Information Sciences 638 (2023) 118915

Contents lists available at ScienceDirect

Information Sciences

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

MuKGB-CRS: Guarantee privacy and authenticity of cross-domain recommendation via multi-feature knowledge graph integrated blockchain

Li-e Wang [a,b], Yuelan Qi [b], Yan Bai [c], Zhigang Sun [b,*], Dongcheng Li [a,b], Xianxian Li [a,b,*]

[a] Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin, China
[b] Guangxi Key Laboratory of Multi-source Information Mining & Security, College of Computer Science and Information Engineering, Guangxi Normal University, Guilin 541004, China
[c] School of Engineering and Technology, University of Washington Tacoma, Tacoma, WA, USA

ARTICLE INFO

Keywords:
Cross-domain recommendation
Knowledge graph
Privacy protection
Blockchain
Trusted sharing
Data fusion

ABSTRACT

In distributed cross-domain recommendation systems, current privacy protection techniques are ineffective at verifying data correctness and existing authenticity techniques have limitations in privacy protection. This poses a conflict between security and credibility. This paper designs a trusted cross-domain recommendation model based on **mu**lti-feature **k**nowledge **g**raph (MuKG) and blockchain to boost recommendation accuracy, data credibility and security. To the best of our knowledge, MuKG is proposed, for the first time, to realize a unified representation of multi-source heterogeneous data in a secure manner. Under this model, federated learning integrated with blockchain is used to implement a co-trust mechanism for distributed learning while guaranteeing the credibility of data through blockchain traceability. Our model can guarantee both data security and authenticity with no need of generalization for privacy protection and coordination of centralized servers for distributed controls. Experiments are performed on the two classic datasets, MovieLens and Amazon, with different sparsity. The results have shown that MuKG improves the recommendation accuracy by 1.5 % and diversity by 18 % while ensuring data security and credibility in sparse data. We also verify that different data characteristics have different influence on the recommendation results.

1. Introduction

Recommendation systems can effectively handle the explosion of information. However, they face the issues of data sparsity and cold start as user scale and number of items rapidly grow. Due to the lack of sufficient data to construct accurate user portraits, recommendation quality is greatly compromised, and the recommendation results are heavily biased. Cross-domain recommendation can address the aforementioned issues by fusing auxiliary data. It leverages the data correlation between target domain and auxiliary domains to obtain valid knowledge so as to enrich or complement the target domain's data. Based on the multidimensional data from different sources, a more accurate user portrait can be constructed to accurately predict the user's behavior. Consequently, the problem

* Corresponding authors.
E-mail addresses: szung@163.com (Z. Sun), lixx@gxnu.edu.cn (X. Li).

https://doi.org/10.1016/j.ins.2023.03.140
Received 30 December 2022; Received in revised form 14 March 2023; Accepted 26 March 2023
Available online 30 March 2023
0020-0255/

of data sparsity and cold start are alleviated. Apparently, users' data in different domains can be directly or indirectly reflect users' interest from different perspectives. For example, friends in social networking domains normally have similar interests. Users who enjoy sports often want to buy sportswear and fitness products; and users who love food often sign up for POIs such as restaurants or food web celebrity points. These data in different domains are correlated, crossed and fused with complex features such as high dimensionality, heterogeneity, dynamics and cross correlation. However, user data contains a large number of sensitive data, including individual information (e.g., personal address, phone number.) and group privacy (e.g., marketing secrets of various platforms). The correlations between cross-domain data considerably increase attackers' background knowledge, making cross-domain recommendations subject to more severe privacy issues. Moreover, data provided by different participants may not be trustworthy due to privacy concerns or selfish nature in a cross-domain context. If different participants provide false or fake data, it causes a skewness in a recommendation model and reduces the recommendation quality. Therefore, protecting privacy while verifying data authenticity effectively in an open distributed environment is a problem that needs to be solved in the current recommendation domain.

The main issues are as follows: (1) The privacy problems in cross-domain recommendations are more challenging due to diversified sensitive information and multiple privacy associations. The associations between multi-source data are characterized by diversity and uncertainty, which are critical for cross-domain recommendation. However, it also considerably increases the attackers' background knowledge and is conducive to cyber attacks. (2) The open distributed scenarios in cross-domain recommendations require verifying data authenticity while guaranteeing data privacy. Present solutions are limited by considering only one aspect or the other, not both. Current anonymous methods typically employ data modification to hide the association between sensitive attributes and individuals' identities, which makes it impossible to verify data authenticity. While existing verification methods mainly use the distributed access control and cryptography to verify data integrity. Essentially, they are identification-based methods to implement data authenticity guarantees and consequently, leading to disclosure of users' private information.

To the best of our knowledge, recent solutions in addressing both privacy and credibility issues are still in their infancy stage. For instance, Li et al. [1] proposed a QoS-Aware Web service Recommendation framework based on Blockchain to guarantee credibility. They separated the traditional Matrix Factorization model into two disjoint parts of private factors and public factors, and trained the public factors collaboratively while keeping the private factors private. This method only considers the case of single-domain prediction, which is difficult when being applied to dynamic and decentralized scenarios. Zhou et al. [2] proposed a location recommendation based on the homomorphic property of Paillier cryptosystem without assuming any trustworthy entity. They devised three location recommendation protocols over ciphertexts to protect the privacy of user check-in data. The location recommendation server calculated the recommendation results without knowing any explicit user check-in information. However, this method does not verify data authenticity. Zhang et al. [3] presented a platoon recommendation scheme which avoids selecting the malicious vehicles among potential user vehicles in a privacy-preserving way. Each user vehicle holds a trust value computed via a truth discovery process. Meanwhile, a pseudonym and Paillier cryptosystem is applied to preserve vehicles' privacy. However, the truth discovery process and the encryption algorithm require additional computational power, and limit the types of computations that can be performed for recommendation analysis. Clearly, it is not suitable for dynamic cross-domain scenarios.

In light of the above, this paper focuses on the privacy and authenticity of recommendation systems for cross-domain scenarios. Specifically, we propose a novel framework for trusted cross-domain recommendation with multiple guarantees of privacy protection and authenticity verification, by constructing a unified representation of complex knowledge integrating federated learning and blockchain. On the one hand, we introduce a concept of **mu**lti-features **k**nowledge **g**raph (MuKG for short) with privacy protection by replacing users with users' features. It provides a secure and unified representation of multiple knowledge by hiding individuals' sensitive information. MuKG, with multi-domain collaboration, enables effective analysis for distributed recommendations while protecting users' privacy and enhancing the interpretability of the recommendation results. On the other hand, blockchain techniques are introduced to resist poisoning attacks so as to ensure that the operation records cannot be tampered with, and thus, enabling traceability of upstream data and ensuring the authenticity and integrity of data source.

The main contributions of this work can be summarized as follows.

1. The concept of MuKG is proposed for the first time to realize unified knowledge representation of multi-source heterogeneous data with privacy protection and incremental updates.
2. Based on MuKG, we also propose a distributed secure collaborative computing method integrated with federated learning. It can store users' data locally and only upload users' features. Different from traditional federated learning, we adopt feature encryption with statistical weights instead of gradient encryption to guarantee both data privacy and flexibility in recommendation calculation.
3. Moreover, we design a blockchain-based multi-party trust verification mechanism to ensure data authenticity by storing data operational records on the chain in a secure and tamper-proof manner. A consensus mechanism based on recommendation server (POR) is designed for better efficiency.
4. Furthermore, we present a personalized recommendation strategy based on MuKG and convolutional neural network (MuKGCN) to improve the quality of recommendation results. We verify that our method can improve the recommendation accuracy by 1.5% and diversity by 18% while ensuring data security and credibility when handling sparse data.

The rest of the paper is organized as follows. Related works on privacy and authenticity in recommendation systems are reviewed in Section 2. The preliminaries of our work are presented in Sections 3. Our solution and algorithm are described in Sections 4. Performance and experimental analysis are reported in Section 5 and 6. Finally, Section 7 concludes this paper.

2. Related work

2.1. Privacy problem of recommendation systems

As recommendation system is a tool to solve the problem of users' information overloading, it needs to be open to the users. Consequently, a malicious attacker can infer users' privacy by exploiting the relationships between multiple data inputs and recommendation results. When the number of users and items increases, users' personalized data is crucial for recommendation quality. How to balance data security and recommendation quality became an important research question. Existing privacy protection approaches can be classified into categories of anonymization-based, encryption-based, federated learning-based, and blockchain-based.

2.1.1. Security recommendations based on anonymization

Existing anonymization technologies mainly include generalization [4], obfuscation [5], perturbation[6], differential privacy[67] and local differential privacy(LDP)[8]. However, these methods based on data modification cannot resist attribute inference attacks. The information loss caused by modification operation largely affects the accuracy of recommendation results. Beigi et al. [9] proposed a property protection recommended (RAP) model to address private property attacks and develop Bayesian inference personalized recommendation. Assuming that an attacker attempts to infer a user's private attribute information based on a list of user's entries and recommended results, the RAP model takes advantage of an attacker component to regulate the recommendation process while extracting the user's interests, thus protecting the user from inference attacks on private attributes during recommendation. Zhou et al. [10] proposed a USST framework to ensure the privacy of distributed recommendation with two phases. It first trains a shared model with a small set of records contributed by sampled users, and then trains a personalized model using individual user's information. This method can provide the strongest level of protection for all unsampled users.

The above related works mainly focus on the privacy issues of single-domain recommendation systems. They are not applicable to privacy protection in cross-domain recommendation scenarios. The privacy issues involving multiple data sources are more challenging as they involve not only privacy issues in the data fusion process between different domains, but also security of the inter-domain auxiliary information transmission [11]. Gao et al. [12] aimed at the privacy risk of data sharing in cross-domain recommendation. They proposed a neural attention transfer recommendation that only shares one side of the items, and establishes target domain association based on overlapping items to achieve cross-domain transfer. The method can extract useful information from item transfer and perform auxiliary prediction analysis on the target domain to effectively alleviate the data sparsity problem. However, this method is only applicable to cross-domain scenarios with overlapping items. Additionally, there is a risk of privacy leakage during domain transfer.

2.1.2. Security recommendations based on data encryption

Encryption or coding techniques are mainly used to hide sensitive data for privacy protection. Presently, there are two main types of applications in these areas: homomorphic encryption and locally sensitive hashing. Homomorphic encryption techniques enabling ciphertext computation have attracted the attention of researchers for secure cooperative computing in untrusted distributed scenarios. Ogunseyi et al. [13] designed a privacy protection based on matrix factorization cross-domain recommendation. They encrypted user ratings by adopting the homomorphic encryption scheme so as to ensure user privacy while sharing potential factors between the different domains. The extracted knowledge can be securely transferred from source domain to target domain. Ogunseyi et al. [14] also proposed another homomorphic encryption-based scheme for privacy protection in cross-domain recommendation systems, called locally sensitive hashing (LSH). It compresses data into a compact hash code in a secure way and maintains the similarity of the original data, which has been widely used for similarity computation on large-scale data. Qi et al. [15] proposed an enhanced location-sensitive LSH method based on classical location-sensitive hashing technology to solve the privacy problems in distributed service recommendation. Qi et al. [16] also proposed to apply MinHash, an instance of location-sensitive hash, to service recommendation. They further designed a privacy-protected and extensible mobile service recommendation method based on two-stage location-sensitive hash to calculate the similarity between users in different fields and platforms. However, the methods require a large amount of computation resources, especially in the scenarios where multiple parties need to interact with each other.

2.1.3. Security recommendations based on federated learning

With the proposal of federated learning[17], the practice of storing user data locally to protect privacy is favored by researchers. Wang et al. [18] proposed a cross-domain recommendation POI (Point of Interest) framework with federated learning and privacy protection. Participants do not need to upload individuals' raw data, instead, only the gradient of model training is uploaded to the server for model learning, and thus guaranteeing user privacy. Perifanis et al. [19] presented FedPOIRec, another privacy-preserving federated learning approach enhanced with features from users' social circles to generate top-N POI recommendations.

However, Zhu et al. [20] pointed out that the gradients of model training also have the problem of privacy leakage during uploading process. Researchers apply differential privacy in federated learning frameworks to protect gradient security. Qi et al. [21] proposed a privacy protection method of news recommendation model based on federated learning. It trains accurate news recommendation model by using enormous behavioral data locally stored by users, and eliminating the need for a centralized storage. Meanwhile, local differential privacy is used to protect the gradient information of user-server interaction to ensure the security of model training. Chen et al. [22] proposed a POI recommendation framework for privacy protection (called PriRec). It combines differential privacy and federated learning to protect public data privacy. Wu et al. [23] proposed a joint recommended framework

based on GNN (Graph Neural Network) and privacy protection. The framework trains local GNN model based on local data and uploaded local gradients with added differential privacy noise to server in order to obtain a globally trained GNN model, which can better protect the high-order interactions between users and items. Truex et al. [24] also proposed a method of federated learning with differential privacy, which can effectively prevent individual privacy disclosure while generating machine learning models with elevated accuracy. Although differential privacy can better protect the privacy of the gradients, the effect of federated learning will deviate significantly because the noise accumulates during iterations.

