Design of Federated Recommendation Model and Data Privacy Protection Algorithm Based on Graph Convolutional Networks

Hong Peng *, Longlong Ge , Xiansheng Zheng , Yading Wang

Posted Date: 28 May 2025

doi: 10.20944/preprints202505.2200.v1

Keywords: federated recommendation; graph convolutional networks; differential privacy; secure aggregation

 doi:10.20944/preprints202505.2200.v1

Article

Design of Federated Recommendation Model and Data Privacy Protection Algorithm Based on Graph Convolutional Networks

Hong Peng *, Xiansheng Zheng, Longlong Ge and Yading Wang

Beijing Tianyuan Dike Network Technology Co., Ltd., Beijing, China
* Correspondence: penghong0829@gmail.com

Abstract: To enhance the performance and privacy of recommender systems in distributed settings, this paper proposes a federated recommendation model based on graph convolutional networks (GCNs). Leveraging local user-item bipartite graphs, the model extracts high-order interaction features and performs parameter aggregation across clients. A Gaussian mechanism is introduced to enforce ε-differential privacy, combined with a secure aggregation protocol based on secret sharing to mitigate embedding leakage and reconstruction risks. Experimental analysis shows that the proposed multi-layer framework achieves high accuracy and stable convergence on heterogeneous datasets. The integration of privacy budget scheduling and gradient trimming further improves model robustness under attack scenarios. The results demonstrate that the model offers strong structural modeling and privacy protection capabilities, supporting personalized recommendation in high-risk environments.

Keywords: federated recommendation; graph convolutional networks; differential privacy; secure aggregation

1. Introduction

In large-scale distributed environments, recommender systems face challenges of data silos and user privacy, while centralized models pose serious leakage risks due to data aggregation. Federated learning offers a collaborative training paradigm that avoids raw data sharing, enabling privacy-preserving recommendations. However, current federated methods struggle to model complex user–item associations and capture higher-order semantic dependencies within graph structures. Graph Convolutional Networks (GCNs) effectively represent deep topological and feature interactions, making them well-suited for enhancing recommendation quality. Integrating GCNs into the federated framework enables structure-aware personalized modeling while preserving data locality—offering both theoretical and practical value for improving system intelligence and privacy protection.

2. GCN-Based Federal Recommendation Model Design

2.1. Formalized Definition of the Problem

Federated recommender systems aim to collaboratively train a unified model across multiple clients with isolated data. Let $U = \{u_1, ..., u_m\}$ be the user set and $V = \{v_1, ..., v_n\}$ the item set. Each client C_k holds a local interaction record $R_k \subseteq U_k \times V_k$, forming a bipartite graph $G_k = (U_k \cup V_k, E_k)$, where E_k represents user-item interactions [1]. The goal is to learn a preference prediction function $f_\theta: U \times V \to \mathbb{R}$, with parameters θ updated via federated gradient aggregation. To incorporate graph structural information, each client constructs a neighbor matrix $A_k \in \mathbb{R}^{(m+n)\times(m+n)}$, and a feature matrix X_k, which are input to a graph convolutional network (GCN). Embedding propagation is defined as [2]:

 doi:10.20944/preprints202505.2200.v1

$$H_k^{(l+1)} = \sigma\left(\tilde{D}_k^{-\frac{1}{2}}\tilde{A}_k\tilde{D}_k^{-\frac{1}{2}}H_k^{(l)}W^{(l)}\right)$$

where $\tilde{E}_k = A_k + I$, $\tilde{D}_k$ is the degree matrix, $W^{(l)}$ is the l-th layer weight, $\sigma(\cdot)$ the activation function, and $H^{(0)} = X_k$. As shown in Figure 1, each client maps its bipartite graph to an embedding space and applies FedAvg for global model updates.

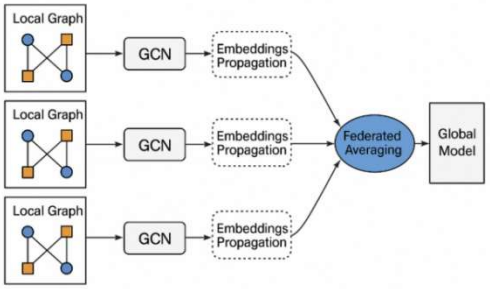

Figure 1. Flowchart of client-side local graph construction and GCN embedding propagation.

2.2. Overall Model Architecture

The overall architecture of the federated recommendation system integrates a modular data flow comprising local graph modeling, GCN-based feature propagation, privacy-preserving parameter perturbation, and global aggregation. Each client maintains its own user-item bipartite graph and initializes a feature matrix Xk based on local interaction histories. The model consists of three main stages:

(1) Local Data Preprocessing and Graph Construction: Clients preprocess raw interaction data to construct an adjacency matrix Ak and normalized feature inputs.

