




A trusted recommendation scheme for privacy protection based on federated learning

Yichuan Wang¹ · Yuying Tian¹ · Xinyue Yin¹ · Xinhong Hei¹ 

Received: 15 June 2020 / Accepted: 31 October 2020 / Published online: 23 November 2020
© China Computer Federation (CCF) 2020

Abstract

With the convergence of the era of global news and the era of big data, the daily amount of news sent to the world is exploding. Users also face the problem of information overloads when they get massive information, which leads to how cloud servers push personalized data to users among massive data have become the focus of news companies. In order to obtain the push accuracy, the traditional recommendation system often makes deep mining of users' privacy data, which makes users' privacy cannot be guaranteed. In order to solve the above problems, this paper proposes a collaborative filtering algorithm recommendation system based on federated learning on end-edge-cloud. The exposure of data privacy was further prevented by adding Laplace noise to the training model through differential privacy technology. Finally, the training model and recommendation information is stored to the blockchain network to provide permanent storage, evidence chain and real-time traceability services. On the premise of protecting data privacy, this system provides cloud server with solutions to alleviate computing pressure, bandwidth pressure and improve news push accuracy through end-edge-cloud distributed learning.

Keywords Federated learning · Blockchain · Differential privacy · Recommendation system

1 Introduction

The emergence of personalized recommendation technology is to solve the problem of accurate distribution of information and help users find the information they want. Personalized recommendation technology plays an important role in reducing user cost, improving work efficiency, optimizing content and improving background performance. Recommender system is a kind of information filtering tool, which can present the most relevant content of interest to specific users by using the user characteristics and habits of the whole cluster. With the explosive growth of information on the Internet, it is impossible for users to browse all the content. The recommendation system can help users find the

real interesting content according to their taste by analyzing the data.

However, the recommendation system is based on the centralized processing mode of cloud computing, which inevitably consumes a lot of communication resources and causes a large time delay in the data transmission process; moreover, the physical limited communication resources (such as bandwidth) will limit the scalability of the system. The transmission of user data in the public network will also bring higher risk of data security and privacy disclosure. Therefore, this scheme introduces federated learning and blockchain to solve the above problems. Traditional machine learning uses centralized training method and requires that the training data be gathered to the data center in the cloud. But this centralized training method will involve the privacy of user data. For example, the mobile phone user's mobile terminal usually keeps the user's privacy sensitive data. In order to obtain a better machine learning model under this centralized training method, mobile phone users have to trade their privacy by sending the personal data stored in the mobile phone to the enterprise cloud. Federated learning can construct machine learning system without directly accessing training data. Data is kept in its original location, which helps to ensure privacy and reduce communication costs.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s42045-020-00045-8>) contains supplementary material, which is available to authorized users.

✉ Xinhong Hei
heixinhong@xaut.edu.cn

Yichuan Wang
chuan@xaut.edu.cn

¹ Xi'an University of Technology, Xi'an, China

Therefore, we propose a privacy preserving news recommendation scheme based on federated learning, which adds Laplace noise to the training model through differential privacy technology to further prevent data privacy exposure. Finally, the training model and recommendation news are saved to the licensed blockchain network to provide permanent preservation, evidence chain and real-time traceability services. In the experimental part of this paper, we have carried out relevant work on licensed blockchain storage, the accuracy and recall rate of collaborative filtering algorithm recommendation, and the usability of federated learning algorithm based on DP-SDG.

In summary, our main contributions are as follows:

1. We propose a blockchain enabled recommendation system based on Federated learning, which combines blockchain with federated learning, and provides security protection and technical support for the recommendation system by making use of their unforgeability and privacy.
2. We implement differential privacy by adding Laplace noise to the training model to further prevent data privacy exposure. Finally, the training model and recommended news are saved to the licensed blockchain network to achieve permanent preservation and real-time traceability service.
3. We theoretically analyzed the security and performance of the system model we designed, and evaluated the efficiency and possible effect of our model from related experiments.

The rest of the paper is organized as follows. The second part introduces the background knowledge and related work, and introduces the concept of federated learning, blockchain, content delivery network and end-edge-cloud orchestrated computing systems. In the third section, we propose the system architecture and network model. Then, the fourth section describes the collaborative filtering algorithm and the federated learning algorithm based on DP-SDG in detail. The fifth section analyzes the model storage efficiency and algorithm availability. Finally, the article is summarized in the sixth part.

2 Related work

2.1 Federated learning

Federated learning (Yang et al. 2019) provides a distributed machine learning framework built across multiple devices, which is divided into local training and global training. D transfers parameter weights to G and keeps sensitive data locally, thus achieving user privacy protection. Federated

learning consists of global server G and multiple local devices D , device $D = \{D_1, D_2, \dots, D_n\}, |D| = D_n$. In the first round of training, the global server transmits the initial training model M_0 to the member devices, and the t -th($t > 0$) training consists of N participant C_k ($k = 1, \dots, N$), $N \leq |D_n|$. C_k^t owns data set x_k , C_k^t following Stochastic Gradient Descent(SGD) algorithm training M_k^t to obtain local parameter weight $W_k^t \in R^n$. The global server G aggregates the participant C_k^t of this round of training by means of secure communication protocol, carries out FederatedAvg algorithm with W_k^t .

$$W_G^t = \frac{1}{N} \sum_{k=1}^N W_k^t \quad (1)$$

$$W_k^{(t+1)} = W_k^t - \lambda_t \nabla J(W_k^t, D_k)$$

∇J is loss function. G transfers the new model $M_G^{(t+1)}$ and global parameter weights W_G^t to $(t+1)$ -th training participants, a perfect model of training that requires multiple iterations of the process. λ_t is the learning rate, and the optimal value λ improves the accuracy of the model.

In our scheme, the idea of federated learning is applied to the recommendation process. Each user uses local data and global user news model to update the local user news model to achieve privacy protection in the recommendation process. A federated learning scheme of collaborative filters is proposed (Ammad-Ud-Din et al. 2019) to explore a personalized recommendation system based on implicit user feedback. Federated training can collaborate filters without loss of accuracy, while enhancing user privacy.

