

Research on knowledge graph-based personalized recommendation algorithms and privacy protection in distributed networks

Jingfu Li^{1,*}

¹School of International Education, HuangHuai University, Zhumadian 463000, China

Abstract

To improve the personalized recommendation performance of online platforms in a distributed environment while also considering user privacy protection, this paper proposes a hyperbolic attention knowledge graph recommendation algorithm for distributed architectures, integrating contextual information and privacy protection mechanisms. First, In the design of the recommendation model, a hyperbolic attention model is employed to organically integrate semantic features with an attention mechanism, thereby fully capturing the inter-relationships between products. Next, a context-aware model is designed by combining an LSTM-based memory model with AFM-based feature crossing. Furthermore, to address the aforementioned security threats, differential privacy techniques are introduced to inject noise into user features, and collaborative model training across multiple nodes is implemented within a federated learning framework. Experimental results show that, under distributed and privacy-preserving constraints, knowledge graph recommendation algorithms can achieve better precision, recall, ACC, and AUC under the same conditions. Comparing our algorithm with other algorithms, our proposed algorithm improves the ACC by 2.9 to 11 percentage points and the AUC by 3.2 to 9.1 percentage points. Similarly, our proposed algorithm achieves RMSE and MAE of 0.960 and 0.858, respectively, which can demonstrate a significant advantage. It also performs well in terms of privacy protection (such as ϵ -differential privacy parameters) and communication efficiency. Furthermore, system security evaluation experiments demonstrate that when the privacy budget $\epsilon \geq 2.0$, the model effectively reduces the success rate of membership inference attacks and significantly degrades the reconstruction quality of gradient inversion attacks; it also exhibits a mitigating effect against model poisoning attacks under certain conditions, thereby validating the algorithm's security and practicality in real-world distributed recommendation scenarios. Comparative results across the various algorithms show that the proposed algorithm remains the best among all models in terms of precision and recall.

Keywords: knowledge graph, attention mechanism, deep learning, recommendation algorithm, distributed network, privacy protection, federated learning

Received on 28 January 2026, accepted on 01 September 2026, published on 23 October 2026

Copyright © 2026 Jingfu Li, licensed to EAI. This is an open access article distributed under the terms of the [CC BY-NC-SA 4.0](#), which permits copying, redistributing, remixing, transformation, and building upon the material in any medium so long as the original work is properly cited.

doi: 10.4108/eetsis.11735

*Corresponding author. Email: 20070524@huanghuai.edu.cn

1. Introduction

The advancement of internet technology has drastically changed people's lifestyles, with traditional mall shopping

gradually being replaced by online e-commerce [1]. Especially during large-scale e-commerce events, a large number of consumers are attracted, generating massive amounts of data, leaving users overwhelmed by redundant information when selecting products [2]. In distributed network environments, user data is often scattered across

different nodes or edge devices. Traditional centralized recommendation systems face multiple challenges, including data silos, high communication costs, and user privacy leaks. In a distributed environment, data is heterogeneously distributed across different nodes. The communication overhead and synchronization latency between nodes become key bottlenecks affecting model training efficiency. How to achieve efficient node collaboration while ensuring recommendation accuracy is one of the core challenges faced by distributed recommendation systems.

How to quickly and effectively find the products users need, improve shopping efficiency, and simultaneously protect user privacy and data security has become a key concern for businesses [3]. Recommendation systems, based on existing information data, can use algorithms to identify user needs from a complex and diverse range of products, providing personalized services and achieving a win-win situation for both parties. This has attracted considerable attention from researchers. However, in distributed scenarios, ensuring recommendation accuracy while preventing the leakage of sensitive user information remains one of the key challenges in current recommendation system research.

The product recommendation system greatly improves the accuracy of product recommendations by analysing users' potential needs and using big data technology and deep learning mechanisms. The attention mechanism analyses users' personal behaviours and effectively distinguishes the important parts of the data, thus improving the accuracy of product recommendations. Lu P et al. [4] designed a DNN dual-tower model for personalized recommendation, aiming to provide users with more precise content and services. The network model was built using the TensorFlow framework to generate embeddings for users and items. Additionally, the system employs a collaborative filtering algorithm combined with techniques such as Word2Vec for feature engineering, thereby balancing the exploration-exploitation trade-off within the recommendation system. Knowledge graph technology is also used for product recommendations. Because it can map product attribute information and obtain the internal associations between items, it can better capture users' personal preferences and alleviate the sparsity problem in the data processing process. Tang Hong et al. [5] combined knowledge graphs with optimized graph attention networks to achieve the fusion and synergy of the two, and finally achieved good results in product recommendations. Hongwei Wang et al. [6] combined knowledge graphs with deep learning, used embedding tasks to achieve the mutual associations between different task compression units, and then obtained their internal features, which also achieved good results. In terms of distributed recommendation systems, Peng P et al. [7] proposed a lightweight federated recommendation framework for edge computing; Kang Wei et al. [8] systematically analysed the privacy budget allocation strategy of differential privacy in federated learning; Li Y et al. [9] designed a cross-domain recommendation model based on knowledge graphs, which can provide a reference for knowledge transfer in distributed scenarios.

Although the above algorithms can play a role in the product recommendation process, they still face challenges such as the cold start problem of traditional algorithms and data sparsity. Especially in a distributed environment, they also face challenges such as data heterogeneity, communication constraints, and privacy protection. As a result, the results achieved by the above algorithms are still unsatisfactory.

To address the efficiency issues in e-commerce product recommendations and the privacy protection requirements in distributed network environments, this paper, building upon previous research, proposes a hyperbolic attention knowledge graph recommendation algorithm for distributed architectures that integrates contextual information and privacy protection mechanisms. Furthermore, personalized features of users and products are extracted and privatized locally or at the edge. A secure aggregation mechanism is used to update the global model, which can generate recommendation results tailored to users' deep interests. Compared with existing methods, the main contributions of this paper are: 1) For distributed knowledge graph recommendation systems, the system defines threat models and security boundaries; 2) Recommendation algorithms are designed for distributed network environments, supporting local feature extraction and collaborative model training; 3) Differential privacy and federated learning mechanisms are integrated to ensure that user data does not leave the local machine while maintaining recommendation accuracy; 4) System security assessments, including that privacy-utility trade-offs, member inference defence, gradient inversion defence, and robustness against poisoning attacks, are completed on public datasets. Experiments show that the model can effectively improve recommendation performance and maintain high recommendation performance while protecting user privacy, providing insights for solving similar problems.

2. Personalized recommendation algorithms based on knowledge graphs in distributed networks and privacy protection

2.1. Knowledge Graphs, Attention Mechanisms and Distributed Privacy Protection

Knowledge graph is a heterogeneous network that is mainly used to describe the internal connections between multiple entities. It can be used for the association of objective objects as well as the relationship between abstract concepts. Applying it to the recommendation algorithm can effectively construct the relevant attributes between products, especially for analysing the association between products. Combined with the user's information data attributes, it can effectively connect users with products, analyse users' personal preferences, realize personalized recommendations, and improve the recommendation accuracy of the algorithm model. In distributed networks, knowledge graphs can be

broken down and stored across multiple nodes. Through distributed computing and privacy-preserving aggregation techniques embedded in the graph, semantic association mining across nodes can be achieved while avoiding the leakage of original graph data.

