ITC 3/55
Information Technology and Control
Vol. 55 / No. 3/ 2026
pp. 1316-1335
DOI 10.5755/j01.itc.55.3.44443

Improved Fast-moving Consumer Goods Recommendation Algorithm Integrating Lightweight Graph Convolution

Received 2026/03/03 | Accepted after revision 2026/06/26

HOW TO CITE: Zhang, L., Li, C., Huang, F., Huang, X., Feng, Y., Wang, P., Bai, Y. (2026) Improved Fast-moving Consumer Goods Recommendation Algorithm Integrating Lightweight Graph Convolution. *Information Technology and Control*, 55(3), 1316-1335. https://doi.org/10.5755/j01.itc.55.3.44443

Improved Fast-moving Consumer Goods Recommendation Algorithm Integrating Lightweight Graph Convolution

Lijie Zhang, Cuizhi Li

College of Modern Logistics, Shanxi Vocational University of Engineering Science and Technology, Jinzhong, 30000, China; e-mails: zhanglijie@sxgkd.edu.cn, licuizhi@sxgkd.edu.cn

Fei Huang*

Seoul Business School, Seoul School of Integrated Sciences and Technologies, Seoul, 085086, Korea; e-mail: FeiHuangfh@outlook.com

Xiaohui Huang, Yulan Feng, Peng Wang, Yufeng Bai

College of Modern Logistics, Shanxi Vocational University of Engineering Science and Technology, Jinzhong, 30000, China; e-mails: huangxiaohui@sxgkd.edu.cn, fengyulan@sxgkd.edu.cn, wangpeng@sxgkd.edu.cn, baiyufeng@sxgkd.edu.cn

Corresponding author: FeiHuangfh@outlook.com

This paper proposes an optimization algorithm that integrates lightweight graph convolutional network (GCN) and Transformer for recommending fast-moving consumer goods products in e-commerce. The method captures user-item interaction information through Lightweight GCN and optimizes the algorithm using the factorization machine recommendation mechanism. Experimental results show that, under the full settings, the model's logarithmic loss is reduced to 0.358, while the top 50 hit rate and the cumulative gain of the top 50 normalized loss are increased to 0.681 and 0.241, respectively. In experiments comparing recommendations for different fast-moving consumer goods categories and resource

consumption, the model demonstrates better practicality in terms of repeat purchase hit rate, average ranking, and computational efficiency. Experimental outcomes indicate that the introduced approach can improve the performance of e-commerce fast-moving consumer goods recommendation systems and has good application prospects.

KEYWORDS: E-commerce recommendation system, Fast-moving consumer goods recommendation, Graph convolutional network, Transformer, Factorization machine

1. Introduction

In e-commerce platforms, fast-moving consumer goods (FMCG) are one of the most representative scenarios in the application of recommendation systems [5]. How to accurately characterize user preferences under large-scale data conditions is the core problem faced by current recommendation systems [27]. Traditional methods often have difficulty in effectively modeling the high-order relationship between users and products, and also have difficulty in making full use of time-series information [28]. Some scholars have explored from the perspective of hybrid recommendation. Singh et al. [23] combined collaborative filtering and content-based filtering and used recurrent neural networks to identify attack behavior in hybrid recommendation. Experiments verified the effective detection capability of this method against various attacks [23]. Li et al. [16] proposed an IoT intelligent recommendation model for consumer electronics products. The model first utilizes graph embedding techniques and combines product auxiliary information to alleviate data sparsity and cold start problems. Secondly, by modeling the user's historical purchase dependency relationship through multi head self attention and position encoding, and introducing an attention update gate loop unit to filter the interest evolution path, the user's interests can be more accurately represented. The experimental results show that the model has good performance in consumer electronics product recommendation [16]. Ullah et al. [26] proposed an entropy based multi criteria decision-making method for optimizing personalized services in e-commerce. This method improves the interface, recommendation system, and information transparency to help the platform accurately recommend cost-effective products based on user preferences. The study simultaneously analyzed the impact of electronic devices and energy-saving sensor networks on inventory management, user preferences, and shopping sustainability, and revealed the complex relationship between data privacy, energy efficiency, and platform governance. Experimental results have shown that this method outperforms existing advanced schemes in multiple indicators [26]. It is highly consistent with the lightweight deployment goal pursued in this article, and can provide richer theoretical support for the contribution of balancing lightweight design and robustness in this article.