2.2 Blockchain choice

Blockchain (Min et al. 2017) is essentially a decentralized and distributed database. data in the blockchain is stored as blocks, and the blocks are chained together with an Hashed-link. That is, each block head contains the hash value of all transactions in the block and the hash value of the previous block head. Blockchain protects data transmission and access through cryptography. All blockchain nodes maintain the same ledger, and all data operations based on the blockchain are open, transparent, and permanently recorded Lundbaek et al. (2017). Blockchains can be divided into public, private and permissioned blockchains. Advantages of permissioned blockchain (Thomas and Ned 2016): (1) set channels, organizations, endorsement nodes, and extending executable business scope, such as node authentication, data transmission privacy, etc. (2) Set specific chaincodes and consensus mechanisms according to different business scenarios. (3) Less tokens and computation than a public blockchain, and a significant execution speed.

The mainstream alliance chain projects include Hyperledger Fabric (Androulaki et al. 2018) and FISCO BCOS. FISCO BCOS platform is a financial blockchain cooperation alliance. The FISCO BCOS platform eliminates the complexities of many functions of Ethereum and retains the core features of the blockchain. However, FISCO BCOS is mainly aimed at the financial sector and has a small application range. Hyperledger Fabric is an open source blockchain project for enterprise-level applications. The project is sponsored by the Linux Foundation. The goal is to help companies build private or alliance-licensed blockchain networks. Hyperledger Fabric's innovations in multi-channel, multi-ledger, authority management, consensus mechanism and other modules provide new ideas for the development of executable business on the blockchain. Therefore, in our solution, the consortium chain Hyperledger Fabric is used to save the training model and recommended news to the permissioned blockchain network to achieve permanent storage and real-time traceability services.

2.3 Content delivery network

Content Delivery Network (CDN) is a layer of intelligent virtual network built on top of the existing Internet. It consists of a main server and nodes deployed throughout the global distributed network Ahmed et al. (2017). It adds a new CACHE layer to publish the content of the website to the node closest to the user's network edge. Its purpose is to shorten the access delay, improve the response speed and website availability, solve the problem of uneven distribution of outlets, and distribute and store the content of the source site to all CDN nodes to enable users to access the data stored on the node closest to the user. This is different from the traditional method of all users accessing a server or a single node to store content. The client accesses a copy of the data near the client. Location is the key to the speed of content delivery. The farther the user is from the server, the longer it takes for the content to reach the user, and the greater the impact on the user experience Chen et al. (2018).

The function of CDN mainly has the following two aspects. For the first time, it can keep important content distributed to multiple distributed data centers. The data centers are globally network-wide in order to be closer to end users, so download and access speeds are faster. The second is that it uses server optimization based on content type to provide content to users most efficiently (Chen et al. 2019). In addition, CDN alleviate the traffic that provides services directly from the original infrastructure of content providers. At the same time, CDN can effectively resist DDoS attacks because CDN provides a large distributed server infrastructure to absorb the amount of attacks. The CDN's role in optimizing the network is mainly reflecting in solving the "first mile" problem on

the server side, optimizing the distribution of online hot content, alleviating or even eliminating the impact of the bottleneck of interconnection between different operators, reducing the pressure on the export bandwidth of the provinces, and the pressure on the backbone network.

2.4 End-edge-cloud orchestrated computing systems

Edge computing De Donno et al. (2019) is a new type of computing model that performs computing at the edge of the network. Edge of edge computing refers to any computing resource and network resource from the data source to the cloud computing center (Sun et al. 2019). Objects targeted by edge computing include upstream data from the Internet of Things and downstream data from cloud services. Edge computing allows terminal devices to migrate storage and computing tasks to network edge nodes, which can not only meet the expansion needs of computing power of terminal devices, but also effectively save the transmission link resources of computing tasks between terminal devices and cloud servers (De Donno et al. 2019).

At present, domestic and foreign scholars have done various researches on end-edge-cloud orchestrated computing systems. Ren et al. proposed transparent computing on the basis of edge computing (Ren et al. 2017). Its core is to separate storage and operation, and software and terminal. It uses cached "streaming" operations to restore computing to "unconscious, user-controllable" personalized services. It enables software services to flow between the cloud, edge, and lightweight devices, providing a seamless computing experience for end users, so that users do not know where the services come from. Peng et al. proposed a block-stream as a service loading model (Peng et al. 2018). By dividing the application into blocks, the client can dynamically apply for the required application block according to the demand, increasing flexibility to adapt to smaller devices. Sharma et al. (2018) proposed a distributed cloud architecture based on blockchain, which combines SDN and blockchain technology to achieve high performance and cost-effective computing. It has the advantages of being able to be accessed on demand, low cost and high reliability. Taking advantage of the commonalities between federated learning and edge computing, Lu et al. (2019) proposed a safe and robust scheme that combines federated learning and edge computing, and added local differential privacy to the scheme to protect the privacy of the local model. Al-Abbasi et al. (2019) combined edge computing with CDN and proposed a system model for video streaming. The components of the model include a centralized server, CDN node and edge cache closer to the terminal

3 System structures

3.1 Network model

Figure 1 is the network topology of this solution. From the perspective of the network topology, the edge server and the terminal equipment interact with each other through a master-slave structure. This is the idea of federal learning. The edge servers in the same CDN cluster adopt a non-master-slave structure for model interaction, and update the model between devices through point-to-point interaction of local model update parameters, and eventually converge to a unified set of model parameters.

The client is composed of terminal devices operable by users such as mobile phones, tablets, and computers. As the computing resources of user terminals become more and more powerful, model training is shifting from the cloud and data centers to the terminals. To protect user privacy, users can independently build their own machine learning models and upload model update parameters (original data that does not contain user privacy information) to the edge server.

The edge server consists of server nodes in the CDN cluster. The CDN node is a content-providing device for end users, which can cache static Web content and streaming media content to achieve the edge propagation and storage of content, so that users can access nearby based on geographic location. There is one SUPER-CDN node and multiple ordinary CDN nodes in a CDN cluster. The SUPER-CDN section has the function of an ordinary CDN

node and at the same time acts as a blockchain node to interact with the blockchain network.

The cloud server is composed of multiple powerful servers. The cloud server is responsible for data aggregation, global information analysis, massive data storage, and large-scale data calculation.