The combination of knowledge graph and recommendation algorithm mainly includes three forms: sequential learning, joint learning and alternating learning [10]. The main reason is that the learning order of knowledge graph module and recommendation algorithm module is different, so the results are not consistent. Sequential learning refers to taking the vector relationship obtained from knowledge graph as the input of recommendation algorithm, so as to achieve the prediction and evaluation of recommended products [11]. Joint learning is to structure the two and give full play to their advantages. The recommendation effect is relatively better, but the calculation amount is also larger. Alternating learning is to share information data as two independent training tasks, and the parameters of the two modules need to be trained alternately [12]. In distributed scenarios, the above learning methods can be combined with federated learning frameworks to achieve collaborative training of knowledge graphs and recommendation models through parameter exchange rather than raw data sharing, significantly improving privacy and security.

The attention mechanism is a mechanism that simulates human behaviour. It focuses on important information to improve the emphasis on key information in the data [13]. The attention mechanism in the algorithm model achieves the division of the importance of input data by allocating weights. It can be used for different levels of importance, which can effectively improve the capture and processing of key information in the algorithm and reduce the computational workload of non-key information. In distributed recommendation, attention mechanisms can be deployed on the user side for local feature selection and privacy filtering, with only the de-identified attention weights uploaded to the aggregation node, reducing the risk of privacy exposure.

As a feature enhancement and screening mechanism, the attention mechanism is widely used in deep learning fields such as information recognition and data processing. The attention mechanism can effectively record the positional relationship between data information and divide the importance of features according to the degree of weight, thereby improving the efficiency of the deep learning algorithm model and the accuracy of the recommendation algorithm [14].

2.2. Knowledge graph recommendation algorithm based on attention mechanism and privacy protection

For personalized product recommendations on online platforms, there is a lot of information collection involved. Relevant modules are used to identify the product information that users are interested in, and the relevant data is used for training subsequent recommendation algorithms

[15]. In distributed networks [16], user data is no longer stored centrally, but distributed on terminal devices or edge servers. For users, the products they purchase mainly include popular products and some similar products. Therefore, the frequency of similar data with relevance increases, leading to the increasing popularity of such products and a decrease in the number of times unpopular products are recommended. There may even be cases where products that should be recommended are not recommended, which can result in a popular product bias. This bias may be further aggravated in a distributed environment due to uneven data distribution.

Meanwhile, for daily purchasing behaviour, there will be a lot of usage noise, and some products may be accidentally clicked or not clicked by the user, which may cause the information collection system to misjudge the user's behaviour. The emergence of such data will have a negative impact on the training of the model [17]. Current recommendation systems mostly focus on the relationship between users and products, lacking analysis of the relationships between products themselves. This leads to errors in the recommendation results, and the conflict between constructing cross-node product relationships and protecting privacy becomes even more pronounced in distributed environments. Therefore, this paper studies a knowledge graph recommendation algorithm based on an attention mechanism that integrates privacy protection for distributed environments. The algorithm's architecture is shown in Figure 1.

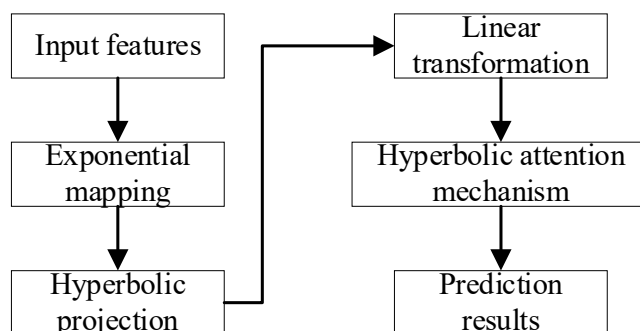


Figure 1. Knowledge graph recommendation algorithm architecture based on attention mechanism and privacy protection in distributed environments

As shown in Figure 1, for personalized product recommendations on online platforms, this model utilizes hyperbolic property mapping to decompose the substitution and complementarity relationships between products, thus forming two sets of embedding vectors to predict the two types of relationships respectively. This mapping process can be executed locally on the user's device or at edge nodes, without needing to upload the original product relationship data to the central server. Networks are constructed separately for substitution and complementarity relationships based on the relationships between products, assuming strong correlations between them. Therefore, its predictive

performance can be improved by adding a correlation module, ultimately generating the corresponding prediction results. Furthermore, differential privacy technology is used to add noise to the output embedding vectors to ensure that relationship information cannot be inferred backwards.

The useful of attention mechanism can improve the ability to capture product features, and the use of multiple self-attentions can better improve the effect of product recommendations. The output of self-attention is cascaded, and the final result can be expressed as:

$$v = \parallel_{k=1}^K v^{(k)} \quad (1)$$

In the formula (1), the operator $\parallel$ indicates that this is a cascade operation, $v(k)$ represents the result of the k -th self-attention in the algorithm, and the number of self-attentions used in the model is represented by K .

For the multi-head attention mechanism in hyperbolic space, it is necessary to design it differently from the Euclidean space. Assume that the potential representation of node i is H_i , which contains K m -dimensional features, namely:

$$H_i = \parallel_{k=1}^K h_i^{(k)} \quad (2)$$

At this point, the improved attention coefficient can be expressed as:

$$\alpha_{ij} = -d_p(H_i, H_j) \quad (3)$$

This attention coefficient, by measuring the distance between nodes in hyperbolic space, can more naturally capture the hierarchical correlations between goods, and is particularly suitable for modelling substitution and complementarity relationships. Compared with attention mechanisms in traditional Euclidean space, hyperbolic space can express complex hierarchical semantics with lower dimensions. Where d_p is the distance between two points in the hyperbolic space, which can be expressed as:

$$d_p(H_i, H_j) = \sqrt{\sum_{k=1}^K d_c^2(h_i^{(k)}, h_j^{(k)})} \quad (4)$$

Where d_c is the vector distance between two points in hyperbolic space. The attention coefficients of all neighbours belonging to node i are normalized and expressed as:

$$w_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{i \in N_i} \exp(\alpha_{ij})} \quad (5)$$

Addition and multiplication calculations in hyperbolic space is applied to aggregate each attention feature, it can be expressed as:

$$h_i'^{(k)} = \sigma^{\otimes_c} \left(\sum_{j \in N_i}^{\oplus_c} w_{ij} \otimes_c h_j'^{(k)} \right) \quad (6)$$

Where σ represents a nonlinear activation function, and N_i represents the set of neighbouring nodes of node i . Finally, in this hyperbolic space, the multi-head attention mechanism used in this algorithm can be expressed as follows:

$$H_i' = \parallel_{k=1}^K h_i'^{(k)} \quad (7)$$

The above attention calculation process can be performed locally on distributed nodes. Each node only needs to exchange the aggregated attention weight vector, rather than the original features, which effectively protects the privacy of local node data.

