Electronic Research Archive

ERA, 31(12): 7559–7577.
DOI: 10.3934/era.2023382
Received: 16 September 2023
Revised: 14 November 2023
Accepted: 17 November 2023
Published: 04 December 2023

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

Research article

A privacy preserving recommendation and fraud detection method based on graph convolution

Yunfei Tan, Shuyu Li* and Zehua Li

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China

* **Correspondence:** Email: lishuyu@snnu.edu.cn; Tel: +862985310161.

Abstract: As a typical deep learning technique, Graph Convolutional Networks (GCN) has been successfully applied to the recommendation systems. Aiming at the leakage risk of user privacy and the problem of fraudulent data in the recommendation systems, a Privacy Preserving Recommendation and Fraud Detection method based on Graph Convolution (PPRFD-GC) is proposed in the paper. The PPRFD-GC method adopts encoder/decoder framework to generate the synthesized graph of rating information which satisfies edge differential privacy, next applies graph-based matrix completion technique for rating prediction according to the synthesized graph. After calculating user's Mean Square Error (MSE) of rating prediction and generating dense representation of the user, then a fraud detection classifier based on AdaBoost is presented to identify possible fraudsters. Finally, the loss functions of both rating prediction module and fraud detection module are linearly combined as the overall loss function. The experimental analysis on two real datasets shows that the proposed method has good recommendation accuracy and anti-fraud attack characteristics on the basis of preserving users' link privacy.

Keywords: recommendation system; graph convolution; matrix completion; fraud detection; differential privacy

1. Introduction

Human society has entered the era of big data, experiencing a massive growth in data at the level of Zettabytes (ZB) annually. Recommendation systems have become a powerful tool for helping users filter useful information from massive data and solving the problem of information overload.

Collaborative filtering, expert-based, and rule-based recommendations are typical methods used for providing recommendations.

Currently, with the rapid development of various learning methods including deep learning and federated learning [1,2], research combining with learning approaches is in full swing, and gains success in many fields such Internet of Things (IoT) [3–5] and Industrial IOT (IIOT) [6], edge computing [7–10] and recommendation systems etc. As a typical deep learning technique, graph convolution overcomes the limitations of manually selecting sample features and excels in capturing the link relationship between users and items, it has been successfully applied in recommendation systems.

Most existing recommendation systems provide recommendation service based on users' preferences which are analyzed by users' rating data and other information. While users enjoy the personalized recommendation services, they also face the risks of privacy leakage to a large extent. For example, link relationship between certain user and an item is formed when the user rates the item, adversaries may use techniques such as reverse inference to obtain the graph structure and link information. From a graph perspective, many privacy-preserving recommendation methods currently focus on protecting the nodes and their attributes but tend to neglect the graph's link relationships (edges).

Further, due to the lack of trust between users and the recommendation system, as well as collusion between merchants and the recommendation system, it is difficult to ensure the authenticity of rating data source. The recommendation system may recommend an item that is intentionally highly rated by fraudsters. For example, certain merchants hire a large number of fraudsters to write false positive reviews for themselves in order to improve their rating, and even delete and modify the rating data used for recommendation. This makes it difficult for the recommendation system to effectively verify and control the quality of the data provided by users, and further reduces users' trust and the accuracy of the recommendation system. Currently, researchers have developed several methods for fraud detection [11,12], with most focusing on fraud detection in financial settings. For instance, Wang et al. [13] employed multi-view data to alleviate the issue of insufficient labeled data, utilizing both semi-supervised and supervised learning for fraud detection. Fraud detection in recommendation systems presents different characteristics than financial fraud detection. For instance, fraud attacks in recommendation systems are becoming increasingly common and simple to execute. Developing a sturdy recommendation system that considers fraud attacks remains an ongoing challenge.

Therefore, constructing a recommendation system that takes into account both the preservation of user privacy and fraud detection holds significant value.

To address the aforementioned issues, a Privacy Preserving Recommendation and Fraud Detection method based on Graph Convolution (PPRFD-GC) is proposed in this paper. The PPRFD-GC method operates under the assumption that a trusted data collector is accountable for collecting user rating data and generating the original graph of rating data. Taking the original graph as the input, a semi-honest recommendation system provides both recommendation service and fraud detection service. The main contributions of this paper can be summarized as follows:

1) To provide effective recommendation in condition of preserving graph structure and link privacy of users, a privacy-preserving recommendation algorithm utilizing the centralized differential privacy and Differentially Private Graph Generative Adversarial Networks (DPGGAN) model [14] is designed. The given algorithm firstly adopts decoder/encoder framework to generate the synthesized graph satisfying edge differential privacy, then applies graph based matrix completion technique for rating prediction. The given algorithm alleviates the problem of data sparsity and achieves a better balance between privacy preservation and data utility.

2) To solve the problem of fraud data and improve the robustness of recommendation system, a fraud detection algorithm based on AdaBoost is presented to detect possible fraudsters. The designed algorithm firstly calculates user's Mean Square Error (MSE) of rating prediction and generates dense representation of the user. Taking dense representation as the input, the fraud detection classifier is trained to detect whether certain user is a fraudster or not. Finally, the loss functions of above two algorithms are linearly combined to form the final loss function of the PPRFD-GC method.

3) Experiments were conducted on actual datasets from Yelp and Amazon, and the findings exhibit the efficacy of the proposed method.

The paper is structured as follows. In Section 2, related work is discussed. Section 3 provides a review of preliminary knowledge about graph convolution and differential privacy. The design of the PPRFD-GC method is presented in Section 4. In Section 5, the results of experiments conducted on two real datasets are analyzed. Finally, Section 6 concludes the paper.

2. Related work

2.1. Graph-based recommendation and fraud detection

Recommendation systems possess strong graph structure properties, for example, user-items can form a bipartite graph or be used as heterogeneous graphs. Further, users can form a social network among themselves. Therefore, some research have applied Graph Neural Network (GNN) to recommendation system. He et al. [15] proposed a new lightweight Graph Convolutional Network (GCN) model for collaborative filtering, which contained only the neighborhood aggregation component of graph convolution, specifying the aggregation function in traditional GCN as a simple weighting and aggregator to ensure good feature representation performance. Mao et al. [16] proposed the simplified GCN model named UltraGCN for collaborative filtering. Instead of using explicit message passing, the proposed model omits feature transformation and nonlinear activation, and directly approximated the limit of infinite-layer graph convolution through constraint loss. Yu et al. [17] proposed a multi-channel hypergraph convolutional network, which compensated the aggregation loss of multi-channel embeddings in the network with self-supervised learning and leveraged comprehensive high-order user relations to enhance social recommendation.

In the field of fraud detection, graphs are constructed based on business scenarios and then combined with GNN to detect fraud. Liu et al. [18] considered the problem of fraud detection in graphs as an unbalanced node classification task, and proposed a Pick and Choose Graph Neural Network (PC-GNN) to resolve the class imbalance problem on graphs in fraud detection. To address the identification of fraud requests, Shen et al. [19] proposed an evolutionary privacy-preserving learning schema for edge computing based IoT data sharing problem. The schema introduces evolutionary game theory and constructs a payoff matrix to represent the mutual communication between IoT devices and edge nodes, then achieves the evolution of the privacy preserving strategy of edge nodes through the game theory. Zhang et al. [20] proposed a user representation learning GCN framework named GraphRfi for robust recommendation and fraud detection. In its end-to-end learning process, the recommendation component and the fraudster detection component mutually enhance by passing parameters to each other.