3.2 Threat model

In the current mainstream recommendation algorithms, some algorithms need to collect and analyze user behavior and identity information, such as user purchase records, information access records, geographic location, gender and age, and even social relations. The algorithm uses the collected information to build user profile, preference or network information table, so as to achieve accurate recommendation. In order to improve the accuracy of recommendation, users of recommendation algorithm will try their best to expand the number and scope of user information collection. These continuously strengthened and gradually accurate user profile information and behavior preference information, which are stored on the recommendation algorithm platform, may become the target of network attacks. Once the information is leaked or improperly used, it may bring huge security risks to users. Therefore, the introduction of federated learning combined with the existing recommendation system can improve the usability and security of the system.

Fig. 1 Network model

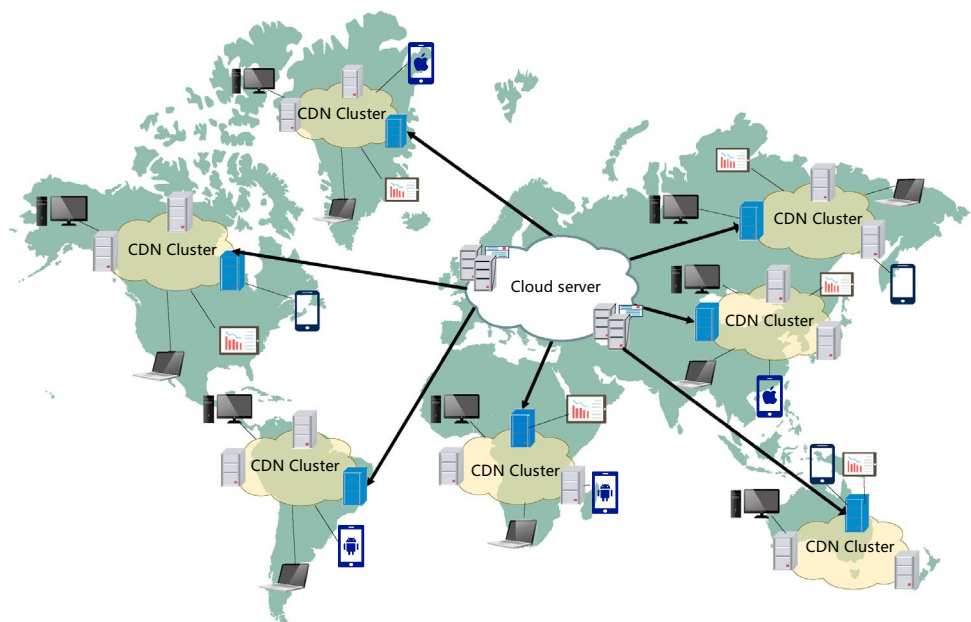
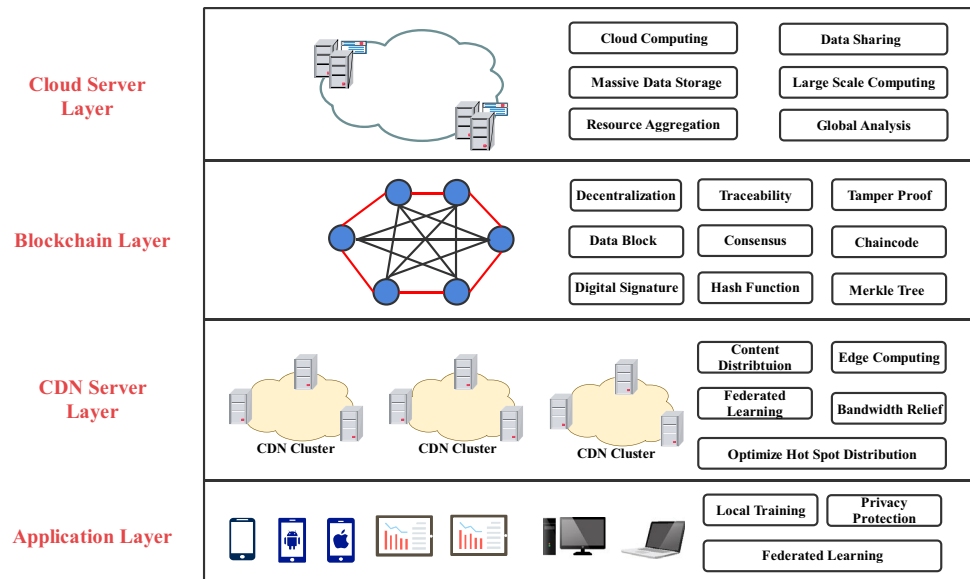


Fig. 2 Flow chart of the system

3.3 Scheme architecture

Figure 2 is an optimization scheme architecture diagram based on edge learning, including the application layer, edge service layer, blockchain layer and cloud service layer. In each CDN cluster in the edge service layer, there is a super node, SUPER-CDN, responsible for learning and interacting with the other three layers. SUPER-CDN has more computing power within the cluster, memory capacity, and can perform more functions than content distribution.

The application layer is the client's set C . The client and the edge service layer perform master-slave (MSFL) federated learning, that is, the super-CDN in the same LAN transfers the global model M_G to the client, and the client uses the local data set for training to obtain the local parameter weight W_k . The edge service layer SUPER-CDN aggregation multiple W_k within the LAN, after many iterations of training the new model. On the premise of not exposing users' sensitive data, distributed machine learning is completed to obtain users' behavioral intention.