Since there is a certain correlation between recommended products, it is necessary to analyse the strength of the relationship between products. How to effectively identify the strong and weak relationships between products will affect the effect of the recommendation algorithm and the training efficiency of the model. This paper designs a feature correlation module to further improve the attribute distance between products, thereby improving the prediction efficiency of alternative or complementary products. In a distributed architecture, this module collaboratively performs global similarity modelling without directly exposing raw product features; it achieves this by calculating the cosine similarity of local product features at each node and uploading the de-identified similarity statistics to a central server for weighted aggregation.

The category features and configuration features between products are used as word vectors to create a vector label for each product [18]. The higher the feature similarity of the products, the higher the similarity of their feature vectors, which is then measured using cosine similarity. The formula is:

$$similarity = \frac{A \cdot B}{\|A\| \times \|B\|} \quad (8)$$

In the model training process, the graph attention mechanism is used to strengthen the adjacent aggregation of network node information. By calculating the cosine similarity of the product feature vector, the similarity distance between products can be obtained, and then the weighted information is used to aggregate them. In the training process, the similarity relationship between products is considered, and the sequence of recommended products can be corrected to make the recommendation results more reasonable. This aggregation process is carried out within a federated learning framework, where a central server is responsible for aggregating gradients or model parameter updates from each node, rather than the original similarity data, thereby achieving privacy protection in collaborative training.

2.3. Deep learning recommendation model that integrates contextual information and privacy protection

For personalized product recommendations, a large amount of information needs to be processed, especially some auxiliary information will be considered, such as the user's rating of the purchased products, which often has a higher weight in the product recommendation process [19]. However, since the user's rating is random, there are also cases where some products have been rated falsely, which affects the accuracy of the rating in obtaining user characteristics, thereby affecting the product recommendation effect [20].

The existing product recommendation model lacks effective use of user comment information and lacks time-series processing of relevant information [21], which leads to the recommendation model not considering the timeliness of user personal interests and the comprehensive evaluation is not comprehensive enough. Meanwhile, the user's attribute information cannot be deeply mined. User's occupational characteristics, age characteristics and other information can also be used for product recommendation. Deeply combining them can improve the recommendation effect of the model. Most of the existing recommendation algorithms consider the user's historical behaviour, but lack deep mining of features, especially cross-features with certain connections. In distributed scenarios, this contextual information is scattered across different devices, and directly using it centrally would pose serious privacy risks. Therefore, this paper proposes a deep learning recommendation model for distributed environments that integrates contextual information with privacy protection techniques. The model structure is shown in Figure 2.

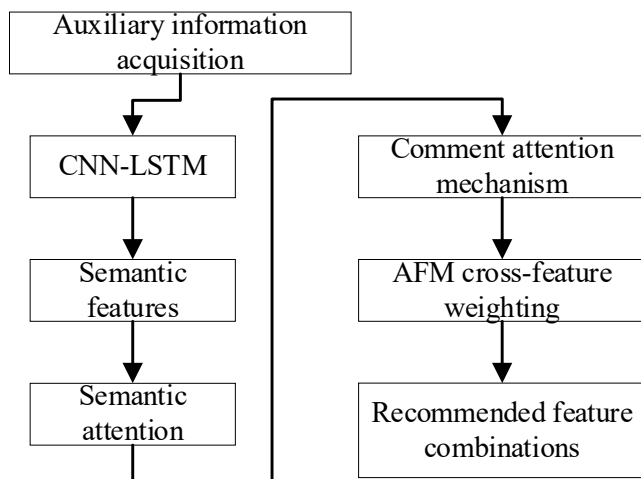


Figure 2. Recommendation models that integrate contextual information and privacy protection in distributed environments

As shown in Figure 2, the model structure fully integrates user auxiliary information features, summarizing user preferences through information such as age, occupation, and reviews, and improving the model's recommendation performance based on the complementary and substitutive relationships between products. Sensitive personal information (such as age and occupation) is processed locally using differential privacy or anonymization techniques before participating in model training. The model organically combines CNN and LSTM networks to capture key information from both short and long sequences. LSTM units can be deployed on the user's end for local sequence modelling, with only the final state vector or gradient being encrypted and uploaded. The AFM model is used to weight cross-feature combinations to obtain the most effective recommendation feature combinations. The attention weights of the AFM can be calculated locally and securely participate in global feature cross-weight learning through homomorphic encryption and other methods.

2.4. Combining recommendation algorithms and distributed privacy protection frameworks

Finally, by comprehensively utilizing the recommendation algorithms described above and embedding them into a federated learning or distributed learning framework, the final algorithm model structure can be obtained as shown in Figure 3.

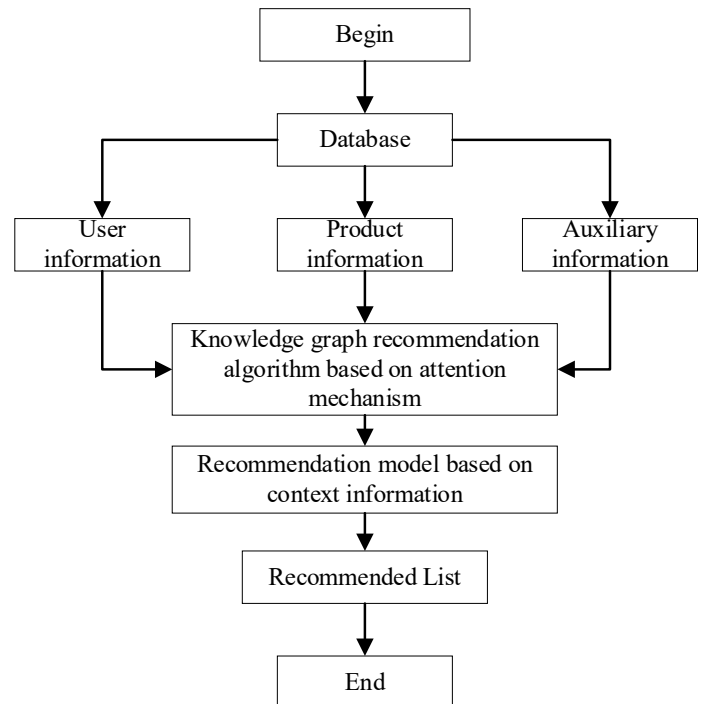


Figure 3. Distributed privacy protection framework model

Figure 3 illustrates the overall structure of the algorithm presented in this paper within a distributed privacy protection framework. Each client (such as a user terminal or edge node) stores its private data locally, trains its local model using the methods described in Sections 2.2 and 2.3, and uploads updated model parameters to the central server. The central server uses a federated averaging algorithm (FedAvg) to securely aggregate the parameters, generate a global model, and distribute it to each client. Throughout the entire process, the original data remains locally, and collaborative training is achieved through model parameter exchange, thus balancing recommendation performance and privacy protection in a distributed environment.