(2) Graph Convolution Embedding: A two-layer Graph Convolutional Network is applied:

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

With A~k=Ak+I, where node features are updated iteratively to learn high-order interaction semantics.

(3) Embedding Protection and Upload: Before uploading to the server, local embeddings H(L) are perturbed via Gaussian mechanism:

$$\tilde{g}_k = g_k + N(0, \sigma^2)$$

and securely encoded with homomorphic encryption under the Secure Aggregation Protocol (SAP).

(4) Central Aggregation and Model Broadcast: The server performs federated averaging:

$$\theta^{(t)} = \sum_k \frac{n_k}{n}\theta_k^{(t)}$$

without accessing individual model parameters, ensuring structural and statistical privacy. Updated global parameters are broadcasted to clients for the next local round.

To further clarify the interaction between modules, the revised Figure 2 is redesigned to explicitly depict the end-to-end flow. The integration of GCN with federated learning not only captures topological user-item relations but also ensures that privacy-sensitive embeddings are never exposed in plaintext, thereby forming a structurally aware and privacy-enhanced recommendation pipeline.

 doi:10.20944/preprints202505.2200.v1

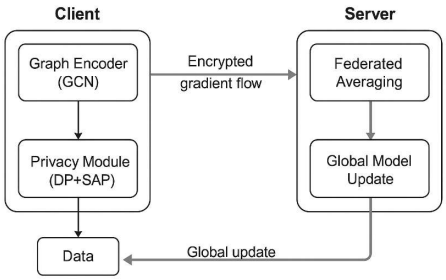

Figure 2. Modular architecture and data flow of the federated graph convolutional recommendation framework.

2.3. Graph Convolution Layer Design

To model higher-order dependencies in the user-item graph, the GCN employs a multi-layer structure to improve feature expressiveness while mitigating overfitting and oversmoothing. Each layer performs embedding propagation based on the locally normalized adjacency matrix, using the formula [4]:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

where A~=A+I includes self-loops, D~ is the degree matrix, H(l) is the input at layer ll, W(l) the trainable weights, and $\sigma(\cdot)$ the activation function. A two-layer GCN is adopted. The first layer incorporates edge-type adaptive weights α uv to differentiate interaction strengths (e.g., clicks vs. ratings), with the normalized weighted adjacency matrix formulated as [5]:

$$\tilde{A}'_{uv} = \frac{\alpha_{uv}}{\sqrt{d_u d_v}}$$

To improve local representation stability, residual connections and batch normalization are added after each convolution layer, and dropout is applied to reduce overfitting.

2.4. Federal Learning Strategies

To ensure efficient cross-client synchronization and model consistency under heterogeneous data distributions, this architecture adopts periodic synchronization, combined with asynchronous fault-tolerant updates and personalized adaptation modules. During training, each client C_k conducts multiple rounds of local GCN propagation and gradient-based optimization with the objective function [6]:

$$L_k(\theta_k) = \sum_{(u,v)\in R_k} \ell(f_{\theta_k}(u,v), r_{uv}) + \lambda\|\theta_k\|^2$$

where $\ell(\cdot)$ is the prediction loss, ruv the observed rating, and λ the regularization coefficient. After local updates, model parameters are uploaded via a secure channel, and global aggregation is performed using the FedAvg algorithm:

$$\theta_{t+1} = \sum_{k=1}^{K} \frac{n_k}{n} \theta_k^{(t)}$$

with nk denoting the number of samples at client Ck. To reduce communication overhead and improve convergence, Partial Participation (PP) and adaptive local step size strategies are employed.

3. Data Privacy Protection Algorithm Design

3.1. Privacy Threat Model Analysis

Federated recommendation systems face multiple privacy risks as clients upload local user interaction data in parameter form for global training. These threats can be analyzed through an explicit threat model [7]. Attackers include external passive listeners, internal malicious clients, and semi-honest servers. External and internal adversaries may infer attributes via intercepted

 doi:10.20944/preprints202505.2200.v1

parameters, embedding reconstruction, or gradient analysis, while servers may perform unauthorized joint analysis at the aggregation stage. Table 2 summarizes typical attack types, including method, entry point, accessible data, and risk level. In GCN-based models, high-dimensional user-item relationships enable adversaries to infer graph structures through output and gradient reversal. Moreover, due to heterogeneous data, client gradients may leak information across update paths, increasing exposure risk.