2.1.4. Security recommendations based on blockchain

Some recent works put forward a privacy protection based on blockchain for distributed recommendations. Wang et al. [25] proposed a collaborative filtering news recommendation based on end-to-end cloud joint learning to solve the privacy problem. It stores the training model and recommendation information into a blockchain for permanent storage, evidence chain, and real-time traceability, while using differential privacy to protect privacy. Bosri et al. [26] proposed a privacy protection recommendation platform (called private-Rec) combining artificial intelligence and blockchain. In private-Rec, the blockchain provides a secure distributed environment for users through its properties. A corresponding reward mechanism is designed to motivate users to share their data for recommendation computation. Each user can safely use the data with permission. Lin et al. [27] proposed a recommendation mechanism for privacy protection also based on blockchain, and established a completely distributed model combining with IPFS (Inter Planetary File System). Meanwhile, they introduced local sensitive hashing and local differential privacy to provide strong privacy protection. Himeur et al. [28] pointed out that blockchain possesses the characteristics of security, high adaptability, fault tolerance and trust, which can effectively promote data security in decentralized recommendation system. Although distributed secure recommendation frameworks can be designed by utilizing the advantages of blockchain for storing and separating data, the introduction of blockchain also raises new privacy issues and storage challenges for recommendations.

2.2. Data authenticity problems of distributed recommendation

In addition to the privacy problems of original data, recommendation systems in distributed environment are also prone to poisoning attacks by participants [11,29]. Malicious users can influence or destroy the recommendation results of other users through false data injection. Fang et al. [30] proved that an attacker can launch a data poisoning attack on the recommendation system by injecting fake users and elaborate user-item interaction data, thus providing recommendations according to the attacker's wishes. Specifically, an attacker can trick the recommendation system into recommending targeted items to as many common users as possible. Huang et al. [31] conducted a systematic study of data poisoning attacks on recommendation system based on deep learning for the first time. Poisoning manipulates recommendation system by injecting fake users with carefully crafted ratings. Consequently, the target item chosen by the attacker will be recommended to a large number of users. Shilling attacks [32] have also attracted the attention of researchers recently. In Shilling attacks, selfish or malicious attackers inject a large number of user profiles to promote the degradation of a certain target item. Since recommendation system plays a key role in guiding customers to purchase, they have strong motivation to provide users with fraudulent recommendation results for profit.

Omar et al. [33] pointed out that data tampering is one of the most noticeable personal information security problems in online business portals. Current recommendation systems as an essential component business portals lack the sufficient security control to protect customer information. Even so, there are still few studies on data reliability verification in the recommendation systems. Bandara et al.[34] pointed out that a blockchain-empowered tracing platform can provide effective tracking while maintaining privacy via introducing Self-Sovereign Identity proofs. It also indirectly confirms the privacy and traceability of blockchain. Wang et al. [35] applied blockchain technology for the first time to ensure the credibility of data in recommendation systems, and proposed an anonymous recommendation system based on blockchain. It separates the storage of users, items and transactions through a multi-chain structure to protect privacy. The technique guarantees data source authenticity with user signatures. However, it is mainly based on overlapping users to establish associations, which cannot be applied to cases with few or non overlapping users.

To address the above issues, in this paper, MuKG is constructed by extracting features from multiple parties, which is a unified representation for multi-source and heterogeneous data with privacy protection. MuKG can be applied to multiple cases with or without users overlaps to achieve secure and reliable fusion of multi-source data, enhance the interpretability of recommendation results, and improve user trustworthy.

3. Preliminaries

3.1. Relevant knowledge

Federated learning [17] can extract characteristic parameters from data scattered in a large number of different clients through model training, and train a high-quality global shared model. It is an effective way to address the privacy preserving data sharing requirement in distributed environments by keeping data locally where it belongs. However, federated learning still has the issue of model training skewness due to differences in data distribution. Since federated learning does not concern verifying data authenticity,

it is unable to resist the challenges of data poisoning and model poisoning attacks in distributed scenarios.

TEE[1] (Trusted Execution Environment) creates a secure container to realize data isolation between trusted and ordinary environment. Execution process and calculation are not interfered by conventional operating systems to ensure internal data integrity and security. It can effectively prevent potential malicious users from controlling or observing internal data, providing powerful support for blockchain-based data sharing schemes. Currently popular TEE solutions include SGX (SoftwareGuard Extensions) based on Intel x86 architecture, TrustZone based on ARM architecture, and MultiZone based on RISC-V. TEE is widely used in many different areas to improve privacy protection and security properties. For instance, Chen et al. [36] used TEE to defend causative attacks and achieve data privacy in federated learning systems.

Homomorphic encryption is a key technique for achieving data privacy computing. It has been widely used in cloud computing, blockchain, privacy computing and other areas. Homomorphic Encryption (HE) is an encryption algorithm that meets the homomorphic operation properties of ciphertext. After homomorphic data encryption, specific calculation of ciphertext is performed. The plaintext computation results obtained after corresponding homomorphic decryption are equivalent to the results of computation on plaintext directly. The "computable invisibility" of data is realized.

Fully Homomorphic Encryption (FHE), which supports any form of ciphertext computation, is still in the exploratory stage. Existing algorithms face performance issues, such as low efficiency, large keys and ciphertext explosion. They are not quite feasible for engineering applications. Multiple selection of HE in practice only supports partially computed semi-homomorphic encryption for ciphertext (e.g., additive homomorphic encryption). It is used to realize limited homomorphic computation in some application scenarios. An additive homomorphic encryption represented by Paillier is suitable for federated learning frameworks.

Blockchain applications place the information that needs to be stored into a chain, and ensure data validity and tamper-proofness through verification and storage of numerous blockchain nodes. In Bitcoin, for example, users broadcast money transfers that are verified by blockchain nodes and packaged them up to ensure a transaction is legitimate. In Ethereum, it relies on the correct execution of smart contracts by blockchain nodes to achieve information consistency and correctness on the chain. However, whether it is a public chain or an affiliate chain, publishing blockchain directly in plaintext often discloses sensitive data to some extent [37].

In order to protect the privacy of information on the chain while realizing computability of relevant information by blockchain nodes, homomorphic encryption of data can be performed to transform the computation process into a homomorphic manipulation process. Nodes can implement ciphertext calculation without knowing the plaintext data. For example, Regueiro et al. [38] designed a combination of blockchain and homomorphic encryption to enhance the confidentiality in data transmission, storage and computation for distributed data aggregation. Since the underlying application blockchain platforms, especially the public chain platforms, are mostly based on token transaction models, it is reasonable to consider the use of additive homomorphic encryption to compute the transaction amounts and other operations that support privacy protection.

The scenario in this paper requires both data confidentiality and verifiability on a chain. Homomorphic encryption can only solve the computation of ciphertexts on the chain. Since the private key cannot be revealed, it is difficult to verify the plaintext computation results on the chain by homomorphic encryption alone. Therefore, in the multi-source data fusion process, we only use the homomorphic encryption technology to encrypt the features in the feature subgraph uploading by the participants. The edges of the weights are not encrypted. In this way, we can guarantee privacy while breaking the limitations of current cryptographic computing. We implement lossless fusion on the weights in plaintext, and capture the features of knowledge graph when recording information on blockchain. Every participant can verify data authenticity based on locally existing feature codes without concerning non-existing features' codes. Privacy computing requirements in data fusion can be protected. In the meantime, the verification requirements of on-chain data can also be achieved.

3.2. *Knowledge graph and Multi-feature knowledge graph*

Definition 1: Knowledge graph is a semantic network suitable for representing complex relations. Nodes represent entities, and edges represent relationships between entities. Triples (h, r, t) are usually used to represent knowledge in knowledge graph, where h and t represent head and tail entities, respectively, and r represents relationships between entities.

Several models such as TransE, TransH and TransR [39] have been proposed. TransE makes the sum of the H and R vectors as close as possible to the T vector, and use the norm of $L1$ or $L2$ to measure their closeness. TransH is designed to handle one-to-many/many-to-one/many-to-many relationships without increasing the complexity and difficulty of a training pattern. It interprets relationships as transformation operations on a hyperplane. Each relation has two vectors, the norm vector W_r of the hyperplane and the translation vector (d_r) on the hyperplane. Each vector of h and t are projected onto the hyperplane, yielding new vectors ($h_\perp$ and $t_\perp$). There exists a relation (d_r) in this hyperplane, which we can train the TransE model. While each entity can have multiple modalities, different relations focus on the different modalities of the entity. To solve this problem, TransR learns a structured representation for each object by projecting entities and relations into different spaces: one entity space and multiple relation spaces, and building the embeddings of entities and relations in the entity space and relation spaces, as shown in Fig. 1.

For each triplet (h, r, t), the entities in the entity space are projected into the r relation space through the matrix M_r, which are respectively represented as h_r and t_r ($h_r + r \approx t_r$). The training method f_r (h,t) and the loss function L of knowledge representation are shown in formula (1–3).

[1] TEE Committee (formerly Device Committee). https://globalplatform.org/technical-committees/trusted-execution-environment-tee-committee/ .

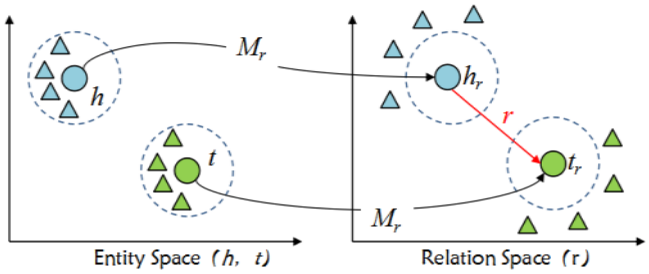


Fig. 1. A typical TransR example diagram.

$$h_r = hM_r \quad t_r = tM_r \tag{1}$$

$$f_r(h\ t) = ||h_r + r \quad t_r||_2^2 \tag{2}$$

$$L = \sum_{(h\ r\ t)\in S} \sum_{(h'\ r\ t')\in S'} \max(0\ f_r(h\ t) + \gamma \quad f_r(h'\ t')) \tag{3}$$

where h and t are entity embedding, r is relation embedding, S and S' are sets of correct and incorrect triples respectively, and γ is the threshold value.

From a practical application point of view, we aim to remove individual information, which is easy to reveal privacy, screen out the associations between attributes or features, or items concerned in recommendation computation, and construct a multi-feature knowledge graph through multi-domain collaboration.

Definition 2: Privacy disclosure refers to the ability to infer more features of a target user or a purchased item based on the background knowledge of a user's individual features and purchased items.

Definition 3: Multi-feature knowledge graph (MuKG) to be constructed in this paper contains various types of nodes and edges, which can be expressed as a quad (feature, r, w, item). In the quad, feature and item are entities, r is the relation between these entities and w is the weight of r.

In MuKG, *feature* is the feature nodes that describe multiple users' characteristics, such as specific values of *region*, *occupation*, *age*, etc. *Item* refers to the item nodes, representing the items related to domain features, such as *books*, *movies*, *goods*, *diseases*, etc. In order to distinguish the two different types of nodes, different shapes or colors are plotted in the related figures. *r* is the relation between nodes. It is represented by edges. Edges can exist between features, items, and between features and items. The weight of an edge *w* represents the statistical probability that an individual with a certain feature value has another feature value or purchased an associated item at the same time. Data can be obtained from a single domain or multiple domains.

Different from traditional knowledge graph, MuKG treats user features as entities associate with items in knowledge graph. It hides the relationships between users and items for privacy protection. The reason is that the nature of privacy is primarily related to an individual user, whereas recommendations mainly focus on the features of users and the associations between items rather than specific behavior of an individual user.

4. Trusted Cross-domain recommendation model based on MuKG and blockchain

4.1. Overview of system

Aiming at the problem of multi-source data security and authenticity verification in distributed data processing, we propose a safe recommendation with data fusion from different domains. When extracting multi-dimensional features to securely build an unified MuKG with multi-domain collaborations, we can find more complete and deeper hidden knowledge to improve accuracy and diversity of recommendation results. At the same time, the knowledge graph can provide interpretability of recommendation results to enhance users' trust and satisfaction.

Our technique is mainly used in distributed environments to integrate federated learning frameworks for fusing multi-domain data security. In order to protect privacy, we separate user features from users. When multiple participants upload local user features, feature-item associations are built into a blockchain for constructing a multi-party co-trust mechanism. We employ homomorphic encryption techniques to encrypt the features and fuse the feature weights of multi-participants for security. Furthermore, iterative learning and updating are performed based on a federated learning framework to build a more comprehensive multi-feature collaborative knowledge graph and establish multi-dimensional and accurate user portraits without touching user private data. As a result, a win–win situation of data security and accurate recommendation is achieved. Associated recommendation analysis of new users and new items can be conducted based on the knowledge graph, which effectively alleviates the problem of cold start. Moreover, learning based on data from different domains makes it easier to discover the potential or hidden interests of users, which can effectively improve the long term effect and stability of the recommendation system.

Users can provide their own encrypted features as query conditions without storage needs when they initiate recommendation

requirements. The recommendation server combines users' features and unified MuKG to perform item prediction analysis and obtain ranked candidates as the recommended list. Since MuKG is constructed based on users' features and item associations, features or items are not associated with specific individuals. We can therefore protect users' private information and fully personalize users' intelligent recommendations. The framework is divided into four modules as shown in Fig. 2.

1. Preprocessing of local data module. The raw data is stored locally on a client and does not need to be uploaded to a server. Instead, the local data is preprocessed in a local Trusted Execution Environment (TEE) to extract users' features subgraph, item knowledge graph and feature-item interaction graph. We encrypt the feature to protect privacy and upload these graphs to the server for fusion. Users' features subgraph includes user attribute information such as *age*, *region*, *occupation*, and *salary*; Item knowledge graph is a relational graph generated from item information in local data including items such as *categories* and *labels*; and Feature-Item graph includes the statistical associations between features and items.