CDN server layer is above the application. Non-master-slave distributed learning (NMSDL) is performed within the CDN cluster of the CDN server layer, that is, each CDN node realizes the final model update through point-to-point interaction of local model update weight parameters. Super-cdn is both the starting point and the end point of iterative training. The CDN participating in each round of training needs to be digitally signed as the symbol of completion of training. SUPER-CDN issues local parameter weight W_k^j , initial training model M_0^j and digital signature S^j to adjacent nodes. Different nodes use local data sets to train and update W_k^j , and pass the updated W_k^j and S to the next hop node. The final SUPER-CDN converges the common update parameters of the iteration. When each CDN server receives

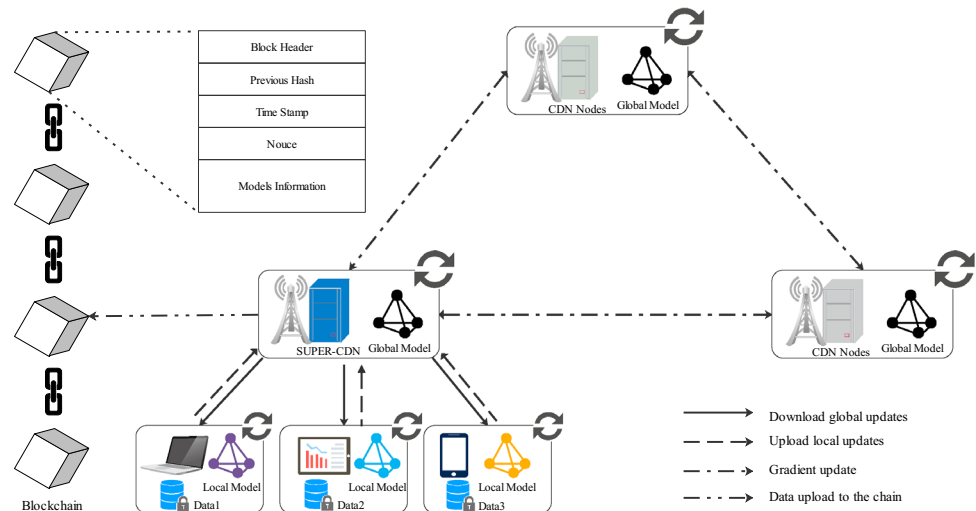
the parameter weight sent by the previous node, it first verifies its digital signature to ensure the parameter integrity and then carries out the training of this node. At the same time, by verifying whether there are nodes with incomplete learning in the signature information, the nodes are urged to participate in the training process to obtain the optimal training model. SUPER-CDN can predict the hotspot trend of users in various regions through NMSDL, and make an application to cloud server, and keep similar news sets in the distributed cluster in advance. The edge service layer has the functions of content distribution, bandwidth mitigation, and optimization of hot topic news distribution.

Blockchain layer provides a secure and reliable information recording data structure for the scheme. Super-cdn sends a brief description of the hot topics within the LAN cluster to the block chain in a certain period. Blockchain ensures that the data on the chain cannot be tampered with through cryptography (Hash algorithm, digital signature, Merkle Tree) and decentralized structure, and prevents the attacker from maliciously deleting hot topics. At the same time, hot topics can be backtracked based on the unique hash value of the transaction, and the region where the source event originated first can be traced.

Cloud services layer provides global data aggregation and sharing services, with powerful computing power and storage space. The cloud server can view the hot events of each blockchain from the blockchain and make predictions on the hot topics around the world. Timely send hot topics to each CDN cluster to enhance user experience.

3.4 Scheme overview

Our scheme includes a total of five steps, which are initialization, downloading global updates, Uploading local

Fig. 3 Flow chart of the system

updates, gradient update and data upload to the chain. Figure 3 is a flow chart of the scheme.

Initialization The main goal of the initialization phase is to build a completely decentralized peer-to-peer network and prepare the network for the next layered learning process. The initialization process mainly includes two parts: Blockchain network construction: At the beginning of initialization, the SUPER-CDNs in each CDN cluster are connected to form a peer-to-peer network. Parameter initialization: SUPER-CDN sends local parameter weights W_k^j , initial training model M_S^j and digital signature S^j to neighboring nodes.

Download global updates Each user's local device downloads a randomly initialized model or pre-trained model from the CDN node.

Upload local updates Each user aggregates local explicit data and implicit data. Explicit data includes user feedback, such as likes, comments, and sharing of news. Implicit data consists of information such as the user's browsing history, click history, and search logs. Users use local data sets for training to generate local parameter weights. Each user updates the local user-new model with local data and a global user-new model. Each user uses local data and local user-new models to calculate the local updates of the global user-new model, and uploads their updates to SUPER-CDN through a differential privacy algorithm.

Gradient update SUPER-CDN aggregates local model updates uploaded from various users through federal weighted algorithms (such as the federal average algorithm), and uses the results of the aggregation to update the global user-new model. SUPER-CDN performs gradient learning with ordinary CDN nodes in the same CDN cluster to update the global user-new model.

Data upload to the chain SUPER-CDN sends a brief description of the hot topics within the CDN cluster to the blockchain at a certain period.

4 Scheme method

4.1 Federated learning algorithm based on DP-SGD

The security of the gradient propagation of federated learning remains to be considered, as an attacker can capture gradient parameters and reverse them to perhaps private data sample sets (Song et al. 2013; Ma et al. 2020). We propose to combine differential privacy (DP) (McMahan et al. 2017; Geyer et al. 2017) with stochastic descent gradient (SGD) (Koskela and Honkela 2018) algorithm of model training in federated learning and enable this scheme to provide a more in-depth sensitive data protection mechanism.

Definition 1 (Neighboring Databases) Database D and D' with the same attribute structure, and the symmetric difference between them is denoted as DD' , $|DD'|$ to indicate the number of symmetric differences. If $|DD'|=1$, D and D' are called Neighboring Databases.

Theorem 1 (Differential Private) Given a random function L , R is all possible output fields of L , any Neighboring Databases D and D' for outputs $S \in R$ in L , while

$$\frac{Pr[L(D) \in S]}{Pr[L(D') \in S]} \leq \exp(\epsilon) \quad (2)$$

So random function L meet ϵ -differential privacy. ϵ is difference of privacy protection budget. When function L outputs a real value vector, sensitivity s is defined as:

$$s = \max_{D, D'} \|L(D) - L(D')\|_p \quad (3)$$

where the $\|\cdot\|_p$ is L_p norm.

