quality of service. The framework of the federated learning system can fully meet the framework requirements of the MCS. Task requirements of task participants can be published as initialization models. The perception data of task participants can be trained locally to become a local model and submitted as task data. The aggregate training of the global model is the data processing of the task participants by the crowdsourcing platform in the MCS. The federated learning system enables task participants to analyze data in a decentralized manner and can process the data locally. The federated learning system uploads the gradient or model and there is no need to upload the data to a centralized server. In addition, the MCS based on federated learning always stores the perception data of the task participants locally. Compared with the method of uploading the data to the central server, it can effectively protect the data privacy of the task participants and improve the quality of service. Liu et al. [22] proposed a sketch-based federated learning system to protect the private information of task participants while ensuring the accuracy of prediction. However, Melis et al. [23] proved that even if the data is updated with gradients, important information data for task participants training may still be leaked. The attackers can restore the original data from the task data (gradient) uploaded and submitted by task participants [24]. To this end, Hao et al. [25] proposed a privacy-enhanced federated learning system to solve the problem of private information of task participants. The solution also helped to improve the efficiency of the system in processing data. However, none of the above algorithms consider the untrustworthy issue of third-party platforms.

Some scholars have found that the decentralization, immutability and traceability of the Blockchain can effectively solve the above problems. Therefore, the method of applying Blockchain and federated learning technology to the MCS system has been widely concerned and studied by many scholars [26]. Blockchain is a data structure composed of data blocks in a manner similar to a linked list in chronological order. It is a distributed decentralized ledger that is not tamper-able, cannot be forged, has a robust network, flexible application, and is secure and reliable [27]. The framework form of the Blockchain is similar to that of MCSs. Task publisher can publish tasks and rewards on the Blockchain. Task participants can also upload local task data through the Blockchain to get corresponding rewards. The verification and processing of data are done by a large number of miner nodes. In addition, the immutability and traceability of Blockchain can effectively ensure the quality of data submitted by task participants. The use of Blockchain technology can effectively solve the problems of poor reliability, low security, high cost, and low efficiency in the current centralized MCSs. Weng et al. [28] proposed a distributed, secure and fair collaborative learning system called DeepChain. They used Blockchain to replace traditional third-party service platforms, effectively solving the risk of privacy leakage caused by untrusted third parties. Awan et al. [29] proposed a Blockchain-based privacy protection federated learning framework. It can use the characteristics of Blockchain to protect the data privacy of task participants while also ensuring the security of model updates. However, the framework did not protect data privacy and the system can easily be directly attacked by malicious attackers. Lu et al. [30] proposed a Blockchain-based secure federated learning framework. The framework protects the privacy of task participants by processing task data using localized differential privacy techniques. But none of the above algorithms and frameworks take into account whether the task participants are malicious attackers. The limited storage of blocks in the Blockchain is also not considered.

In addition, with the rapid development of MCSs, the amount of task data generated by task participants is increasing every day. Traditional centralized MCSs can no longer effectively meet the data transmission bandwidth requirements and data real-time requirements required by the system [31]. Therefore, some scholars have introduced fog computing or edge computing into the MCSs to improve its computing performance and data processing efficiency [32]. The basic idea of edge computing is to offload computing tasks to computing resources close to task participants [33], which can effectively reduce system latency and data transmission bandwidth, ease the pressure on cloud computing centers, and improve data availability [34]. It enables task participants to process a large amount of temporary data at the edge of the network without uploading all of them to the crowdsourcing platform, which greatly reduces the pressure on network bandwidth and crowdsourcing platform power consumption. In addition, since the data processing is carried out near the task participants, there is no need to request the central platform to respond through the network, which effectively reduces the system delay and improves the service response capability. For example, He et al. [35] applied the computing power of fog computing to the crowdsourcing bus service system and constructed a privacy protection model. The system not only protects the privacy of task participants but also improves the efficiency of system data processing. Zhao et al. [36] introduced edge computing and Blockchain technology in the federal system and performed differential privacy protection processing on the data of task participants locally. It not only protects the privacy of task participants, but also improves the system data processing efficiency. However, all the algorithms and frameworks described above are aimed at the processing of data from a single modality. The problem of multi-modality of perceptual data is not taken into account.

However, the above-mentioned various algorithms only protected the data privacy of task participants and failed to consider the location privacy protection of task participants. The location privacy protection in the MCS system is also extremely important. Nowadays, the location privacy protection methods in MCS are mainly divided into three categories: anonymity [37], perturbation [38] and encryption [39]. Anonymity, Wang et al. [40] proposed a privacy protection algorithm based on effective false trajectories to protect the location data privacy by implementing k-anonymity. Wang et al. [41] proposed a privacy protection method based on information entropy suppression. This method achieves the purpose of protecting the privacy of task participants by not publishing sensitive or frequently accessed information. However, anonymous algorithms can cause serious data quality loss and cannot effectively deal with inference attacks. Perturbation, Huang et al. [42] proposed a differential privacy mechanism to protect the privacy of workers. By adding noise interference to the location data of task participants, the effect of protecting the privacy of task participants is achieved. Li et al. [43] proposed a differentiated privacy data aggregation method based on personnel division and location confusion. This method also protected the location privacy of task participants by adding differential privacy noise to the real location data. Although interference algorithms can effectively avoid the leakage of data privacy, they may cause serious data quality loss. Encryption, Zhou et al. [44] used homomorphic encryption technology to encrypt the data of task participants. It effectively avoided the data quality loss in the process of privacy protection. Azhar et al. [45] proposed a novel Privacy-preserving and utility-aware participant selection scheme PUPS, which used a lightweight homomorphic encryption method to protect the privacy of task participants. Compared with other encryption algorithms, the resource consumption of this algorithm is relatively low. However, encryption algorithms usually consume a lot of resources and are not suitable for MCS systems.

Table 1
Main notations.

Symbol	Definition
ε	Privacy protection budget
u_i	Task participant
v_j	Task candidate participant
l_i	Real location of task participant
$l_i^{\wedge}$	The final used location of task participant
$\tilde{l}_i$	The attacker's inferred real location of task participant
F_i	The original eigenvalues of each modal data of task participant
X_i	The eigenvalues of each modal data of task participant after interference and multi-modal fusion processing
ES_i	The edge server to which the task participant belongs
w_i	Local model trained by task participant using local data
w	The aggregated global model is the final task data
SL_i	The ID address returned by the task participant storing the local model in the Swarm
Cr_i	Current reputation value of task participant

In response to all the above problems, this paper proposes a privacy protection federated learning system based on Blockchain and edge computing. We adopt federated learning as the overall framework of the system and utilize consortium Blockchain to distribute tasks and collect models in a distributed way. It can effectively avoid the risk of privacy leakage caused by untrusted third parties. We also introduce edge computing technology into the system to improve the execution efficiency of the entire system. We use the DLD-LDP algorithm proposed in this paper to process the perceptual data. It can not only protect data privacy but also effectively reduce data quality loss and resource consumption. We use the proposed Sig-RCU algorithm to compute the reputation of task candidates. Selecting task participants by reputation value can greatly reduce the risk of selecting malicious task candidates to become participants. We also introduce Swarm as an external data storage solution for this system. It can effectively solve the problem of limited storage of blocks in the Blockchain. Finally, we introduce the MulT algorithm into the system. It can solve the problem of multimodality of perceptual data through multimodal fusion.

3. System overview

In this section, we introduce our proposed privacy protection federated learning system based on Blockchain and edge computing and conduct security and performance analysis on it.

3.1. System framework

The MCS system based on federated learning proposed in this paper is shown in Fig. 1. We use the federated learning framework as the overall architecture of the system. The process of the entire system can be roughly divided into four processes: Task publishing, Data collection and processing, Data upload, Data submission. Task publishing: Task publisher utilizes the Blockchain to publish task (initial model) in a distributed manner. Data collection and processing: Task participants perceive task data locally, process it with perturbation and multi-modal fusion, and then train it into a local model through edge servers. Data upload: The task participant saves the local model in Swarm and uploads the returned ID address to the Blockchain. Data submission: The leader node aggregates all qualified local models to the global model, and finally submits the updated global model to the task publisher. The main notations used in this paper are shown in Table 1.

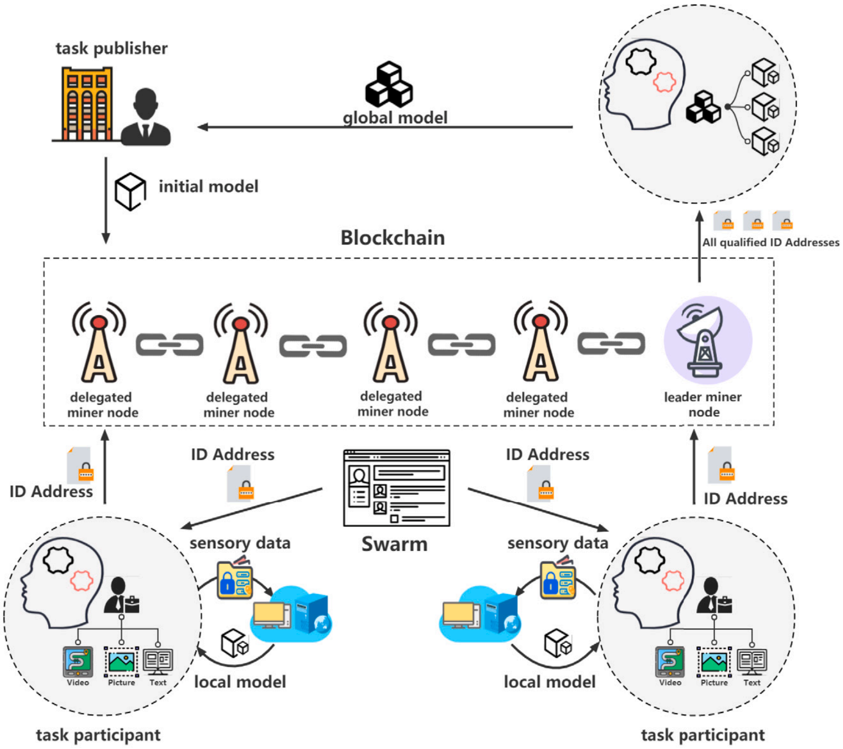

Fig. 1. The framework of a privacy protection federated learning system based on Blockchain and edge computing.

3.2. System model

Our system mainly consists of four parts: task publisher, crowdsourcing platform, edge server and task participants.

Task publisher. Mainly governments, companies or other users who need to obtain task requirements, they can obtain the required data or other requirements by publishing tasks and rewards on the crowdsourcing platform. In this paper, we use the task requirements of the task publisher as the initial model to publish on the Blockchain, and the data or requirements that the task publisher ultimately wants to obtain as the global model.

Crowdsourcing platform. As a bridge between the task publisher and the task participants, its function is to release the task publisher's required tasks and rewards, select qualified task participants to perform the task, and submit the task data submitted by all task participants to the task publisher after processing. As an untrusted central service platform, traditional cloud platforms often bring risks of leakage to users' privacy. For this reason, we abandon the traditional cloud platform. The task release, data submission, and data aggregation processing in the mobile crowdsourcing system are all carried out on the Blockchain, which effectively makes up for the shortcomings of the traditional cloud platform. Task participants can download initial models (accept tasks) and submit local models (submit task data) on the Blockchain. Finally, the leader miner node aggregates and updates the global model (task data aggregation processing) and sends it to the task publisher.

Edge server. A semi-trusted third-party server with relatively high computing power and data processing power and close to the task participants. Some data that cannot be processed by the user can be processed by the edge server, and the data submitted by the user to the cloud platform can also be preprocessed by the edge server, which can effectively reduce the data processing pressure and service delay of the central cloud platform, and improve the operating efficiency of the system. In addition, since the mobile devices used by the task participants cannot meet the requirements of model training, we offload the training task of the local models of the task participants to the edge server.