2. Blockchain-based verification module. In this paper, blockchain is introduced as a multi-party co-trust mechanism in a distributed setting. All participants register and join a blockchain. All interactive data is uploaded and stored to a recommendation server, and all operation records are written to the blockchain ledger. We adopt a consensus mechanism based on blockchain to solve the problem of data anomaly. According to the consensus results, we also track and penalize the nodes that provided abnormal data or perform abnormal actions. Based on the chain operation records, dishonest nodes can be traced back and punished by confiscating deposits, declining to participate in or recommend. As a result, we can reduce the probability of dishonest nodes to provide false data. In order to stimulate the enthusiasm of multiple participants; different incentives are provided depending on their contributions. For example, incentives could be points, tokens, accurate recommendation services, etc.

3. Unified MuKG construction module. In this paper, MuKG is proposed for the first time. In combination with data analysis required by a recommendation system, we construct MuKG by extracting users' statistical features as entities and formulating the relationships between features and items as edge weights to hide sensitive information about specific individuals. We also develop the methods for edges' weights updating and node matching locally. Moreover, we integrate the features of multi-participants in ciphertext with weights in plaintext to guarantee privacy during data transmission. The advantage of our constructed unified MuKG is to enable secure multi-domain data collaboration for discovering more knowledge in different scenarios whether data overlap or not. In data interaction process, multi-participants are not required to provide their raw data, but only the corresponding users' feature subgraphs and the associations between users' features and items. The server has no way to know which features a certain user has. Usually, users with a certain feature are not unique. In this way, the presentation of MuKG can effectively protect the sensitive information and sensitive associations of the users.

4. Personalized recommendation module based on MuKGCN. When a user requests a recommendation, the recommendation server can provide its own feature vector as input to a graph convolutional neural network. The recommendation server can train a recommendation list that conforms to the user's characteristics by combining the unified MuKG. Moreover, the MuKG can build more accurate feature-item relationships by integrating features from multiple data sources. Therefore, the output results obtained by the graph neural network based on the unified multi-domain collaboration MuKG can be more accurate and closer to the users' interests and thus improves the recommendation accuracy.

4.2. *Trusted Cross-domain recommendation with MuKG and blockchain*

In this section, we discuss the implementation details of the four modules described in Section 4.1.

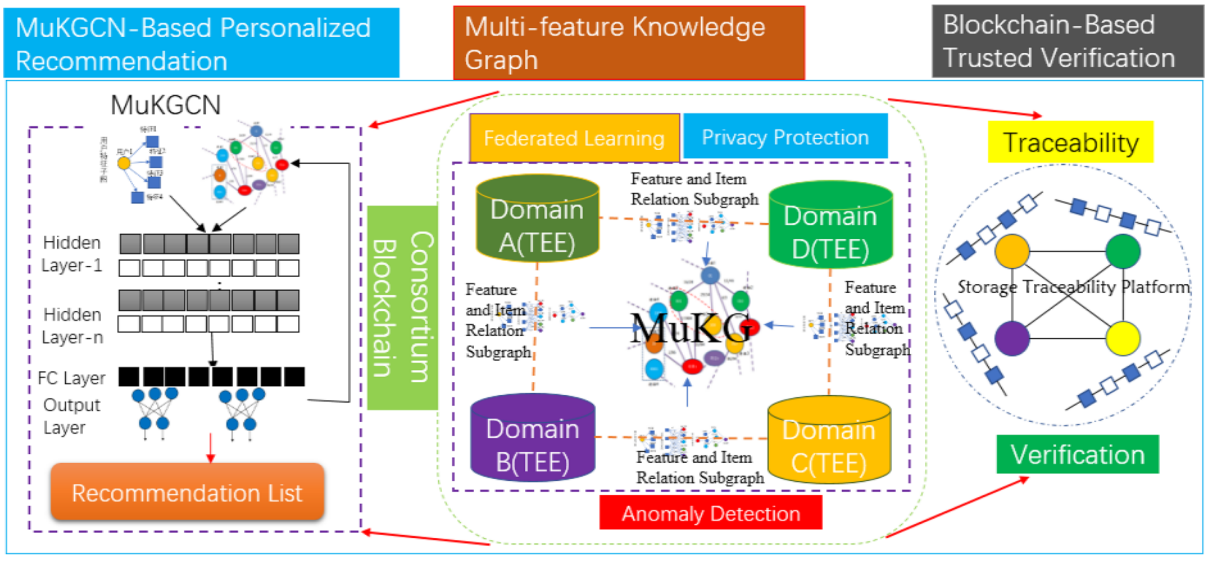


Fig. 2. Trusted cross-domain recommendation framework based on MuKG and Blockchain.

4.2.1. Preprocessing of Clients' local data

For privacy protection, all user data is stored locally instead of being uploaded to a server. The platform of users/clients registered in a blockchain is used as an edge server for fundamental analysis and feature extraction of the local data. In order to resist fake data or model poisoning attacks in distributed environment, participants perform feature extraction and redundant information filtering in local trusted execution environment (TEE). TEE can guarantee that the whole execution process is not controlled by humans, thus ensuring data authenticity. Redundant information filtering in the clients can considerably reduce the amount of information transmission. It can also distribute the computing load of the server and greatly improve the computational efficiency.

To illustrate the feature extraction process more clearly, a concrete example is given in Fig. 3. Fig. 3(a)-(d), respectively, represents local original client data, the extracted user's feature subgraph, the knowledge graph of items, and the feature-item interaction graph. As shown in Fig. 3(b), user's features subgraph includes the user's attributes information, such as *sex*, *age*, *region*, *occupation*, and *salary*. A user's attribute information in different fields can be different. In our study, only the features that match perfectly are aggregated. This method is mainly applicable to classification attributes, and can be segmented for numerical attributes. Fig. 3(c), knowledge graph of items, is a subgraph based on the items in local data, including the items' *category*, *label* and other information. The design of association rules can be specified by domain experts or extracted from background knowledge. Typically, there is a relationship among the items in the same category, which are likely to have been purchased by the same user. For example, televisions and refrigerators both belong to the category of household appliances, and users who have already bought a television often tend to also buy a refrigerator. In addition, items with the same label are also related, such as mobile phones and mobile phone cases. It is highly possible that a user who bought a certain brand of mobile phone is interested in the same brand of a mobile phone case. In Fig. 3(d), feature-item interaction graph extracts information from local data mainly includes the associations between users' features and the items, with the statistics of associations as the weights of edges. For instance, statistically 70% of teachers bought glasses. The weight of the edge between the teachers' occupational feature and the glasses is 0.7, indicating the degree of association between the two. It is clear that the glasses can also be further divided into different brands, and the item generalization relations can be established using item classification trees.

One-dimensional feature associations are counted in the first iteration. The multidimensional feature relations can be further extracted in an iterative process. For instance, associations of two-dimensional attributes can be extracted in the second iteration, and associations among three-dimensional attributes can be extracted in the third iteration, and so on. We take the associations extraction of two-dimensional features as an example. Fig. 4(a) shows the relationships between the features of different dimensions. Fig. 4(b) statistically analyzes the associations between the two-dimensional features' combination and the related items. The local clients upload the multi-dimensional feature associations to perform multi-feature fusion and update the unified MuKG in the recommendation server, and record in the blockchain.

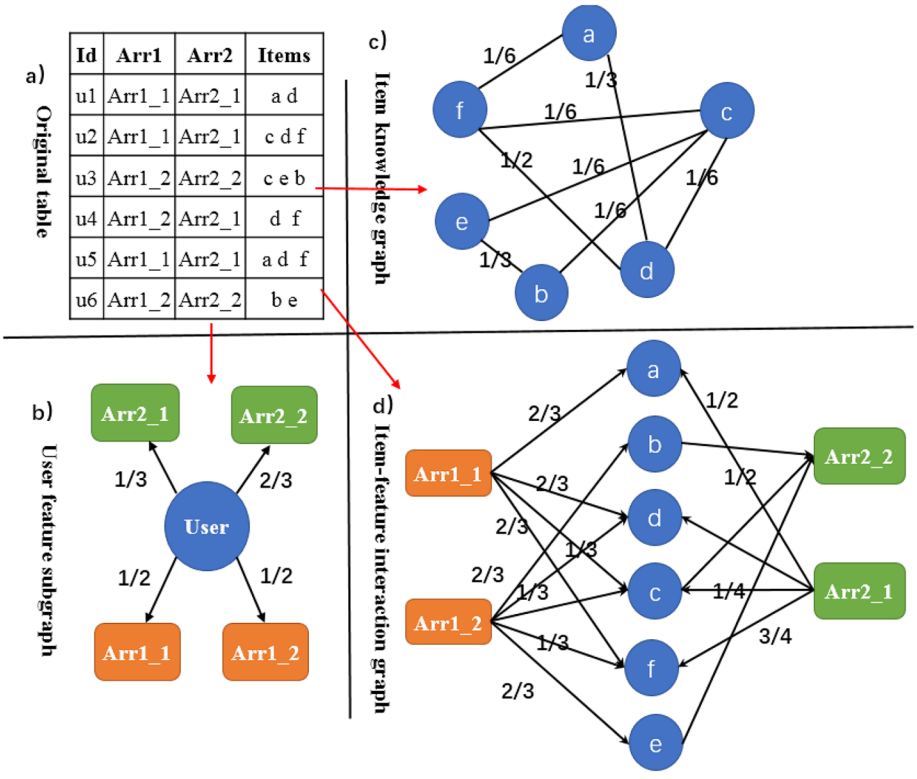


Fig. 3. An example of users' feature subgraph extracting from clients.

4.2.2. Security features fusion in Cross-domain scenario

To ensure privacy during data fusion, each participant encrypts the feature subgraphs extracted locally. Only users' features in the subgraph are encrypted to support trusted data verification. The weights of the edges between the features are not encrypted. The feature encryption with statistical weights method has two advantages: 1) data privacy can be guaranteed, and the flexibility of recommendation analysis is not limited; and 2) efficient data verification can be performed, and the amount of encryption computation is considerably reduced. The participant will send the user features subgraph, items subgraph, and item-feature relation subgraph to the recommendation server with the ciphertext of the features and the plaintext of weights. After receiving the multi-party information, the server will store them for validation and traceability, and perform feature matching and weight fusion on the edges. We consider fusion of the following three data distributions: 1) When there exists more overlap between the items and less overlap between users' features, MuKG can be fused based on the associations between the same feature and the related item. The new edge weight can be obtained through aggregation calculation, and updated in the MuKG of aggregation. For example, after the fusion of the relationship between "Arr1- 1″ and "item 1″ in the fields of A and B in Fig. 5, the weight is updated as $(0.5 + 0.62)\ /2 = 0.56$. Clearly, the fusion of weights can be averaged or computed by setting adaptive adjustment factors depending on the dataset size of different domains. 2) Conversely, when there exists more overlap between users' features and less overlap between items, a small intersection according users' features appears. We can update the weights of edges based on the overlaps of users' features for establishing the associations among the items in different fields and capturing the new relationships. For instance, a new association can be established based on the common feature "Arr1-1″ after the fusion of fields B and C in Fig. 5. 3) When less overlap between users and items occurs, indirect relationship can be obtained by inferring with other associated features. As shown in Fig. 5, there is no crossover between the fields B and D. However, indirect correlations can be established with the help of the common neighborhood C, whose weights can be calculated from the weight coefficients of the multi-hop edges.

In the process of data fusion, the features involved in data fusion are extracted from individuals and are hidden by homomorphic encryption techniques. Feature fusion in ciphertext state enables a secure sharing of multi-source data, while plaintext weight fusion makes data verifiable and computations more flexible. The data fusion process can safely protect private information of individual users and the sensitive relations between users and the related items.

4.2.3. Blockchain-Based mechanism for data verification

To build a distributed trusted sharing mechanism, our framework requires all clients to register and join the permissioned blockchain as participants, and the recommendation server to register and join in the permissioned blockchain as a server, as shown in Fig. 6. Participants upload the local features of the preprocessing data to the server, the upload operation will be written to the blockchain as a record, including the upload source and time. The local features uploaded by participants will be stored in the recommendation server so that its sources can be traced when data anomalies are detected. After collecting the features of multiple participants, the recommendation server will perform multi-party feature classification and fusion in the ciphertext state. Participants check whether there are any anomalies in the blockchain based on the distribution of all participants' local data. If there is no abnormality, they will sign for confirmation. When all data is signed for confirmation, the fused MuKG can be written to the blockchain. When there is abnormal feedback on data, in order to ensure the data correctness, it is necessary to conduct multi-party negotiations on the questionable data, which are jointly executed by all nodes of the registered blockchain.

Common consensus mechanisms for permissioned blockchains mainly include PBFT, DPOS, etc. PBFT (Practical Byzantine Fault Tolerance) achieves correct consensus of data when all nodes are not trusted, which requires more informative interactions. DPOS (Delegated Proof of Stake) mainly conducts elections and consensus based on their tokens, which is only applicable to the case of token issuance. However, the situation described in this paper is different. The recommendation server does not provide original data or touch original data in the entire process. It is only responsible for aggregation and recommendation analysis of ciphertext of data features. Essentially, the recommendation server can be regarded as a trusted third party in the consensus process, and other participants who provide data have the same voting rights.