2.5. Threat models and security assumptions

To rigorously demonstrate the privacy protection capabilities of this paper, this section clarifies the threat model and security boundaries considered. The entire system is based on a federated learning framework, with participants including user clients (edge nodes) and a central server. It is assumed that the central server is "semi-honest," meaning it correctly executes the aggregation protocol but may attempt to analyse received gradients to infer user privacy. Clients are "honest but curious" by default, but some clients may be controlled by attackers and become malicious nodes.

Attackers primarily employ three types of attack methods: first, member inference attacks, attempting to determine whether a piece of data exists in the training set; second, gradient inversion attacks, attempting to reconstruct the original input from uploaded gradients; and third, model poisoning attacks, where malicious clients upload forged gradients to reduce model performance or implant backdoors. To address these attacks, this paper employs differential privacy to mitigate the first two types of attacks and leverages the aggregation properties of Federated Averaging to alleviate the impact of the third type of attack to some extent.

The goal of privacy protection is that, after gradients uploaded by any client are processed by a differential privacy mechanism, attackers cannot distinguish them with a significant advantage over any adjacent dataset. This paper adopts the standard (ϵ, δ) -differential privacy definition, with δ set to 10^{-5} , and adjusts ϵ to control the trade-off between privacy strength and recommendation accuracy. Simultaneously, the original user data never leaves the local device; the central server can only access the noisy model parameters or gradients, and cannot directly obtain user behaviour. These assumptions and defence objectives form the basis of the security analysis in this paper.

3. Experimental results and data analysis

3.1. Experimental setup

To verify the effectiveness of the algorithm in this study, the algorithm was verified using e-commerce open-source data. The selected datasets are the Retail-rocket dataset and the

Tian-chi dataset. Both datasets are derived from publicly available user behavior data from e-commerce platforms. The Retail-rocket dataset is sourced from the Kaggle public data platform (<https://www.kaggle.com/retailrocket/e-commerce-dataset>). It contains fields such as user ID, item ID, item category, event type (view, add-to-cart, or transaction), and timestamp, but does not include explicit ratings. To conduct rating prediction experiments, we followed standard practice by mapping transaction events to a rating of 5 and "add-to-cart" events to a rating of 4. Viewing events were mapped to ratings of 1–3 based on duration, thereby converting implicit feedback into explicit ratings. The Tianchi dataset originates from the Alibaba Tian-chi competition platform (<https://tianchi.aliyun.com/competition/entrance/532043/information>). It includes fields such as user ID, item ID, item category, brand ID, behaviour type (click, purchase, add-to-favourite, or add-to-cart), and timestamp; the same conversion strategy was applied to derive explicit ratings. This allows for the validation of personalized recommendations on online platforms. To simulate a distributed environment, the dataset was partitioned into multiple subsets based on user ID, with each subset treated as data belonging to an independent client. Specifically, user data was randomly assigned to 100 clients. A Dirichlet (0.5) distribution was used to simulate a Non-IID (non-independent and identically distributed) data partition; while the data volume across clients was roughly balanced, the distribution of categories varied. A 1:1 positive-to-negative sample ratio was adopted for negative sampling, and the value of K for the Top-K recommendation list was set to 10. Regarding knowledge graph construction, items and item categories served as entities, while substitution/complementary relationships between items and item-category membership served as relations. The knowledge graph constructed from the Retail-rocket dataset contains approximately 52,000 entities and 38,000 relations, whereas the graph from the Tian-chi dataset contains approximately 86,000 entities and 61,000 relations.

The personal PC configuration used in the experiment is: i7-11800H, NVIDIA RTX3060, 16G memory, Windows 10, and the deep learning framework is TensorFlow 2.4.0. 70% of the dataset is selected as the training set, 20% as the validation set, and the remaining 10% as the test set. Distributed training uses the FedAvg algorithm, which randomly selects a certain proportion of clients to participate in training in each round. In each round, 30% of the clients are randomly selected to participate in the training, and a total of 200 rounds of communication are conducted. The number of local training rounds is set to 5, and the client learning rate is set to 0.01. The random seed was set to 2024; all experiments were repeated five times, and the mean and standard deviation were reported. Regarding differential privacy configuration, δ was set to 10^{-5} and the gradient clipping threshold C to 1.0. The noise multiplier was derived from the target ϵ value using the Rényi Differential Privacy (RDP) accounting method, with a per-round sampling rate q of 0.3. Cumulative privacy expenditure was calculated via the RDP composition theorem to ensure the total privacy budget adhered to the preset ϵ value. A paired t-test was used for

significance testing, with $p < 0.05$ considered statistically significant.

To evaluate the effect of the recommendation algorithm, the selected evaluation indicators are ACC, AUC, Precision@K and Recall@K, and the final scoring results are sorted to obtain the Top-K recommendation list. ACC and AUC represent the accuracy and area under the ROC curve in machine learning, respectively. ACC is the proportion of accurately classified samples to the total number of samples in the sample classification process. AUC represents the probability that the predicted value of the positive sample is greater than the predicted value of the negative sample in the prediction result. For rating prediction tasks, RMSE and MAE are used as evaluation metrics; for click-through rate (CTR) prediction tasks, ACC and AUC are used; and for Top-K recommendation tasks, Precision@K and Recall@K are used.

Precision@K and Recall@K can be expressed as:

$$Precision@K = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \quad (9)$$

$$Recall@K = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \quad (10)$$

Where $R(u)$ is the recommendation list returned to the user based on the user u behaviour in the training set, and $T(u)$ is the behaviour list of user u in the test set.

According to the above ratings, the evaluation indicators of the entire algorithm are the mean absolute error (MAE) and the root mean square error (RMSE), both of which are measured based on the error between the actual rating data and the previous rating data. The indicators are expressed by the formulas as follows:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (obs_i - pre_i)^2} \quad (11)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N |(obs_i - pre_i)| \quad (12)$$

Differential privacy is implemented using a Gaussian mechanism, with the noise scale set according to the privacy budget ϵ and a pruning threshold $C=1.0$. For attack simulation, the member inference attack uses a shadow model method, training a set of shadow models with the same structure as the target model, and evaluating the attack accuracy and AUC. The gradient inversion attack uses the DLG algorithm (Deep Leakage from Gradients), reconstructing the input samples from the uploaded gradients and calculating the reconstructed PSNR. The model poisoning attack randomly selects 20% of malicious clients and uploads stochastic gradients amplified 10 times. Simultaneously, the changes in ACC and AUC, as well as the member inference attack success rate and gradient reconstruction error, are recorded at different ϵ values, including 0.1, 0.5, 1, 2, 5, and 10.

3.2. Performance of knowledge graph recommendation algorithms based on attention mechanism and privacy protection

3.2.1 Algorithm comparison

The distributed privacy-preserving knowledge graph recommendation algorithm of this study is compared with other algorithms. The experimental comparison results are shown in Table 1 under the same distributed simulation environment and privacy constraints. To ensure a fair comparison, all baseline models (MKR, RippleNet, and KGCN) are re-implemented within the same distributed simulation environment used in this paper; they uniformly employ the Federated Averaging (FedAvg) framework for distributed training and adhered to identical differential privacy budgets ($\epsilon=2.0$, $\delta=10^{-5}$) and client partitioning schemes. Hyperparameters for each baseline model are set according to the recommendations in their respective original papers.