Task participants. Users who get paid for submitting local task data. When task participants expect to participate in the task, the initial model can be downloaded from the Blockchain and then trained the local data. Since the task participants will upload the data required to train the model to the edge server, and the data usually contain some sensitive information of the task participants. The edge server is a semi-trusted third party, which may lead to the leakage of the privacy of task participants, thus the DLD-LDP algorithm locally is used to protect the data required for model training.

3.3. Threat model

In the traditional federated learning system, both the task participants and the central server have a high degree of autonomy. Malicious attacks by task participants, central servers and external malicious attackers will all have serious impacts on system operation and data quality. Below is our summary of the relevant threat models. They can be divided into three types: malicious attack model, semi-trusted (curious) attack model and other attack model.

Malicious attack model. The attack model is mainly from the internal or external malicious attacks of the system. The specific attack methods are as follows. (1) Data interception attack [46]: This attack is also called network interception attack. After the task participant uploads the data, the external attacker intercepts the uploaded data information of the task participant in the network transmission process by some means. This will lead to the direct disclosure of the data privacy of the task participants. (2) Witch attack [47]: It can also be called collusion attack [48]. The adversary can simulate multiple task participant accounts, or multiple task participants conspire to launch a more powerful attack on the federated learning system, resulting in a serious reduction in the accuracy of the global model. (3) Byzantine attack [49]: Malicious task participants provide fake or low-quality local models to the central server, thereby disrupting the aggregation and updating of the global model of the central server, resulting in a sharp drop in the accuracy of the global model. (4) Label flipping attack [50]: During the execution of the TCN neural network, malicious task participants may modify local dataset labels (the target of mobile crowdsourcing model training), providing low-quality local models, thereby affecting global model update and accuracy. (5) Data Poisoning Attack [51]: Task participants have all permissions to process local perception task data, and malicious task participants may add extreme noise that destroys the quality of local task data. As a result, unreliable local models are generated and uploaded, affecting the accuracy of the global model. (6) Malicious attackers attack [52]: Malicious attackers try to understand the private state of task participants and modify, replay or delete task data (local models) submitted by task participants, thereby affecting the accuracy of the global model and system operation.

Semi-trusted (curious) attack model. This model mainly comes from the attacks launched by semi-trusted roles in the system. The main semi-trusted attack in mobile crowdsourcing system is the member inference attack. Member Inference Attack [53]: Because the central server holds the data information of all the local models submitted by the task participants, the central server can restore the original data of the task participants through the changes of the local models submitted by the task participants many times [23,24]. This will seriously lead to the leakage of the privacy of the task participants.

Other attack models. These models mainly come from uncontrollable attacks such as damage of software and hardware equipment outside the system. The most significant attack mode is the single point of failure attack. Single point of failure attack [54]: The central server is an important part of the federation system, which aggregates all local models and updates the global model. If the central server is attacked or fails to work properly, the aggregation and update of the global model will fail, and the new global model will not be distributed to task participants for subsequent local model training. This will lead to the termination of the entire federated learning system.

3.4. Security analysis

The security threats of the system mainly come from various attacks launched by malicious attack model, semi-trusted (curious) model and other attack models. The system proposed in this paper can effectively solve the attack problems of the above threat models. The specific analysis is as follows.

First, the task participants use the DLD-LDP algorithm to perturb the eigenvalues of the perception task data locally, and then perform the subsequent model training and uploading steps. In this way, even if the attacker can obtain the data submitted by the task participants, the data is perturbed, and the attacker cannot obtain the real data of the task participants. This can effectively resolve member inference attack caused by cloud platform or edge server and malicious attack by external attackers. Secondly, we use Blockchain instead of traditional cloud platform, the distribution and interaction of data are completed through Blockchain. The identity verification and internal data immutability of the Blockchain can effectively reduce the possibility of sybil attacks and malicious adversary attacks. We attach a reputation value to each task participant through the reputation calculation method proposed in this paper, and the dBFT algorithm [15] is used to verify and screen the identities and submitted data of all task participants. Validation and screening results directly affect the task evaluation of task participants. This can help the system select high-quality task participants and task data, thus effectively reducing the possibility of Byzantine attacks, tag flipping attacks, and data poisoning attacks. In addition, we use the Algorand algorithm to randomly select some miner nodes as temporarily delegated miner nodes, and select temporary leader nodes to update the global model through our proposed algorithm. This can make the leader node that processes data different each time, effectively reducing the possibility of member inference attacks. If the temporary leader node does not update the global model within the specified time, select a new temporary leader miner node to update the global model among the temporarily delegated miner nodes. This can effectively solve the threat to the system caused by a single point of failure attack.

In summary, the system proposed in this paper has good security and reliability, and can effectively resist various attacks caused by malicious attacker model, semi-trusted (curious) attack model and other attack models.

3.5. Performance analysis

The distribution of traditional Blockchain (public Blockchain) can reduce system performance, so that it cannot meet the performance requirements of mobile crowdsourcing system. So we utilize the consortium Blockchain with higher performance to distribute tasks and collect models in a distributed manner. In addition, the main reason for the performance degradation is broadcast communication, information encryption and decryption, consensus mechanism, transaction verification, and other links. The first two are mainly related to the amount of data processed. For this reason, we use Swarm technology as an external storage method for Blockchain data. Task participants can store raw task data (local model) in Swarm, only the Swarm ID address is stored on the Blockchain. This can effectively improve system performance. For the consensus mechanism, a new consensus mechanism is proposed through combining Algorand and dBFT algorithms. Each time we update the global model, we randomly select some miner nodes as temporary entrusting miners to complete the consensus work, and use our proposed algorithm to select temporary leader miner node to aggregate and update the global model. This not only ensures fairness, but also improves performance. We did not deal with the transaction verification stage, as transaction verification is an important means of addressing various threat models. In addition, edge computing technology is also referenced in our system, and the task data of task participants can be partially processed in advance through the nearby edge server. This can also effectively reduce the data processing pressure in the system center and improve the overall performance of the system. Although the distribution of the Blockchain will seriously reduce the performance, after the improvement of the above methods, it has been able to effectively meet the performance requirements of the mobile crowdsourcing system.

4. System construction scheme

This section mainly introduces in detail the key algorithmic mechanisms used in privacy protection federated learning system based on Blockchain and edge computing.

4.1. Double local disturbance localized differential privacy (DLD-LDP)

Localized differential privacy (LDP) technology is an important privacy protection processing method in the current MCS field. For LDP technology, each task participates in the privacy processing of task data locally, and the process of privacy processing no longer requires the intervention of a trusted third party. It fully considers the risk of privacy leakage caused by external malicious attackers and untrusted servers during data collection and processing. The formal definition of LDP is as follows.

Definition: ε**-LDP** [55]: Given n task participants, each task participant corresponds to only one record. Given a privacy algorithm M, its domain is $Dom(M)$ and the value range is $Ran(M)$. If the algorithm M is in any two same output result is obtained on records t and $t'(t, t' \in Dom(M))$, $t, t^* \in Ran(M))$ satisfies Eq. (1), then satisfies ε-LDP .

$$Pr[M(t) = t^*] \le e^{\varepsilon} \times Pr[M(t') = t^*] \tag{1}$$

Among them, the probability $Pr[\cdot]$ is controlled by the randomness of the algorithm and also represents the risk of privacy being leaked. The privacy budget parameter ε represents the degree of privacy protection. The smaller ε, the higher the degree of privacy protection. The properties of the LDP algorithm are as follows.

Property 1. *Given a data set D and n privacy algorithms $\{M_1, M_2, \dots, M_n\}$, where $M_i (1 \le i \le n)$ satisfies ε-LDP [56]. The sequence combination of $\{M_1, M_2, \dots, M_n\}$ on D satisfies ε-LDP, where $\varepsilon = \sum_{i=1}^{n} \varepsilon_i$.*

Property 2. *Given a data set D, divide it into disjoint subsets D_i, $D = \{D_1, D_2, \dots, D_n\}$. Let be any privacy algorithm that satisfies ε-LDP, then algorithm M satisfies ε-LDP on D_i, $D = \{D_1, D_2, \dots, D_n\}$ [56].*

At present, the random response mechanism is the mainstream perturbation mechanism of the LDP. According to the privacy budget parameter ε of LDP given by the random response mechanism, the number of interference locations in the candidate location set is $h-1$ and the number of real locations is e^ε. For any input real location l_i, the result of its response to output interference location l_i' is shown by Eq. (2), $l_i, l_i' \in \theta$.

$$P(l_i|l_i') = \begin{cases} \dfrac{e^\varepsilon}{h-1+e^\varepsilon}, l_i' = l_i \\ \dfrac{1}{h-1+e^\varepsilon}, l_i' \ne l_i \end{cases} \tag{2}$$

It means that the algorithm will have a probability response of $\frac{e^\varepsilon}{h-1+e^\varepsilon}$ to output the true result. It will have a probability response of $\frac{1}{h-1+e^\varepsilon}$ to output any one of the other $h-1$ interference locations in the candidate location set.

Algorithm 1 The DLD-LDP algorithm

```
Input: $l_i, \varepsilon$
Output: Final used location $l_i^\wedge$
1: $u_i$ gets the number of generated interference data $k-1$
2: Convert $l_i$ to a binary array $B_i$ of length $2log_2 k$
3: for each $b$ from 1 to $len(B_i)/2$ do
4:     if this set of centrosymmetric data($B_i[b]$ and $B_i[len(B_i)-b-1]$)
       satisfies the exchange condition then
5:         $a = B_i[b], B_i[b] = B_i[len(B_i)-b-1], B_i[len(B_i)-b-1] = a$
6:     end if
7: end for
8: Convert $B_i$ to the original data type again to obtain the interference
   location $l_i'$
9: $u_i$ chooses one of $l_i$ or $l_i'$ with probability $P(l_i|l_i')$ as the final used
   location $l_i^\wedge$
10: return $l_i^\wedge$
```

Although the LDP mechanism has a strong ability to protect the privacy of task participants, it will cause relatively high losses to the quality of task data uploaded by task participants. For this reason, this paper improves the LDP algorithm implemented in the traditional WRR and kRR [57] algorithms by combining Properties 1 and 2. The traditional WRR and kRR algorithms convert the original data into binary arrays with a length of $log_2 k$. Then random perturbation is performed on the binary data of each bit, and k different results can be obtained, one of which is the original data, so that the definition of LDP in kRR can be satisfied. However, this approach results in a relatively high data quality loss. The DLD-LDP algorithm converts the original data into a binary array of length $2log_2 k$, and realizes random perturbation by randomly deciding whether to exchange each group of centrosymmetric binary data. Because the array length is $2log_2 k$, there will be $log_2 k$ sets of centrosymmetric binary data. Therefore, k data results can still be obtained in this way. Its specific realization is shown in Algorithm 1.

The specific realization steps of Algorithm 1 are as follows. The input includes the real location l_i of the task participant u_i and the location privacy budget ε. The output is the final used location $l_i^\wedge$ of the task participant u_i. Step 1 and Step 2: convert the real location l_i of the task participant u_i into a binary array B_i. Steps 3–7: perform perturbation exchange for each group of centrosymmetric binary data in B_i. Step 8: convert B_i to primitive data type to get the interference location l_i'. Step 9, task participant u_i chooses one of l_i or l_i' with probability $P(l_i|l_i')$ as the final used location $l_i^\wedge$. Step 10: return to the final used location $l_i^\wedge$.

According to Algorithm 1, we can get the probability that the final upload location of the task participant is the real location is shown in Eq. (3), the probability of uploading the interference location is shown in Eq. (4) [58].

$$Pr[DLD-LDP(l_i,\varepsilon) = l_i] = \frac{e^\varepsilon}{k-1+e^\varepsilon} \tag{3}$$

$$Pr[DLD-LDP(l_i,\varepsilon) = l_i'] = \frac{1}{k-1+e^\varepsilon} \tag{4}$$

where k represents the number of results generated by the algorithm, the number of interference results is $k-1$, and the number of real results is 1.