In order to better protect the user's sensitive data set, the calibrated Laplace noise is added to the training gradient g

of the random SGD, and the local weight parameter W_k^i with noise was output through local training.

$$\tilde{g} = g + \text{Laplace}\left(\frac{s}{\epsilon}\right) \quad (4)$$

Federated learning algorithm based on DP-SGD (DP-SGDFL Algorithm): In the epoch t -th, local participants received the global model M_G^t sent by the central server G , and performed SGD algorithm training. Optimize the weight parameter W_k^t by minimizing the loss function J for all data samples. The randomly generated sample gradient $g = \nabla J(W_k^t, D_i)$ was calculated, and the gradient \bar{g} was obtained by calculating the L_2 norm of the generated gradient (Abadi et al. 2016). Add Laplace noise to \bar{g} to get noise gradient \tilde{g} . Finally, the local weight parameter W_k^i with noise is calculated.

Algorithm 1 Federated learning algorithm based on DP-SGD

Input: Database D_k , $i \in k$, learning rate λ_t , gradient norm bound C , Candidate C_k , protect buget ϵ , L_2 sensitivity s , loss function $\nabla J(W_k^i) = \frac{1}{N} \sum_i \nabla J(W_k^i, D_k)$.

Output: noise-added \tilde{W}_k^i

```

1: for epoch  $t \in [T]$  do
2:    $C_k$  receive  $W_G^t$  from  $G$ 
3:   while Accuracy( $W_G^t$ )  $\leq$  Threshold do
4:     Compute the random database gradient
5:      $g_t(D_i) \leftarrow \nabla J(W_k^t, D_i)$ 
6:     Clip gradient in  $L_2$  norm
7:      $\bar{g}(D_i) \leftarrow g_t(D_i) / \max(1, \frac{\|g_t(D_i)\|_2}{C})$ 
8:     Add noise to  $\bar{g}(D_i)$ 
9:      $\tilde{g}_t \leftarrow \bar{g}_t + \text{Laplace}(\frac{s}{\epsilon})$ 
10:    Training local weight parameter
11:     $\tilde{W}_k^{(t+1)} = W_G^t - \lambda_t \tilde{g}_t$ 
12:  end while
13:  Broadcast  $\tilde{W}_k^{(t+1)}$  to other Candidates
14: end for
15: return send  $\sum_k \tilde{W}_k^{(t+1)}$  to  $G$ 

```

4.2 Collaborative filtering algorithm

The basis of recommendation technology is recommendation algorithm. Among many recommendation algorithms, collaborative filtering algorithm is most widely used, including user based collaborative filtering algorithm and item based collaborative filtering algorithm (UserCF and ItemCF) (Huang et al. 2014). Both of them are neighborhood based algorithms. First, the correlation degree is calculated, and then the recommendation list is generated according to the correlation degree. The reason why news is recommended to use UserCF is that news has strong

timeliness, and users are often more interested in the latest news or popular hot news. Usercf has high timeliness, it can reflect the hot spots and the relationship between users' interests and groups, and does not need to calculate news similarity. Therefore, this paper is based on UserCF and combined with federated learning algorithm.

First, we need to build a user-new model, that is, a user's preference model for news (Hong-xia 2019). There are three ways for users to obtain data about the news preference model. The first type is the initiative provided by the user, that is, the basic information provided by the user at the time of registration, including gender, age, preferred news classification, etc. The second type is explicit data, some active operations that users generate on news, such as likes, sharing, and comments. The third type is private data, the behavior of users during browsing, including factors such as browsing time. In the cold start phase, we establish a local initial model based on the data provided by users, and the system continuously updates the local user model through the global model and user behavior.

User-new model calculating the user's preference value for news is the key to successful news recommendation. We build a local *UPVM* (User Preference Vector Model) based on user behavior preference values. The expression of *UPVM* is as follows.

$$UPVM = \{(n_1, p_1), (n_2, p_2) \dots (n_i, p_i)\} \quad (5)$$

Among them, n_i represents the news with the label i , and p_i is the comprehensive preference value of the user behavior of the news n_i . The calculation formula of the comprehensive preference value of user behavior of news n_i is as follows.

$$p_i = \frac{\sum_{k=1}^n \omega_k \cdot value_k}{k} + bt_i \quad (6)$$

Where ω_k is the weight of the user's behavior k , $value_k$ is the value of the user's behavior k , and bt_i is the time when the user browses the news n_i . User behavior usually includes clicking, reading time, interest, dislike and comment. The latter three belong to explicit acquisition. If the news n_i is labeled as not interested, its comprehensive preference value is $p_i = 0$. In user behavior, the weight of comments is greater than that of clicking.

News set The news model mainly determines the recommendation list output after combining the user's similarity with the news collection of neighbor users and the global model. In the past news recommendation systems, most of them used content-based news recommendation schemes. The advantage of this method is that the similarity between news can be obtained through the weighting of topics in the news content, and the system can accurately recommend the news of interest to the user. The disadvantage is that the news update speed is too fast, resulting

in too much calculation, so it is impossible to update hot news in time.

We take the classification of news, release time, length of news and reading volume as elements of the news model. The expression of NVS (News Vector Set) is as follows:

$$NVS = \{ \text{Classification}, \text{Release} - \text{time}, \\ \text{Length}, \text{News} - \text{clicks} \} \quad (7)$$

Among them:

Classification is the classification of news. During the cold start of the system, we need to provide the initial news recommendation activities in combination with the interest classification data provided by the user. For example, if the user chooses to be interested in the entertainment classification, the hottest entertainment news is recommended.

Release - time is the time of news release, which determines the degree of news old and new. It plays a key role in the entire system and affects the recommendation to users.

Length is the length of the news. The length of the news and the reading time of the user together determine the user's reading time-consuming factor, which affects the user's actual score calculation results for the news.

News - clicks indicates the number of news reads, that is, the number of news clicks, which determines the popularity of the news.