Over recent years, deep learning (DL) techniques have seen extensive application in recommender system domains. Among them, graph convolutional networks (GCNs) can mine the collaborative relationship between users and items by modeling the recommendation problem as a user-item interaction graph [3]. Fu et al. [7] proposed a self-supervised recommendation model based on GCN. This method uses a multi-task learning framework to integrate self-supervised tasks and main recommendation tasks, thereby optimizing recommendation performance. Experimental results show that the model outperforms mainstream baseline methods in both recall and normalized discounted cumulative gain (NDCG) metrics [7]. Tong et al. [25] proposed a knowledge recommendation model that integrates context and user behavior. This model integrated knowledge base structure, user preferences and semantic information of knowledge graph, and usedGCNs to quantify user preferences layer by layer. Experiments verified its effectiveness in knowledge recommendation [25].

In the field of sequence recommendation, the Transformer model can effectively capture the long-term dependence and dynamic interest evolution of user behavior, and is therefore widely

used [18]. Zheng et al. [29] proposed a bidirectional multi-behavior recommendation model for online recruitment. The model encoded the user's historical behavior sequence through Transformer to obtain the expected representation, and used heterogeneous graph Transformer to construct a recruitment market graph to extract the competitive representation. Experiments showed that the model was better than the current mainstream baseline methods [29]. Kim et al. [13] used a Transformer-based model to determine the optimal architecture for aspect-level sentiment analysis. The model effectively captured the aspect-level similarity between items. The results showed that the integration of DL sentiment analysis and graph learning technology helped to enhance the performance of recommendation systems [13].

Existing research on recommendation algorithms for e-commerce FMCG still suffers from high model complexity and insufficient engineering deployability. To address this, this study proposes a lightweight GCN approach combined with Transformer-based sequence modeling, and innovatively integrates factorization machines (FMs) and deep neural network (DNN) models to construct a recommendation algorithm framework for e-commerce FMCG scenarios. This method fully explores high-order user-item relationships while effectively characterizing the dynamic evolution of user interests. It improves model stability and practical application value while maintaining recommendation performance.

2. Methods

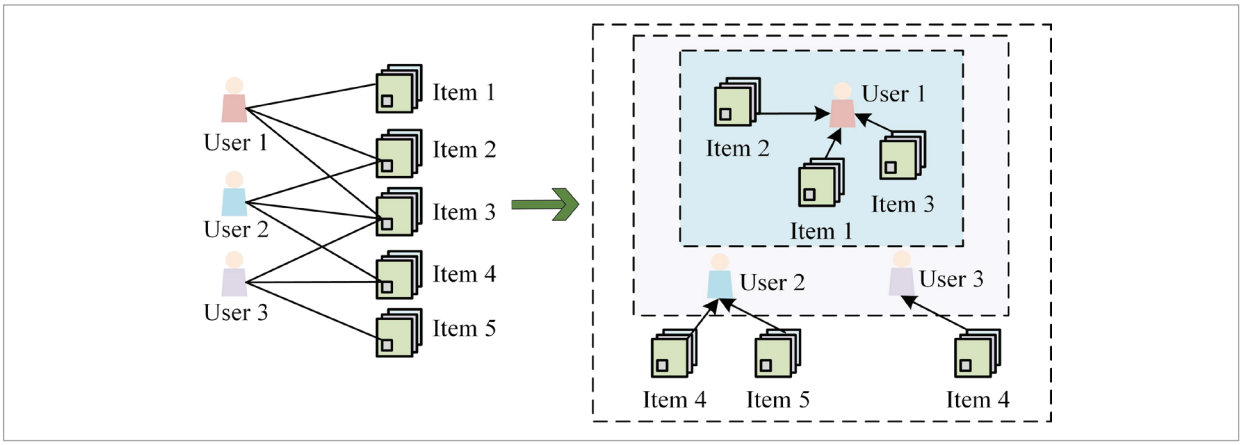
2.1. A Two-stage Recommendation Framework Based on GCN and Sequence Modeling

2.1.1. User-item Representation Learning Based on Lightweight GCN

In the recall phase, the study first constructs a bipartite graph structure based on user-item historical interaction data, and uses lightweight GCN to learn the representation of user and item nodes. Compared with traditional graph convolutional models, lightweight GCN introduces a simple neighborhood aggregation mechanism by removing unnecessary feature transformations and complex parameters. While reducing model complexity and computational overhead, it can still effectively capture the high-order relationships between nodes [15]. The user-item interaction relationship is shown in Figure 1 [12].