Proof. According to Eqs. (3) and (4), the following Eq. (5) could be obtained. From Eq. (5), it could be concluded that the DLD-LDP algorithm meets the definition of ε-LDP under any conditions.

$$\frac{Pr[DLD-LDP(l_i,\varepsilon) = l_i]}{Pr[DLD-LDP(l_i,\varepsilon) = l_i']} \le e^\varepsilon \tag{5}$$

In addition, suppose that the attacker has a certain prior knowledge $\pi(l_i)$ of the probability distribution of the real location l_i of the task participant. The sensing area of the task participant is θ. The attacker passes the prior knowledge $\pi(l_i)$ and the interference location of the task participant. The probability that l_i' knows the real location l_i of the task participant is $P(l_i'|l_i)$. Then, by observing the interference location l_i' of the task participant. The attacker predicts that the posterior probability of the real location l_i of the task participant is $\tau(l_i|l_i')$, which is expressed by Eq. (6).

$$\tau(l_i|l_i') = \frac{P(l_i'|l_i) \times \pi(l_i)}{\sum_{l_i^* \in \theta} P(l_i'|l_i) \times \pi(l_i^*)}, \forall l_i^*, l_i, l_i' \in \theta \tag{6}$$

where l_i^* is represented as a similar interference location of another task participant that is similar to the interference location of the task participant l_i'. $\tau(\hat{l}_i|l_i')$ represents the inference attack carried out by task participants based on a certain experience, that is, when the attacker observes the interference location of the task participant l_i'. The real location of the task participant l_i is inferred to be the posterior probability of $\tilde{l}_i$ by the attacker [59]. In this paper, the real location of the task participant u_i is represented as l_i and the expected reasoning error E_i is represented by Eq. (7).

$$E_i = \sum_{l_i' \in \theta} P(l_i'|l_i) \sum_{\tilde{l}_i \in \theta} \tau(\tilde{l}_i|l_i') \times d(\tilde{l}_i, l_i) > 0 \tag{7}$$

where $d(\tilde{l}_i, l_i)$ represents the distance between the predicted location $\tilde{l}_i$ and the real location l_i. Therefore, it can be concluded that the algorithm proposed in this paper can effectively deal with the inference attack of the attacker.

4.2. MulT (Multimodal Transformer)

For task data collection and processing, most of the existing crowdsourcing systems only perform tasks on single modal data. In MCS, the sensing data obtained by task participants are usually multi-modal, such as text, video and audio. Therefore, this paper introduces the MulT algorithm in the crowdsourcing system. MulT [16] is a multi-modal conversion algorithm for analyzing human multi-modal languages. The core of MulT is the cross-modal attention mechanism, which provides a potential cross-modal adaptation to fuse multi-modal information by directly focusing on the low-level features of other modalities.

There are usually three main modalities involved in multi-modal language sequences: text (L), video (V) and audio (A). The initial extraction features of the three modalities are: $F_{\{L,V,A\}} \in R^{T_{\{L,V,A\}} \times d_{\{L,V,A\}}}$, where d is a common dimension. Since direct data processing will cause serious data quality loss, this paper directly performs privacy processing on feature F and the result is shown by Eq. (8).

$$F'_{\{L',V',A'\}} = DLD - LDP(F_{\{L,V,A\}}, \varepsilon) \tag{8}$$

where $F'_{\{L',V',A'\}} \in R^{T_{\{L',V',A'\}} \times d_{\{L',V',A'\}}}$ represents the modal characteristics after processing by the privacy protection algorithm. Through computing the time distribution and location embedding processing for each modal feature, the result can be calculated by Eqs. (9)–(11) [16].

$$\hat{F}'_{\{L',V',A'\}} = Conv1D(F'_{\{L',V',A'\}}, k_{F'_{\{L',V',A'\}}}) \tag{9}$$

$$X^{[0]}_{\{L',V',A'\}} = \hat{F}'_{\{L',V',A'\}} + PE(T_{\{L',V',A'\}}, d) \tag{10}$$

$$PE(m,k) = \begin{cases} \sin(\frac{m}{10000^{\frac{k}{d}}}), \frac{k}{2} = 0 \\ \cos(\frac{m}{10000^{\frac{k-1}{d}}}), \frac{k}{2} \neq 0 \end{cases} \tag{11}$$

where $Conv1D$ represents one-dimensional convolution. $k_{F'_{\{L',V',A'\}}}$ represents the size of the convolution kernel of the $\{L', V', A'\}$ modal. $\hat{F}'_{\{L',V',A'\}} \in R^{T_{\{L',V',A'\}} \times d}$ represents the modal characteristics after time convolution processing. d is a common dimension. $PE(T_{\{L',V',A'\}}, d) \in R^{T_{\{L',V',A'\}} \times d}$ is used to calculate the (fixed) embedding of each location index. The conversion of each modal data feature based on the cross-modal attention block is shown by Eqs. (12)–(14) [16].

$$X^{[0]}_{G_i \to G_j} = X^{[0]}_{G_j} \tag{12}$$

$$\hat{X}^{[m]}_{G_i \to G_j} = CM^{[m],mul}_{G_i \to G_j}(LN(X^{[m-1]}_{G_i \to G_j}), LN(X^{[0]}_{G_i})) + LN(X^{[m-1]}_{G_i \to G_j}) \tag{13}$$

$$X^{[m]}_{G_i \to G_j} = f_{\theta^{[m]}_{G_i \to G_j}}(LN(\hat{X}^{[m]}_{G_i \to G_j})) + LN(\hat{X}^{[m]}_{G_i \to G_j}) \tag{14}$$

where $X^{[0]}_{G_i, G_j \in \{L',V',A'\}}$ represents the low-level location sensing features produced by different modalities. $m \in \{1, 2, \dots, M\}$ represents the number of feedforward layers calculated by the cross-mode transformer. f_θ represents the location feedforward network. LN stands for layer normalization. $CM^{[m],mul}_{G_i \to G_j}$ means a multi-head version of CM at layer m, which could be calculated by Eq. (15) [16].

$$\begin{aligned} CM_{b \to a}(Y_a, Y_b) &= softmax(\frac{Q_a C_b^T}{\sqrt{d_k}}) \\ &= softmax(\frac{Y_a W_{Q_a} W_{C_b}^T Y_b^T}{\sqrt{d_k}}) Y_b W_{V_b} \end{aligned} \tag{15}$$

where $Y_a \in R^{T_a \times d_a}, Y_b \in R^{T_b \times d_b}$ represent two different modal data. T represents the sequence length, d represents the feature dimension, $Q_r = Y_r W_{Q_r}$ is $Query$, $C_r^T = W_{C_r}^T Y_r^T$ means Key, $V_r = Y_r W_{V_r}$ is $Value$. $W_{Q_r}, W_{C_r}, W_{V_r} \in R^{T_r \times d_r}$ represents their weight. r indicates the modalities a, b. Finally, the output of MulT with the same target modality is connected to obtain the fusion feature $X_{\{L',V',A'\}} \in R^{T_{\{L',V',A'\}} \times 2d}$. For example, $X_a = [X_a^{[D]} : X_{b \to a}^{[D]} : X_{c \to a}^{[D]}]$, $a, b, c \in \{L', V', A'\}$, D represents the number of layers of cross-modal attention blocks in each MulT algorithm.

4.3. Federated learning

The concept of federated learning was originally proposed by the Google team [60]. The main idea of federated learning is to establish a machine learning model based on data sets distributed on multiple devices while preventing data leakage. Federated learning is a decentralized learning method. Task participants can independently train the model locally, learn and share the global model by aggregating local model calculation updates, without uploading data to the central server. Federated learning provides a brand-new solution for privacy protection in the MCS system. The execution process of federated learning is rough as follows.

Algorithm 2 Multi-source disturbance FedAvg learning algorithm

```
Input: $w_0, \varepsilon, I, E, B, \gamma, \nabla\xi(;), t, V = \{V_1, V_2, ..., V_M\}$
Output: $w$
initialize $w_1 = w_0$
for each round $t = 1, 2...$ do
    Select $I$ candidates from $V$ as task participants to join the task participant set $U$,$U = u_1, u_2, ..., u_I$
    The task publisher publishes the initial model $w_1$ as a task to the task participants $U$ through the Blockchain and edge server $ES$
    for each $u_i \in U$ do
        $u_i \leftarrow DLD - LDP\_M(u_i, \varepsilon)$
        $u_i$ offloads the training task to its own edge server $ES_i$
        Edge server $ES_i$ performs $w^i_{t+1} \leftarrow LocalUser(u_i, w_t)$ operation
    end for
    Choose a leading miner node to perform operation $w = w_{t+1} \leftarrow \frac{1}{|\{U\}|} \sum_{i=1}^{I} w^i_{t+1}$operation
end for
return $w$ to Blockchain
$DLD - LDP\_M(u_i, \varepsilon)$ :
$F'_i = \{\}$
for $f_i^k \in F_i = \{f_L^i, f_V^i, f_A^i\}$ do
    $\hat{f}_i^k = DLD - LDP(f_i^k, \varepsilon)$
    $F'_i.append(\hat{f}_k^i)$
end for
Use MulT algorithm $F'_i$ to process to get unified feature $X_i$
Replace the original $F_i$ in the with $X_i$
return $u_i$
$LocalUser(u_i, w_t)$ ://Run on task participant
$\beta \leftarrow (split\ p_i\ into\ B)$
for each local epoch $e$ from 1 to $E$ do
    for batch $b \in B$ do
        $w \leftarrow w - \gamma \cdot \nabla\xi(w; b)$
    end for
end for
return $w$ to $u_i$
```

Initialize the model: the task publisher define the released task as the initialization model w_0 and publishes it through the Blockchain.

Task participant local training: the task participant $u_i \in \{u_1, u_2, \dots, u_I\}$ uses the task data he perceives locally to train the local model. And he obtains the updated local model w^i_{t+1} through $w^i_{t+1} \leftarrow w_t$, that is, the task participant u_i solves the optimization problem of the loss function. This paper trains the model based on Eq. (16).

$$\underset{w \in R}{argmin} F_i(w), F_i(w) \overset{def}{=} \frac{1}{N_i} \sum_{n=1}^{N_i} f_n(w) \overset{def}{=} \frac{1}{2N_i} \sum_{n=1}^{N_i} (x_n^T w - y_n) \tag{16}$$

where x_n and y_n represent the input and output vectors of the local data set LD_i of task participant u_i. N_i represents the division size of LD_i, w represents the local model trained by task participants. $f_n(w)$ represents the loss function of local model training.

Central server aggregation: after the task participant u_i uploads the trained local model w_t^i to the Blockchain. The leader miner node in the Blockchain will use the FedAvg algorithm to aggregate the local models trained by all task participants and finally update a new global model w_{t+1}. The update process is shown by Eq. (17).

$$w_{t+1} \leftarrow w_t - \frac{1}{I} \sum_{i=1}^{I} F_i(w) \tag{17}$$

Algorithm 2 is a multi-source disturbance Federated Average learning algorithm after adding the DLD-LDP algorithm and MulT algorithm. The input is the initialization model (task) w_0, the privacy protection budget ε, batch size B, local epoch E, the learning rate γ, the optimization function $\nabla\xi(;)$, the number of task interactions t and the task candidate participants set $V = \{V_1, V_2, \dots, V_M\}$. The output is the final global model w. Step 1: initialize the model. Steps 2–10: perform multiple interactions to obtain the final global model w. Step 3: select the task participant set U from the task candidate participants set V according to the reputation mechanism. Step 4: send w_1 as a task to the task participant u_i through the Blockchain and edge server. Steps 5–9: all participants use their sensing data and w_t locally to train the local model w^i_{t+1}. Step 10: aggregate all local models to obtain a unique global model w. Step 11: upload the final global model w to the Blockchain. Steps 12–21: perform perturbation and multi-modal fusion processing on the characteristics of the u_i's sensed data. Steps 22–29: u_i performs local training through its sensed data's characteristics and the current initial model w_t, so as to obtain the trained local model w^i_{t+1}.