News has strong timeliness, and users are often more interested in the latest news or popular hotspots. At this time, personalization is not as important as timeliness and hotspots in news recommendation. In the same time period, two users acted on the same news. If the news is relatively unpopular or has strong professional news, it shows that their interests are similar. If you have read some hot news, it is reluctant to think that the two have similar interests at this time. Therefore, the system should start from the time context factors to modify the similarity. Considering the time factor, we can add a time attenuation factor α to the model. The farther the user u and user v generate behavior to news n_i , the less similarity of interest between the two users will be. Among them, $N(u)$ is the news set read by user u , and $N(v)$ is the news set read by user v . At this time, the cosine similarity formula should be modified as follows:

$$sim_{uv} = \frac{\sum_{i \in N(u) \cap N(v)} \frac{1}{1 + \alpha |t_{un_i} - t_{vn_i}|}}{\sqrt{|N(u)| \cup |N(v)|}} \quad (8)$$

TOP-N recommendation stage The TOP-N (Yanxiang et al. 2013) recommendation stage refers to after the user has acted on the news, and the user gets the trained global news model. Calculate the predicted preference value of news that the user has not generated behavior. The solution generates a recommendation list by predicting preference values and pushes the first n news items in the list to the user. The

calculation formula of the expected preference value of news that users have not yet generated behavior is as follows:

$$exp_{(u,i)} = \sum_{v \in [U(u,j) \cup N(i)]} bt_i sim_{uv} \quad (9)$$

With the data representation model based on user model as input, we also need the collaborative filtering recommendation engine provided by taste. Taste defines five basic interfaces: datamodel, preferencetransform, usercorrelation, itemcorrelation, userneighborhood and recommender. Datamodel is the user's evaluation matrix model of content, and user evaluation data in files or databases can be loaded through specific implementation classes. Taste supports two types of datamodels: filedatamodel and databasedatamodel. The system uses the principle of UserCF and the usersimilarity interface. Taking the datamodel of filedatamodel as an example, the user-new model is applied to implement the UserCF based on Taste.

Algorithm 2 Recommendation algorithm

```

1: DataModel model = new FileDataModel(new
   File("data.txt"));
2: UserSimilarity sim = new PersonCorrelationSimilarity
   (model);
3: UserNeighborhood neighborhood = new Nearest-
   NUserNeighborhood(3, sim, model);
4: Recommender recommender = new GenericUserBase-
   dRecommender(model, neighborhood, sim);
5: Recommender cachingRecommender = new
6: CachingRecommender(recommender);
7: List <RecommendedItem> recItem-
   List=rec.recommend(1,2,3);
8: For(RecommendedItem item:recItemList)

```

5 Experimental configuration and analysis

5.1 Blockchain storage

In the experimental environment of blockchain network, the operating system use Ubuntu 18.04, and the block-chain is built based on the Hyperledger Fabric 1.4.4 Alliance Block-chain sterling, Experimental data for Hyperledger Fabric showed that the time rate for storage of data size below 128KB was lower than 100ms overall, while the storage rate for public blockchain Ethernet lane and allied blockchain FISCO BCOS remained increasing. Ethereum took longer, with an upload rate of almost 820 ms at 128 KB. When the upload data size was 128 KB, the average upload time of Hyperledger Fabric was 1/8 of Ethereum and 1/3 of FISCO BCOS. For public blockchain, alliance blockchain have better advantages in node redundancy, identity authentication and upload rate. This is

why we chose Hyperledger Fabric as the underlying architecture of the blockchain. The detailed configuration of the experiment is shown in Table 1. Figure 4 shows the storage performance comparison of three different types of blockchains.

5.2 System usability analysis

In order to verify the effectiveness of the algorithm proposed in this paper and to evaluate the performance of the system, we collected 5000 news information from various news websites by using crawler technology on the Internet. Through analyzing user behavior logs, we collected the necessary data of the model, including news ID, classification, release time, length, news—clicks, so as to generate the news data set of this scheme, which is used as the test set of the experiment.

Accuracy rate and recall rate are important indicators to measure the prediction accuracy of a recommendation system. We tested the impact of setting the number of nodes in the CDN cluster on the recommendation results. Figure 5 is a graph of our experimental results. From the experimental results, we can see that if the number of nodes in the CDN cluster is small, the accuracy of the recommendation result is relatively low. As the number of CDN nodes in the cluster increases, the recommendation effect also increases. When it reaches a certain value. The recall rate and accuracy rate of recommendation results tended to be flat.

For the DP-SGDFL algorithm proposed in Sect. 4, the L_p Norm is intended to be discussed in the experimental part. According to the experimental results, the classification Accuracy value of noiseless SGD and SGD under different normal forms was compared, and the optimal normal form was selected. In this section, we constructed *logisticRegression* (refer to LogR) (Chaudhuri and Monteleoni 2008) and *LinearPegasosSVM* (refer to PSVM) (Shalev-Shwartz et al. 2010), optimization algorithms widely used in SGD, for experimental analysis respectively.

We used the Eq. (8) to update SGD rules, and the learning rate is set to $\lambda_t = t^{-\frac{1}{2}}$, private budget ϵ is set 1.

$$\tilde{W}_k^{(t+1)} = W_G^t - \lambda_t \tilde{g}_t \quad (10)$$

The experiment is divided into substitute samples (sub) and non-substitute samples (no-sub), because the sample size of SGD is random at each step, and the sample size may be

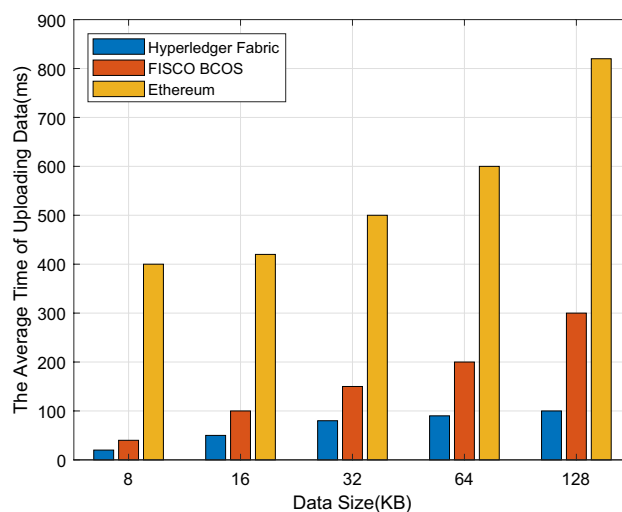


Fig. 4 Blockchain storage

exhausted. When the non-replaceable sample set is used up, we sample the sample again. Differential privacy is usually the L_1 norm and L_2 norm. The L_1 norm is the sum of the absolute values of $Laplace(\frac{\epsilon}{n})$ noise of all dimensions. L_2 norm is the result of quadratic term operation of $Laplace(\frac{\epsilon}{n})$ noise vector of each dimension. L_2 norm is significantly better than other parameter settings in each test environment. Figures 6 and 7 are the accuracy of the logR and PSVM optimization algorithms in the news test set. According to the data in the figure, the accuracy of L_2 norm is better than L_1 norm in any test environment. Under L_1 norm, the increased noise is sparse and easier to feature selection. However, the news data of this system has been classified by certain characteristics. In contrast, the advantage of L_2 norm is that the data is more uniformly distributed on the coordinate system, which can effectively reduce the interference on the gradient steps. Through the experimental data, we found that in the testing