Table 1. Comparison results of different algorithms

Algorithms	Retail-rocket				Tian-chi			
	ACC	AUC	Precision@K	Recall@K	ACC	AUC	Precision@K	Recall@K
MKR	0.758	0.809	0.093	0.091	0.719	0.728	0.092	0.085
RippleNet	0.743	0.758	0.088	0.085	0.640	0.670	0.083	0.081
KGCN	0.768	0.804	0.087	0.083	0.716	0.691	0.080	0.083
The algorithm in this paper	0.795	0.842	0.109	0.106	0.751	0.761	0.112	0.107

As shown in Table 1, comparing the attention-based and privacy-preserving knowledge graph recommendation algorithm described in Section 2.2 with other similar algorithms, it is evident that our algorithm achieves the highest ACC and AUC scores among all models. It demonstrates excellent performance on both the Retailrocket and Tianchi datasets. This proves that even with the introduction of a distributed architecture and privacy-preserving mechanisms, the algorithm maintains superior recommendation performance.

From the results in Table 1, we can see that the ACC and AUC indicators of the RippNet algorithm in all models are lower than those of other algorithms, and are the lowest among all algorithm models. The main reason is that its algorithm model adopts a path-based recommendation algorithm, which leads to insufficient efficiency in user behavior and product recommendation. KGCN also uses the knowledge graph and graph neural network algorithm used in this study. However, in its recommendation process, it mainly uses the relationship between neighbor vectors to reduce the sparsity of the matrix, without fully considering the data relationship between related products, so the results it achieves are not ideal. The MKR algorithm fully leverages the advantages of knowledge graphs and performs cross-learning with recommendation tasks, achieving good recommendation results, but overall, it is not as good as the algorithm in this study. This algorithm, by employing a hyperbolic attention mechanism and incorporating distributed training and privacy protection techniques, fully utilizes the correlation between substitute and complementary goods, which can enable the recommended products to better reflect user expectations. From the results, the algorithm in this paper has improved by 2.9 percentage points to 11 percentage points in the ACC indicator and 3.2 percentage points to 9.1 percentage points in the AUC indicator. Similarly, by comparing the results of each model, we can see that in terms of accuracy and recall, the algorithm in this paper is still the best among all models.

3.2.2 Ablation Experiment of Knowledge Graph Recommendation Algorithm

Ablation experiments were conducted on the knowledge graph recommendation algorithm based on attention mechanisms and privacy protection to examine performance changes resulting from the removal of the attention mechanism, the hyperbolic feature projection, and—individually—federated learning and differential privacy; the results are presented in Table 2. To evaluate the independent contribution of each privacy protection module in greater detail, three additional fine-grained ablation experiments were introduced: "FedAvg only" (federated learning without DP), "DP only" (differential privacy with centralized training), and "FedAvg+DP" (federated learning combined with differential privacy, but without the attention mechanism).

Table 2. Comparison of ablation experiment results

Algorithms	Retail-rocket		Tian-chi	
	ACC	AUC	ACC	AUC
No attention mechanism	0.765	0.802	0.735	0.728
No hyperbolic feature projection	0.772	0.821	0.742	0.754
FedAvg only (no DP, distributed)	0.803	0.851	0.760	0.770
DP only (centralized+DP, $\epsilon=2.0$)	0.791	0.839	0.748	0.758
FedAvg+DP ($\epsilon=2.0$, variant without attention)	0.778	0.825	0.739	0.747
The algorithm of this study (fully distributed privacy-protection)	0.795	0.842	0.751	0.761

The ablation study results in Table 2 demonstrate that models lacking either the attention mechanism or the hyperbolic feature projection algorithm underperform compared to the proposed algorithm in terms of ACC and AUC metrics, thereby validating the effectiveness of the proposed approach. As shown in Table 2, the "FedAvg only" configuration (without privacy protection) achieves the highest metrics—with ACC=0.803 and AUC=0.851 on the Retail-rocket dataset—while the complete model (ACC=0.795, AUC=0.842) shows a relative decline of approximately 1.0%, indicating that the privacy protection mechanism incurs an acceptable performance cost. The performance of "DP only" (centralized training) (ACC=0.791) is slightly lower than that of the complete model; this suggests that, given the same level of privacy protection, the distributed training framework does not significantly compromise accuracy but instead enhances model generalization through data from multiple nodes. The performance of the "FedAvg+DP" variant (lacking the attention mechanism) (ACC=0.778) is notably lower than that of the complete model, further confirming the effectiveness of the hyperbolic attention mechanism. These results demonstrate that the proposed method achieves an effective balance between recommendation accuracy and privacy protection.

3.2.3 Impact of model parameters

To comprehensively compare the effects of different algorithm parameters on the model, the model effects under different algorithm learning rates and knowledge graph learning rates are considered respectively. We also

analysed the impact of the privacy budget ϵ on model performance and privacy protection level.

To explore the impact of the knowledge graph learning rate parameter on the performance of the entire model, the learning effects under different knowledge graph learning rates in the two data sets are compared respectively, and the results are shown in Figure 4.

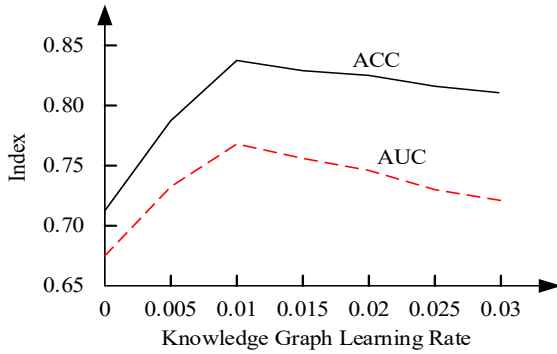


Figure 4. The impact of knowledge graph learning rate on evaluation indicators

As shown in Figure 4, when the learning rate of the knowledge graph is different, the evaluation indicators ACC and AUC of the model in the two data sets have changed significantly, and the change trends in the two data sets are similar. As the learning rate increases, the evaluation indicators first increase and then decrease. The main reason is that when the model learning rate is low, the algorithm's learning efficiency for the data is too low, and when the learning rate increases, the model will not converge. Therefore, among all the results, when the knowledge graph learning rate is 0.01, its recommendation algorithm model achieves the best performance.

Similarly, under the same circumstances, the learning rate of the recommendation model is changed to analyse its impact on the learning effect on different data sets. The final results are shown in Figure 5.