4.4. Blockchain and swram

Blockchain is a decentralized distributed technology that can effectively avoid the risk of privacy leakage caused by untrusted third parties. The data written into the Blockchain can only perform *add* and search *operations*. It cannot perform *delete* and *modify* operations. Each miner node can obtain a backup of the entire Blockchain's complete database [61]. The information transmission of the Blockchain adopts the form of broadcasting and only after 51% of the miner nodes agree, can the task or sensor information be released. If there is a problem with this task or sensor information, it will be discarded and it will no longer be broadcast [62]. This feature of Blockchain guarantees the reliability of task data to a certain extent. The content of the Blockchain is non-tamperable, non-forgeable and non-repudiation. Combining it with the reputation mechanism can effectively solve the single-point attacks and Byzantine attacks in federated learning. Since the data on the Blockchain is transparent, but the nodes are anonymous. It has a good effect on protecting the identity and privacy of task participants without affecting the use of system data.

Blockchain mainly includes underlying transaction data, narrowly-defined distributed ledgers, important consensus mechanisms, complete and reliable distributed networks and distributed applications on the network [63]. A block is a data structure formed by the underlying data organization. The Blockchain is made up of multiple blocks linked in chronological order. The task data exists in the form of a Merkle tree on the Blockchain. The construction process is a process of recursively calculating the hash value, that is, the hash value is calculated recursively layer by layer until the only Merkle root is left. For the calculation of the hash value, this paper uses the $Double-SHA256$ encryption algorithm to hash the data block sl^i_k. sl^i_k is cut from the Swarm ID address SL_i uploaded by the task participants. The calculation is as follows:

$$Hash_i = SHA256(SHA256(sl^i_k)) \tag{18}$$

From Eq. (18), it can be seen that the $Double-SHA256$ algorithm is actually the data obtained after $SHA256$ processing and then $SHA256$ processing again. The specific implementation process is shown in Algorithm 3.

In the Algorithm 3, $\wedge$ represents bitwise and. $\neg$ represents bitwise $complement$. $\oplus$ represents bitwise exclusive OR. S^n represents rotate right n bytes. R^n represents shift right n bytes. The input is the small data block sl^i_k after the uploaded data from u_i is divided by the internal mechanism of the Blockchain. The output is the hash value of sl^i_k. Step 1: perform two processing on sl^i_k. Step 2: return the hash value obtained in step 1 to output. Step 4: initialize hash initial value and hash constant. Step 5 and Step 6: perform information preprocessing by adding padding bits and length values. Step 7: divide the message into blocks. Steps 8–26: process all the divided blocks. Steps 10–14: calculate the value of each element in $\{W[16], \dots, W[63]\}$. Step 15: use the initialized initial hash value as the initial hash value of each block. Steps 16–22: calculate the final value of each element in $\{h_0, \dots, h_7\}$ through 64 cycles. Steps 23–25: assign each value of the element in the current $\{h_0, \dots, h_7\}$ to each element in $\{H_0, \dots, H_7\}$. Step 27: add the values of the final $\{H_0, \dots, H_7\}$ elements in order to obtain the final value and assign it to $HASH$. Step 28: return and output the final value of $HASH$.

Algorithm 3 Use Double-SHA256 algorithm to find the hash value

```
Input: $sl^i_k$
Output: $w$
 1: $Hash_i = SHA256(SHA256(sl^i_k))$
 2: return $Hash_i$
 3: SHA256 $(sl^i_k)$
 4: Initialize hash initial value $\{H_0, ..., H_7\}$ and hash constant $\{C_0, ..., C_{63}\}$
 5: The first bit of the message is first filled with '1' and the rest are filled with '0' until the message length satisfies the modulus of 512 and the remainder is 448
 6: Append length of message, as 64-bit big-endian integer
 7: Break message into 512-bit chunks
 8: for each chunk do
 9:     Break chunk into sixteen 32-bit big-endian words $\{W[0], ..., W[15]\}$
10:     for $j$ from 16 to 63 do
11:         $s_0 = S^7(W[j-15]) \oplus S^{18}(W[j-15]) \oplus R^3(W[j-15]$
12:         $s_1 = S^{17}(W[j-2]) \oplus S^{19}(W[j-2]) \oplus R^{10}(W[j-2]$
13:         $W[j] = W[j-16] + s_0 + W[j-7] + s_1$
14:     end for
15:     Assign $\{H_0, ..., H_7\}$ to $\{h_0, ..., h_7\}$ as the initial hash value of the block
16:     for $i$ from 0 to 63 do
17:         $s_0 = S^2(h_0) \oplus S^{13}(h_0) \oplus S^{22}(h_0), s_1 = S^6(h_4) \oplus S^{11}(h_4) \oplus = S^{25}(h_4)$
18:         $Maj = (h_0 \wedge h_1) \oplus (h_0 \wedge h_2) \oplus (h_1 \wedge h_2)$
19:         $Ch = (h_4 \wedge h_5) \oplus (\neg h_4 \wedge h_6)$
20:         $t_1 = h_7 + s_1 + Ch + C[i] + W[i], t_2 = s_0 + Maj$
21:         $h_7 = h_6, h_6 = h_5, h_5 = h_4, h_4 = h_3 + t_1, h_3 = h_2, h_2 = h_1, h_1 = h_0, h_0 = t_1 + t_2$
22:     end for
23:     for $j$ from 0 to 7 do
24:         $H_j = h_j$
25:     end for
26: end for
27: $HASH = H_0\ append\ H_1 append\ H_2\ append\ H_3\ append\ H_4\ append\ H_5\ append\ H_6\ append\ H_7$
28: return $HASH$
```

In addition, $Merkle\ tree$ not only has good privacy protection capabilities, but it also has extremely high extensibility. No matter how much data information is, it can finally generate a fixed-length $Merkle\ root$. The formation process of $Merkle\ root$ is shown in Algorithm 4. The input is the task participant u_i and the output is the $Merkle\ root$ stored in the Blockchain. Step 1 and Step 2: store and upload the local model of the task participant u_i. Step 3: the internal mechanism of the Blockchain divides the data into several data blocks $SL_i = \{sl^i_1, sl^i_2, \dots, sl^i_K\}$. Steps 5–8: hash all data blocks in the and save them in the set SL_i. Step 9: start constructing $Merkle\ tree$ with all elements in $Hash$ as nodes. Step 10: judge the number of leaf nodes. If it is 0, it proves that the data is empty and proceed directly to Step 11, otherwise repeat Steps 13–23. Step 11: return 0 to prove that the u_i uploaded data is empty. Step 13: judge whether there is only one leaf node at present, if there is one, go to Step 24, otherwise go to

Algorithm 4 Swarm-based Merkle tree generation algorithm

```
Input: $u_i$
Output: $Merkle\ root$
1: The task participant $u_i$ stores its trained local model $w_i$ in Swarm and returns an ID address $SL_i$
2: $u_i$ uploads $SL_i$ to the Blockchain through the edge server $ES_i$
3: The internal mechanism of the Blockchain divides $SL_i$ into $K$ small data blocks, namely $SL_i = \{sl_1^i, sl_2^i, ..., sl_K^i\}$
4: $Hash = \{\}$
5: for each $sl_k^i \in SL_i$ do
6:     $Hash_k = Double - SHA256(sl_k^i)$
7:     $Hash.append(Hash_k)$
8: end for
9: Use the elements in $Hash$ as the current leaf nodes of $Merkle\ tree$
10: if he current $Merkle\ tree$ has no leaf nodes then
11:     return 0
12: end if
13: while the number of leaf nodes of the current $Merkle\ tree$ is not 1 do
14:     if the number of leaf nodes is not a multiple of 2 then
15:         $Hash.append(Hash_{len(Hash)})$, that is, add a leaf node the same as it after the last leaf node of $Merkle\ tree$
16:     end if
17:     $newHash = \{\}$
18:     for each $j$ from 1 to $len(Hash/2)$ do
19:         $newHash.append(Double - SHA256(Hash_{2j-1}Hash_{2j}))$
20:     end for
21:     $Hash = newHash$
22:     $Merkle\ tree\ layers = Merkle\ tree\ layers + 1$
23: end while
24: $Merkle\ root = Hash_1$
25: return $Merkle\ root$
```

Steps 15–23. Step 14: determine whether the number of leaf nodes is an odd number. If it is an odd number, perform Step 15 and Step 16. Step 15 and Step 16: copy the last element in $Hash$ and add it to $Hash$. Steps 17–21: combine all the elements in the $Hash$ pair by pair for hash processing and use the result as an element in the new $Hash$. Step 24: the only value in $Hash$ is $Merkle\ root$. Step 25: return the that outputs u_i.

In this system, we use edge servers as miner nodes, and propose a new consensus algorithm (AdBFT) combining Algorand algorithm and dBFT [15] algorithm. First, when a task participant uploads their local model or Swarm ID address, the miner node will verify and screen the task participant's identity and uploaded data through the verification algorithm in the dBFT algorithm. Data that passes validation and screening will be put into the transaction pool. Task participants who submit unqualified data or whose identity has not been verified will be marked and appropriately punished (such as lowering the score of this task, deducting rewards, etc.). Second, every time the global model is updated, the Algorand algorithm is used to randomly select some miner nodes as temporary commissioned miners, and they all aggregate the data in the transaction pool to obtain the global model. Then, according to Eq. (19), the temporary leader node is selected from all the temporary delegated miners, which uploads the global model to the Blockchain and updates the global model. If the temporary leader node does not update the global model within the constraint time, a new temporary leader node will be re-selected in the temporary delegated miner node to update the global model.

$$p_g = \frac{\sigma S_g + \mu N_g^*}{\sum_{g=1}^{G}(\sigma S_g + \mu N_g^*)}, \sigma + \mu = 1 \tag{19}$$

where p_g is the probability of choosing edge server ES_g as the leading miner node. S_g is the account balance of the edge server, that is, the margin N_g^* is the number of task participants in the edge server ES_g. S_g is the security deposit for ES_g as an honest leader of the miner node. $\sigma S_g + \mu N_g^*$ is the bargaining chip for the edge server ES_g to campaign to lead the miner node. σ and μ are the weights of S_g and N_g^* in the bargaining chip, respectively.

Due to the limited block storage in the Blockchain, this paper introduces Swarm as an off-chain storage method to solve this problem. Swarm is a decentralized distributed storage technology. Swarm is hot storage, with the characteristics of timely storage, processing and distribution, which can fully meet the real-time data requirements of mobile crowdsourcing. Swarm interacts through nodes and has low requirements for hardware equipment and the mobile terminal equipment of task participants can meet the conditions. Based on the above characteristics of Swarm, Swarm can effectively reduce the storage pressure of blocks in the Blockchain while avoiding the risk of privacy leakage caused by untrusted third parties.