2.2. Differentially private recommendation methods

Differential privacy is a privacy preservation method based on data perturbance and rigorous mathematical proof. Zheng et al. [21] proposed a model of a decentralized GNN for privacy-preserving recommendations named DGREC. The proposed model constructed local inner-item graph as well as the global user graph for each user, and securely shared the gradient computed locally using a mechanism of local differential privacy (LDP), achieving strong privacy protection for user data. Wu et al. [22] had proposed a GNN-based privacy preserving federated learning recommendation framework. In the framework, each client employed LDP to preserve local gradients and uploaded them to a central server for aggregation. To protect the items that interacted with the user, a random sampling method was used to generate anonymous pseudo-interaction items and participate in embedding generation. To achieve a trade-off between recommendation quality and privacy protection against inference attacks, Xiao et al. [23] proposed a deep reinforcement learning (RL) based user profile perturbation for recommendation systems. The scheme uses differential privacy to perturb user profiles and uses an evaluated neural network (NN) and a target neural network to select the privacy budget, achieving a balance between user privacy protection and recommendation quality. To address the privacy leakage issue in Collaborative Filtering (CF) recommendation systems, Chen et al. [24] proposed a differentially private CF recommendation system based on *K*-Means clustering, called KDPCF. KDPCF clusters the dataset into categories using *K*-Means clustering and efficiently selects neighbors for recommendation using the exponential mechanism.

From above analysis, most current research focus on either non-privacy preserving/privacy preserving recommendation issue or fraud detection problem. There are few research work taking these two issues together into consideration.

3. Preliminaries

3.1. Graph convolutional networks

Graph Convolutional Networks (GCN) [25] is a type of graph neural network that utilizes convolutional operations for representation learning. GCN's layer-by-layer update formula is obtained from the embedded aggregation of the node itself and its neighboring nodes followed by a non-linear transformation. The convolution in GCN differs from the commonly used convolution in deep learning. It pertains to the aggregation of data for a node and its neighbors in a non-Euclidean domain.

Spectral-based graph convolution methods define operators for each layer by using the convolution theorem. The convolution kernel is applied to aggregate information between nodes in the input signal. Multiple layers of neural networks are stacked using a particular non-linear activation function.

For a graph $G=(V,E)$, where V is the set of nodes and E represents the set of edges. The formula of GCN can be defined as follows:

$$H^{l+1} = f(H^l, A) \tag{1}$$

where H^l is the embedded representation of the l layer, A denotes the adjacency matrix, and f represents an activation function. Specifically, it can be realized in GCN by the following formula:

$$H^{l+1} = \sigma(D^{-\frac{1}{2}}\hat{A}D^{-\frac{1}{2}}H^{l}W^{l}) \tag{2}$$

where D represents the degree matrix, σ denotes a nonlinear activation function, such as Leaky ReLU; W^l stands for the parameter matrix of the l layer, $\hat{A}$ is the adjacency matrix with self-loop.

3.2. Edge differential privacy

Differential privacy is a privacy protection technology proposed by Dwork [17]. In simple terms, differential privacy adds controllable noise to the original dataset, allowing the noisy dataset to retain the original statistical attributes without disclosing sensitive information.

Definition 1 (Edge Differential Privacy [14]). Given two neighboring graphs $G=(V,E)$ and $G'=(V',E')$, where $V=V'$ and $E'=E-E_x:|E_x|=k$ (i.e., the two edge sets differ by k edges), and there is a randomized algorithm M, P_M represents the set comprising all possible outputs of M. For any subset of P_M, if M satisfies:

$$\Pr[M(G)\in S]\le e^{\varepsilon}\times\Pr[M(G')\in S]+\delta \tag{3}$$

Then M is said to satisfy (ε,δ)-edge differential privacy. Where $\Pr[\cdot]$ denotes the probability of the event, ε is the privacy budget and δ represents a probability of failure.

Theorem 1 (Sequential Combination). Supposing there is a set of random algorithms $\{M_1,M_2,...,M_n\}$, each $M_i(1\le i\le n)$ satisfies ε_i-differential privacy on the dataset D. Then, the set of M_i sequence privacy mechanisms provides $(\sum_{i=1}^{n}\varepsilon_i)$-differential privacy.

Theorem 2 (Parallel Combination). Supposing the dataset D can be divided into a series of independent and non-overlapped subsets $\{D_1,D_2,...,D_n\}$ and there is a set of random algorithms $\{M_1,M_2,...,M_n\}$. If each $M_i(1\le i\le n)$ satisfies ε_i-differential privacy on $D_i(1\le i\le n)$, the set of randomized algorithms can provide $\max\{\varepsilon_i\}$-differential privacy on the dataset D.

Theorem 3 (Post-Processing [26]). Supposing an randomized algorithm M satisfies ε-differential privacy, and let f be an arbitrary mapping from the set of possible outputs of M to an arbitrary set, then $f\circ M$ achieves ε-differential privacy.

4. The PPRFD-GC method

4.1. Framework

Aiming at the possible risks of privacy disclosure and the problems of fraudulent data for recommendation system, a Privacy Preserving Recommendation and Fraud Detection method based on Graph Convolution (PPRFD-GC) is proposed in the paper, and the framework of the proposed method is depicted in Figure 1. In the framework, there exists a trustworthy third-party data aggregator and a semi-honest recommendation system. The trusted data aggregator generates the original graph $G(V,E)$ according to the collected users' rating data, where $V=U\cup It$ is the set of vertices, U is the set of users and It is the set of items, E is the set of edges which reflects the links between the users and the items. Taking the original graph G as input, the recommendation system realizes rating prediction and fraud detection service based on GCN.

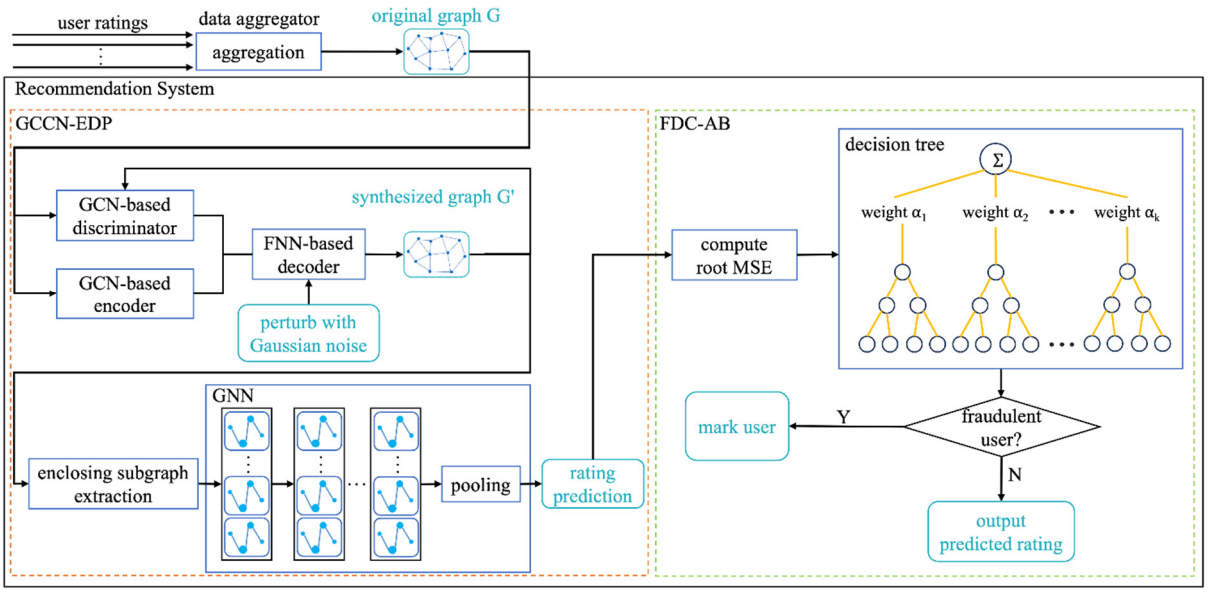


Figure 1. The framework of the PPRFD-GC method.