Figure 1, the user-item interaction relationship is first represented as a bipartite graph structure, where the edges between user nodes and item nodes represent historical interaction behaviors. Based on this, with the target user node as the center, the model aggregates its first-order, second-order and third-order neighborhood information layer by layer, and includes items and other users that have direct or indirect interaction relationships with the user into the receptive field. Through multi-layer neighborhood expansion and information propagation, the model can explore the potential association relationship between users and items through common interaction paths, thereby

Figure 1
User project interaction relationship.



alleviating the issue of inadequate information stemming from the sparse nature of the original interaction data. The study adopts a lightweight GCN model to learn the embedding representation of users and items. GCN can efficiently learn the relationship and feature representation between nodes based on graph structure [8]. The fundamental concept revolves around iteratively refining the node embedding by aggregating neighborhood information, and the specific process is shown in Equation (1) [17].

$$c_u^{k+1} = \mathrm{AGG}\left(c_u^k, \left\{c_i^k : i \in N_u\right\}\right). \quad (1)$$

In Equation (1), c_u^k represents the user u's embedding at layer k, c_u^{k+1} represents the user u's updated embedding at layer k+1, c_i^k is the embedding of the item i adjacent to the user u at layer k, N_u is the set of the user u's neighboring nodes, and AGG(·) is the aggregation function. The lightweight GCN layer consists of message passing and feature aggregation. It performs weighted aggregation of neighbor features through a concise propagation rule, thereby efficiently updating the node representation. After k layers of lightweight GCN computation, the final user and item representations are obtained by weighted combination of the embeddings of each layer, as shown in Equation (2).

$$\begin{cases} c_u = \sum_{k=0}^{k} \omega_k c_u^k \\ c_i = \sum_{k=0}^{k} \omega_k c_i^k \end{cases}. \quad (2)$$

In Equation (2), c_u and c_i represent the final embeddings of user u and item i, respectively, and ω_k represents the weight coefficient of layer k embedding. By concatenating c_u and c_i, the user-item joint representation is obtained, the calculation of which is presented in Equation (3) [10].

$$c_{ui} = \mathrm{ReLU}\left((c_u \oplus c_i) \cdot W + b\right). \quad (3)$$

In Equation (3), c_{ui} represents the user-item joint vector, W is the trainable weight matrix (WM), and b is the bias term. To facilitate the implementation and discussion of existing models, the matrix of the lightweight GCN is set as shown in Equation (4).

$$C^{k+1} = (D^{-\frac{1}{2}} O D^{-\frac{1}{2}}) C^k. \quad (4)$$

In Equation (4), C^k represents the node embedding matrix of the k-th layer, C^{k+1} is the node embedding matrix of the k+1th layer, D is the degree matrix, and O is the adjacency matrix. The final embedding matrix is shown in Equation (5).

$$C = \omega_0 C^0 + \omega_1 C^1 + \omega_2 C^2 + \ldots + \omega_k C^k. \quad (5)$$

In Equation (5), C is the node embedding matrix used for prediction.

2.1.2. Recall Model Integrating Lightweight GCN and Transformer

The study uses a multi-head attention Transformer encoder to encode c_{ui}, to learn user purchasing behavior more effectively. Transformer is a DL architecture grounded in self-attention mechanism. Its encoder is formed by stacking multiple identical sub-modules. Each module contains multi-head attention, normalization and feedforward neural network (FNN) layers [6]. The attention weight of each attention head in the multi-head attention layer is calculated as shown in Equation (6) [19].

$$\omega_{ij}^i = \mathrm{soft\,max}\left(\frac{Q_i K_j^T}{\sqrt{d_k}}\right). \quad (6)$$