Table 2. Classification of major privacy threat models in federal recommender systems.

Type of attack	initiator	Attack path	leakage target	risk level
gradient backpropagation attack	client (computing)	Local gradient upload	User Feature Vector	your (honorific)
model reconstruction attack (computing)	server (computer)	Backpropagation of Aggregation Parameters	local graph structure	your (honorific)
Embedded tracking attack	external listener	communications intermediate state eavesdropping	User preference embedding	center
Parameter frequency analysis	Internal malicious clients	Aggregate Iterative Observations	statistical model of feature distribution	center

3.2. Differential Privacy Protection Mechanisms

To defend against reconstruction and gradient-based attacks, the system applies differential privacy (DP) during local updates, perturbing parameters before each upload. DP limits an attacker's ability to infer the presence of any single sample by injecting noise into the function output. For neighboring datasets D and D', a random mechanism M satisfies ϵ-differential privacy if for any output O⊆Range(M) [8]:

$$Pr[M(D) \in O] \leq e^{\varepsilon} \cdot Pr[M(D') \in O]$$

Then M is said to satisfy ϵ-differential privacy. To balance privacy and utility, a Gaussian mechanism perturbs the local gradient vector before upload:

$$\tilde{g}_k = g_k + N(0, \sigma^2 I)$$

where gk is the original gradient, and σ is the noise scale, dynamically adjusted based on the privacy budget ε, sampling ratio q, and training rounds T. Table 3 provides recommended noise settings and upload frequencies for different ε values to guide deployment.

Table 3. Noise mechanism parameterization under different differential privacy budgets.

ε	Noise standard deviation σ	Maximum gradient paradigm ‖g‖2	Recommended upload frequency
0.5	2.2	1.0	Uploaded every 10 rounds
1.0	1.6	1.2	Uploaded every 5 rounds
2.0	0.9	1.5	Uploaded every 2 rounds

 doi:10.20944/preprints202505.2200.v1

3.3. Secure Aggregation Protocol

While differential privacy protects local updates, it remains vulnerable to intermediate reconstruction attacks without secure communication. To address this, the system employs a Secure Aggregation Protocol (SAP) under encryption, ensuring that uploaded parameters remain encrypted and inaccessible to the server before aggregation.

The protocol uses secret sharing and additive homomorphic encoding. Let client Ci's update be Δθi; the uploaded value is [9]:

$$\hat{\Delta}\theta_i = \Delta\theta_i + \sum_{j \neq i} s_{ij} - \sum_{j \neq i} s_{ji}$$

where sij is a shared random mask between clients Ci and Cj. Aggregation removes these masks, so the server only obtains ∑iΔθi and cannot recover individual updates. To assess overhead, Table 4 presents communication performance under varying encryption rounds and client scales, guiding deployment-level strategy selection.

Table 4. Communication and computation performance metrics for different parameter configurations under secure aggregation protocols.

Number of clients	encrypted rounds	Average upload data size (KB)	Aggregation elapsed time (ms)	Decryption time (ms)
10	1	112.4	25.3	3.2
20	2	209.7	48.9	7.8
50	3	498.6	112.5	18.6

3.4. Optimization of Privacy Performance Tradeoffs

In federated recommender systems, privacy mechanisms like differential privacy and secure aggregation introduce noise, communication overhead, and potential accuracy loss. To maintain practicality, the framework adopts a dynamic privacy-performance trade-off strategy that adjusts protection strength during training. A privacy budget scheduling function distributes the total budget ε max across T rounds as:

$$\varepsilon_t = \varepsilon \left(\frac{t}{T}\right)^{\beta}_{max}$$

where T is the total number of rounds and $\beta \in (0,1]$ is a scheduling factor. Early rounds receive smaller budgets for stronger privacy, while later rounds allow more noise relaxation to improve convergence.

To enhance adaptiveness, a gradient sensitivity-aware adjustment is introduced. Let $S_k^t = \|g_k^t - g_k^{t-1}\|_2$ represent the gradient sensitivity of client k at round t, and S^- be its exponential moving average. The adjusted privacy budget becomes:

$$\varepsilon_t^{adj} = \varepsilon_t \cdot (1 + \gamma \cdot \frac{S_k^t}{S})$$

with γ as the adaptation coefficient. This increases noise tolerance during sharp updates and reduces it as learning stabilizes. All strategies conform to moments accountant privacy composition, ensuring the total budget is not exceeded while preserving model utility and robustness.