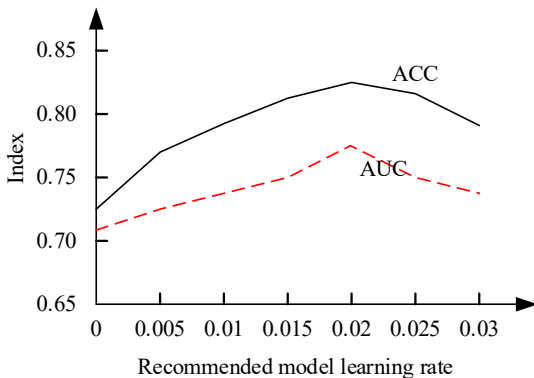


Figure 5. The impact of the learning rate of the recommendation model on the evaluation indicators

As shown in Figure 5, when the learning rate of the recommendation model is different, the evaluation indicators ACC and AUC of the model in the two data sets also show large changes, and the change trends in the two data sets are similar. As the learning rate increases, the evaluation indicators first increase and then decrease. The reason for this phenomenon is similar to the situation of the knowledge graph learning rate. Among all the results, when the learning rate of the recommendation model is 0.02, the recommendation algorithm model achieves the best performance. Figure 6 illustrates the impact of different privacy budgets ϵ (differential privacy noise intensity) on model performance.

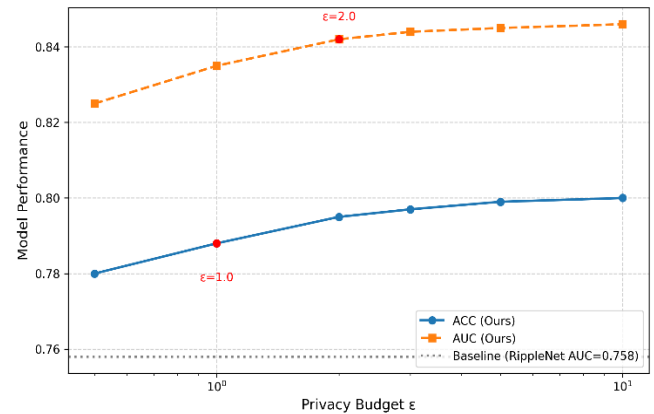


Figure 6. Impact of privacy budget ϵ on evaluation metrics

As shown in Figure 6, as ϵ decreases (indicating stronger privacy protection), the ACC and AUC metrics exhibit a slow downward trend. Even at $\epsilon = 1.0$, the model's performance remains significantly superior to that of the baseline model without privacy protection (such as RippleNet in Table 1), demonstrating that the proposed algorithm maintains high recommendation accuracy under strong privacy protection and achieves an effective trade-off between privacy and performance.

3.2.4 Privacy protection effectiveness and anti-attack capability assessment

(1) Privacy-utility trade-off analysis

Table 3 shows the variations in ACC and AUC for the model on the Retailrocket dataset under different privacy budgets (ϵ), as well as the corresponding privacy protection strength (relative assessment).

Table 3. Privacy-utility trade-off under different privacy budgets

ϵ	ACC	AUC	Privacy protection strength
0.1	0.742	0.791	Extremely strong
0.5	0.768	0.812	Strong
1.0	0.783	0.828	Moderately strong
2.0	0.795	0.842	Moderate
5.0	0.801	0.850	Weak
10.0	0.804	0.853	Extremely weak
No Privacy Protection($\epsilon=\infty$)	0.808	0.856	None

As shown in Table 3, as the privacy budget ϵ decreases from 10 to 0.1, the model's ACC slowly decreases from 0.804 to 0.742, and AUC decreases from 0.853 to 0.791, while the privacy protection strength increases from very weak to very strong. When $\epsilon=2.0$, the ACC and AUC are 0.795 and 0.842, respectively, only decreasing by 0.013 and 0.014 compared to the version without privacy protection, a decrease of less than 1.6 percentage points, but at this point, the privacy protection strength has reached a moderate level. This indicates that the algorithm in this paper can achieve significant privacy gains with a small loss of recommendation accuracy, and the trade-off between privacy and utility is relatively ideal.

(2) Member inference attack and defence assessment

With $\epsilon=0.5, 1.0, 2.0$ settings, the attack accuracy and AUC of member inference attacks are shown in Table 4.

Table 4. Member inference attack effectiveness under different privacy budgets

ϵ	MIA Accuracy	MIA AUC	Random guessing benchmark
0.5	0.53	0.52	0.50
1.0	0.56	0.54	0.50
2.0	0.61	0.59	0.50
No Privacy Protection	0.82	0.87	0.50

As shown in Table 4, without privacy protection, the attack accuracy reaches 0.82, far exceeding the 0.50 of random guessing, indicating a serious risk of member leakage in the model. Introducing differential privacy significantly reduces the attack accuracy. When $\epsilon=0.5$, the

attack accuracy is only 0.53, almost identical to random guessing; even with $\epsilon=2.0$, the attack accuracy is only 0.61. This demonstrates that the differential privacy mechanism proposed in this paper can effectively obfuscate the member information in the training set, significantly reducing the success rate of member inference attacks even under moderate privacy protection.

(3) Gradient inversion attack defence assessment

The DLG algorithm was used to reconstruct the input feature vector from the gradient, and the reconstructed MSE and PSNR were calculated. The results are shown in Table 5.

Table 5. Gradient inversion attack and defence

ϵ	Reconstruct MSE	Reconstruct PSNR (dB)
0.5	0.347	14.6
1.0	0.212	16.7
2.0	0.098	20.1
No Privacy Protection	0.003	35.2

As shown in Table 5, without privacy protection, the PSNR reaches as high as 35.2 dB, meaning that attackers can almost perfectly recover the user's original data. With differential privacy, even at $\epsilon=2.0$, the PSNR drops to 20.1 dB, and the reconstructed MSE is 0.098, indicating severe distortion of the reconstructed image or features. When $\epsilon=1.0$, the PSNR is only 16.7 dB, and the reconstructed content is almost unrecognizable. This demonstrates that the proposed method can effectively defend against gradient inversion attacks and protect the user's original data from being stolen.

(4) Robustness analysis under poisoning attack of the model

With a malicious client ratio of 20%, the global model ACC changes with the number of training rounds as shown in Table 6.

Table 6. Robustness analysis under poisoning attack

Training rounds	No attack ACC	Poison attack ACC (undefended)	Poison attack ACC (ours)
50	0.785	0.612	0.771
100	0.798	0.584	0.782
150	0.802	0.551	0.788
200	0.808	0.523	0.795

Table 6 shows that without specific defences, a poison attack causes the ACC to drop sharply from 0.808 to 0.523, a performance loss exceeding 35%. The proposed algorithm uses federated averaging, which significantly reduces the impact of a single malicious client because the gradients of each client are averaged. As shown in the table, the proposed algorithm consistently maintains an ACC above 0.77 under poison attacks, with a difference of less than 3 percentage points from the no-attack scenario. This indicates that, given a low proportion of malicious clients (20%) and random gradient poisoning as the attack method, the Federated Averaging mechanism can mitigate the impact of poisoning attacks to some extent. It should be noted that this study did not employ specialized robust aggregation algorithms such as Krum, Median, or Trimmed Mean; consequently, the defence effectiveness of this method may decline significantly when facing higher proportions of malicious clients (e.g., over 40%) or more complex backdoor poisoning attacks. This represents a limitation of the current approach and a key direction for future work.