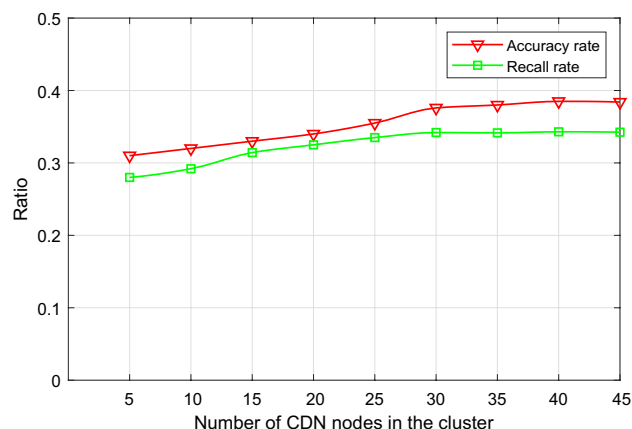


Fig. 5 The impact of the number of CDN nodes on the results

Table 1 Experimental configuration

Environment	Version	Notes
Ubuntu	v18.04	Operating system
Hyperledger Fabric	v1.4.4	Permissioned blockchain
FISCO BCOS	v2.2.0	Permissioned blockchain
Ethereum	v1.9.0	Public blockchain

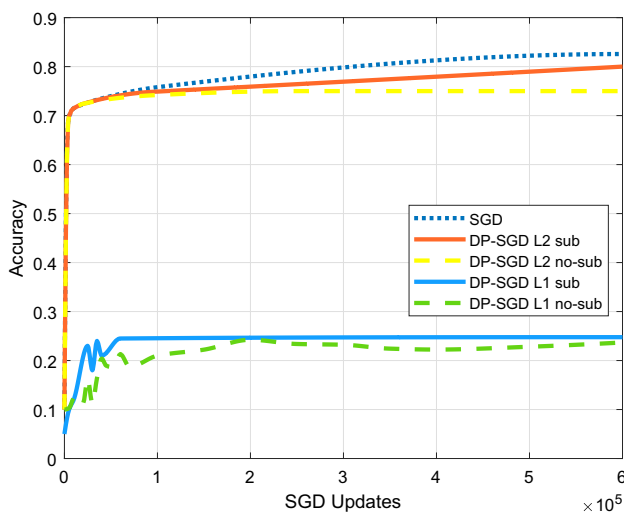


Fig. 6 Algorithm DP-SGDFL with LogR on news test set

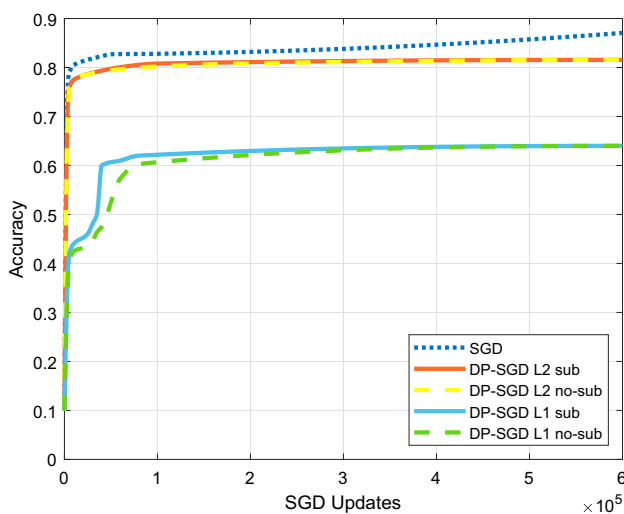


Fig. 7 Algorithm DP-SGDFL with PSVM on news test set

environment of the two classification algorithms, L_2 norm has a higher classification accuracy by adding noise data, which is only 0.1 lower than the noiseless SGD algorithm. Therefore, the DP-SGDFL algorithm proposed in this paper adopts L_2 norm.

6 Conclusion

This paper proposes a collaborative filtering algorithm recommendation system based on federated learning of end-edge-cloud, which is based on cloud side computing scheme and adds multi-layer distributed learning mechanism and blockchain card protection. At the same time, the marginal CDN cluster is allowed to take part of the computing

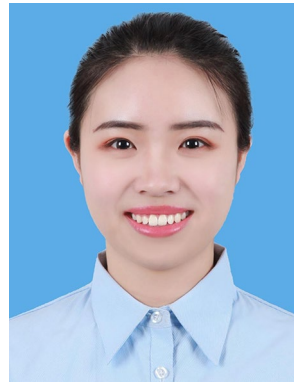
pressure and recommendation tasks for the cloud server. Finally, we obtained the license of the block chain through experimental data, and conducted experimental analysis and discussion on the considerable storage efficiency of the blockchain and the availability of the two algorithms. Compared with the noiseless SGD algorithm, the classification accuracy difference of the SGD algorithm based on differential privacy is only 0.1. The news recommendation system of this scheme has better innovation in sensitive data protection, reducing bandwidth pressure and improving node computing speed.