In order to improve the consensus efficiency, we design a consensus mechanism based on recommendation server (POR). The specific execution procedure is as follows: the recommendation server acts as the leading node to initiate consensus requests to all users. When there exist abnormal data, the recommendation server integrates the operation records on the blockchain and proves the

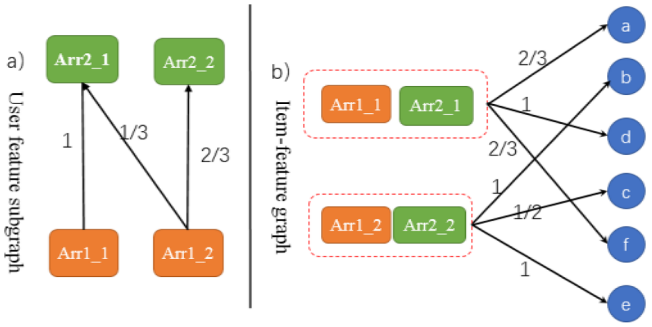


Fig. 4. Iterative updating of interaction subgraphs between multi-dimensional features and items.

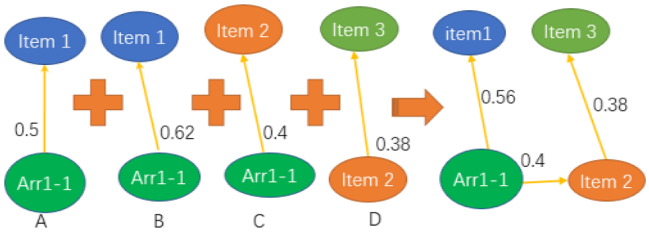


Fig. 5. Global fusion of multi-feature knowledge graph.

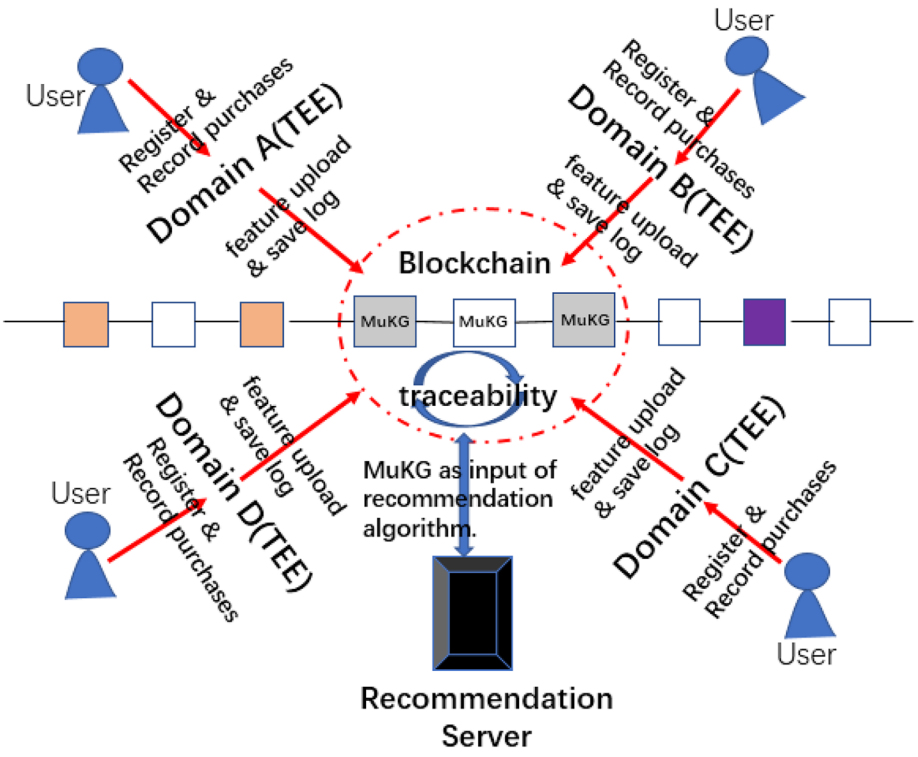


Fig. 6. Data verification mechanism based on blockchain.

source of the questioned data to all participants. It then executes a smart contract to perform a specific combination of queries to the data source, obtain the query results and send a feedback to all participants. Based on the local data distribution, the participants will conduct verification in combination with the query results and the questioned data, and send the acceptance information to all other participating nodes, including the recommendation server, if the questioned data is considered credible. When the data is considered abnormal, the participants will send a rejection message. The recommendation server calculates the final result based on the positive and negative feedbacks received. When an acceptance ratio exceeds 2/3, the data are accepted and written to the blockchain with a consensus timestamp as the benchmark data for the next round. Otherwise, the data is discarded. The fusion of the data is rolled back, and the multi-domain collaboration MuKG is updated accordingly.

As all uploaded data and operation records will be verified by the blockchain, the abnormal behaviors of nodes will be exposed after the consensus is successful. Due to the traceability of the blockchain, the participants who perform abnormal behavior will be penalized, which can greatly reduce participants' malicious behaviors and enhance the efficiency of the consensus process. After the consensus is successfully completed, the multi-domain collaborative MuKG constructed by the recommendation server will be stamped and written to the blockchain. In the subsequent partial update operation due to incremental data, the update operation and the data before and after the update will be also written into the blockchain. The fake data can be uncovered and the influence of false information on the recommendation model can be reduced to the greatest extent.

As shown in Fig. 6, our blockchain-based data verification mechanism has the following advantages:

(1) Privacy-Preserving: during the recommendation analysis and verification process, the recommendation server cannot touch the original data and cannot infer users' sensitive data.

(2) Identity Authentication: unauthorized platforms cannot participate in the permissioned blockchain. All participants can be tracked.
(3) Application Security: it is difficult for adversaries to steal or corrupt other domains' data in the feature fusion process.
(4) Traceability: a participant who provides fake data or violates the consensus protocol can be found and censured.

In summary, our blockchain-based authentication mechanism effectively guarantees participants' privacy during the data fusion process and resists poisoning attacks due to malicious behavior in an open distributed environment. Moreover, it also effectively implements data source tracking to greatly reduce the occurrences of malicious behavior, and subsequently improves recommendation accuracy. The recommendation server does not touch the participants' raw data. It only stores the local data features in ciphertext and the fused MuKG. For participants, the extracted feature data is based on all local users entirely. A specific feature's values are not unique to users. Therefore, the mechanism can hide the sensitive associations between users and items. The mechanism also establishes the MuKG between users' features and items, on which is the recommendation system focuses. The association between an individual and an item is also truncated, which can further protect the security of users' sensitive attributes.

4.2.4. MuKGCN-based personalized recommendation

Knowledge Graph Convolutional Networks (KGCN) is a recommended method to automatically capture high-order structural and semantic information in Knowledge Graph (KG). It summarizes and merges the biased neighborhood information when calculating the representation of a given entity in KG, which is suitable for the cross-domain recommendation scenario in this paper. To ensure the accuracy of a recommendation list, the personalized recommendations discussed in this paper not only need to be learned from different domains based on the features of a target user, but also the higher-order structural information and semantic information of the target user's neighbors based on the MuKG. Thus, we train a recommendation model based on the proposed MuKG to obtain personalized recommendation lists by extracting user features and item subgraphs of the recommendation targets.

As shown in Fig. 7, we take the feature subgraph of the target user as the input of a graph convolutional neural network and combine it with the constructed multi-domain collaboration MuKG to obtain a recommendation list that conforms to the user's interest preferences. In this process, the features provided by the user are homomorphic encrypted ciphertexts for protecting user's privacy. As MuKG integrates associations and knowledge from multiple domains, it can comprehensively and accurately represent the relation between features and items. The graph convolutional neural networks can learn users' features effectively. Therefore, the personalized recommendation based on MuKG with multi-domain collaboration can ensure recommendation accuracy and obtain the interpretability of the recommendation results, further improving the user's trust and click conversion rate.

At this point, the entire recommendation process is completed. Our recommendation puts forward a unified knowledge representation of MuKG that can effectively guarantee privacy during data fusion. Meanwhile, we introduce the blockchain as a common trust mechanism that can effectively resist poisoning attacks in distributed environments by storing data and recording operating records in the chain to keep track of the data sources and punish the attackers. It achieves a win–win guarantee of recommendation accuracy and data privacy in multi-source cross-domain recommendations.

4.3. Algorithmic designs of MuKG and MuKGCN-Based recommendation

This section details the specific algorithmic design of multi-domain collaborative MuKG construction and personalized, intelligent recommendation based on MuKGCN.

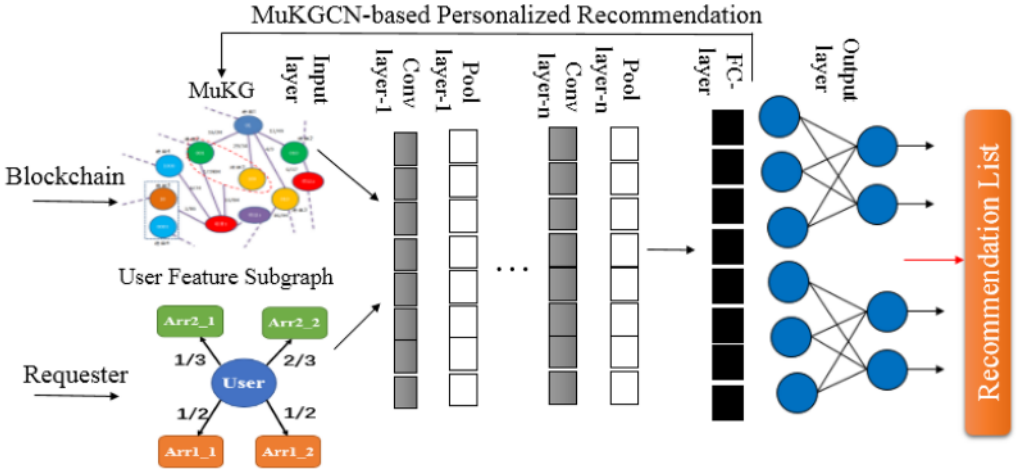


Fig. 7. Personalized recommendation based on MuKGCN.

4.3.1. Construction of MuKG

The specific steps of multi-domain feature fusion are as follows.

Step 1: Align features to build joint feature vectors and joint item vectors. All participants negotiate to agree upon a unified hash function, and send the hash codes of local user features to the recommendation center. The recommendation center concatenates the feature vectors and item vectors from all participants based on hash codes, and returns the IDs of the features to each participant.

Step 2: Record the local distribution of user features in Convolutional layer 1. Suppose there are k user features. m denotes the number of users, n represents the number of items, and ki denotes the one-hot dimension of the i-th user feature. The ith feature vector of the j-th participant is expressed as $U\text{-}F_j^{m\times ki}$. Each participant learns a distribution with a certain feature based on the local data, and the calculation formula is as follows:

$$F_j^{m\times k} = U \quad F_j^{m\times ki} \times A^T \tag{4}$$

where $A=\begin{bmatrix} x_{11} & \cdots & x_{1k_i} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mk_i} \end{bmatrix}$, all values in the matrix are $1/m$.

Step 3: Learn the local user feature-item matrix in Convolutional layer 2. This step learns the probability that a user with a particular feature likes a particular item based on the local data. The item matrix of the user is expressed as $U\text{-}V^{m\times n}$. The feature-item matrix of the j-th participant is expressed as $F_j\text{-}V^{ki\times n} = (F_j^{m\times ki})^T \times U\text{-}V^{m\times n}$.

For instance, $U\text{-}V^{m\times n}=\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$, $x_{ih} = 1$ if user i has brought h, otherwise $x_{ih} = 0$; .

According to the one-hot coding principle, the characteristics of *gender* can be encoded as "male = 10″ and "female = 01″. The gender feature vector $F_{sex} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is obtained, and the feature matrix of m users can be expressed as $U\text{-}F^{m\times ki}$. The formula $F_j\text{-}V^{ki\times n} = (U\text{-}F_j^{m\times ki})^T \times U\text{-}V^{m\times n}$ is used to calculate the probability y_{ih} that a user with feature value i has brought item h.

Step 4: Upload and initiate the aggregation task in Pooling layer. Each participant uploads the feature-item matrix to the recommendation center. The recommendation center sets the initial aggregation parameters based on the number of participants and initiates the aggregation task. z represents the number of participants and α_i represents the aggregation parameter of the ith participant, $\alpha_i \in (0,1)$, $\sum_{i=1}^{z} \alpha_i = 1$. In general, the larger the number of participants, the larger the aggregation parameter. All participating nodes are encouraged to take part in the aggregation. The earliest aggregation and release node obtains task rewards, which can be credits or cash payments.

Step 5: Aggregate calculations. The recommendation center performs an affine transformation on the neurons of the hidden layer based on the feature IDs and the items to aggregate the feature matrix of all participants, which is formulated as

$$F \quad V^{ki\times n} = \sum_{j=1}^{z} \alpha_j \; F_j \quad V^{ki\times n}) + b \tag{5}$$

where b represents the neuron bias and α_i represents the aggregation parameter of the i-th participant. $F\text{-}V^{ki\times n}$ is the global probability of the item purchased by users with a certain feature value.

Step 6: Aggregate updating. The aggregated feature matrix is returned to each participant. They update the local feature model based on the aggregated data.

Step 7: Learn associations between different dimensions of local user features in Convolutional layer 3. The correlation degree between user features is computed based on the returned global user feature-item matrix and the local data. It is then sent to the recommendation center. The recommendation center computes the aggregate association parameters of the global user features based on the feature IDs, and returns them to the participant. The participants update the associations of local user features based on the associations' parameters, and disregard the features that are not present locally.

Step 8: Update aggregation in Pooling layer. Each participant further calculates the probability of combination features based on the local feature matrix and the global features association parameters. This process can be co-iterated many times until the final requirements are met.

Step 9: Learn the associations between the items locally in Convolutional layer 4. The correlation degree between items is calculated based on the returned global user feature-item matrix and the local data, and is sent to the recommendation center. The recommendation center calculates the aggregated association parameters of the global items based on the items and returns them to the participants. The participants update the association of local items based on the parameters of the global association, and discard the items that do not exist locally.