4.5. Reputation mechanism

The idea of MCS is to select part of the task participants to complete the crowdsourcing task. When selecting task participants, the reliability of task participants cannot be guaranteed. It will greatly affect the quality of service obtained by the task publisher and it is very easy to cause poisoning attacks and Byzantine attacks. In order to ensure the reliability of data uploaded by task participants, this paper designs a reputation calculation algorithm (Sig-RCU). The algorithm is based on the historical task scores of task participants and different task scores are given different weights according to the time series and the total number of task scores. The closer the task scoring time distance is to the current time, the higher its weight in reputation calculation. For newly registered task participants, their reputation is equivalent to the average of the current reputations of all non-new task participants. The calculation method of the Sig-RCU algorithm is shown in Eqs. (20) and (21).

$$MAR = \frac{\sum_{j=1}^{len_i} \left| \frac{\sum_{j=1}^{len_{i_j}} h_{i_j}}{len_{i_j}} - h_{i_j} \right|}{len_i} \tag{20}$$

$$Cr_i = \begin{cases} \dfrac{\sum_{j=1}^{len_i}\left(\frac{e^{-(j-(len_i+1)/2)}}{1+e^{-(j-(len_i+1)/2)}} \times h_{i_j}\right)}{\sum_{j=1}^{len_i}\frac{e^{-(j-(len_i+1)/2)}}{1+e^{-(j-(len_i+1)/2)}}}, \\ \quad Non-new\ task\ participant\ and\ MAR < \alpha \\ \dfrac{\sum_{j=1}^{len_i} h_{i_j}}{len_i}, Non-new\ task\ participant\ and\ MAR \geq \alpha \\ \dfrac{\sum_{o=1}^{N} Cr_o}{N}, new\ task\ participant \end{cases} \tag{21}$$

where Cr_i represents the current reputation value of the task participant u_i. len_{i_j} represents the number of historical tasks completed the first j times by u_i. len_i represents the number of historical tasks completed by u_i. h_{i_j} represents the quality score of the previous tasks completed by u_i. N represents the number of non-new registered task participants in this task. Cr_o represents the current reputation value of the non-new registered task participant u_o in this task. α is the credit fluctuation factor, that is, the maximum credit fluctuation difference allowed by the algorithm. MAR is the average absolute error of service quality of task participants. This algorithm not only considers the reputation calculation of task participants with historical task scores, but also considers the cold start problem, that is, the reputation calculation of newly registered task participants. Miner nodes can filter task participants based on their reputation. It can help the system effectively solve the problems of poisoning attacks and Byzantine attacks.

5. Experiment and result analysis

In this section, we conduct comparison experiments to evaluate the performances of the proposed method. All experiments are written in Python language on JetBrains PyCharm 2019.2.6 x64 platform. The hardware environment of all experiments is Intel(R) Core(TM) i5-6200U, CPU @ 2.30 GHz, 8 GB RAM, running Win 10 OS.

5.1. Experimental setup

Experimental data set. This paper used the real New York taxi GPS location data sets (2016) to conduct the comparison experiments on location privacy protection. It includes the ID, timestamp, passenger capacity, longitude and latitude of each driver's location. Use real textual (GloVe word embeddings [64]), visual (Facet [65]), and acoustic (COVAREP [66]) data modalities data sets for comparison experiments. GloVe word embeddings is a 300 dimensional vector. Facet to indicate 35 facial action units, which records facial muscle movement for representing per-frame basic and advanced emotions. The feature of COVAREP includes 12 Mel-frequency cepstral coefficients (MFCCs), pitch tracking and voiced/unvoiced segmenting features, glottal source parameters, peak slope parameters and maxima dispersion quotients the dimension is 74. A comparison experiment of reputation calculations using real Dianping (2018) data set. It includes the IDs of multiple businesses, the historical service scores they obtained, the ID of the rater and the time of the score.

Contrast algorithm. (1) $\varepsilon - Differential\ Privacy\ mechanism$ $(Laplace)$. This mechanism is a centralized Differential Privacy(DP) mechanism. It realizes ε-Differential Privacy protection by adding random noise that obeys the Laplace distribution to the exact query result. We define the Laplace mechanism as follows: for a query function $f : D \to R^d$, where D is a data set. R_d is a d-dimensional real number vector, which is the return result of the query function. In any pair of adjacent data sets D and $D_{\prime}$, their sensitivity Δf is defined by Eq. (23), then the Eq. (22) provides ε-Differential Privacy protection, where Y is random noise.

$$M(D) = F(D) + Y, Y \sim Lap(\Delta f/\varepsilon) \tag{22}$$

$$\Delta f = \max \|f(D) - f(D')\|_1 \tag{23}$$

where $\|f(D)-f(D')\|_1$ represents the Manhattan distance between $f(D)$ and $f(D')$. Y obeys the Laplacian distribution with the scale parameter $\Delta f/\varepsilon$.

(2) $\varepsilon - Differential\ Privacy\ mechanism\ (Gaussian)$. This mechanism is also a centralized Differential Privacy mechanism. It implements ε-Differential Privacy protection by adding random noise that obeys the Gaussian distribution $N(\mu, \sigma^2)$ to the exact query result. $\mu = \sum_{i=1}^{n} x_i p_i$, usually let μ be 0. It conforms to the standard Gaussian distribution, $\sigma = \Delta f/\varepsilon$. Gaussian noise is a statistical noise whose probability density function obeys a normal distribution and its calculation is shown by Eq. (24).

$$Q(x) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x-\mu)^2}{2\sigma^2}) \tag{24}$$

(3) $\varepsilon - Localized\ Differential\ Privacy$. This mechanism is similar to the idea of centralized Differential Privacy, but it is a decentralized Differential Privacy mechanism. Task participants randomly respond to disturbances on personal data locally to achieve the effect of Differential Privacy protection.

(4) $k - anonymity$. The $k - anonymity$ mechanism uses generalization and hiding technology. It enables task participants to have k location data information at a certain moment. Each location record has at least the same quasi-identifier attribute value as the other $k - 1$ records. Therefore, the probability that the attacker learns the true location is reduced to $1/k$, which can effectively reduce the risk of privacy leakage.

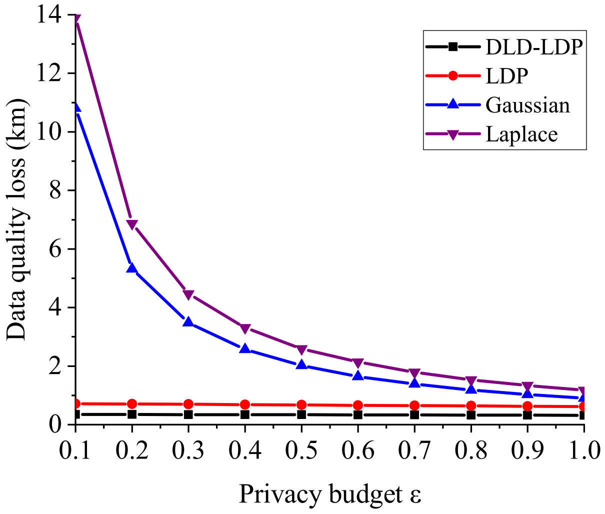

Fig. 2. The impact of privacy budget ε on data quality loss.

5.2. Location privacy protection experiment

In the location privacy protection experiment, a real New York taxi GPS location data set is used for comparison experiments and 500 locations were randomly selected. In the case of ensuring the same level of privacy protection, the level of data quality loss (DQL) is an important index for evaluating the superiority of a privacy protection algorithm.

ε-DP (Laplace/Gaussian) and ε-LDP mechanisms usually a single location data input and a single processed location data output. Therefore, the calculation method of DQL is shown by Eq. (25).

$$DQL(DP/LDP) = \sum_{i=1}^{I} d(l_i, l_i') \tag{25}$$

where I represents the total number of participants in the task, $d(l_i, l_i')$ represents the Euclidean distance between the real location l_i and the output location l_i'.

The $k-anonymity$ mechanism usually returns $k-1$ processed location information for a single location data input. For this reason, the DQL for defining this type of mechanism is shown by Eq. (26).

$$DQL(k - anonymity) = \sum_{i=1}^{I} \sum_{j=1}^{k} d(l_i, l_{i_j}) \tag{26}$$

where I represents the total number of task participants, $k-1$ represents the number of interference locations of each task participant. $d(l_i, l_{i_j})$ represents the Euclidean distance between the real location l_i and any interference location l_{i_j}.

It can be seen from Fig. 2 that although the DQL of each algorithm gradually decreases as the privacy budget ε increases. The DQL of the DLD-LDP algorithm is always lower than the DQL of other algorithms.

Fig. 3 shows the impact of different levels of privacy protection on the DQL. The 100 task participants are randomly selected to process their locations with different levels of privacy protection. It can be seen that the DQL of the DLD-LDP algorithm is always lower than other algorithms.

Fig. 4 shows the impact of the number of task participants on DQL when the privacy protection level is 75%. It can be seen that as the number of task participants increases, the DQL caused by each algorithm also increases steadily. But regardless of the number of task participants, the DQL of the DLD-LDP algorithm is always lower than other algorithms.

Fig. 5 shows the time required for data processing by different algorithms when the privacy protection level is 75%. Regardless of the

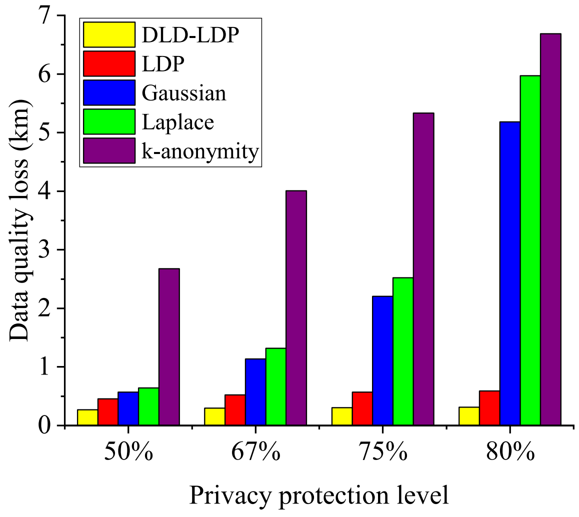

Fig. 3. The impact of privacy protection level on data quality loss.

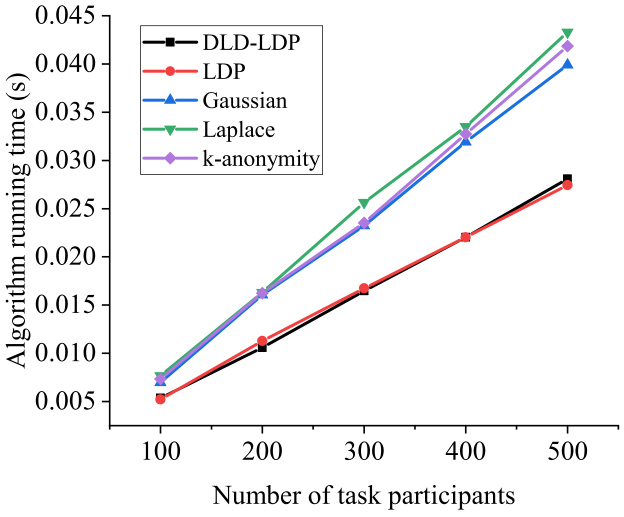

Fig. 5. The running time consumption of data processing by different algorithms.

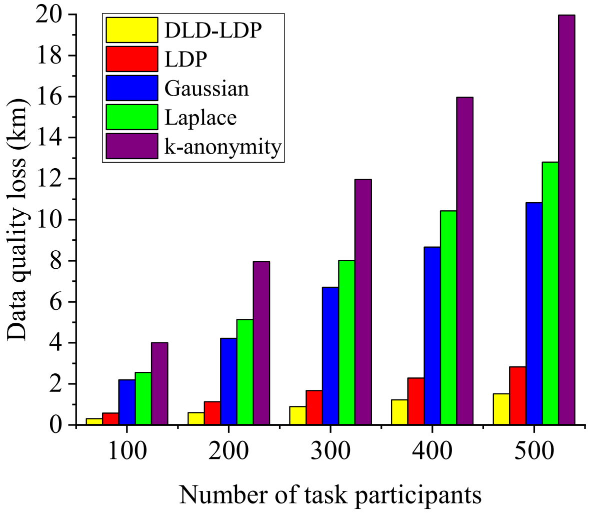

Fig. 4. The impact of task participants on data quality loss.