Acknowledgements This research work is supported by the National Key R&D Program of China (2018YFB1201500), National Natural Science Funds of China (62072368, 61773313, 61702411), National Natural Science Funds of Shaanxi (2017JQ6020, 2016JQ6041), Key Research and Development Program of Shaanxi Province (2020GY-039, 2017ZDXMGY-098, 2019TD-014)

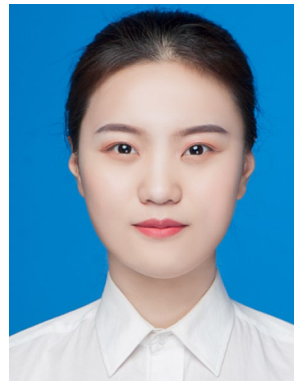
References

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (2016)
- Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: a survey. *IEEE Internet Things J.* **5**, 1 (2017)
- Ahmed, F., Shafiq, M.Z., Khakpour, A.R., Liu, A.X.: Optimizing internet transit routing for content delivery networks. *IEEE/ACM Trans. Netw.* **26**, 1 (2017)
- Al-Abbasi, A.O., Aggarwal, V., Ra, M.: Multi-tier caching analysis in CDN-based over-the-top video streaming systems. *IEEE/ACM Trans. Netw.* **27**, 2 (2019)
- Ammad-ud-din, M., Ivannikova, E., Khan, S.A., Oyomno, W., Fu Q., Tan, K.E., Flanagan, A.: Federated collaborative filtering for privacy-preserving personalized recommendation system. *Statistics* (2019)
- Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: 13th EuroSys Conference (EuroSys) (2018)
- Bagchi, S.: Performance and quality assessment of similarity measures in collaborative filtering using mahout. *Procedia Comput. Sci.* **50**, 229–234 (2015)
- Chaudhuri, K., Monteleoni, C.: Privacy-preserving logistic regression. *Adv. Neural Inf. Process. Syst.* **21**, 289–296 (2008)
- Chen, M., et al.: Analysis and scheduling in a 5G heterogeneous content delivery network. *IEEE Access* **2018**, 6 (2018)
- Chen, M., Wang, L., Chen, J., Wei, X., Lei, L.: A computing and content delivery network in the smart city: scenario, framework, and analysis. *IEEE Netw.* **33**(2), 89–95 (2019). <https://doi.org/10.1109/MNET.2019.1800253>
- De Donno, M., Tange, K., Dragoni, N.: Foundations and evolution of modern computing paradigms: cloud, IoT, Edge, and Fog. *IEEE Access* **2019**, 7 (2019)
- Geyer, R.C., Klein, T., Nabi, M.: Differentially private federated learning: a client level perspective. *Statistics* (2017)
- Huang, S., Jiang, X., Zhang, N., Zhang, C., Dang, D.: Collaborative filtering of web service based on MapReduce. In: Proceedings 2014 International Conference on Service Sciences (ICSS) (2014)

- Koskela, A., Honkela, A.: Learning rate adaptation for differentially private stochastic gradient descent. *Statistics* (2018)
- Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y.: Differentially private asynchronous federated learning for mobile edge computing in urban informatics. *IEEE Trans. Ind. Inform.* **16**, 3 (2019)
- Lundbaek, L.-N., D'Iddio, A.C., Huth, M.: Centrally governed blockchains: optimizing security, cost, and availability. In: *Conference on Models, Algorithms, Logics and Tools in Honour of Kim G. Larsen on the Occasion of his 60th Birthday* (2017)
- Ma, C., Li, J., Ding, M., et al.: On safeguarding privacy and security in the framework of federated learning. *IEEE Netw.* **99**, 1–7 (2020)
- McMahan, H.B., Moore, E., Ramage, D.: Communication-efficient learning of deep networks from decentralized data. In: *20th International Conference on Artificial Intelligence and Statistics (AISTATS)* (2017)
- Min, X., Li, Q., Liu, L., Cui, L.: A Permissioned blockchain framework for supporting instant transaction and dynamic block size. *Trust-com/bigdata/se/ispa*, IEEE (2017)
- Peng, X., Ren, J., She, L., Zhang, D., Li, J., Zhang, Y.: BOAT: a block-streaming app execution scheme for lightweight IoT devices. *IEEE Internet Things J.* **5**(3), 1816–1829 (2018)
- Ren, J., Guo, H., Xu, C., Zhang, Y.: Serving at the edge: a scalable iot architecture based on transparent computing. *IEEE Netw.* **31**(5), 96–105 (2017)
- Shalev-Shwartz, S., Singer, Y., Srebro, N., Cotter, A.: Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program. B* **2010**, 3–30 (2010)
- Sharma, P.K., Chen, M., Park, J.H.: a software defined fog node based distributed blockchain cloud architecture for IoT. *IEEE Access* **6**, 115–124 (2018). <https://doi.org/10.1109/ACCESS.2017.2757955>
- Song, S., Chaudhuri, K., Sarwate, A.D.: Stochastic gradient descent with differentially private updates. In: *IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (2013)
- Sun, H., Yu, Y., Sha, K., Lou, B.: mVideo: edge computing based mobile video processing systems. *IEEE Access* **2019**, 8 (2019)
- Thomas, H., Ned, S.: Cloud-based commissioning of constrained devices using permissioned blockchains. In: *IoTPTS '16: Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security* (2016)
- Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. *ACM Trans. Intell. Syst.* **10**(2), 12.1–12.19 (2019)
- Yanxiang, L., Deke, G., Fei, C., Honghui, C.: User-based Clustering with Top-N Recommendation on Cold-Start Problem. In: *2013 third international conference on intelligent system design and engineering applications*, Hong Kong (2013)
- Zhou, W., Li, R., Liu, W.: Collaborative filtering recommendation algorithm based on improved similarity. In: *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (2020)



Yuying Tian is studying for master's degree in software engineering from the school of computer science and engineering, Xi'an University of technology. Her research area is blockchain, network security and edge computing.



Xinyue Yin is studying for master's degree in computer application in Xi'an University of Technology. Her research area is network security and blockchain.



Xinhong Hei received his B.S. degree and M.S. degree in computer science and technology from Xi'an University of Technology, Xi'an, China, in 1998 and 2003, respectively, and his Ph.D. degree from Nihon University, Tokyo, Japan, in 2008. He is currently a professor with the Faculty of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. His current research interests include intelligent systems, safety-critical system, and train control system.



Yichuan Wang received his Ph.D. degrees in computer system architecture from Xidian University of China in 2014. He is an ACM member and a CCF member. Now he is an associate professor in Xi'an University of Technology and with Shaanxi Key Laboratory of Network Computing and Security Technology. His research areas include networks security and system vulnerability analysis.