Step 10: Update aggregation in Pooling layer. Each participant further calculates the correlation degree between the combined items based on the global association parameters of the items. This process can be co-iterated many times until the final requirements are met.

Step 11: Work in Fully connected layer. To improve the performance of the CNN network, Softmax logistic regression is applied to classify the activation function of each neuron in the Fully connected layer. The recommendation center trains the classification based on the existing user feature-item association matrix and its corresponding combination associations, establishes the associations between all user features and the items, and sends feedbacks to each participant.

At this point, the collaborative fusion process of multi-source features is completed, and the unified MuKG can be constructed based on user features, items and their associations, which will be recorded in the blockchain. It is clearly that the modified operation will also be recorded to enable traceability and rollback operations if necessary.

4.3.2. MuKGCN-Based deep recommendation algorithm

The Pseudocodes of MuKGCN, a deep recommendation algorithm based on MuKG, are shown in Algorithm 1. The algorithm consists of three main parts: reconstruction of the feature interaction matrix, sampling of neighborhoods, and training of the MuKGCN network.

We borrow the training model from KGCN, but improves it by replacing users' entities involved in KGCN with users' features. Prior to training, it is necessary to reconstruct the feature interaction matrix and replace users' entities with users' features to interact with items. Lines 14–26 in Algorithm 1 shows the reconstruction function. In the process of reconstruction, the local weights of the associated edges are obtained. Then the weighted feature interaction matrixes are constructed. It integrates multidimensional higher-order information, and well reflect the degree of associations between features and items, which is the key information used in the recommendation algorithm.

The reconstructed interaction matrix and the items KG are fed into the MuKGCN model for training. In the MuKGCN model, items in the interaction matrix are used as seed nodes to sample and aggregate their neighbors to obtain the higher-order semantic information of the items, and assist in obtaining better user preferences with features. Lines 27–34 in Algorithm 1 shows the design of the neighbor sampling function. In the process of neighbor sampling and aggregation, a fixed-size set of neighbors is used. All its neighbors are sampled randomly in order to improve the algorithm efficiency. When the neighbor size is smaller than the sample set size, the existing neighbors are sampled repeatedly.

The training algorithm based on MuKGCN network is shown in Algorithm 1 in lines 1–13. During the training of the MuKGCN network, the interaction matrix is reconstructed based on user's features. Related sensitive information is extracted. A fixed number of neighbors is obtained by sampling based on the feature entities, and a convolutional neural network is used to capture the neighborhood information. The formula of line 8 $e^u[h \quad 1] \leftarrow \sum_{e' \in S(e)} \tilde{\pi}^u_{r_{e,e'}} e'u[h \quad 1]$ aggregates the entities of the sampled neighbors. The function of the formula in line 9 $e^u[h] \leftarrow agg\left(e^u_{S(e)}[h \quad 1] \; e^u[h \quad 1]\right)$ aggregates the representation of the entity and its neighborhood to form a new entity's representation for the MuKG. The aggregated neighborhood information takes part in the updating of entity's representation. They are used to analyze and predict the score based on the obtained entities' embedded representation and the representation of feature's entity. Finally the items are ranked according to the scores, and the personalized recommendation list is generated.

Algorithm 1. Depth recommendation algorithm based on MuKGCN

Input:Item Knowledge Graph ($\mathscr{G}$), Interaction Matrix (Y), User Feature Table (T), Training Parameters: $\{u\}u \in \mathscr{U}$ $\{e\}e \in \varepsilon$ $\{r\}r \in R$.
Output:Recommendation List

1. $Y' \leftarrow$ ***Restrucct_Y***$(Y \; T)$// Reconstruct the feature interaction matrix
2. **Repeat** // Training MuKGCN network
3. **for** $(u \; v)$ in Y' **do**
4. $\{M[i]\}_i^H \leftarrow$ ***sample_neibor*** // Neighbor sampling
5. $e^u[0] \leftarrow e \; \forall e \in M[0]$
6. **for** $h = 1 \; 2 \; \cdots \; H$ **do**
7. **for** $e \in M[h]$ **do**
8. $e^u[h \quad 1] \leftarrow \sum_{e' \in S(e)} \tilde{\pi}^u_{r_{e,e'}} e'u[h \quad 1]$
9. $e^u[h] \leftarrow agg\left(e^u_{S(e)}[h \quad 1] \; e^u[h \quad 1]\right)$
10. $v^u \leftarrow e^u[H]$
11. Calculating and predicting scores $\widehat{\mathscr{Y}} = f(u \; v^u)$
12. Optimizing parameters via gradient descent
13. **return** $\mathscr{T}$ //Recommendation List
14. **function restruct_Y(**$Y \; T$**)** // Feature interaction matrix reconstruction
15. **for** $(u \; v) \in Yandu \in T$ **do**
16. $F\_u \leftarrow f_1 \; f_2 \; \cdots \; f_n$ //Obtaining the features set of user u
17. **for** f in F_u **do**
18. $(f \; v) \leftarrow v \; f$ //Constructing interaction pairs of users' features and items
19. $S_1 \leftarrow sum(v)$ // Counting the number of interactions between user u and item v
20. $S_2 \leftarrow sum(u \; v_i)$ // Counting the total number of interactions between user u and all items
21. r $ating \leftarrow S_1/S_2$ //Computing the association weight of (u,v)
22. $V_f \leftarrow (v \; rating)$ //Obtaining the set of items each feature interacts with and their weights
23. $V_u \leftarrow V_{f_1}(v) \cap V_{f_2}(v) \cap \cdots \cap V_{f_n}(v) \; f_1 \; f_2 \; \cdots \; f_n \in F\_u$ // Obtaining personalized interaction itemsets via multi-feature intersection itemsets for user u
24. **for** v in V_u **do**
25. $Y' \leftarrow (u \; v \; max(rating)) \; (v \; rating) \in V_f$ // Reconstructing the matrix of user interaction
26. **return** Y'
27. **function** ***sample_neibor***() // Neighbor sampling
28. $M[H] \leftarrow v$
29. **for** $h = H \quad 1 \; \cdots \; 0$ **do**

(*continued on next page*)

(*continued*)

Algorithm 1. Depth recommendation algorithm based on MuKGCN

30. $M[h] \leftarrow M[h+1]$
31. **for** $e \in M[h+1]$ **do**
32. $M[h] \leftarrow M[h] \cup S(e)$
33. **return** $\{M[i]\}_{i=0}^{H}$
34. **until convergence**

5. Performance analysis

5.1. Recommendation Quality

1. The degree of accuracy

The trusted cross-domain recommendation model based on MuKG and blockchain proposed in this paper guarantee the accuracy of recommendation results. The rationales are as follows:

(1) To solve the problem of data sparseness, the method integrates multi-source data as auxiliary knowledge to construct the knowledge graph and does not modify the data in the process of feature extraction and data fusion. It can depict user portrait more completely and accurately, and thus, provide high-quality recommendation results. Although some information may be lost during feature extraction, statistical features required for recommendation analysis remain unchanged. In particular, in the cross-domain fusion scenarios, the amount of data involved is very large, and thus, the impact of information loss on recommendation accuracy can be ignored.
(2) For the purpose of data authenticity, we use the trusted execution environment (TEE) to ensure that the data processing is not interfered by humans. In the data fusion phase, we introduce the blockchain technology to establish a distributed co-trust mechanism, and realize data source traceability through storing participant characteristic data and the chain of operation records, which can greatly counterbalance the attacks of malicious nodes. The possibility of data tampering after fusion is eliminated by writing the knowledge graph of multi-source feature fusion into a shared ledger in the blockchain. Moreover, the multi-party consensus mechanism based on blockchain can effectively identify abnormal data and abnormal operations. In this way, we can promptly prevent the negative impacts of false data and dirty data on model training while improving speed and performance of model training. In summary, the proposed cross-domain recommendation model can effectively resist poisoning attacks and ensure data authenticity, thus guaranteeing the correctness of the recommendation model.
(3) To realize personalized recommendation, we design personalized recommendation based on multi-domain collaboration MuKG and convolutional neural network (MuKGCN). The method achieves personalized recommendation results by combining features from the unified MuKG to improve the quality of the input data of a convolutional neural network to study the relevant items of a specific feature.

2. The degree of diversity and novelty

The unified MuKG constructed in this paper comes from multiple participants and multiple domains. Local feature extraction and secure fusion of user data can break through the single-domain skewness problem, and obtain more complete and accurate association information. Therefore, based on the multi-dimensional data, a more comprehensive and accurate user portrait can be constructed to depict the diverse preference information of the users, thus improving the diversity of recommendations.

In addition, MuKG can not only perform explicit multi-correlation, but also mine users' hidden interests, and migration and change relations of groups from a multi-dimensional perspective. It is based on the multi-correlation of the graph to obtain deeper hidden knowledge and discover users' hidden needs. Theoretically the model discussed in this paper can break through the information cocoon problem and ensure the diversity and novelty of recommendation results. Relevant experiments are designed to verify the above analysis in Section 6.

5.2. Data security

This section focuses on data security analysis from the three stages: local data preprocessing, multi-source data fusion, and personalized recommendation.

In the local data preprocessing stage, the proposed method keeps raw data locally and separates users' private data for feature extraction so as protect users' raw data security. Next, in order to ensure that the extracted feature data is not tampered with, data feature extraction is performed in the local trusted execution environment (TEE), which can ensure the security and credibility of the feature extraction process. Finally, for securing the uploaded feature data, the participants use homomorphic encryption techniques to encrypt the extracted features locally. As a result, the security of the extracted features data during transmission is guaranteed. Local encryption methods can greatly reduce the amount of computation needed during the encryption process. The weights of the edges between features are not encrypted for providing the flexibility of the aggregated computation.

In the multi-source data fusion process, the server matches the ciphertext-based feature codes and fuses the edge weights in plaintext. Since the feature codes are ciphertext, the recommendation system does not know the specific information of users and item.

The meaning of the feature code is not clear for it, which effectively protect against malicious inference attacks on the server. At the same time, the features of the ciphertext state are written to the recommendation server for retrospective verification. The participants can use private key signatures to verify the relevant data based on the locally held feature codes, and discard the non-exist feature codes. The server can write the validated MuKG into the chain for its recommendation analysis and interpretability of the recommendation results. Different from knowledge graphs, MuKG after multi-source feature collaboration only retains the users' features. The users matched with a specific feature are often a group, making it impossible for attackers to associate the feature with a particular user. Therefore, user privacy can be guaranteed.

The third stage is the personalized recommendation stage. Based on MuKG and convolutional neural networks, the personalized recommendation can effectively resist neighbor attacks. Based upon the basic knowledge about a known recommendation algorithm, attackers obtain the recommendations result and analyze it through inserting fake data to disguise as a target neighbor. They further integrate the corresponding user entities in the knowledge graph, the specific association of the items purchased by the target user can be deduced, which leads to privacy leakage. However, in our scheme, attackers cannot reveal individual privacy from the recommendation result because the data source of the recommendation result is a feature vector or the unified MuKG based on multi-source data fusion, rather than from an individual. Apparently, it can effectively resist the injection attack.

In conclusion, our method can provide a higher-level privacy guarantee than the traditional recommendation methods. It can also resist poisoning attacks and neighbor attacks by addressing the problem of data authenticity verification in a cross-domain environment in an evidential and traceability manner.

6. Experimental analysis

For convenience, we will abbreviate the proposed method as MuKGCN. Since the traceability of blockchain cannot be measured experimentally, we can only compare the privacy leakage risk of the MuKG and knowledge graph from a theoretical perspective, and evaluate MuKGCN security based on the traceability properties of the blockchain. First, the combination of sensitive feature ciphertext and statistical feature plaintext can effectively protect data privacy. Second, the source traceability feature and accountability mechanism of the blockchain can reduce the probability of malicious attacks, and provide guarantees for data authenticity to some extent. Therefore, in this section, we evaluate the performance of MuKGCN in terms of data utility.

8.1 Datasets and Evaluation Metrics.

1. Datasets

We choose two classic datasets widely used in the evaluation of recommendation systems for experimental verification: Amazon[2] and MovieLens1M[3]. The Amazon dataset was selected because it contains multi-domain data. To verify the recommendation effect of cross-domain association, we select three relevant datasets (Movies and TV, Kindle Store, and Digital Music) with overlapping user sets. Due to the large amount of data in the datasets, 100,000 data were randomly selected for experimental comparison with sparsity exceeding 99.9%. The cross-domain common dataset of Amazon (Cross-domain) is extracted from the three associated datasets and includes 8943 common users, 68,485 involved items, and 107,601 rating data. The Amazon dataset is very sparse. While there is user overlap between domains, there is no overlap between items within these domains. In order to verify the performance under different scenarios, we also choose the classic dense dataset of Movielens according to the proportion of 3:3:4 between the three different domains, respectively, to simulate two kinds of cases involving users and items overlapping. Table 1 lists the relevant characteristics of the datasets.

2. Evaluation metrics

Metrics, such as precision, recall, F-measure, and diversity, are widely used by most top-k of recommendation frameworks in evaluating data utility. We also use these metrics to measure the recommendation system's performance.