3.3. The effect of recommendation algorithm

3.3.1 Comparison with other algorithms

To compare the performance of the final recommendation algorithm model in this study, we compared the average evaluation results of this study with other widely used recommendation algorithm models and their variants under distributed privacy protection settings in the same conditions on two datasets. The comparison results are shown in Table 7.

Table 7. Comparison of results of different recommendation algorithms

Algorithms	RMSE	MAE
NRPA	1.046	0.908
ConvMF	1.089	0.947
ANR	1.043	0.898
FedNRPA	1.032	0.892
The algorithm model of this paper	0.960	0.858

From the results in Table 7, we can see that among all algorithms, the evaluation indicators RMSE and MAE of the algorithm in this paper are the lowest, proving its superiority. In the NRPA algorithm, only the attention mechanism is used, but the relationship between specific

data and the attribute relationship of users are not analysed, resulting in insufficient accuracy of the model's prediction results. While ConvMF uses the user's evaluation information in the algorithm, it does not consider the situation of mis-evaluation and random evaluation in the algorithm design process, resulting in insufficient accuracy of the model. In the process of algorithm averaging, ANR comprehensively considers the relationship between goods and users, so it has achieved the second-best performance among all models. FedNRPA is a federated learning implementation of NRPA. While it protects privacy through distributed training, its recommendation accuracy is still limited by the structure of the underlying model. This research's algorithm, by comprehensively considering the influence of user-aided information, the timeliness of evaluation metrics, and designing a model architecture deeply integrated with distributed privacy protection mechanisms, achieves good performance. The final RMSE and MAE of this model are 0.960 and 0.858, respectively.

3.3.2 Ablation experiment

Ablation experiments were conducted on the deep learning recommendation model that integrates contextual information and privacy protection. The results are compared in Table 8, considering the cases of only integrating auxiliary information, only using feature cross-interaction, and only using the LSTM algorithm without applying privacy protection in a distributed environment.

Table 8. Comparison of ablation experiment results

Algorithms	RMSE	MAE
Only auxiliary information is integrated	0.986	0.881
Only cross features are used	0.975	0.883
Only LSTM algorithm is used	0.987	0.887
The algorithm in this paper	0.960	0.858

From the results in Table 8, we can see that auxiliary information has a good effect on personalized recommendations on online platforms, and can effectively improve the accuracy of algorithm recommendation results. Compared with other data in Table 3 that only use evaluation results, it has better performance. The use of LSTM network can effectively analyse the timeliness of user evaluation information, save long-term evaluation information, discard useless information, and achieve a good average effect. The algorithm that only uses AFM cross-features can use the attention mechanism to distribute the weight of the algorithm and improve the interpretability of the algorithm recommendation results. However, none of the individual models are as good as the summative algorithm model in this study, and these individual models

do not consider privacy protection in a distributed environment, which poses risks in actual deployment. This demonstrates the effectiveness and superiority of the model that integrates contextual information, distributed computing and privacy protection mechanisms.

3.3.3 The impact of parameters on the algorithm

To comprehensively consider the impact of different algorithm parameters on model prediction results, this study mainly considers the influence of the ID dimension and auxiliary information weights on the prediction results among the aforementioned evaluation metrics, as well as the impact of the client selection ratio in federated learning on training efficiency and model convergence speed. The influence of the ID dimension on the model prediction results is shown in Figure 7.

From the results in Figure 7, we can see that for the evaluation indicators RMSE and MAE, their values first decrease and then increase with the increase of ID dimension, and the two trends are the same. The main reason is that the ID dimension is too low, the attention model cannot better obtain the characteristic information of users and products, and the learning efficiency is insufficient. When the dimension is too high, overfitting will also occur. Therefore, the algorithm in this paper selects the ID dimension as 32 to obtain the best effect.

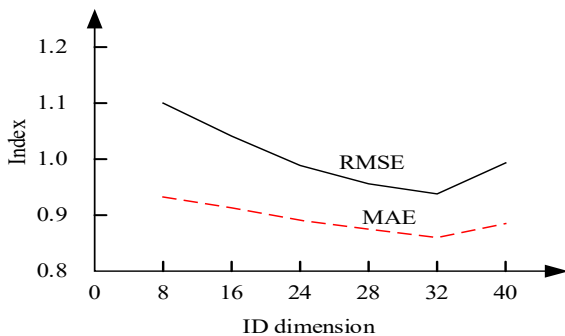


Figure 7. The impact of the ID dimension on the algorithm's prediction indicators

The influence of weight value on prediction results is shown in Figure 8.

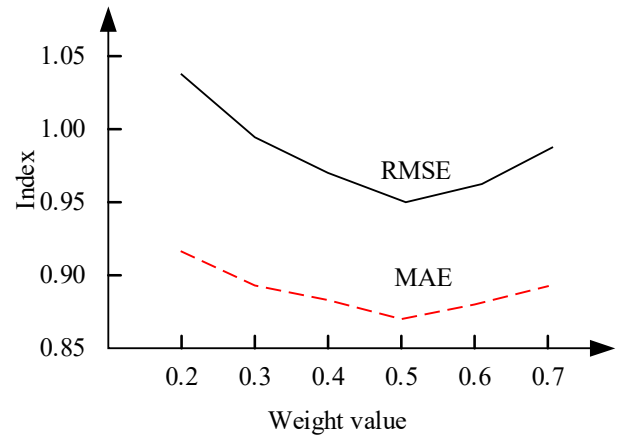


Figure 8. The impact of weight values on predictive indicators

From the results in Figure 8, we can see that for the evaluation indicators RMSE and MAE, their values first decrease and then increase with the increase of weight values, and the trends of the two are also similar. The main reason is that the weight value is too low, the model cannot better obtain the characteristic information of users and products, and the learning efficiency is insufficient. When the weight value is too high, overfitting will also occur. Finally, the algorithm in this paper selects a weight value of 0.5 to obtain the best effect.

Furthermore, experiments have shown that in federated learning settings, selecting 30%-50% of clients to participate in training in each round can achieve a good balance between communication overhead and model convergence speed. Too low a participation rate will slow down convergence, while too high a participation rate will increase the communication burden in each round.

4. Conclusion

To enhance personalized recommendation performance for online platforms in distributed network environments while effectively protecting user privacy, this study investigates a knowledge graph-based recommendation algorithm model that integrates distributed computing with privacy-preserving technologies. The model incorporates an attention mechanism, utilizing a hyperbolic attention model to fully explore item interrelationships, while employing localized processing and differential privacy techniques to prevent the leakage of relational information. The recommendation model employs a deep learning framework that integrates contextual information to extract user attention vectors. It organically combines semantic features with the attention mechanism and utilizes cross-feature modelling for weighting to generate final item recommendations. Implemented within a federated learning framework, the algorithm ensures that user data remains on-device during training, thereby achieving a

balance between distributed collaboration and privacy protection.