The PPRFD-GC method consists of two main algorithms: the Graph Complete Convolution Network with edge differential privacy (GCCN-EDP) and the Fraud Detection Classification based on AdaBoost (FDC-AB).

To preserve the link privacy of users and improve the recommendation efficiency of graph convolution, the GCCN-EDP algorithm firstly follows the concept of the DPGGAN model proposed by Yang et al. [14] and central differential privacy for generating the synthesized graph, which consists of a GCN-based graph encoder, a GCN-based discriminator and a Feedforward Neural Network (FNN) based decoder. The GCN-based graph encoder guides the learning of a fully connected FNN-based adjacency matrix decoder, which can be trained to directly reconstruct a graph with similar links as in the original graph. Further, Gaussian noise is added to the gradients of the decoder to preserve the link privacy. Next, the GCCN-EDP algorithm uses graph based matrix completion technique to carry out rating prediction, which includes two components: enclosing subgraph extraction and graph-level Graph Neural Network (GNN). The first component extracts enclosing subgraphs and labels the nodes in the subgraphs, then these subgraphs are fed to the graph level GNN for training and rating prediction.

According to sociological theory, ratings made by normal users have greater stability and predictability compared to those made by fraudsters, who tend to make multiple extreme ratings. The FDC-AB algorithm is designed to detect fraudsters based on the AdaBoost. For a user u, the FDC-AB algorithm firstly calculates MSE of rating prediction of u, and then generates dense representation h_u^* of u. Taking h_u^* as the input, the fraud detection classifier is trained to detect whether u is a fraudster or not. Finally, the loss function of the GCCN-EDP algorithm is linearly combined with the loss function of the FDC-AB algorithm to form the final loss function of the PPRFD-GC method.

4.2. GCCN-EDP algorithm

The GCCN-EDP algorithm comprises two phases: synthesized graph construction and graph based rating matrix completion.

During the synthesized graph construction phase, the GCCN-EDP algorithm adheres to the

principles of the DPGGAN mode to perturb the original graph G. As a result, a synthesized graph G' is generated that satisfies edge differential privacy. Notably, both G' and G have the same number of vertices, degrees and triangles, but it is hard to deduce whether an edge in G' is also present in G.

Supposing $\mathbf{A}$ is the adjacency matrix of graph G, and $\mathbf{X}$ represents the one-hot node identity matrix. The GCN-based graph encoder guides the learning of the FNN-based adjacency matrix decoder, which can be trained to directly reconstruct a graph G' with similar links as in the original graph G.

The latent representation $\mathbf{Z}$ of $\mathbf{A}$ is defined as follows:

$$q(\mathbf{Z} \mid \mathbf{X}, \mathbf{A}) = \prod_{i=1}^{N} q(\mathbf{Z}_i \mid \mathbf{X}, \mathbf{A}) = \prod_{i}^{N} \mathcal{N}(\mathbf{z}_i \mid u_i, diag(\sigma_i^2)), \tag{4}$$

where N is the number of vertices, μ_i is the mean vector, σ_i is the standard deviation vector.

The GCN-based graph encoder adopts a two-layer GCN model while $\mathrm{g}_{\bullet}(\mathbf{X}, \mathbf{A}) = \tilde{\mathbf{A}}\,\mathrm{ReLU}(\tilde{\mathbf{A}}\mathbf{X}W_0)W_1$, g_{μ} and g_{σ} form the encoder and share the first-layer parameters W_0. $\tilde{\mathbf{A}} = D^{-\frac{1}{2}}\mathbf{A}D^{-\frac{1}{2}}$ is the symmetrically normalized adjacency matrix of G, and D is the degree matrix.

To generate the synthesized graph G', a reconstructed adjacency matrix $\mathbf{A}'$ is computed from $\mathbf{Z}$ by the FNN-based decoder as follows:

$$p(\mathbf{A}' \mid \mathbf{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{N} p(\mathbf{A}'_{ij} \mid \mathbf{z}_i, \mathbf{z}_j) = \prod_{i=1}^{N}\prod_{j=1}^{N} \sigma(\mathbf{f}(\mathbf{z}_i)^T \mathbf{f}(\mathbf{z}_j)), \tag{5}$$

where $\sigma(z) = 1/(1+e^{-z})$, $\mathbf{f}$ is a two-layer FNN appended to $\mathbf{Z}$ before the logistic sigmoid function, and it is helpful for generating the edges in G'.

In the phase of graph based rating matrix completion, to train the graph convolution model, the GCCN-EDP algorithm adopts the idea of inductive matrix completion proposed by Zhang et al. [27], and extracts *h-hop* enclosing subgraphs of the rating matrix of the synthesized graph G'. These subgraphs are then mapped to their corresponding ratings, to achieve the function of rating matrix completion. This phase consists of two components: enclosing subgraph extraction, graph-level GNN training.

In the component of enclosing subgraph extraction, the GCCN-EDP algorithm firstly extracts an h hops enclosing subgraph around nodes u and v of G' for each observed rating $r_{u,v}$. The *h-hop* enclosing subgraph includes nodes u, v and their neighbors within h hops, which contains rich graphical structural information about rating $r_{u,v}$. The basic concept of the enclosing subgraph is simple and intuitive: if item v_0 is liked by a user u_0, it is possible that it will also be liked by a user u_1, who shares similar preferences with u_0. A Breadth First Search (BFS) algorithm can be used for enclosing subgraph extraction, and more details are omitted due to space limitation.

Before feeding these enclosing subgraphs to the GNN, each node in a subgraph is assigned with an integer label, marking the different role of each node in the subgraph. Initially, the target user u and target item v are assigned labels 0 and 1, respectively. For the remaining nodes in the enclosing subgraph, we determine their labels according to at which hop they are included in the subgraph. A label of $2i$ is assigned to a node of user type if it is present in the i-th hop, while a label of $2i+1$ is assigned to a node of item type if it is present in the i-th hop. The purpose of node labeling is to distinguish the target node from context nodes, enabling the GNN to create inferred relationships and forecast ratings among varying users and items. The one-hot encoding of these node labels will be considered as the initial node feature W_0 of the enclosing subgraphs when sending these subgraphs to the GNN.