Precision and recall measure how accurately an algorithm can generate a recommendation list for each user. As given in [40], precision is the percentage of the testing items in the recommendations list for users, and recall is the ratio of recommended items to all testing items. F-measure is the weighted harmonic average of precision and recall. For the readers' convenience, we redefine them as follows:

$$Precision@k = \frac{1}{k|u|}\sum_{u\in U}\left|S_u^T \cap S_u^R\right| \tag{6}$$

$$recall@k = \frac{1}{k|u|}\sum_{u\in U}\frac{|S_u^T \cap S_u^R|}{|S_u^T|} \tag{7}$$

$$F \quad measure@k = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{8}$$

where k is the length of the recommendation item list, S_u^T denotes the testing item list of user u, and S_u^R denotes the recommended list of user u. All experimental results are taken as the average of all users.

[2] https://jmcauley.ucsd.edu/data/amazon.

[3] https://grouplens.org/datasets/movielens/1m/.

Table 1
The characteristics of datasets.

Dataset	User	Item	Ratings
Movielens 1 M	6040	3952	1,000,209
Movies and TV	2,088,620	200,941	4,607,047
Kindle Store	1,406,890	430,530	3,205,467
Digital Music	836,006	478,235	266,414
Amazon (Cross-domain)	8943	68,485	107,601

The metric diversity [41] mainly measures the diversity of recommendation results, i.e. the ability of a recommendation system to recommend a variety of items. Here, the list of recommended items from the previous recommendation is compared with the current recommended list. The ratio of the difference between the two to the length of the recommendation list is expressed as follows: $Diversity = diff/k$, where $diff = list1\text{-}list2$ indicates differences between two adjacent recommendations.

Bedsides, we measure the ability of recommendation systems to discover hidden user needs by using the metric discovery [42]. Here, *discovery* is represented by the ratio of the number of items that first appear in the recommendation list, and are selected by the user, to the length of the recommendation list. The formulation is defined as follows:

$$discovery = \frac{|first \cap selected|}{k} \tag{9}$$

where *first* is a collection of items that never appeared in the previous recommendation list. They are newly discovered items. $|$ *first* $\cap$ *selected* $|$ is the number of items that first appear in the recommendation list and are selected by the user, which reflects the potential interest of the user.

6.1. Experimental settings and comparison methods

In our experiments, three server nodes are configured to simulate three different domains servers, and a Hyperledger Fabric 1.4 based peer node is deployed on the recommendation server. We run Ubuntu 18.04 on three server nodes with a 2-core CPU and 4 GB of RAM, and one node with eight Intel Cores I7 6700 CPUS and 32 GB of RAM, coded in Python 3.8.4. To simulate a cross-domain scenario where both the users and items have overlapping data, we randomly split the Movielens dataset, which is treated as domain data owned by three different servers respectively, to simulate a dense data scenario.

The experimental analysis is carried out based on two datasets with different parameter settings. The length of the recommendation list k ranges from 0 to 50. To ensure the fairness of the results, all experiments take the average of the results over 10 iterations.

To better evaluate the performance, we use the following related works as comparison baselines:

DP-GD [8]: This is a distributed differential privacy recommendation method, which satisfies differential privacy by adding random noise to gradient and ensures stronger privacy. The random projection algorithm is used to reduce noise and improve recommendation accuracy. This is a reference line for comparing privacy protection. Our objective is to examine whether MuKG can effectively reduce the information loss during training as compared to differential privacy.

KGCN [43]: This is a recommendation model based on knowledge graph convolution network (KGCN), which can automatically discover the higher-order structural and semantic information of entities, and effectively capture the associations between the items by mining the association attributes between items. It is based on user entities, but it does not consider privacy issues. This is the enhanced comparison baseline. By comparing with KGCN, we can verify whether removing individual information from MuKG affects the performance of the recommendation system in different aspects.

KGNN-LS[44]: This is a model of Knowledge-aware GNN with Label Smoothness regularization to provide better recommendations, which applies GNN architecture to KGs by using user-specific relation scoring functions and aggregating neighborhood information with different weights.

CGAT[45]: This is a novel recommendation framework, which explicitly exploits both local and non-local graph context information of an entity in KG by a user graph attention mechanism and a biased random walk sampling process, respectively.

DSKReG [46]: This is a model of differentiable sampling on knowledge graphs for recommendation with relational GNN to learn the relevant distribution of connected items from KGs and use the distribution to sample suitable items for recommendation.

MKR [47]: This is a deep end-to-end framework that utilizes knowledge graph embedding tasks to assist recommendations. The two tasks are associated by cross&compress units, which automatically share latent features, and learn high-order interactions between items in recommender systems, and entities in the knowledge graph.

RippleNet[48]: This is an end-to-end framework that naturally incorporates knowledge graph into recommender systems. RippleNet stimulates the propagation of user preferences over a set of knowledge entities by automatically and iteratively extending a user's potential interests along links in the knowledge graph.

To better compare and contrast the relationships between the above baselines, we summarize them in Table 2 in terms of recommendation strategies, with or without privacy consideration, auxiliary knowledge, and single or cross-domain.

6.2. Experimental results

In this section, we will make a comparative analysis of the proposed MuKGCN with the comparison baselines described in Section 6.2 in terms of multiple metrics. First, we evaluated data *accuracy* in term of *Precision*, *Recall* and *F-measure*.

In MuKGCN, the training is based on users' features to obtain users' preferences. A user often has multiple features. Each feature has different effects on different items. In order to better capture the real users' preferences, we conduct ablation experiments considering cases where the users either provide a single feature or multiple features.

(1) When a user only provides a single feature when requesting recommendation, the single feature is directly used to represent the user's real preference and is taken as the input data. The recommendation list obtained through the convolutional network learning of the MuKG is returned to the user as the recommendation list.

(2) When a user provides multiple features when requesting recommendations. Each individual feature is taken as an input. The intersection of the recommendation list of the multiple features is obtained by convolutional network learning of the MuKG. Based on the score of the intersection list, top-k items are selected as the user's final recommendation list. If the length k is insufficient, the intersection will be directly returned as the final recommendation list.

First, we investigate the impacts of recommendation list lengths k on recommendation precision and recall compare **MuKGCN** and other baseline methods. The experimental results are shown in Table 3. Please note that for the purpose of effective comparative analysis, we present the methods that can reproduce consistent results on the same dataset. We see that **MuKGCN** has a clear advantage on both the sparse Amazon dataset and the dense Movielens dataset. For $k = 20$ in the Movielens, the recall metric is 12 % higher than the suboptimal baseline, and the precision metric is 13.5 % higher than the suboptimal baseline. On the Amazon, when $k = 20$, the recall metric increases by 10% and the precision metric increases by 24 % as compared to the suboptimal baseline.

To analyze recommendation performance in detail, we take more sampling points for recommendation length. The effect of changing the recommendation length k on precision and recall in the Amazon dataset is shown in Fig. 8 and Fig. 9. Fig. 8 shows a downward trend where the recommendation result decreases as k increases. This is expected since the longer the recommendation list's length, the fewer items selected by the user, which then implies a lower recommendation accuracy. Fig. 8 also shows that the recommendation accuracy of our method is 24 % better than the other baseline methods when $k = 20$. The MuKGCN method is based on the fusion of multiple domain data to build a unified MuKG with privacy protection. Although individual entities are removed, it can capture higher-order information based on users' features, and discover internal associations between features and the items, which can effectively improve recommendation accuracy. Fig. 9 all shows an upward trend. The recall increases as k increases. This happens since the longer the recommendation length, the more items fall into the recommended range. Compared to other baseline methods, our MuKGCN also has a clear advantage on the recall metric because it incorporates more dimensional information, and is able to model user preferences more accurately.

The effect of varying k on F-measure in Amazon dataset is shown in Fig. 10. F-measure in Fig. 10 shows a stable upward trend except with KGCN and DP-GD. In KGCN, there is no item overlap in Amazon dataset. Many missing data appear in the item knowledge graph, which leads to instability of the associations captured by KGCN. In DP-GD, the added noise has an uncertain effect, making it unstable. MuKGCN presents clear advantages: it is 16 % higher than the other baseline methods when $k = 20$ as expected. MuKGCN integrates multi-dimensional associations based on multi-feature collaborations, which can greatly improve recommendation performance when data is sparse.

Amazon dataset is sparse cross-domain dataset. In order to test the performance of MuKGCN on dense datasets, we use the MovieLens dataset. Since MovieLens does not involve cross-domain data, we segment the dataset at a ratio of 3:3:4 to simulate a cross-domain situation in which both users and items overlap. The recommendation accuracy as k varies on MovieLens are shown in Figs.11–13. We obtain a similar result when we ran experiments on the Amazon dataset. Experimental results show that MuKGCN performs significantly better than the other baseline methods. We observe that when k is equal to 20, the precision is improved by 13.5 % over KGNN-LS and 36.4 % over KGCN, the recall is improved by 26.6 % over KGNN-LS and by 22.5 % over KGCN, and the *F1* is improved by 24 % over KGNN-LS and 30 % over KGCN.

We also evaluate the diversity and discovery of MuKGCN. Since the DSKReG method outputs prediction labels, its diversity and discovery metrics cannot be computed so we removed the DSKReG baseline. Since the longer the recommendation length k, the worse the recommendation performance of diversity and discovery, we change the range of k to 0–40.

Diversity is mainly based on the difference between successive adjacent multiple recommendation lists to measure the ability of a recommendation system to recommend multiple types of items. We add TDRS[41], the first work to consider diversity in evaluating

Table 2
Summary of our model and the comparison baselines.

Baseline	Recommendation strategies	Consider Privacy	Auxiliary knowledge	Cross domain
DP-GD	Matrix Factorization	Yes	None	No
KGCN	GCN-based	No	Item Knowledge graph	No
KGNN-LS	GNN-based	No	Item Knowledge graph	No
CGAT	RNN-based	No	Item Knowledge graph	No
DSKReG	GNN-based	No	Item Knowledge graph	No
MKR	Multi-task feature learning	No	Item Knowledge graph	Yes
RippleNet	Embedding-based and path-based methods	No	Item Knowledge graph	No
Our MuKGCN	**GCN-based**	**Yes**	**Multi-feature Knowledge graph**	**Yes**

Table 3
The impacts on recommendation precision and recall for different recommendation list lengths.

	Baseline	Recall			Precision		
		R@10	**R@20**	**R@50**	**P@10**	**P@20**	**P@50**
Movielens	RippleNet	0.0706	0.0880	0.1261	0.0147	0.0099	0.0060
	MKR	0.0732	0.0920	0.1306	0.0154	0.0105	0.0063
	DP-GD	0.0145	0.0220	0.0670	0.0340	0.0376	0.0335
	KGCN	0.1100	0.1556	0.2889	0.1120	0.0860	0.0620
	KGNN-LS	0.0697	0.1156	0.2244	0.3720	0.3150	0.2524
	DSKReG	0.0043	0.0115	0.0878	0.0464	0.0579	0.0489
	CGAT	0.1674	0.2608	0.4311	0.1575	0.1288	0.0916
	MuKGCN	**0.2066**	**0.3808**	**0.7835**	**0.4880**	**0.4500**	**0.3732**
Amazon	DP-GD	0.0316	0.0374	0.0734	0.0046	0.0030	0.0023
	KGCN	0.0283	0.0283	0.0361	0.0022	0.0015	0.0013
	KGNN-LS	0.0022	0.0028	0.0048	0.0140	0.0100	0.0064
	DSKReG	0.0040	0.0095	0.0150	0.0043	0.0056	0.0035
	MuKGCN	**0.0639**	**0.1255**	**0.2296**	**0.2589**	**0.2539**	**0.2256**

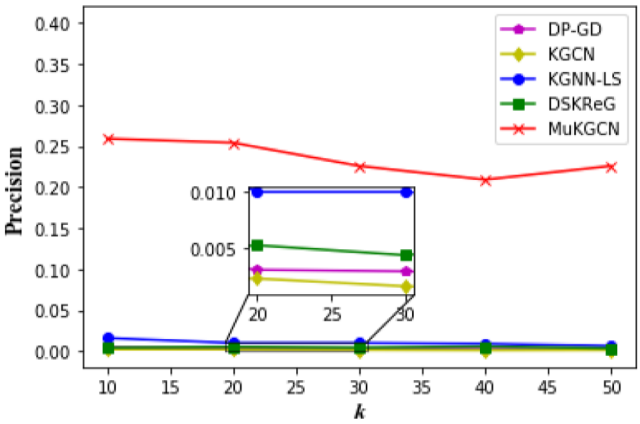


Fig. 8. The influence of varying k on *precision* for Amazon dataset.