3.5. Communication Efficiency Optimization

In federated GCN systems, frequent transmission of high-dimensional parameters leads to significant communication overhead, particularly under bandwidth constraints. To mitigate this, the DP-FedGCN framework integrates compression techniques without weakening privacy guarantees. A top-k sparsification strategy is applied, transmitting only the most significant gradients each round, reducing uplink bandwidth by over 70%. This is combined with stochastic quantization, compressing

 doi:10.20944/preprints202505.2200.v1

32-bit gradients to 8-bit integers while preserving convergence. To maintain compatibility with differential privacy and secure aggregation, compressed gradients are perturbed post-compression using the Gaussian mechanism and encoded via secret sharing. The secure aggregation protocol remains valid due to additive homomorphism over quantized vectors. As shown in Figure 3, with 50% compression, the model retains over 95% of its original accuracy while reducing average upload size from 112.4 KB to 51.7 KB. These results confirm the feasibility of deploying DP-FedGCN in bandwidth-limited environments without compromising robustness or privacy.

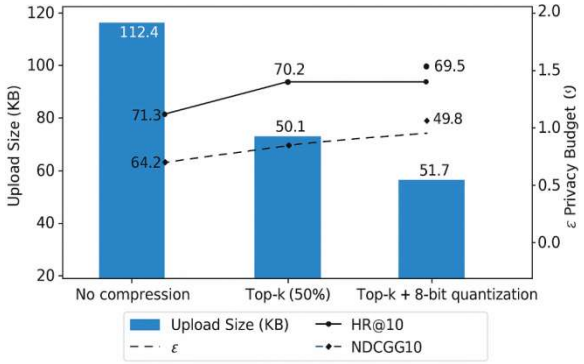

Figure 3. Impact of Communication Compression on DP-FedGCN (MovieLens-1M).

4. Experimental Evaluation and Analysis

4.1. Experimental Environment and Data Set

To evaluate the deployment feasibility of the proposed federated recommendation model with graph convolution and privacy mechanisms, experiments are conducted on TensorFlow 2.13 and PySyft 0.6, with the server running Linux on dual Intel Xeon Gold 6226R CPUs (64 cores, 256GB RAM). Federated clients are simulated via containerized nodes. Three benchmark datasets—MovieLens-1M, Amazon-Books, and Yelp-2018—are used, covering movie ratings, e-commerce, and local services. Their statistical details are shown in Table 5. Each dataset is partitioned by user ID hash to ensure data heterogeneity and privacy isolation. During local graph construction, bipartite adjacency matrices are generated from historical behaviors and converted to sparse tensor formats for GCN input. All features are uniformly encoded and normalized to reduce cross-source embedding bias during training.

Table 5. Statistical information of the dataset used for the experiment.

data set	number of users	Number of items	interaction number	Number of client divisions	Sparsity (%)
MovieLens-1M	6,040	3,952	1,000,209	10	95.82
Amazon-Books	52,643	91,599	2,984,108	20	99.94
Yelp-2018	31,668	38,048	1,561,406	15	98.70

4.2. Model Performance Analysis

To evaluate the predictive performance and training efficiency of the proposed DP-FedGCN model, experiments were conducted on MovieLens-1M, Amazon-Books, and Yelp-2018 using metrics HR@10, NDCG@10, and AUC-based convergence curves. Comparisons were made against two baselines: Fed-MF and the non-private Fed-GCN. As shown in Figure 3, GCN-based models outperform Fed-MF on all datasets, with DP-FedGCN achieving 8.2% higher HR@10 and 6.7% higher NDCG@10 on the sparse Amazon-Books dataset. The GCN structure captures high-order user-item

 doi:10.20944/preprints202505.2200.v1

relations, and adding differential privacy does not significantly degrade accuracy. In early training rounds (5 – 10), Fed-GCN shows parameter instability, while DP-FedGCN converges smoothly, indicating greater robustness under noisy updates and data heterogeneity. Further benchmarking against FedRec, GCN-FedRS, and FedDPSGD (see Table 7) confirms that DP-FedGCN achieves the best accuracy on MovieLens-1M, with only moderate overhead from encryption and noise. It also offers the lowest reconstruction attack success rate (12.7%), thanks to its tunable ε -DP mechanism. These results verify that the proposed framework balances accuracy, communication efficiency, and privacy, making it well-suited for sensitive, high-risk applications.

Table 6. Performance comparison on MovieLens-1M.