In the component of graph-level GNN training, the GNN contains two sub-components: message

passing layers that extract a feature vector for each node in the subgraph, and a pooling layer to summarize a subgraph representation from node features.

To learn different graph patterns in a subgraph according to edge types, the message layer passing function used by the GCCN-EDP algorithm is defined as follows:

$$\mathrm{x}_i^{l+1} = W_0^l x_i^l + \sum_{r\in\Re}\sum_{j\in N_r(i)} \frac{1}{\mathrm{N}_r(u)} W_r^l x_j^l \tag{6}$$

where x_i^l and x_j^l denote the feature vectors of nodes i and j at a given layer l, and $\mathrm{N}_r(u)$ is the neighbor set of node u, where each node has a connection to u with an edge type r. W_0^l and W_r^l are matrices of learnable parameters. $\{W_r^l \mid r \in \Re\}$ represents the parameter matrices of different edge types between nodes, and $\Re$ is the set of all possible ratings. The various edge patterns within the graph can be introduced into the GCN model using the message passing function described above.

To enhance the efficiency of message aggregation, the GCCN-EDP algorithm computes the power average of multiple embeddings to serve as the final representation h_i of node i:

$$\mathrm{h}_i = \left(\frac{1}{L}\sum_{l=1}^{L}(x_i^l)^2\right)^{1/2} \tag{7}$$

where L represents the number of message passing layers.

To pool the node representations into a graph-level feature vector, the GCCN-EDP algorithm adopts a pooling layer. The pooling layer concatenates the final representations of both the target user and the target item as the graph representation, and then a Multi-Layer Perceptron (MLP) layer is utilized to output the predicted rating, as exhibited in the following two formulas, respectively:

$$g = concat(h_u, h_v) \tag{8}$$

$$\hat{r}_{u,v} = w^{\mathrm{T}}\mathrm{ReLU}(Wg) \tag{9}$$

where h_u and h_v are the final representations of the target user u and the target item v, respectively; $\hat{r}_{u,v}$ represents the predicted rating, and w and W are two parameter matrices of MLP, which to map the graph representation g to the rating $\hat{r}_{u,v}$.

4.3. FDC-AB algorithm

The FDC-AB algorithm mainly adopts AdaBoost framework for fraudster detection. AdaBoost utilizes multiple weak classifiers for cascade operation, resulting in high classification accuracy. Additionally, AdaBoost differs from other bagging algorithms, including the Random Forest algorithm, by fully considering the weight of each weak classifier.

To detect whether a user u is fraudster or not, the FDC-AB algorithm firstly computes the MSE of rating prediction of u and generates dense representation h_u^* of u, then fraud detection classifier is trained.

For a user u, the root MSE $\mathbb{E}_u$ of all rating items $S_v(u)$ can be calculated as follows:

$$\mathbb{E}_u = \frac{1}{|S_v(u)|}\sum_{\forall v\in S_v(u)}(|r_{u,v} - \hat{r}_{u,v}|^2) \tag{10}$$

where $r_{u,v}$ represents the observed rating in the dataset and $\hat{r}_{u,v}$ represents the predicted rating calculated according to Eq (7).

Thus, we cascade the previously derived final user representation h_u with $\mathbb{E}_u$ to produce a fraud-detecting user representation $\mathrm{h}_u^{'}$. To achieve full distinguishability of the decision tree, $\mathrm{h}_u^{'}$ is passed through a fully connected layer resulting in a dense representation h_u^{*}, as depicted by the following two formulas:

$$\mathrm{h}_u^{'} = h_u \oplus \mathbb{E}_u \tag{11}$$

$$\mathrm{h}_u^{*} = Sigmoid(W_{z'} h_u^{'} + b_{z'}) \tag{12}$$

where $W_{z'}$ and $b_{z'}$ denote the weight and bias terms, respectively.

At the beginning of model training, the FDC-AB algorithm assigns the equal weight to each weak classifier. Training a weak classifier and if its classification result is correct, its corresponding weight should be reduced when constructing the next training set. Otherwise, its corresponding weight should be increased. This step is iterated until the end of the training process. Eventually, all the weak classifiers obtained are merged to form a strong classifier. In the final objective function, a weak classifier with smaller classification error rate has larger weight, so as to achieve fraud detection classification. In the classifier, the fraud identity label of a user u is labeled as y_u, u is recognized as a fraudster when $y_u = 1$, otherwise $y_u = 0$.

Supposing there are K decision trees, each one is a standard binary tree structure, and a decision tree $T_k (1 \le k \le K)$ contains two types of nodes: prediction nodes (leaf nodes) and decision nodes (non-leaf nodes). For a prediction node $n_p \in LS$ (LS is the set of leaf nodes), there exists a probability distribution ℓ_{p_y} of label y_u. For a decision node $n_q \in NLS$ (NLS is the set of non-leaf nodes), there exists a decision function $f_q(\mathrm{h}_u^{*};\theta)$ which decides whether to assign the received h_u^{*} to the left subtree or not, where θ is a parameter term. The decision function $f_q(\mathrm{h}_u^{*};\theta)$ generates a ground truth label, which is defined as follows:

$$f_q(\mathrm{h}_u^{*};\theta) = Sigmoid(\mathrm{w}_q^{\mathrm{T}} \mathrm{h}_u^{*}) \tag{13}$$

Hence, the classification probability of the decision tree T_k about user u is:

$$\mathrm{P}_{T_k}[y_u \mid \mathrm{h}_u^{*}, \theta, \ell] = \sum_{n_p \in LS} \ell_{p_y} \left(\prod_{n_q \in NLS} f_q(\mathrm{h}_u^{*};\theta)^{\Gamma_{left}} \bar{f}_q(\mathrm{h}_u^{*};\theta)^{1-\Gamma_{left}} \right) \tag{14}$$

where ℓ_{p_y} is the probability of label y on the prediction node n_p, $\bar{f}_q(\mathrm{h}_u^{*};\theta) = 1 - f_q(\mathrm{h}_u^{*};\theta)$, Γ_{left} indicates whether to visit the left subtree or not, with $\Gamma_{\text{left}} = 1$ for left subtree and $\Gamma_{\text{left}} = 0$ for right subtree.

For the decision tree T_k, its error rate e_k on the training set is defined as follows:

$$e_k = P(T_k(x_i) \neq y_i) = \sum_i w_{k,i} I(T_k(x_i) \neq y_i) \tag{15}$$

where TN represents the size of the training set, $w_{k,i}$ denotes the weight factor of training sample x_i, which has an initial value of $w_{1,i} = 1/TN$.

After obtaining e_k of the decision tree T_k, the decision weight α_k for the corresponding weak classifier can be calculated and the weight distribution of x_i can be updated:

$$\alpha_k = \frac{1}{2}\log\frac{1-e_k}{e_k} \tag{16}$$

$$w_{k+1,i} = \frac{w_{k,i}}{Z_k}e^{-\alpha_k y_i T_k(x_i)} \tag{17}$$

where Z_k denotes the normalization constant that satisfies $Z_k = 2\sqrt{e_k(1-e_k)}$.