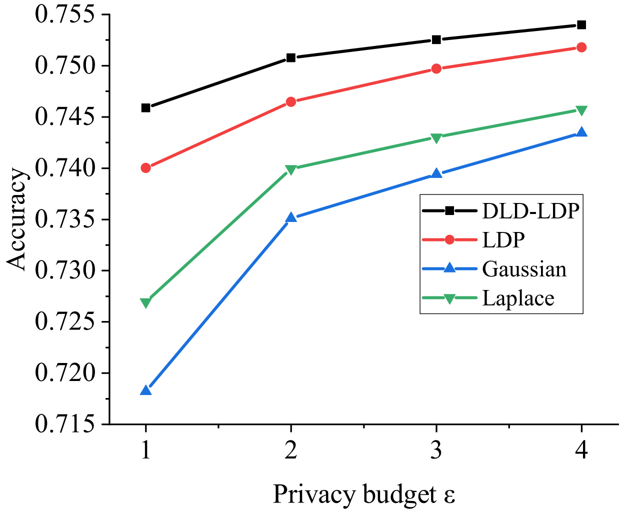

Fig. 6. The influence of privacy budget ε for accuracy.

number of task participants, the time required for DLD-LDP and LDP is always the lowest. The main reasons for the results shown in Figs. 3–5 are discussed as follows. k-anonymity algorithm will produce $k-1$ relatively similar data when processing the data, resulting in high DQL and time consumption. Centralized Differential Privacy (Laplace/Gaussian) is the processing of privacy protection by adding noise. So it will also cause relatively high DQL and time consumption. Both DLD-LDP and LDP implement data privacy protection processing through random responses. So the DQL and time consumption caused are relatively low. DLD-LDP performs data protection processing by locally perturbing the data array multiple times. The total number of perturbations is the same as that of LDP, so the algorithm requires a similar time. However, the changes in the array elements finally obtained by the DLD-LDP algorithm are relatively less than that of the LDP algorithm. So it has lower DQL under the same privacy protection level.

5.3. Data privacy protection experiment

According to the evaluation of data privacy protection, we use real text, video, audio data modalities data sets to conduct data privacy protection experiments. During the experiment, the default global epoch is 1, the local epoch is 8, the learning rate is 0.001, the batch size is 24, the privacy budget ε is 2 and the number of task participants is 4. In the case of ensuring the same level of privacy protection, the accuracy is an important index to measure the superiority of the model. So we take $Accuracy$ as one of the criteria to evaluate our algorithm.

Because the privacy budget ε has an important impact on the level of data privacy protection, experiments are conducted on the impact of different privacy budgets on the accuracy of the model. Fig. 6 shows the impact of different privacy budgets ε on model accuracy. It could be seen that as the privacy budget ε increases, the accuracy of the models obtained by each algorithm gradually tends to be similar. However, the accuracy of DLD-LDP is always higher than other algorithms. It is because that the data quality required to train the model has an important impact on the accuracy and the DLD-LDP algorithm cause lower data quality loss compared to other algorithms. So a relatively higher-precision model is finally obtained. As the privacy budget ε increases, the privacy protection capability will gradually decrease. Meanwhile, the accuracy obtained by each algorithm gradually tends to a similar value.

Batch size is an important parameter in machine learning, it represents the batch size of the divided data set D. Fig. 7 shows the accuracy obtained by each algorithm under different batch sizes. It could be seen that as the batch size increases, the accuracy obtained by each

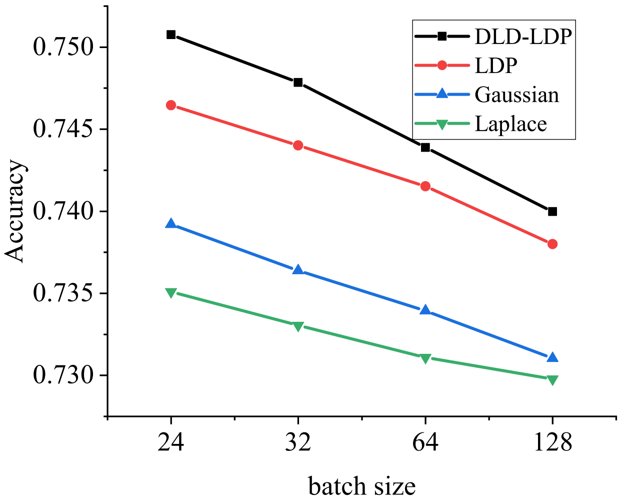

Fig. 7. The influence of batch size for accuracy.

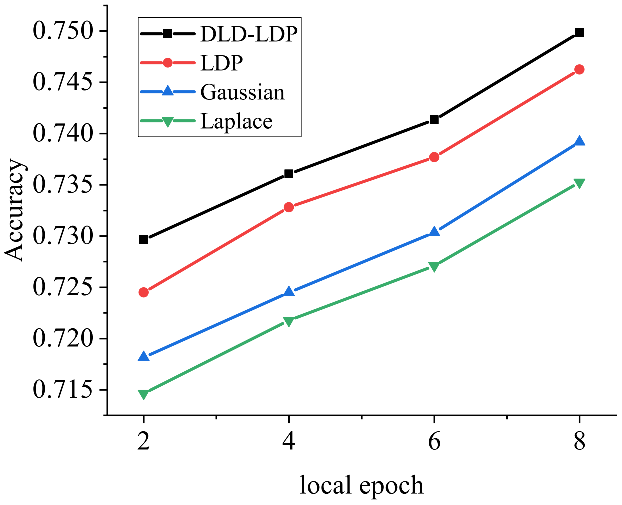

Fig. 8. The impact of local epoch on model accuracy.

algorithm gradually decreases. However, the accuracy obtained by the DLD-LDP algorithm is still always higher than that of other algorithms. This is because as the number of divided batches increases, the higher the data quality loss caused during the division process, the lower the accuracy. The data quality loss of DLD-LDP is lower than other algorithms under the same batch size, so the final accuracy is higher than other algorithms.

Local epoch and global epoch are important parameters in the federated learning system. Figs. 8 and 9 respectively show the influence of local epoch and global epoch on the accuracy obtained by the system algorithm.

From Fig. 8, it could be seen that as the local epoch increases, the model of each algorithm gradually increases. This is because the more local epoch training data is, the more comprehensive the model is, which makes the model be more accurate. Fig. 9 shows that as the global epoch increases, the model of each algorithm gradually increases. Because as the number of epochs increases, the number of weight update iterations in the neural network also increases. It promotes the accuracy of the model to gradually increase until it finally reaches the fitting state. It can be seen from Figs. 8 and 9 that no matter what the values of the local epoch and global epoch are. The accuracy of the model obtained by the DLD-LDP algorithm is always higher than that of other algorithms. This is because the higher the data quality

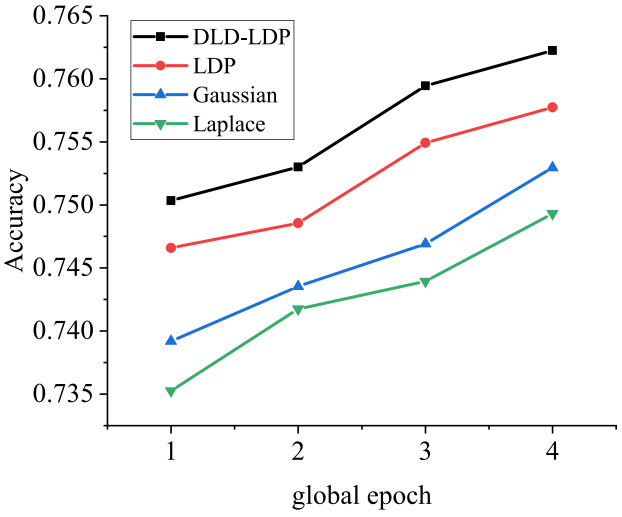

Fig. 9. The impact of global epoch on model accuracy.

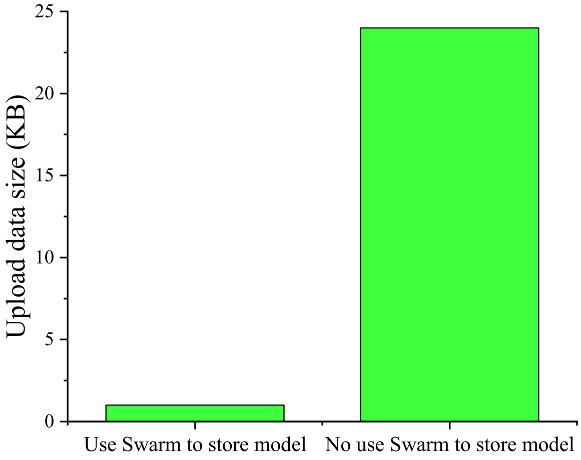

Fig. 10. The effect of Swarm on the size of uploaded data.

used for training, the higher the accuracy of the final model under the same other conditions. DLD-LDP causes the lowest data quality loss under the same privacy budget, resulting in relatively high accuracy of the final model.

Fig. 10 shows the impact of whether u_i uses Swarm to store w_i data on the size of the data finally uploaded to the Blockchain. It can be seen from Fig. 10 that the amount of data uploaded to the Blockchain by the former is significantly smaller than the latter. Fig. 11 shows the impact of whether the u_i uses Swarm to store w_i data on the time required to generate the Merkle tree. It can be seen from Fig. 11 that the time required for the former to generate a Merkle tree is also significantly lower than the latter. This is because the former uploads only the Swarm ID address SL_i to the Blockchain and the latter must upload the complete training model of the task participants w_i.

5.4. Reputation calculation experiment

The experiment used the real public comment (2018) data set to compare reputation calculation. The credit fluctuation factor $\alpha = 0.25$. The main comparison algorithms are Ebbinghaus [67], Yan [68], Avg, Random. The Avg algorithm used the average of all historical service quality scores of task participants as the current reputation value. The Random algorithm randomly selects task participants to perform the task. The choice of task participants is an important index that

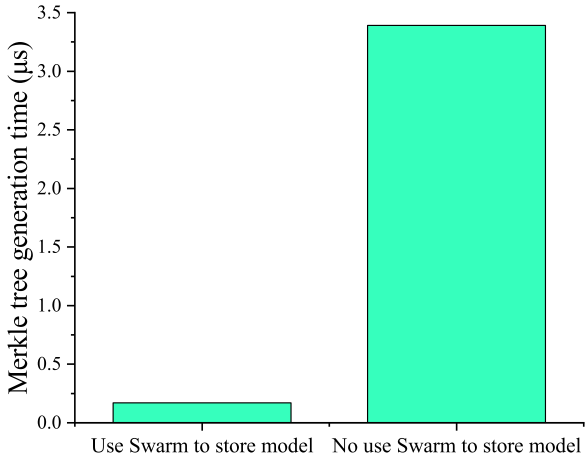

Fig. 11. The effect of Swarm on the generation time of Merkle tree.