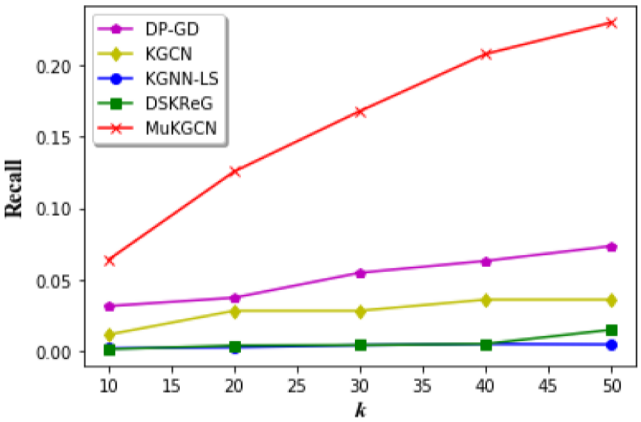


Fig. 9. The influence of varying k on *recall* with for Amazon dataset.

recommendation systems, as a baseline for comparing of diversity. By comparing with TDRS, we can examine whether the cross-domain association we proposed can improve the diversity of recommendation results. Fig. 14 shows the effect of varying k on diversity when two datasets are used. The curves in the figures are relatively stable when k increases. In line with our expectations, MuKGCN shows a very stable diversity performance, maintaining values around 0.8–0.9 on both Amazon and MovieLens datasets. It can efficiently aggregate data correlations between different domains through which the MuKG extracts related features. However by comparing Fig. 14(a) and (b), we observe that the fluctuations of KGCN-LS are relatively large: the diversity of KGCN-LS is close to 1 when running on Amazon dataset and is close to 0.2 when running on MovieLens dataset. This occurs since the Amazon dataset

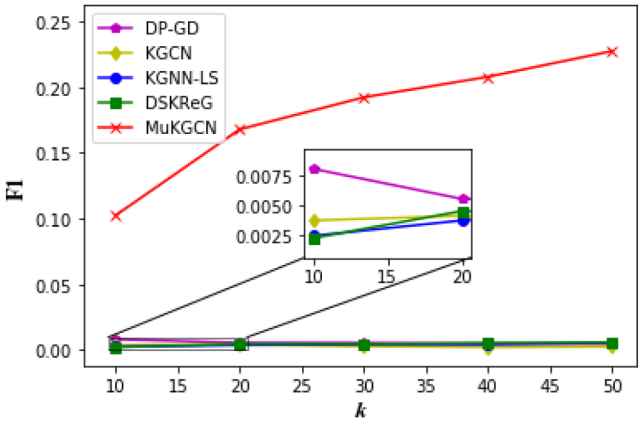


Fig. 10. The influence of varying k on *F-measure* with for Amazon dataset.

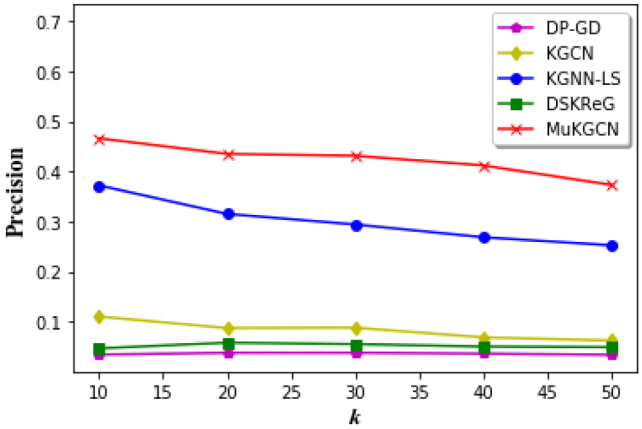


Fig. 11. The influence of varying k *on precision* for MovieLens dataset.

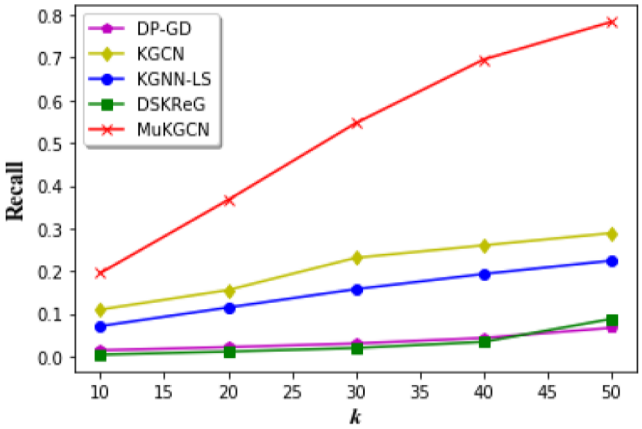


Fig. 12. The influence of varying k on *recall* for MovieLens dataset.

involves cross-domain data and there is no overlap between items in different domains. It is difficult to accurately capture the associations between these items from different domains, resulting in many missing data in the items knowledge graph, which affects score prediction. When running on Amazon dataset, the precision and recall of KGCN-LS are relatively low due to the lack of sufficient data relational information. The two neighboring recommendation lists are very different. We also observe that KGCN shows poor diversity even on the MovieLens dataset. The KGCN method does not distinguish the weight of different associations, leading to huge redundant

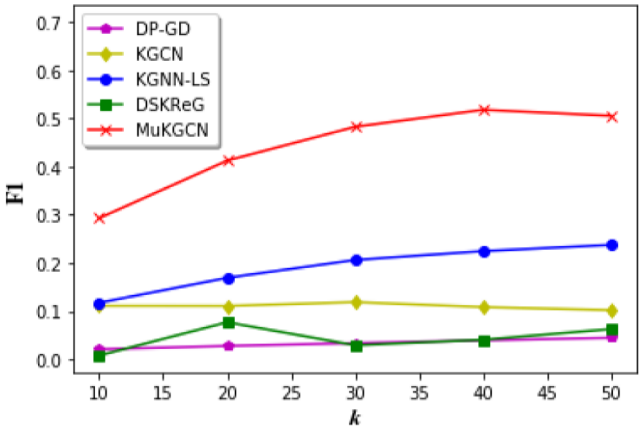
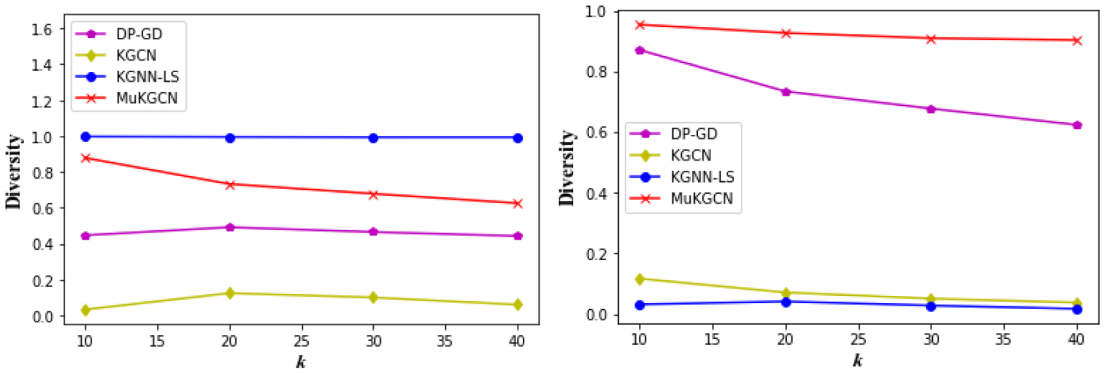


Fig. 13. The influence of varying k on *F-measure* for MovieLens dataset.

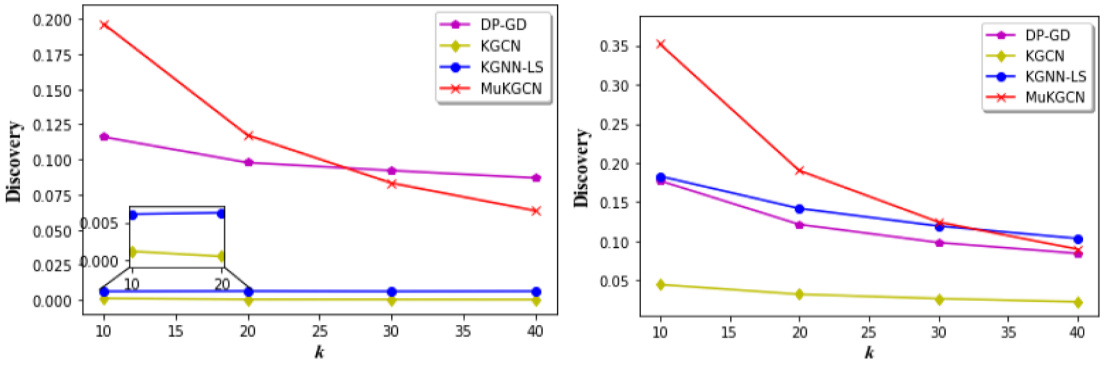


(a) Amazon dataset (b) MovieLens dataset

Fig. 14. The influence of varying k on *diversity*.

information, which in turn affects recommendation performance. The performance of DP-GD is unstable because it adds noise to the original data based on differential privacy model, which in turn increases the uncertainty of users' historical data and affects the analysis and prediction of the recommendation results.

Discovery is used to measure the ability of discovering items that are not within the user's explicit interest based on the ratio of the number of items that first appear and are selected by a user to the recommendation list length. The influence of varying k on discovery on the two datasets are shown in the Fig. 15. MuKGCN shows a clear advantage and a deceasing trend as k increases. But it intersects



(a) Amazon dataset (b) MovieLens dataset

Fig. 15. The influence of varying k on *discovery*.

with the other baseline methods and is even lower than the other methods at $k = 40$. When the recommendation length k is long enough, more and more items will fall into the recommended list, and fewer items will appear for the first time. By comparing the experimental results, we see that the MuKGCN has clear advantages, especially when the recommendation list length is very limited. When $k = 10$, the *discovery* of the MuKGCN method is 8 % higher than those of DP-GD (the best among the others) on Amazon dataset, and 17 % higher than those of KGNN-LS (the best among the others) on MovieLens dataset. The results demonstrate that MuKGCN can be applied to resource-constrained recommendation scenarios. In reality, the length of the recommendation list should be limited to improve the efficiency of recommendation algorithms and satisfy the real-time recommendation requirements. Therefore, the proposed MuKGCN has good practicability.

Furthermore to examine the effects of different parameters on recommendation performance, we also conduct ablation experiments and comparative analysis on the effects of different dimensions, different aggregations, different sizes of neighbor sampling, and different features.

Ablation experiments with different dimensions and different aggregations are performed on Amazon and Movielens datasets. The experimental results for varying feature dimension from 8 to 128 are shown in Table 4. We can see that the best recommendation accuracy is achieved when the dimension is 64 on both datasets. It indicates that have more user features does not imply better recommendation performance. Too many features introduce noise and degrade the recommendation performance. Table 5 shows the results of transforming the aggregation method of weighted aggregated neighbor vector and user vector into the following three different methods: sum of user vector and neighbor vector, concat of user vector and neighbor vector, directly representing with neighbor vector (neighbor). The other experimental parameters are kept to the same values. We observe that, the sum method has an advantage on both datasets. The F1 index on Movielens dataset increases by 4 % compared to the suboptimal results when $k = 40$. We therefore set the dimension value to 64 and use the sum method when aggregating of the center vector and the neighbor vector except for special designation.

We also conduct ablation experiments with different sampling neighbors on both datasets. The experimental results of varying the neighbor size from 2 to 16 are shown in Table 6. On the Amazon dataset, the best recommendation accuracy is achieved when the sampling neighbor size is 8. When we run experiments on Movielens dataset, the best sampling neighbor size is mostly 2, but it is not stable. We conclude that for dense dataset Movielens, sufficient information can be obtained for training even when the neighbor sampling is 2, while for relatively sparse dataset, Amazon, more neighbor information needs to be sampled for a better representation.

Ablation experiments with different features are conducted in terms of precision, recall, and F-measure running on the Amazon dataset. We randomly select five categorical attributes of Location, Education, Occupation, Sex and Marriage and a numerical attribute of Age for testing. The experimental results are shown in Fig. 16, where we record that the optimal performance regardless of precision, recall, or F-measure is the Location feature, followed by Education. This indicates that the two features of region and education level have the greatest influence on users' preferences. Comparatively, marital status has little influence on users' preferences.

Experimental results show that MuKGCN can effectively improve recommendation performance and is good at discovering users' implicit needs in both dense or sparse datasets.

Regarding the security MuKGCN as analyzed in Section 5.2, individuals are separated from sensitive associations by replacing individual users with features when securing local data. Local data preprocessing is secured by the TEE environment. The security of uploaded features is protected in an encrypted manner. In summary, the MuKGCN method can effectively protect users' privacy by removing their entities and encrypting their features, while preserving the statistical ratio of features to items for lossless aggregation. Our MuKGCN can also effectively improve data reliability by exploiting the incentive mechanism and the traceability of blockchain since blockchain's traceability enables the non-repudiation of data sources. Therefore, the security of the proposed method is guaranteed.

7. Conclusion

In this paper, we focused on the impact of data privacy security, anonymous data authenticity verification, and recommendation result accuracy on cross-domain recommendation scenarios, while proposing a trusted cross-domain recommendation model based on MuKG and blockchain to address the complex coupling relations and conflicts among the above three. The proposed method improves the accuracy of recommendation results while protecting both the security of private data and reliable verification of anonymous data. First, we proposed MuKG which enables lossless fusion of multi-source feature data with privacy protection. Then, we designed a trusted verification mechanism, based on the blockchain and MuKG to ensure the source traceability of anonymous data and the authenticity of the data by preventing tampering of the MuKG in the blockchain. Meanwhile, we designed an adaptive recommendation algorithm based on MuKGCN which combines the user's personalized features and the unified MuKG to achieve safe and accurate personalized recommendations. Finally, theoretical analysis and experimental results show that our MuKGCN can overcome the

Table 4
The impacts of different dimensions on recommendation accuracy.