Finally, these individual weak classifiers can be combined to form the strong classifier according to their weights

$$f_{\text{final}}(x) = \sum_{k=1}^{K}\alpha_k T_k(x) \tag{18}$$

Consequently, the classification probability y_u of user u is given as follows:

$$y_u(x) = \arg\max f_{final}(x) \tag{19}$$

4.4. Execution flow of the PPRFD-GC method

The loss function of the GCCN-EDP algorithm adopts the squared error to predict the loss, using the probability of classification as the normal user multiplied by the squared error term. And the loss function of the FDC-AB algorithm uses the Cross Entropy Loss function. Finally, these two loss functions are linearly combined to form the final loss function of the PPRFD-GC method. These three loss functions are respectively defined as follows:

$$L_{GCCN-EDP} = \frac{1}{|E'|}\sum_{edge(u,v)\in E'}\Pr[y_u = 0 \mid h_u^*, \theta, \ell] * (\hat{r}_{u,v} - r_{u,v})^2 \tag{20}$$

$$L_{FDC-AB} = \frac{1}{|U|}\sum_{\forall u\in U} -\log\Pr[y = y_u \mid h_u^*, \theta, \ell] \tag{21}$$

$$L = L_{GCCN-EDP} + \lambda L_{FDC-AB} \tag{22}$$

where $|E'|$ and $|U|$ are the number of edges and the number of users of the synthesized graph G', $edge(u,v) \in E'$ means there exists an edge between user u and item v, and λ is a hyperparameter to balance the effects of the two components.

The pseudo-code of the PPRFD-GC method is shown in the Algorithm 1.

Algorithm 1：The PPRFD-GC method

Input: original graph $G(V = U \cup It, E)$, clipping parameter C, privacy budget (ε, δ), noise scale σ, and the number of hops h

Output: rating prediction $\hat{r}_{u,v}$

// The GCCN-EDP algorithm

// **stage 1** synthesized graph construction

1: obtain the adjacency matrix A of G;

2: generate the latent representation Z of A according to Eq (4);

3: add Gaussian noise obeying $N(0, \sigma^2 C^2 \mathrm{I})$ to the gradients of the GCN-based decoder during the gradient update process;

4: reconstruct the adjacency matrix A' according to Eq (5);

5: generate the synthesized graph $G'(V, E')$ based on A';

6: **do**

// **stage 2** graph-level GNN training

7: **for each** user-item pair (u, v) **in** G'

8: generate h-hop enclosing subgraph $G'^{h}_{u,v}$;

9: train graph-level GNN based on $G'^{h}_{u,v}$ according to Eqs (6) to (8);

10: predict rating $\hat{r}_{u,v}$ according to Eq (9);

11: **end for**

// **The FDC-AB algorithm**

12: **for each** $u \in U$ // U is the set of users

13: calculate $error_u$ and h^*_u according to Eqs (10) to (12);

14: train the weak classifiers according to Eqs (13) to (16);

15: construct the strong classifier according to Eq (18);

16: calculate the classification probability y_u of user u according to Eq (19);

17: **end for**

18: **until** the loss function defined by Eq (22) converges.

From the perspective of privacy analysis, we briefly prove that the PPRFD-GC method satisfies (ε, δ)-edge differential privacy.

The PPRFD-GC method contains two algorithms: GCCN-EDP and FDC-AB. The GCCN-EDP algorithm consists of two phases. In the first phase, it follows the concept of the DPGGAN model, and applies the DP-SGD algorithm to generate the synthesized graph with the addition of Gaussian noise. This phase does not modify the DP-SGD algorithm, and thus satisfies (ε, δ)-edge differential privacy. The specific proof process can be referred to the literature [14]. The second phase of the GCCN-EDP algorithm and the FDC-AB algorithm both rely on the synthesized graph of the first stage, and according to the post-processing theorem of differential privacy, the second stage of the GCCN-EDP algorithm and the FDC-AB algorithm also satisfy the (ε, δ)-edge differential privacy. Therefore, the PPRFD-GC method satisfies (ε, δ)-edge differential privacy.

5. Experiments

5.1. Experimental settings

We implemented the proposed method in Python programming language (version 3.6.9) and

TensorFlow (version 1.6.0), and conducted experiments on a machine with Intel (R) i7-10700KF/3.8 GHz/64 GB hardware configuration and a 64-bit Windows 10 operating system.

Our experimental study utilized two real datasets, Yelp [28] and Amazon [29], which differentiate normal users from fraudsters with the help of labels. Table 1 outlines the statistics of the two datasets.

Table 1. Statistics of datasets.

Dataset	Users	Ratio of normal users and fraudsters	Items	Ratings
Yelp	32,393	(70%, 30%)	4670	293,936
Amazon	12,643	(70%, 30%)	4746	250,423

5.2. Experimental results

To evaluate the accuracy and robustness of the proposed method, we applied two metrics commonly used in recommendation system: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). A greater degree of accuracy in the recommendation system is reflected by lower values for these two metrics. Furthermore, the accuracy of ratings is affected by the presence of fraudster.

MAE: the mean value of the absolute error, indicating the average distance between the predicted value of the model and the true value of the sample. MAE is defined as follows:

$$\text{MAE}=\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i| \tag{23}$$

RMSE: measures the square root of the average squared difference between the estimated values and the actual values of a dataset. RMSE is used to measure the deviation of the estimated value from the true value and defined as below:

$$\text{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i-\hat{y}_i|^2} \tag{24}$$

To validate the effectiveness of the PPRFD-GC method, the proposed method is compared with the following methods:

PPRFD-GC2: the non-privacy-preserving version of the PPRFD-GC method;

DP-GCMC: a framework for message-passing graph autoencoders based on bipartite interaction graph [30], with a differentially private version of GCMC that adds Laplace noise during gradient descent process.

DP-GraphRec: a framework for social recommendation based on GNN [31], in which the user-item interaction module is used in the experiments, with a differentially private version of GraphRec that adds Laplace noise during gradient descent process.

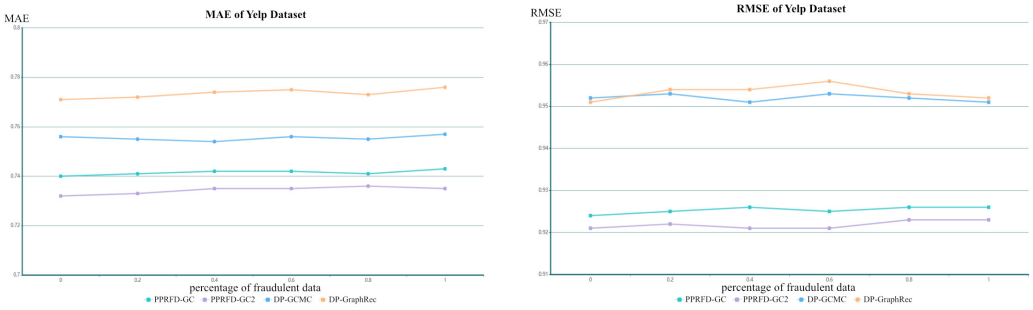
The PPRFD-GC method follows specific parameters: the GCCN-EDP algorithm incorporates a three-layer graph neural network connection with a hidden layer size of 100. The model parameters are initialized using a Gaussian distribution and updated through the Adaptive Moment Estimation (Adam) algorithm. The optimization value interval for both α and β is set at $[10^{-4},10]$, and the depth of the decision tree in the FDC-AB algorithm is set at 3.