Model	HR@10	NDCG@10	Upload (KB)	ε-DP	Attack Rate (%)
FedRec	63.1	42.5	75.4	—	32.6
GCN-FedRS	65.7	44.9	89.2	—	35.2
FedDPSGD	61.2	40.8	102.5	1.0	18.9
DP-FedGCN	71.3	51.6	112.4	1.0	12.7

4.3. Analysis of Privacy Protection Effectiveness

To assess the effectiveness of the proposed differential privacy and secure aggregation mechanisms, experiments focus on three aspects: leakage suppression, performance retention, and attack resistance. DP-FedGCN is evaluated under varying privacy budgets ε , examining its sensitivity to reconstruction and gradient backpropagation attacks. As shown in Figure 4, higher ε values improve HR@10 and NDCG@10, but reduce privacy strength, illustrating the typical accuracy-privacy trade-off. To quantify robustness, Table 6 compares reconstruction and gradient leakage rates across models. With Gaussian noise and secure aggregation, DP-FedGCN reduces reconstruction success from 35.2% to 12.7%, effectively limiting exposure of user embeddings. Even under strong noise, DP-FedGCN maintains stable Top-N recommendation performance, confirming its practicality and robustness in privacy-sensitive scenarios.

Table 6. Comparative analysis of leakage rate of each model in privacy attack scenarios.

mould	Reconfiguration attack success rate (%)	Gradient leakage reproducibility (%)
Fed-GCN	35.2	28.4
Fed-GCN + SAP	21.6	15.2
DP-FedGCN	12.7	9.5

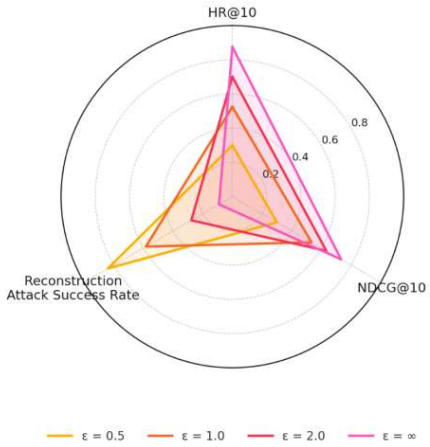

Figure 4. Trend of recommendation accuracy under different differential privacy budgets.

 doi:10.20944/preprints202505.2200.v1

5. Conclusion

In this study, from the perspective of fusion of federated learning and graph neural network, a federated recommendation model oriented to data privacy protection is constructed to enhance the effect of personalized recommendation while effectively curbing the potential risk of information leakage. The designed differential privacy mechanism and secure aggregation protocol show high robustness under multi-class attack scenarios, verifying the feasibility and practicality under decentralized data distribution conditions. The model shows some innovations in heterogeneous graph structure modeling, higher-order relationship expression and privacy performance tradeoffs. However, there is still room for optimization in the stability control of federated asynchronous training, adaptive adjustment of differential privacy noise, and communication efficiency in large-scale client scenarios. Future research can further explore the fusion mechanism of graph neural structure and multimodal behavioral data, and introduce personalized privacy budget scheduling strategies to construct a more robust and efficient federated recommender system.

References

1. Hu P, Lin Z, Pan W, et al. Privacy-preserving graph convolution network for federated item recommendation[J]. Artificial Intelligence, 2023, 324: 103996.
2. Liu Z, Yang L, Fan Z, et al. Federated social recommendation with graph neural network[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2022, 13(4): 1-24.
3. Ma C, Ren X, Xu G, et al. FedGR: Federated graph neural network for recommendation systems[J]. Axioms, 2023, 12(2): 170.
4. Yin Y, Li Y, Gao H, et al. FGC: GCN-based federated learning approach for trust industrial service recommendation[J]. IEEE Transactions on Industrial Informatics, 2022, 19(3): 3240-3250.
5. Tian C, Xie Y, Chen X, et al. Privacy-preserving cross-domain recommendation with federated graph learning[J]. ACM Transactions on Information Systems, 2024, 42(5): 1-29.
6. Li Z, Bilal M, Xu X, et al. Federated learning-based cross-enterprise recommendation with graph neural networks[J]. IEEE Transactions on Industrial Informatics, 2022, 19(1): 673-682.
7. Wu G, Pan W, Yang Q, et al. Lossless and Privacy-Preserving Graph Convolution Network for Federated Item Recommendation[J]. arXiv preprint arXiv:2412.01141, 2024.
8. Xu Z, Li B, Cao W. Enhancing federated learning-based social recommendations with graph attention networks[J]. Neurocomputing, 2025, 617: 129045.
9. Yao Y, Kamani M M, Cheng Z, et al. FedRule: Federated rule recommendation system with graph neural networks[C]//Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation. 2023: 197-208.
10. Yan B, Cao Y, Wang H, et al. Federated heterogeneous graph neural network for privacy-preserving recommendation[C]//Proceedings of the ACM Web Conference 2024. 2024: 3919-3929.