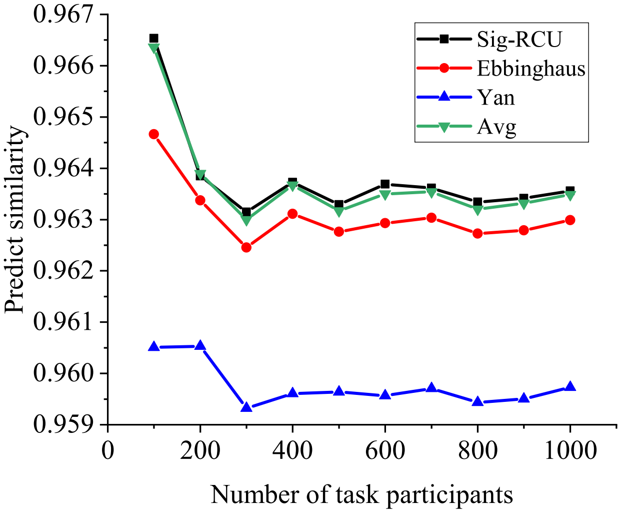

Fig. 12. The similarity between the predicted results of each algorithm and the real results (PS) when there is no newly registered task candidate participant.

affects the service quality of the entire task. Their reputations also an important index for measuring selectivity. This paper is defined PS is the similarity between the results of the reputation algorithm and the data service quality of the real task participants. The higher the similarity, the more accurate the prediction of the algorithm. This paper defined $ARSQ$ the average real service quality as the task participants selected by the reputation calculation method. The calculation methods are shown in Eqs. (27) and (28).

$$PS = \frac{1}{N}\sum_{i=1}^{N}\frac{S-|P_i-T_i|}{S} \tag{27}$$

$$ARSQ = \frac{1}{I}\sum_{i=1}^{I}T'_i, T' \leftarrow maxSort(P) \tag{28}$$

where S represents the highest score of the service quality standard of the task participant. Let $S = 5$, that is, the service quality score of the task participant varies from 0 to 5. N represents the number of task candidate participants. I represents the number of task participants finally selected. P_i represents the reputation value of task participants obtained by the reputation calculation algorithm. T_i represents the real service quality of the task participants' current task. $T' \leftarrow maxSort(P)$ indicates that all task candidate participants are selected in descending order according to their reputation value P_i. T'_i indicates the real service quality of the selected task participants.

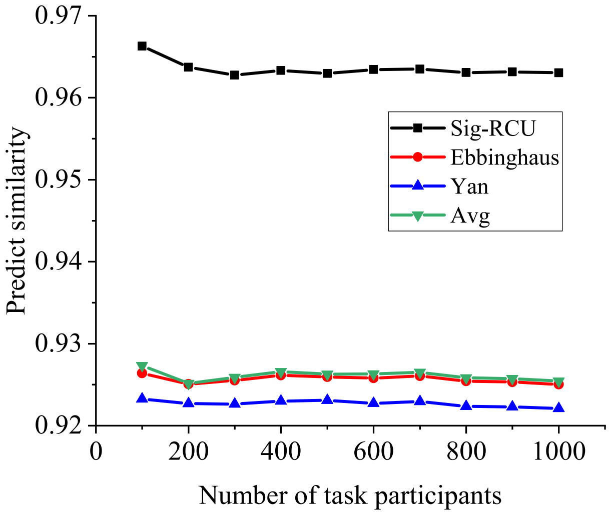

Fig. 13. The similarity between the predicted results of each algorithm and the real results (PS) in the presence of newly registered task candidate participants.

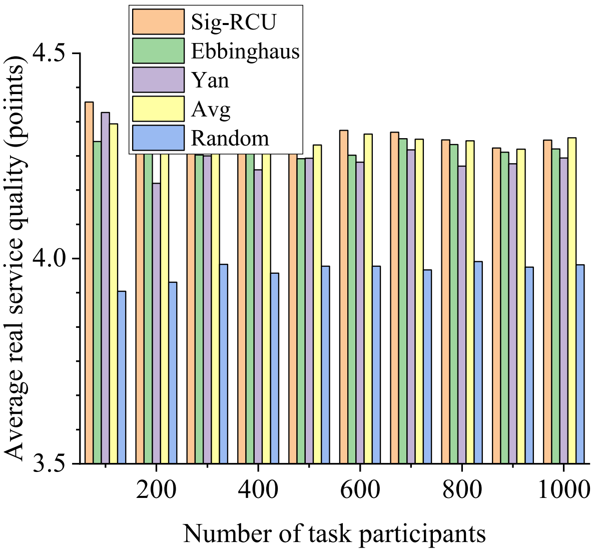

Fig. 14. The average real service quality (ARSQ) of task participants selected by each algorithm without newly registered task candidate participants.

The experiment sets the number of task candidate participants from 100 to 1000. Newly registered task candidate participants account for 5% of the total number of task candidate participants. The final number of task participants selected by the system is 25% of the total number of task candidate participants.

It can be seen from Figs. 12 and 13 that regardless of whether there is a newly registered task candidate participant. The PS of the Sigmid-RCU algorithm is higher than that of other algorithms, and it has better adaptability than other algorithms. This is because the Sigmid-RCU algorithm not only considers the reputation calculation problem of newly registered task candidate participants, but also fully considers the stability of the service quality of task candidate participants. The PS of 100 task candidate participants in Figs. 12 and 13 are relatively high. This is because the historical service quality of these task candidate participants is more in line with the calculation rules of each algorithm.

It can be seen from Figs. 14 and 15 that regardless of whether there is a newly registered task candidate participant. The ARSQ of the task participant selected by the Sigmid-RCU algorithm is relatively higher than that of other algorithms. This is because the higher the ARSQ of the algorithm, the higher the possibility of selecting task participants

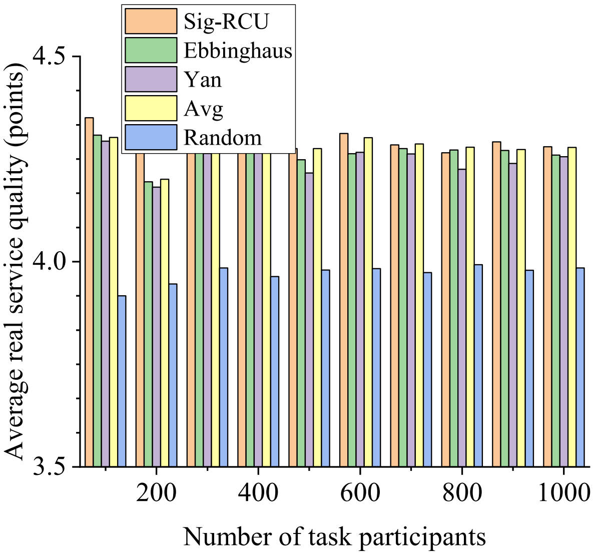

Fig. 15. The average real service quality (ARSQ) of task participants selected by each algorithm in the presence of newly registered task candidate participants.

with high service quality through the calculated reputation value. It leads to the higher the ARSQ of the algorithm.

6. Conclusion

In this paper, an MCS federated learning system based on Blockchain and edge computing is proposed for the privacy protection problem in MCS. Specifically, this article uses federated learning as the framework of the MCS system. Both data privacy and location privacy are independently protected locally by task participants using the DLD-LDP algorithm. Since the sensed data exists in multiple modalities (text, video, audio, etc.), this paper used the MulT method to merge the multi-modal data before subsequent operations. This system offloads model training and aggregation tasks to the edge server to solve the problem of insufficient computing on the mobile terminal and the data processing efficiency of the crowdsourcing platform. Utilize the Blockchain to distribute tasks and collection models in a distributed manner to solve the privacy leakage problem caused by untrusted third parties. This paper designs a reputation calculation method (Sig-RCU) to improve the service quality of task participants. Finally, through experiments on real data sets, the effectiveness and adaptation of the proposed MCS federated learning system are verified.

In the future, we will design an effective incentive mechanism to improve task participants' enthusiasm and service quality. We will also combine other advanced technologies with the privacy protection mechanism in MCS to further improve privacy protection capabilities and reduce data quality loss.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62073201, Grant 62072392, Grant 62172351, the Youth Innovation Science and Technology Support Program of Shandong Provincial under Grant 2021KJ080, the Key projects of Shandong Natural Science Foundation under Grant ZR2020K-F019, the Major scientific and technological innovation projects of Shandong Province under Grant 2019JZZY020131, Yantai Science and Technology Innovation Development Plan Project under Grant 2021YT06000645. All authors have seen and approved the final version of the manuscript being submitted.

References

[1] Z. Liang, J. Du, C. Li, Abstractive social media text summarization using selective reinforced Seq2Seq attention model, Neurocomputing 410 (2020) 432–440.

[2] F. Kou, J. Du, C. Yang, Y. Shi, W. Cui, M. Liang, Y. Geng, Hashtag recommendation based on multi-features of microblogs, J. Comput. Sci. Tech. 33 (4) (2018) 711–726.

[3] Z. Lu, Y. Wang, Y. Li, X. Tong, C. Yu, Data-driven many-objective crowd worker selection for mobile crowdsourcing in industrial IoT, IEEE Trans. Ind. Inf. 99 (2021) 1.

[4] Y. Wang, Z. Cai, Z. Zhan, B. Zhao, L. Qi, Walrasian equilibrium-based multi-objective optimization for task allocation in mobile crowdsourcing, IEEE Trans. Comput. Soc. Syst. 99 (2020) 1–14.

[5] Y. Wang, Y. Gao, Y. Li, X. Tong, A worker-selection incentive mechanism for optimizing platform-centric mobile crowdsourcing systems, Comput. Netw. 107 (2020) 107144.

[6] X. Zheng, Z. Cai, Privacy-preserved data sharing towards multiple parties in industrial IoTs, IEEE J. Sel. Areas Commun. 99 (2020) 1.

[7] Z. Cai, X. Zheng, J. Yu, A differential-private framework for urban traffic flows estimation via taxi companies, IEEE Trans. Ind. Inf. 99 (2019) 1.

[8] Z. Cai, X. Zheng, A private and efficient mechanism for data uploading in smart cyber-physical systems, IEEE Trans. Netw. Sci. Eng. 99 (2018) 1.

[9] Z. Cai, Z. Xiong, H. Xu, P. Wang, Y. Pan, Generative adversarial networks: A survey toward private and secure applications, ACM Comput. Surv. 54 (6) (2021) 1–38.

[10] S. Shaham, M. Ding, B. Liu, S. Dang, Z. Lin, J. Li, Privacy preserving location data publishing: A machine learning approach, IEEE Trans. Knowl. Data Eng. 99 (2020) 1.

[11] D. Tao, T. Wu, S. Zhu, M. Guizani, Privacy protection-based incentive mechanism for mobile crowdsensing, Comput. Commun. 156 (2020) 201–210.

[12] J. An, H. Yang, X. Gui, W. Zhang, R. Gui, J. Kang, TCNS: Node selection with privacy protection in crowdsensing based on twice consensuses of blockchain, IEEE Trans. Netw. Serv. Manag. 16 (3) (2019) 1255–1267.

[13] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, PPFA: Privacy preserving fog-enabled aggregation in smart grid, IEEE Trans. Ind. Inf. 14 (8) (2018) 3733–3744.

[14] P. Arachchige, P. Bertok, I. Khalil, D. Liu, M. Atiquzzaman, A trustworthy privacy preserving framework for machine learning in industrial IoT systems, IEEE Trans. Ind. Inf. 99 (2020) 1.

[15] Y. Qi, M. Hossain, J. Nie, X. Li, Privacy-preserving blockchain-based federated learning for traffic flow prediction, Future Gener. Comput. Syst. 117 (2946) (2021) 328–337.

[16] Y. Tsai, S. Bai, P. Liang, J. Kolter, R. Salakhutdinov, Multimodal transformer for unaligned multimodal language sequences, 2019, arXiv.

[17] Y. Yang, J. Du, Y. Ping, Ontology-based intelligent information retrieval system, J. Softw. 26 (7) (2015) 1675–1687.

[18] W. Wang, Y. Wang, P. Duan, T. Liu, X. Tong, Z. Cai, A triple real-time trajectory privacy protection mechanism based on edge computing and blockchain in mobile crowdsourcing, IEEE Trans. Mob. Comput. (01) (2022) 1–18.

[19] F. Kou, J. Du, Z. Lin, M. Liang, H. Li, L. Shi, C. Yang, A semantic modeling method for social network short text based on spatial and temporal characteristics, J. Comput. Sci. 28 (2018) 281–293.