In this section, two comparative experiments are conducted. The first experiment selects the real data of the original dataset to form the training dataset, and then inserts different percentages of the fraudulent

data of the original dataset into the training dataset for comparison. The second experiment inserts artificially constructed fake data into the training set and makes comparisons under fixed privacy budget. Specifically, hate attacks and random attacks are chosen to populate the training dataset. Hate attacks are used to generate as many rating extremes as possible and they are inserted into the dataset. While random attacks generate ratings randomly and insert them into the dataset. The experiments adopt a five-fold cross-validation strategy to test the accuracy of the above methods and take the average as the result.

5.2.1. Result analysis of the first experiment

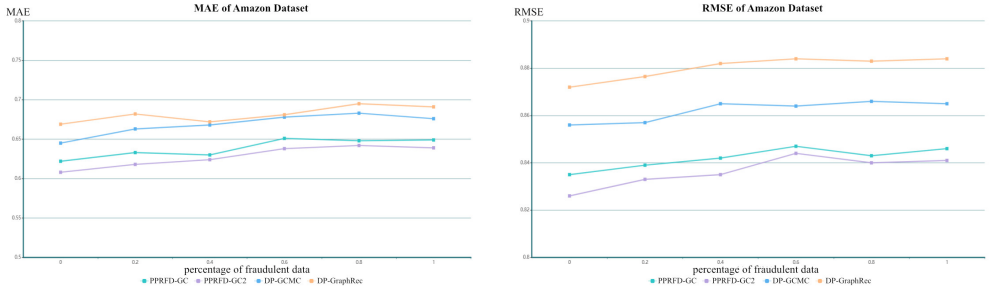
In this experiment, the privacy budgets of all three differentially private methods are set to ε=1. Figure 2 illustrates the experimental results on the Yelp dataset. The trend in two metrics, MAE and RMSE, of the PPRFD-GC method is slightly lower than that of the other two methods as the proportion of fraudulent data increases, and the non-privacy preserving baseline method, PPRFD-GC2, performs best. Overall, edge differential privacy has a minimal negative impact on data utility. The PPRFD-GC method surpasses the other two privacy-preserving methods, with an average improvement in MAE of 1.9% and 4.2% over DP-GCMC and DP-GraphRec, respectively; and an average improvement in RMSE of 2.8% and 3.0% over DP-GCMC and DP-GraphRec, respectively.



(a) MAE comparison (b) RMSE comparison

Figure 2. Comparison of MAE and RMSE on Yelp under different proportion of fraud data.

Figure 3 shows the experimental results on the Amazon dataset, where the PPRFD-GC method still outperforms the other two privacy-preserving methods, with an average MAE improvement of 4.6% and 6.4% over DP-GCMC and DP-GraphRec, respectively; and an average RMSE improvement of 2.3% and 4.4% over DP-GCMC and DP-GraphRec, respectively.

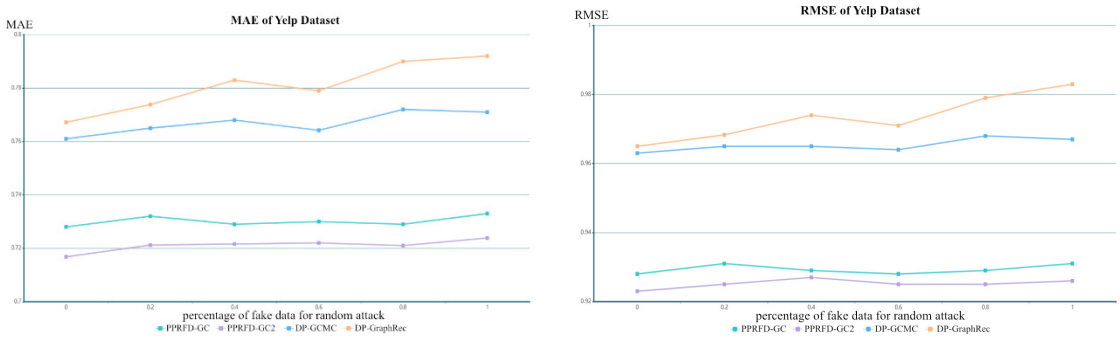


(a) MAE comparison (b) RMSE comparison

Figure 3. Comparison of MAE and RMSE on Amazon under different proportion of fraud data.

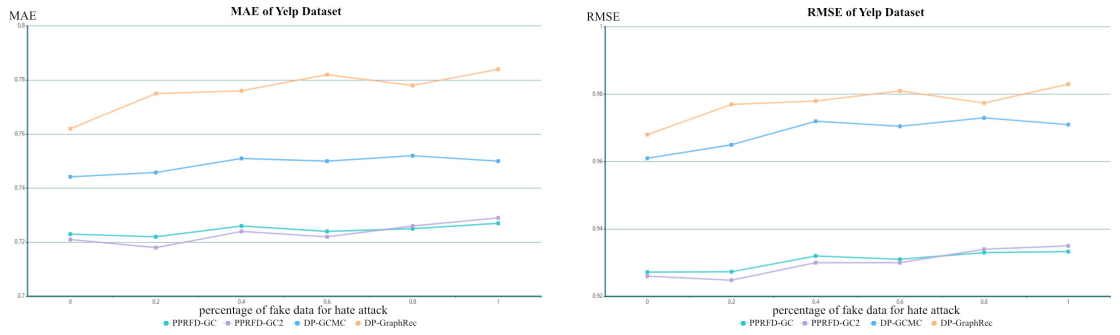
Further, it can also be clearly seen from Figures 2 and 3 that the trend line of each method basically keep a slightly upward trend. But with the rapid increase of fraud data, for the DP-GCMC method, the MAE value of Amazon and RMSE values of both Yelp and Amazon decrease. The DP-GraphRec method shows similar but minor phenomenon. And the proposed method shows a better resistance to fraudulent data.

Figures 4 and 5 illustrate the MAE and RMSE trends of the four methods on the Yelp dataset, after inserting different proportions of artificially constructed fake data. From the figures, the PPRFD-GC method achieves better protection against random attacks and hate attacks. With the increase of inserted fake data, the PPRFD-GC method outperforms the other two privacy-preserving methods in both MAE and RMSE. Additionally, the proposed method exhibits less fluctuation for these two metrics compared to the other two privacy-preserving methods. With the rapid increase of fake data, the RMSE value of proposed method is slightly lower than that of the PPRFD-GC2 method.



(a) MAE comparison (b) RMSE comparison

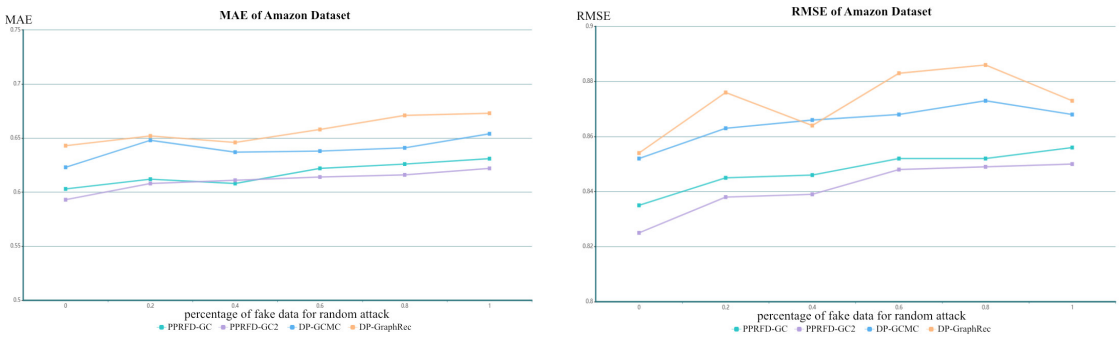
Figure 4. Comparison of MAE and RMSE on Yelp under random attack.