The proposed model is compared against other existing models. Experimental results demonstrate that, under identical distributed and privacy constraints, the proposed model achieves superior performance metrics; with an RMSE of 0.960 and an MAE of 0.858, it exhibits significant advantages over other models. System security assessments indicate that when the privacy budget $\epsilon \geq 2.0$, the model effectively reduces the success rate of membership inference attacks (with attack accuracy not exceeding 0.61) and lowers the reconstruction quality of gradient inversion attacks (with reconstructed PSNR not exceeding 20.6 dB). Furthermore, the model maintains robust resistance against model poisoning attacks, with performance degradation of less than 3 percentage points, thereby validating its security and practicality in real-world distributed recommendation scenarios.

Future work will focus on the following directions: first, studying the adaptability of algorithms in more complex heterogeneous network environments; second, exploring dynamic adjustment mechanisms for privacy budgets to adaptively optimize the strength of privacy protection during training; and third, further reducing communication overhead in federated learning and improving training efficiency in large-scale node scenarios. Fourth, robust aggregation algorithms—such as Krum, Median, and Trimmed Mean—are introduced to enhance the model's defence against poisoning attacks when facing a higher proportion of malicious clients. Fifth, practical deployment schemes for stronger privacy-preserving technologies, such as Secure Multi-Party Computation (SMPC) and Homomorphic Encryption (HE), are explored, alongside an evaluation of their computational and communication costs.

Acknowledgements

This work is supported by the Henan Province Science and Technology Project (Grant No.232102210117), the Research and Practice Project on Higher Education Teaching Reform of Henan Province (Grant No. 2024SJGLX0478).

References

- [1] Zhang Y, Cai J, Li C, Li T, Wang H. KMPR-AEP: Knowledge-enhanced multi-task parallelized recommendation algorithm incorporating attention-embedded propagation. *International Journal of Computational Intelligence Systems*. 2024;17(1):213. Doi: <https://doi.org/10.1007/s44196-024-00625-2>
- [2] Shen B, Zhang Y. Research on personalised knowledge graph recommendation algorithm based on self-supervised learning. *Discover Applied Sciences*. 2024 Aug 7;6(8):434. Doi: <https://doi.org/10.1007/s42452-024-06135-2>
- [3] Sankar JG, David A, Valan P. Examining user understanding and perceptions of e-commerce data privacy, security, and protection. In *Confronting security and privacy challenges in digital marketing 2023* (pp. 159-185). IGI Global Scientific Publishing. Doi: <https://doi.org/10.4018/978-1-6684-8958-1.ch008>
- [4] Lu P, Liu P. Product recommendation system based on deep learning. *International Journal of Advanced Network, Monitoring and Controls*. 2023;8(1):1-9. Doi: <https://doi.org/10.2478/ijanmc-2023-0041>
- [5] Tang Hong, Fan Sen, Tang Fan. Recommendation Algorithm Integrating Collaborative Knowledge Graph and Optimizing Graph Attention Network, *Computer Engineering and Applications*, 2022; 58(19): 98-106(in chinese). Doi: <https://doi.org/10.3778/j.issn.1002-8331.2201-0183>
- [6] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *The World Wide Web Conference (WWW '19)*. Association for Computing Machinery, New York, NY, USA, 2019; 2000–2010. Doi: <https://doi.org/10.1145/3308558.3313411>
- [7] Peng P, Moghimi MM. Hash-based lightweight recommendation in edge computing for privacy-preserving sport communities. *Journal of Cloud Computing*. 2025 6;15(1):7. Doi: <https://doi.org/10.1186/s13677-025-00809-x>
- [8] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H. Yang, Farhad Farokhi. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 2020, 15: 3454-3469. Doi: <https://doi.org/10.1109/TIFS.2020.2988575>
- [9] Li Y, Hou L, Li J. Towards knowledge-aware and deep reinforced cross-domain recommendation over collaborative knowledge graph. *IEEE Transactions on Knowledge and Data Engineering*. 2024 Apr 19;36(11):7171-87. Doi: [10.1109/TKDE.2024.3391268](https://doi.org/10.1109/TKDE.2024.3391268)
- [10] Huang W, Wu J, Song W, Wang Z. Cross attention fusion for knowledge graph optimized recommendation. *Applied Intelligence*. 2022;52(9):10297-306. Doi: <https://doi.org/10.1007/s10489-021-02930-1>
- [11] Zhao N, Long Z, Wang J, Zhao ZD. AGRE: A knowledge graph recommendation algorithm based on multiple paths embeddings RNN encoder. *Knowledge-Based Systems*. 2023; 259:110078. Doi: <https://doi.org/10.1016/j.knosys.2022.110078>
- [12] Chen L, Bi X, Fan G, Sun H. A multitask recommendation algorithm based on DeepFM and Graph Convolutional Network. *Concurrency and computation: practice and experience*. 2023;35(2):e7498. Doi: <https://doi.org/10.1002/cpe.7498>
- [13] Ghogh B, Ghodsi A. Attention mechanism and transformers. In *Elements of deep learning 2026 Jun 2* (pp. 231-257). Cham: Springer Nature Switzerland. Doi: https://doi.org/10.1007/978-3-032-10738-1_9
- [14] Wu C, Liu S, Zeng Z, Chen M, Alhudhaif A, Tang X, Alenezi F, Alnaim N, Peng X. Knowledge graph-based multi-context-aware recommendation algorithm. *Information Sciences*. 2022; 595:179-94. Doi: <https://doi.org/10.1016/j.ins.2022.02.054>
- [15] Yao C, Zhao C. Knowledge graph and GNN-based news recommendation algorithm with edge computing support. *International Journal of Distributed Systems and Technologies (IJ DST)*. 2022;13(2):1-1. Doi: <https://doi.org/10.4018/IJ DST.291080>
- [16] Ye PG, Wang W, Mi B, Chen K. EdgeStreaming: Secure computation intelligence in distributed edge networks for streaming analytics. *ACM Transactions on Multimedia Computing, Communications and Applications*. 2025 Aug 13;21(8):1-5. Doi: <https://doi.org/10.1145/3686161>

- [17] Hammoudeh Z, Lowd D. Training data influence analysis and estimation: A survey. *Machine Learning*. 2024; 113(5):2351-403. Doi: <https://doi.org/10.1007/s10994-023-06495-7>
- [18] Chicaiza J, Valdiviezo-Diaz P. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information*. 2021, 12(6):232. Doi: <https://doi.org/10.3390/info12060232>
- [19] Shankar A, Perumal P, Subramanian M, Ramu N, Natesan D, Kulkarni VR, Stephan T. An intelligent recommendation system in e-commerce using ensemble learning. *Multimedia Tools and Applications*. 2024 May;83(16):48521-37. Doi: <https://doi.org/10.1007/s11042-023-17415-1>
- [20] Wu L, Zheng Z, Qiu Z, Wang H, Gu H, Shen T, Qin C, Zhu C, Zhu H, Liu Q, Xiong H. A survey on large language models for recommendation. *World Wide Web*. 2024;27(5):60. Doi: <https://doi.org/10.1007/s11280-024-01291-2>
- [21] Li T, Kou G, Peng Y. Learning user's dynamic and personalized interests from e-commerce reviews of verified purchases: a time series based framework. *Journal of the Operational Research Society*. 2025;76(12):2464-73. Doi: <https://doi.org/10.1080/01605682.2025.2476076>