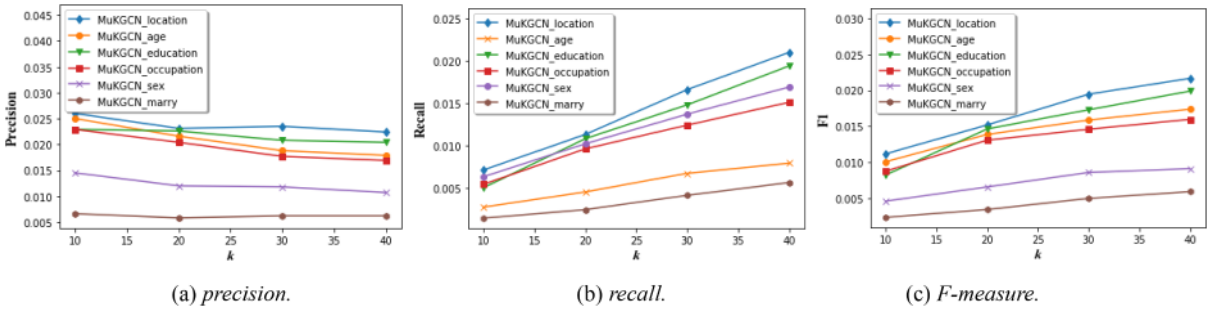
Dataset	Metric	dim = 8	dim = 16	dim = 32	dim = 64	dim = 128
Movielens	precision@20	0.2910	0.2650	0.3810	**0.4350**	0.4300
	recall@20	0.2556	0.2238	0.3247	**0.3666**	0.3632
Amazon	precision@20	0.2383	0.2383	0.2406	**0.2422**	0.2411
	recall@20	0.1184	0.1178	0.1191	**0.1196**	0.1188

Table 5
The effects of varying aggregation on F-measure.

Dataset	Aggregation	F1@10	F1@20	F1@30	F1@40	F1@50
Movielens	concat	0.2660	0.3937	0.4535	0.4774	0.4648
	sum	**0.2903**	**0.4125**	**0.4827**	**0.5180**	**0.5056**
	neighbor	0.2560	0.4075	0.4747	0.4860	0.4666
Amazon	concat	**0.1032**	0.1672	0.1920	0.2068	**0.2057**
	sum	0.1025	**0.1680**	**0.1924**	**0.2082**	0.2053
	neighbor	0.0006	0.0004	0.0004	0.0003	0.0003

Table 6
The effects of varying sampling neighbor size on F-measure.

Dataset	Neighbor size	F1@10	F1@20	F1@30	F1@40	F1@50
Movielens	2	0.2805	**0.4137**	**0.4878**	**0.5248**	0.5000
	4	**0.2903**	0.4125	0.4827	0.5180	0.5056
	8	0.2754	0.3979	0.4821	0.5174	**0.5128**
	16	0.2298	0.3690	0.4480	0.4910	0.4865
Amazon	2	0.0961	0.1413	0.1612	0.1718	0.1775
	4	0.1003	0.1540	0.1783	0.1866	0.1905
	8	0.1025	**0.1680**	**0.1924**	**0.2082**	**0.2053**
	16	**0.1059**	0.1582	0.1838	0.1956	0.2008



(a) *precision.* (b) *recall.* (c) *F-measure.*

Fig. 16. The influence of different features changing on recommendation performance.

conflict between privacy protection, recommendation accuracy, and data authenticity verification to achieve multiple guarantees of security and accuracy. MuKGCN will advance the development and application of cross-domain recommendation techniques.

CRediT authorship contribution statement

Li-e Wang: Writing – original draft, Validation, Formal analysis, Visualization, Software, Methodology, Investigation, Conceptualization, Data curation. **Yuelan Qi:** Investigation, Software, Validation, Visualization. **Yan Bai:** Writing – review & editing. **Zhigang Sun:** Resources, Software, Validation, Writing – review & editing, Data curation. **Dongcheng Li:** Software, Validation. **Xianxian Li:** Methodology, Writing – review & editing, Funding acquisition, Supervision, Project administration.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgements

The authors wish to thank the editors and anonymous reviewers for their helpful comments on and suggestions for this paper. This work is supported in part by the National Natural Science Foundation of China (Nos. 62262003 and U21A20474) and the Guangxi

Science and Technology Major Project (No. AA22387), the Guangxi Natural Science Foundation (No.2020GXNSFAA297075), the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, the Guangxi Collaborative Innovation Center of Multi-source Information Integration and Intelligent Processing, and the National Science Foundation (NSF) (award number 1921576).

References

[1] X. Li, E. Du, C. Chen, et al., Blockchain-Based Credible and Privacy-Preserving QoS-Aware Web Service Recommendation, in: International Conference on Blockchain and Trustworthy Systems (BlockSys 2019), 2019, pp. 621–635.
[2] C. Zhou, J. Peng, Y. Ma, et al., A Privacy-preserving Location Recommendation Scheme without Trustworthy Entity, in: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), 2021, pp. 444–451. DOI: 10.1109/TrustCom53373.2021.00073.
[3] C. Zhang, L. Zhu, C. Xu, K. Sharif, K. Ding, X. Liu, X. Du, M. Guizani, TPPR: A trust-based and privacy-preserving platoon recommendation scheme in VANET, IEEE Trans. Serv. Comput. 15 (2) (2022) 806–818.
[4] X. Li, P. Sun, Y. Bai, et al., M-generalization for multipurpose transcational data publication, Front. Comp. Sci. 12 (6) (2018) 1241–1254.
[5] D. Yang, B. Qu, P. Cudre-Mauroux, Privacy-preserving social media data publishing for personalized ranking-based recommendation, IEEE Trans. Knowl. Data Eng. 31 (3) (2019) 507–520.
[6] J.Y. Jiang, C.T. Li, S.D. Lin, Towards a more reliable privacy-preserving recommender system[J], Inf. Sci. 482 (2019) 248–265.
[7] S. Zhang, H. Yin, T. Chen, et al., Graph embedding for recommendation against attribute inference attacks, in: Proceedings of the Web Conference 2021, 2021, pp. 3002–3014.
[8] H. Shin, S. Kim, J. Shin, et al., Privacy enhanced matrix factorization for recommendation with local differential privacy, IEEE Trans. Knowl. Data Eng. 30 (9) (2018) 1770–1782.
[9] G. Beigi, A. Mosallanezhad, R. Guo, et al., Privacy-aware recommendation with private-attribute protection using adversarial learning, in: Proceedings of the 13th International Conference on Web Search and Data Mining, 2020, pp. 34–42.
[10] Y. Zhou, J. Liu, J.H. Wang, et al., Usst: A two-phase privacy-preserving framework for personalized recommendation with semi-distributed training, Inf. Sci. 606 (2022) 688–701.
[11] Y. Himeur, S.S. Sohail, F. Bensaali, et al., Latest trends of security and privacy in recommender systems: a comprehensive review and future perspectives, Comput. Secur. 102746 (2022) 1–28.
[12] C. Gao, Y. Li, F. Feng, et al., Cross-domain recommendation with bridge-item embeddings, ACM Trans. Knowledge Discov. Data 16 (1) (2022), https://doi.org/10.1145/3447683.
[13] T. Ogunseyi, C. Avoussoukpo, Y. Jiang, Privacy-preserving matrix factorization for cross-domain recommendation, IEEE Access 9 (2021) 91027–91037.
[14] T.B. Ogunseyi, B. Tang, Y. Cheng, A privacy-preserving framework for cross-domain recommender systems, Comput. Electr. Eng. 93 (2021) 1–15, https://doi.org/10.1016/j.compeleceng.2021.107213.
[15] L. Qi, X. Wang, X. Xu, et al., Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing, IEEE Trans. Network Sci. Eng. 8 (2) (2021) 1145–1153.
[16] L. Qi, X. Zhang, W. Dou, C. Hu, et al., A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment, Futur. Gener. Comput. Syst. 88 (2018) 636–643.
[17] Q. Yang, Y. Liu, T. Chen, et al., Federated machine learning: Concept and applications, ACM Trans. Intell. Syst. Technol. 10 (2) (2019) 1–19.
[18] L. Wang, Y. Wang, Y. Bai, et al., POI Recommendation with Federated Learning and Privacy Preserving in Cross Domain Recommendation, In Proceedings of IEEE Conference on Computer Communications Workshops (2021) 1–6.
[19] V. Perifanis, G. Drosatos, G. Stamatelatos, et al., FedPOIRec: Privacy-preserving federated poi recommendation with social influence, Inf. Sci. 623 (2023) 767–790.
[20] L. Zhu, Z. Liu, S. Han, Deep leakage from gradients, in: Proceedings of Conference and Workshop on Neural Information Processing Systems, 2019, pp. 17–31.
[21] T. Qi, F. Wu, C. Wu, et al., Privacy-preserving news recommendation model learning, in: Proceedings of EMNLP (Findings), 2020, pp. 1423–1432.
[22] C. Chen, J. Zhou, B. Wu, W. Fang, et al., Practical privacy preserving POI recommendation, ACM Trans. Intell. Syst. Technol. 11 (5) (2020) 1–20.
[23] C. Wu, F. Wu, Y. Cao, et al., FedGNN: federated graph neural network for privacy-preserving recommendation, in: Proceedings of International Conference on Machine Learning, 2021, pp. 1–9, 10.48550/arXiv.2102.04925.
[24] S. Truex, N. Baracaldo, A. Anwar, et al., A hybrid approach to privacy-preserving federated learning, Informatik Spektrum 42 (5) (2019) 356–357.
[25] Y. Wang, Y. Tian, X. Yin, et al., A trusted recommendation scheme for privacy protection based on federated learning, CCF Trans. Network. 3 (3–4) (2020) 218–228.
[26] R. Bosri, M.S. Rahman, M.Z.A. Bhuiyan, et al., Integrating blockchain with artificial intelligence for privacy-preserving recommender systems, IEEE Trans. Network Sci. Eng. 8 (2) (2021) 1009–1018.
[27] L. Lin, Y. Tian, Y. Liu, A blockchain-based privacy-preserving recommendation mechanism, in: Proceedings of 2021 IEEE 5th International Conference on Cryptography, Security and Privacy, 2021, pp. 74–78.
[28] Y. Himeur, A. Sayed, A. Alsalemi, et al., Blockchain-based recommender systems: Applications, challenges and future opportunities, Comput. Sci. Rev. 43 (2022) 1–21.
[29] H. Huang, J. Mu, N.Z. Gong, et al., Data poisoning attacks to deep learning based recommender systems, in: Proceedings of the Network and Distributed System Security, 2021, pp. 1–17.
[30] Fang M, Gong N Z, Liu J. Influence Function based Data Poisoning Attacks to Top-N Recommender Systems. In Proceedings of The Web Conference 2020 (WWW '20), 2020, pp.3019-3025.
[31] H. Huang, J. Mu, N.Z. Gong, et al., Data poisoning attacks to deep learning based recommender systems, in: Proceedings of the Network and Distributed System Security, 2021, pp. 1–17.
[32] L. Chen, S. Chen, H. Li, et al., Qian Yang. attacking recommender systems with augmented user profiles, in: Proceedings of the Conference on Information and Knowledge Management, 2020, pp. 855–864.
[33] A.A. Omar, R. Bosri, M.S. Rahman, et al., Towards privacy-preserving recommender system with blockchains, in: Proceedings of International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications, 2019, pp. 106–118.
[34] E. Bandara, X. Liang, P. Foytik, et al., A blockchain empowered and privacy preserving digital contact tracing platform, Inf. Process. Manag. 58 (4) (2021) 1–17.
[35] L. Wang, D. Li, P. Liu, X. Li, Bam_crs,, Blockchain-based anonymous model for cross-domain recommendation systems, J. Comput. Sci. Technol. (2021). https://jcst.ict.ac.cn/EN/10.1007/s11390-021-0657-9.
[36] Y. Chen, F. Luo, T. Li, et al., A training-integrity privacy preserving federated learning scheme with trusted execution environment, Inf. Sci. 522 (2020) 69–79.
[37] X. Li, P. Jiang, T. Chen, et al., A Survey on the security of blockchain systems, Futur. Gener. Comput. Syst. 107 (2020) 841–853.
[38] C. Regueiro, I. Seco, S. Diego, et al., Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption, Inf. Process. Manag. 58 (6) (2021) 1–17.
[39] Y. Lin, Z. Liu, M. Sun, et al., Learning entity and relation embeddings for knowledge graph completion, in: In Proceedings of the Twenty-ninth AAAI conference on artificial intelligence, 2015, pp. 2181–2187.
[40] Z. Zolaktaf, R. Babanezhad, R. Pottinger, A Generic Top-N Recommendation Framework for Trading-off Accuracy, Novelty, and Coverage, in: In Proceedings of the IEEE 34th International Conference on Data Engineering, 2018, pp. 149–160.
[41] N. Lathia, S. Hailes, L. Capra, et al., Temporal Diversity in Recommender Systems, in: In Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2010, pp. 210–217.

[42] L. Wang, D. Li, X. Li, Deep recommendation model with cross-domain association and privacy protection, J. Software (2022) 1–20, https://doi.org/10.13328/j.cnki.jos.006533 (In Chinese).
[43] H. Wang, M. Zhao, X. Xie, et al., Knowledge graph convolutional networks for recommender systems, in: In Proceedings of the World Wide Web Conference, 2019, pp. 3307–3313.
[44] H. Wang, F. Zhang, M. Zhang, et al., Knowledge-aware graph neural networks with label smoothness regularization for recommender systems, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 968–977.
[45] Y. Liu, S. Yang, Y. Xu, et al., Contextualized graph attention network for recommendation with item knowledge graph, IEEE Trans. Knowl. Data Eng. 35 (1) (2023) 181–195, https://doi.org/10.1109/TKDE.2021.3082948.
[46] Y. Wang, Z. Liu, Z. Fan, et al., Dskreg: Differentiable sampling on knowledge graph for recommendation with relational GNN, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3513–3517, https://doi.org/10.1145/3459637.3482092.
[47] H. Wang, F. Zhang, M. Zhao, et al., Multi-task feature learning for knowledge graph enhanced recommendation, in: The World Wide Web Conference, 2019, pp. 2000–2010.
[48] H. Wang, F. Zhang, J. Wang, et al., Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM international conference on information and knowledge management, 2018, pp. 417–426.