(a) MAE comparison (b) RMSE comparison

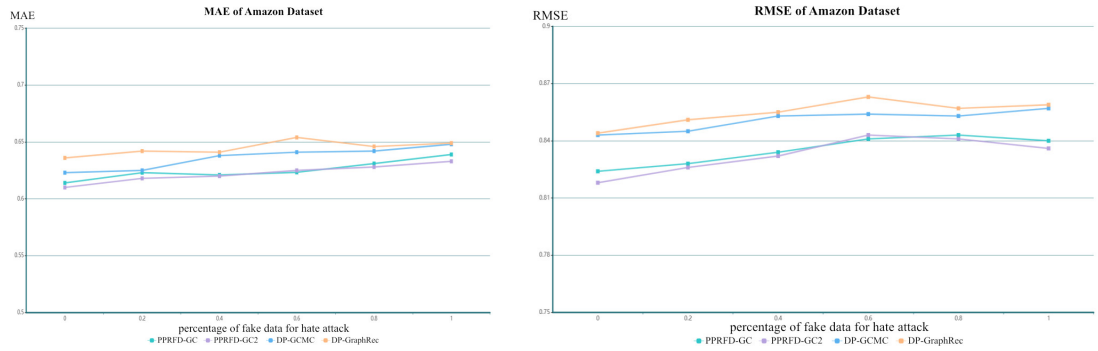
Figure 5. Comparison of MAE and RMSE on Yelp under hate attack.

Figures 6 and 7 show the trends in MAE values and RMSE values of the four methods on the Amazon dataset after inserting different proportions of artificially constructed fake data. Similar to the results on the Yelp dataset, the PPRFD-GC method outperforms the other two privacy-preserving methods in both MAE and RMSE, and the fluctuation of the proposed method for these two metrics is lower than that of the other two privacy-preserving methods. Moreover, the performance of proposed method on Amazon dataset is very close to that of the PPRFD-GC2 method.



(a) MAE comparison (b) RMSE comparison

Figure 6. Comparison of MAE and RMSE on Amazon under random attack.



(a) MAE comparison (b) RMSE comparison

Figure 7. Comparison of MAE and RMSE on Amazon under hate attack.

The findings of the two aforementioned experiments demonstrate that the proposed method is successful in mitigating recommendation distortion caused by fraudulent data while simultaneously maintaining satisfactory recommendation accuracy levels.

5.2.2. Ablation study

Here, we provide an ablation study that demonstrates the effectiveness of respective parts in the PPRFD-GC method. We compare the proposed method with three weaken versions, including: 1) replacing the two layer FNN-based decoder by a two layer GCN-based decoder in the GCCN-EDP algorithm; 2) replacing the graph based matrix completion by F-EAE [32] in the GCCN-EDP algorithm, and F-EAE is also a commonly used matrix completion technique which uses exchangeable matrix layers to perform inductive matrix completion; 3) replacing the group of weaker classifiers in the AdaBoost by a group of decision trees of random forest (RF) in the FDC-AB algorithm.

Two groups of comparative experiments are performed. The first group of experiments compares MAE value of above four versions of the proposed method while the second one compares RMSE value. In these two experiments, privacy budget ε is fixed to 1 and percentage of fraudulent data is set to 0.2, 0.4 and 0.6. The experimental results of the ablation study are reported in Tables 2 and 3, respectively.

Table 2. Ablation Study of the PPRFD-GC method on MAE.

	MAE of Yelp (ε = 1)			MAE of Amazon (ε = 1)		
Percentage of fraud data	0.2	0.4	0.6	0.2	0.4	0.6
PPRFD-GC	0.741	0.743	0.743	0.632	0.630	0.649
Weaken Version1	0.742	0.747	0.745	0.636	0.633	0.651
Weaken Version2	0.762	0.764	0.768	0.647	0.649	0.661
Weaken Version3	0.765	0.776	0.779	0.659	0.664	0.665

Table 3. Ablation Study of the PPRFD-GC method on RMSE.

	RMSE of Yelp (ε = 1)			RMSE of Amazon (ε = 1)		
Percentage of fraud data	0.2	0.4	0.6	0.2	0.4	0.6
PPRFD-GC	0.925	0.926	0.925	0.839	0.842	0.846
Weaken Version1	0.927	0.929	0.930	0.840	0.843	0.849
Weaken Version2	0.936	0.939	0.940	0.847	0.848	0.851
Weaken Version3	0.939	0.942	0.943	0.850	0.855	0.855

As shown in Tables 2 and 3, the original GCCN-EDP method always achieves the best performance. The performance of the first weaken version is very slightly lower than the original one, since FNN is more effective for static data than GCN. The performance of the second weaken version is medium, implying graph based matrix completion technique is suitable for our scenario. And the performance of the last weaken version is the worst, which may imply that assigning the same weight to each decision tree is not a good strategy.

6. Conclusions

To solve the risk of privacy leakage and the problem of fraud detection faced by the recommendation system, the PPRFD-GC method consisting of two algorithms: GCCN-EDP and FDC-AB, is proposed in the paper. According to users' rating data, the GCCN-EDP algorithm firstly construct a synthesized graph which satisfies edge differential privacy, then adopts graph based matrix completion technique for GNN based rating prediction. The FDC-AB algorithm is designed to detect fraudsters based on AdaBoost. By calculating the MSE of user's rating prediction and generating the dense representation of user as the input of the fraud detection classifier, a user can be classified into a fraudster or a normal user. Finally, the loss functions of these two algorithms are linearly combined to form the final loss function of the PPRFD-GC method.

In the future, we plan to continue the research in the following aspects: 1) The GCCN-EDP algorithm mainly takes users' rating data into consideration for constructing a single layer graph. To improve recommendation quality, more useful information such as social interactions should be utilized. The adoption of multiplex graph for GNN-based recommendation should be our future attention because each layer of the graph may contain different type of information; 2) The fraud detection part of the proposed method is currently applicable to offline/static scenarios. But most real-world networks evolve and fraudsters leverage their dynamics to evade detection, thus the design of our fraud detection solution should consider employing a time-evolving network structure to