[20] Y. Wang, M. Wang, Q. Meng, X. Tong, Z. Cai, New crowd sensing computing in space-air-ground integrated networks, in: IEEE SAGC 2021, 2021.

[21] Z. Cai, Z. He, X. Guan, Y. Li, Collective data-sanitization for preventing sensitive information inference attacks in social networks, IEEE Trans. Dependable Secure Comput. 99 (2018) 1.

[22] Z. Liu, T. Li, V. Smith, V. Sekar, Enhancing the privacy of federated learning with sketching, 2019, arXiv.

[23] L. Melis, C. Song, E. Cristofaro, V. Shmatikov, Exploiting unintended feature leakage in collaborative learning, in: 2019 IEEE Symposium on Security and Privacy (SP), 2019.

[24] B. Hitaj, G. Ateniese, F. Perez-Cruz, Deep models under the GAN: information leakage from collaborative deep learning, in: 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 603–618.

[25] M. Hao, H. Li, X. Luo, G. Xu, S. Liu, Efficient and privacy-enhanced federated learning for industrial artificial intelligence, IEEE Trans. Ind. Inf. 16 (10) (2020) 6532–6542.

[26] Z. Li, J. Liu, J. Hao, H. Wang, M. Xian, CrowdSFL: A secure crowd computing framework based on blockchain and federated learning, Electronics 9 (5) (2020) 773.

[27] X. Shen, Q. Pei, X. Liu, Survey of block chain, Chin. J. Netw. Inf. Secur. 1 (11) (2016) 11–20.

[28] J. Weng, J. Weng, J. Zhang, M. Li, W. Luo, DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive, IEEE Trans. Dependable Secure Comput. 18 (5) (2021) 2438–2455.

[29] S. Awan, F. Li, B. Luo, M. Liu, Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain, in: The 2019 ACM SIGSAC Conference, 2019.

[30] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, IEEE Trans. Ind. Inf. 16 (6) (2020) 4177–4186.

[31] S. Xia, Z. Yao, Y. Li, S. Mao, Online distributed offloading and computing resource management with energy harvesting for heterogeneous MEC-enabled IoT, IEEE Trans. Wireless Commun. 20 (10) (2021) 6743–6757.

[32] F. Wu, T. Luo, Crowdprivacy: Publish more useful data with less privacy exposure in crowdsourced location-based services, ACM Trans. Priv. Secur. 23 (6) (2020) 1–25.

[33] Y. Li, H. Ma, L. Wang, S. Mao, G. Wang, Optimized content caching and user association for edge computing in densely deployed heterogeneous networks, IEEE Trans. Mob. Comput. 21 (6) (2022) 2130–2142.

[34] T. Zhu, T. Shi, J. Li, Z. Cai, X. Zhou, Task scheduling in deadline-aware mobile edge computing systems, IEEE Internet Things J. 6 (3) (2019) 4854–4866.

[35] D. Yhab, E. Jn, B. Bn, C. Flb, D. Xss, Privbus: A privacy-enhanced crowdsourced bus service via fog computing, J. Parallel Distrib. Comput. 135 (2020) 156–168.

[36] Y. Zhao, J. Zhao, L. Jiang, R. Tan, Y. Liu, Privacy-preserving blockchain-based federated learning for IoT devices, IEEE Internet Things J. 99 (2020) 1.

[37] B. Luo, X. Li, J. Weng, J. Guo, J. Ma, Blockchain enabled trust-based location privacy protection scheme in VANET, IEEE Trans. Veh. Technol. 69 (2) (2020) 2034–2048.

[38] Z. Cai, Z. He, Trading private range counting over big IoT data, in: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019.

[39] T. Liu, Y. Wang, Y. Li, X. Tong, Privacy protection based on stream cipher for spatio-temporal data in IoT, IEEE Internet Things J. 99 (2020) 1.

[40] Y. Wang, Y. Luo, Q. Yu, Q. Liu, W. Chen, Trajectory privacy-preserving method based on information entropy suppression, J. Comput. Appl. 38 (11) (2018) 3252–3257.

[41] L. Ma, X. Liu, Q. Pei, X. Yong, Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing, IEEE Trans. Serv. Comput. 12 (5) (2019) 786–799.

[42] S. Hong, H. Kim, Qoe-aware computation offloading to capture energy-latency-pricing tradeoff in mobile clouds, IEEE Trans. Mob. Comput. 18 (9) (2019) 2174–2189.

[43] S. Li, G. Zhang, A differentially private data aggregation method based on worker partition and location obfuscation for mobile crowdsensing, Comput. Mater. Contin. 63 (1) (2020) 223–241.

[44] T. Zhou, Z. Cai, Q. Xia, B. Xiao, Location privacy-preserving data recovery for moblie crowdsensing, in: Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, Vol. 2, 2018, pp. 1–23.

[45] G. Ping, X. Ye, A survey of research on network attack model, J. Inf. Secur. Res. 6 (12) (2020) 1058–1067.

[46] L. Lyu, J. Yu, K. Nandakumar, Y. Li, K. Ng, Towards fair and privacy-preserving federated deep models, IEEE Trans. Parallel Distrib. Syst. 31 (11) (2020) 2524–2541.

[47] K. Arun, K. Rajendra, A layer-wise security analysis for internet of things network: Challenges and countermeasures, Int. J. Manage. IT Eng. 9 (6) (2019) 118–133.

[48] L. Zhang, G. Ding, Q. Wu, Y. Zou, Z. Han, J. Wang, Byzantine attack and defense in cognitive radio networks: A survey, IEEE Commun. Surv. Tutor. 17 (3) (2015) 1342–1363.

[49] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, B. Li, Manipulating machine learning: Poisoning attacks and countermeasures for regression learning, in: Proceedings of 2018IEEE Symposium on Security and Privacy(SP), 2018, pp. 19–35.

[50] Y. He, X. Hu, G. He, K. Chen, Privacy and security issues in machine learning systems: a survey, J. Comput. Res. Dev. 56 (10) (2019) 2049–2070.

[51] K. Wang, J. Liu, C. Li, Y. Zhao, H. Lyu, P. Li, B. Liu, A survey on threats to federated learning, J. Inf. Secur. Res. 8 (3) (2022) 223–234.

[52] S. Azhar, S. Chang, Y. Liu, Y. Tao, G. Liu, Privacy-preserving and utility-aware participant selection for mobile crowd sensing, Mob. Netw. Appl. (9) (2020) 290–302.

[53] J. Douceur, The sybil attack, Peer-to-Peer Syst. (2002) 251–260.

[54] M. Alazab, S. Venkatraman, Detecting malicious behaviour using supervised learning algorithms of the function calls, IEEE Trans. Parallel Distrib. Syst. (2013) 90–109.

[55] Q. Ye, X. Meng, M. Zhu, Z. Huo, Survey on local differential privacy, J. Softw. 29 (7) (2018) 1981–2005.

[56] J. Wang, Y. Wang, G. Zhao, Z. Zhao, Location protection method for mobile crowd sensing based on local differential privacy preference, Peer-to-Peer Netw. Appl. (1) (2019) 1–13.

[57] S. Warner, Randomized response: a survey technique for eliminating evasive answer bias, Publ. Amer. Statist. Assoc. 60 (309) (1965) 63–69.

[58] Z. Sun, Y. Wang, Z. Cai, T. Liu, N. Jiang, A twotage privacy protection mechanism based on blockchain in mobile crowdsourcing, Int. J. Intell. Syst. 36 (5) (2021) 2058–2080.

[59] R. Lan, Y. Zhou, Z. Liu, X. Luo, Prior knowledge-based probabilistic collaborative representation for visual recognition, IEEE Trans. Cybern. 60 (4) (2018) 1498–1508.

[60] Q. Yang, Y. Liu, T. Chen, Y. Tong, Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. (TIST) 10 (2) (2019) 1–19.

[61] Y. Wang, Z. Cai, Z.H. Zhan, Y.J. Gong, X. Tong, An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing, IEEE Trans. Comput. Soc. Syst. 6 (3) (2019) 414–429.

[62] X. Yang, Y. Chen, X. Chen, Effective scheme against 51% attack on proof-of-work blockchain with history weighted information, in: 2019 IEEE International Conference on Blockchain (Blockchain), 2019.

[63] A. Alphonse, M. Starvin, Blockchain and internet of things: An overview, in: Handbook of Research on Blockchain Technology, 2020, pp. 295–322.

[64] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.

[65] Facial expression analysis, in: IMotions, 2017.

[66] G. DeGottex, J. Kane, T. Drugman, T. Raitio, S. Scherer, COVAREP: A collaborative voice analysis repository for speech technologies, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014.

[67] G. Zhu, L. Zhou, Hybrid recommendation based on forgetting curve and domain nearest neighbor, J. Manag. Sci. China. 15 (2012) 55–64.

[68] J. Yan, K. Shaoping, Y. Chu, Reputation model of crowdsourcing workers based on active degree, J. Comput. Appl. 37 (7) (2017) 2039–2043.

Weilong Wang received the Bachelor degree in the School of Information Science and Engineering, Shandong Agricultural Engineering College. He is currently pursuing the Master degree in the School of Computer and Control Engineering, Yantai University. His research interests are mobile crowdsourcing and privacy protection.

Yingjie Wang received the Ph.D. degree in College of Computer Science and Technology from Harbin Engineering University. She visited Georgia State University from 2013/09 to 2014/09 as a visiting scholar. Dr. Wang is currently an Associate Professor in the School of Computer and Control Engineering at Yantai University. She is a Post-doc in South China University of Technology. Her research interests are mobile crowdsourcing, privacy protection and trust computing. She has published more than 50 papers in well known journals and conferences in her research field, which includes an ESI high cited paper. In addition, she has presided 1 National Natural Science Foundation of China project, 2 China Postdoctoral Science Foundation projects. Dr. Wang obtained the Shandong Province Artificial Intelligence Outstanding Youth Award.

Yan Huang is currently an Assistant Professor in the Department of Software Enportraitgineering & Game Development at Kennesaw State University (KSU). Dr. Huang received his Ph.D. degree in the Department of Computing Science at Georgia State University. He is broadly interested in privacy and security, with particular emphasis on deep learning aided privacy protection solutions and cybersecurity challenges in the IoT environment. Dr. Huang is/was a TPC member for many conferences, including IEEE GLOBECOM 2020, IEEE Blockchain 2019, COCOA 2019. He is/was Technical Track Chair of CyberSciTech 2021/2020 and Editor of WCMC.

Chunxiao Mu received the Master degree in School of Software Engineering Major from Shanghai Jiaotong University. His research interests are marine information system. He is now in the school of computer and control engineering of Yantai University, and is the director of the marine engineering equipment and intelligent technology Key Laboratory Yantai.

Zice Sun received the Bachelor degree in the School of Computer and Control Engineering, Yantai University. He is currently pursuing the Master degree in the School of Computer and Control Engineering, Yantai University. His research interests are mobile crowdsourcing and blockchain.

Xiangrong Tong received the Ph.D. degree in School of Computer and Information Technology from Beijing Jiaotong University. Currently, he is a Full Professor of Yantai University. His research interests are computer science, intelligent information processing and social networks. He has published more than 50 papers in well known journals and conferences. In addition, he has presided and joined 3 national projects and 3 provincial projects.

Zhipeng Cai received his Ph.D. and M.S. degrees in the Department of Computing Science at University of Alberta, and B.S. degree from Beijing Institute of Technology. Dr. Cai is currently an Associate Professor in the Department of Computer Science at Georgia State University. Dr. Cai's research areas focus on Networking, Privacy and Big data. Dr. Cai is the recipient of an NSF CAREER Award. Dr. Cai is now a Steering Committee Co-Chair for WASA. He is an editor/guest editor for Algorithmica, Theoretical Computer Science, Journal of Combinatorial Optimization, IEEE/ACM Transactions on Computational Biology and Bioinformatics. He is a senior member of the IEEE.