continuously track suspicious activities across different time-based snapshots.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by "Key Research and Development Program of Shaanxi Province", grant number 2021GY-090.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. Lu, B. Pan, A. M. Seid, B. Li, G. Hu, S. Wan, Truthful incentive mechanism design via internalizing externalities and lp relaxation for vertical federated learning, *IEEE Trans. Comput. Social Syst.*, 2022. https://doi.org/10.1109/TCSS.2022.3227270
2. S. Liu, J. Yu, X. Deng, S. Wan, FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks, *IEEE Trans. Comput. Social Syst.*, **23** (2021), 1616–1629. https://doi.org/10.1109/TITS.2021.3099368
3. C. Wang, C. Jiang, J. Wang, S. Shen, S. Guo, P. Zhang, Blockchain-aided network resource orchestration in intelligent internet of things, *IEEE Internet Things J.*, **10** (2022), 6151–6163. https://doi.org/10.1109/JIOT.2022.3222911
4. J. Lu, H. Liu, R. Jia, J. Wang, L. Sun, S. Wan, Towards personalized federated learning via group collaboration in IIoT, *IEEE Trans. Ind. Inf.*, **19** (2022), 8923–8932. https://doi.org/10.1109/TII.2022.3223234
5. G. Wu, L. Xie, H. Zhang, J. Wang, S. Shen, S. Yu, STSIR: An individual-group game-based model for disclosing virus spread in Social Internet of Things, *J. Network Comput. Appl.*, **214** (2023), 103608. https://doi.org/10.1016/j.jnca.2023.103608
6. S. Shen, L. Xie, Y. Zhang, G. Wu, H. Zhang, S. Yu, Joint differential game and double deep q-networks for suppressing malware spread in industrial internet of things, *IEEE Trans. Inf. Forensics Secur.*, **18** (2023), 5302–5315. https://doi.org/10.1109/TIFS.2023.3307956
7. G. Wu, Z. Xu, H. Zhang, S. Shen, S. Yu, Multi-agent DRL for joint completion delay and energy consumption with queuing theory in MEC-based IIoT, *J. Parallel Distrib. Comput.*, **176** (2023), 80–94. https://doi.org/10.1016/j.jpdc.2023.02.008
8. G. Wu, H. Wang, H. Zhang, Y. Zhao, S. Yu, S. Shen, Computation offloading method using stochastic games for software defined network-based multi-agent mobile edge computing, *IEEE Internet Things J.*, **10** (2023), 17620–17634. https://doi.org/10.1109/JIOT.2023.3277541
9. G. Wu, X. Chen, Z. Gao, H. Zhang, S. Yu, S. Shen, Privacy-preserving offloading scheme in multi-access mobile edge computing based on MADRL, *J. Parallel Distrib. Comput.*, **183** (2024), 104775. https://doi.org/10.1016/j.jpdc.2023.104775

10. S. Shen, X. Wu, P. Sun, H. Zhou, Z. Wu, S. Yu, Optimal privacy preservation strategies with signaling Q-learning for edge-computing-based IoT resource grant systems, *Expert Syst. Appl.*, **225** (2023), 120192. https://doi.org/10.1016/j.eswa.2023.120192
11. H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah, Q. Kang, Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection, *Neurocomputing*, **407** (2020), 50–62. https://doi.org/10.1016/j.neucom.2020.04.078
12. Y. Xie, G. Liu, C. Yan, C. Jiang, M. Zhou, Time-aware attention-based gated network for credit card fraud detection by extracting transactional behaviors, *IEEE Trans. Comput. Social Syst.*, **10** (2022), 1004–1016. https://doi.org/10.1109/TCSS.2022.3158318
13. D. Wang, J. Lin, P. Cui, Q. Jia, Z. Wang, Y. Fang, et al., A semi-supervised graph attentive network for financial fraud detection, in *2019 IEEE International Conference on Data Mining (ICDM)*, IEEE, (2019), 598–607. https://doi.org/10.1109/ICDM.2019.00070
14. C. Yang, H. Wang, K. Zhang, L. Sun, Secure network release with link privacy, preprint, arXiv:2005.00455.
15. X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in *SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information*, (2020), 639–648. https://doi.org/10.1145/3397271.3401063
16. K. Mao, J. Zhu, X. Xiao, B. Lu, Z. Wang, X. He, UltraGCN: Ultra simplification of graph convolutional networks for recommendation, in *CIKM '21: Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, (2021), 1253–1262. https://doi.org/10.1145/3459637.3482291
17. J. Yu, H. Yin, J. Li, Q. Wang, N. V. Hung, X. Zhang, Self-supervised multi-channel hypergraph convolutional network for social recommendation, in *WWW '21: Proceedings of the Web Conference 2021*, (2021), 413–424. https://doi.org/10.1145/3442381.3449844
18. Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, et al., Pick and choose: A GNN-based imbalanced learning approach for fraud detection, in *WWW '21: Proceedings of the Web Conference 2021*, (2021), 3168–3177. https://doi.org/10.1145/3442381.3449989
19. Y. Shen, S. Shen, Q. Li, H. Zhou, Z. Wu, Y. Qu, Evolutionary privacy-preserving learning strategies for edge-based IoT data sharing schemes, *Digital Commun. Networks*, **9** (2023), 906–919. https://doi.org/10.1016/j.dcan.2022.05.004
20. S. Zhang, H. Yin, T. Chen, N. V. Hung, Z. Huang, L. Cui, Gcn-based user representation learning for unifying robust recommendation and fraudster detection, in *SIGIR '20: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, (2020), 689–698. https://doi.org/10.1145/3397271.3401165
21. X. Zheng, Z. Wang, C. Chen, J. Qian, Y. Yang, Decentralized graph neural network for privacy-preserving recommendation, in *CIKM '23: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, (2023), 3494–3504. https://doi.org/10.1145/3583780.3614834
22. C. Wu, F. Wu, Y. Cao, Y. Huang, X. Xie, Fedgnn: Federated graph neural network for privacy-preserving recommendation, preprint, arXiv:2102.04925.
23. Y. Xiao, L. Xiao, X. Lu, H. Zhang, S. Yu, H. V. Poor, Deep-reinforcement-learning-based user profile perturbation for privacy-aware recommendation, *IEEE Internet Things J.*, **8** (2020), 4560–4568. https://doi.org/10.1109/JIOT.2020.3027586

24. Z. Chen, Y. Wang, S. Zhang, H. Zhong, L. Chen, Differentially private user-based collaborative filtering recommendation based on k-means clustering, *Expert Syst. Appl.*, **168** (2021), 114366. https://doi.org/10.1016/j.eswa.2020.114366
25. T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, preprint, arXiv:1609.02907.
26. C. Dwork, A. Roth, The algorithmic foundations of differential privacy, *Found. Trends Theor. Comput. Sci.*, **9** (2014), 211–407. http://dx.doi.org/10.1561/0400000042
27. M. Zhang, Y. Chen, Inductive matrix completion based on graph neural networks, preprint, arXiv:1904.12058.
28. *Yelp Open Dataset*. Available from: https://www.yelp.com/dataset.
29. J. Ni, J. Li, J. McAuley, Justifying recommendations using distantly-labeled reviews and fine-grained aspects, in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, (2019), 188–197. https://doi.org/10.18653/v1/D19-1018
30. R. Berg, T. N. Kipf, M. Welling, Graph convolutional matrix completion, preprint, arXiv:1706.02263.
31. W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, et al., Graph neural networks for social recommendation, in *WWW '19: The World Wide Web Conference*, (2019), 417–426. https://doi.org/10.1145/3308558.3313488
32. J. Hartford, D. Graham, K. Leyton-Brown, S. Ravanbakhsh, Deep models of interactions across sets, in *Proceedings of the 35th International Conference on Machine Learning*, **80** (2018), 1909–1918. Available from: http://proceedings.mlr.press/v80/hartford18a/hartford18a.pdf.

AIMS Press

©2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)