In Equation (6), ω_{ij}^i represents the contribution weight of position j in the i-th attention head to the current position. Q_i is the query vector, K_j^T is the transpose of the key vector, and d_K represents the dimension of the key vector. After obtaining the attention weights, the corresponding value vectors are weighted and summed to obtain the output of the i-th attention head, as presented in Equation (7).

$$H_i = \sum_{j=1}^{m} \omega_{ij}^i V_j. \quad (7)$$

In Equation (7), H_i represents the output of the i-th attention head, V_j is the value vector at the j-th position, and m is the sequence length. The outputs of each attention head are concatenated and trans-

formed using a linear projection matrix, as shown in Equation (8).

$$\text{Multi}H(e_{ui}^{w}) = \text{Concat}(H_1, H_2, \ldots, H_i)W_o . \quad (8)$$

In Equation (8), e_{ui}^{w} represents the user u's behavior sequence input vector, W_o is a trainable linear mapping matrix. To further train the model, a normalization layer is used for calculation, as shown in Equation (9) [2].

$$Hide = \frac{\text{Multi}H(e_{ui}^{w})) - \mu}{\sigma} \odot \alpha + \beta . \quad (9)$$

In Equation (9), *Hide* represents the hidden vector after layer normalization, and μ and σ are the mean and standard deviation of the input features in the current layer, respectively, and α and β are both trainable parameters. The expression of the FNN layer is shown in Equation (10) [21].

$$F(Hide) = \text{ReLU}(HideW_1 + b_1)W_2 + b_2 . \quad (10)$$

In Equation (10), $F(Hide)$ represents the output of the FNN layer, W_1 and b_1 are the WM and bias of the first layer of the FNN, respectively, and W_2 and b_2 are the WM and bias of the second layer. Finally, the two-layer FNN is used to predict the user's rating as shown in Equation (11).

$$P_a^{\mu} = \text{Sigmoid}(\text{ReLU}(Hide \cdot W_{P1} + b_{P1}) \cdot W_{P2} + b_{P2}) . \quad (11)$$

In Equation (11), P_a^{u} represents the probability that item a will be selected in the user u's next action, W_{P1} and W_{P2} are the weight matrices of the rating prediction layer, and b_{P1} and b_{P2} are the bias terms. Through the joint learning of temporal modeling of user behavior sequences and user-item graph structure information, a recommendation model integrating lightweight GCN and Transformer (LGTRM)is constructed. Among them, the user item embeddings learned by lightweight graph convolution and the behavior sequence features are jointly input into the Transformer encoder, which is not a simple feature stacking, but a complementary design based on the semantic level of the two types of information. Graph convolution embedding mainly encodes the high-order collaborative relationship between users and items in the interaction graph, reflecting the static community structure of user preferences; The behavioral sequence features record the temporal sequence of user interactions, reflecting the dynamic evolution process of interests. Both inputs are simultaneously fed into the Transformer, allowing the self attention mechanism to jointly model two types of information in a unified space: "who the user is similar to" and "what the user may like next", without redundancy or conflict. This is because the former provides global neighbor constraints in the interaction graph, while the latter provides local temporal dependencies.

Furthermore, the relative importance of the two information sources depends on the specific patterns of user historical behavior. When the user interaction sequence is long and the behavior pattern is stable, sequence information dominates, and Transformer can effectively capture long-distance dependencies and predict the next behavior; When user interaction data is sparse or behavior patterns are irregular, the collaborative filtering information provided by graph convolution embedding becomes the main basis, compensating for the lack of individual behavior sequences through similar user interaction paths. In addition, in the cold start scenario, the behavior sequence of new users is extremely short. At this time, graph convolution relies on the neighborhood information associated with the user's demographic characteristics or contextual attributes, and its contribution weight is significantly higher than the sequence information; For active users, their rich interaction history makes the ability to predict sequence information stronger. At present, this study has not conducted rigorous theoretical condition analysis or hypothesis testing on this dynamic weight relationship, which is a limitation of the current work. Subsequent research can introduce gating mechanisms or adaptive fusion strategies to dynamically adjust the fusion weights of two types of information based on indicators such as user behavior entropy, sequence length, and neighborhood density, and design corresponding statistical testing methods to verify the significant differences in information source dominance under different conditions. The structural framework of the LGTRM model is shown in Figure 2.

Figure 2
LGTRM structure.

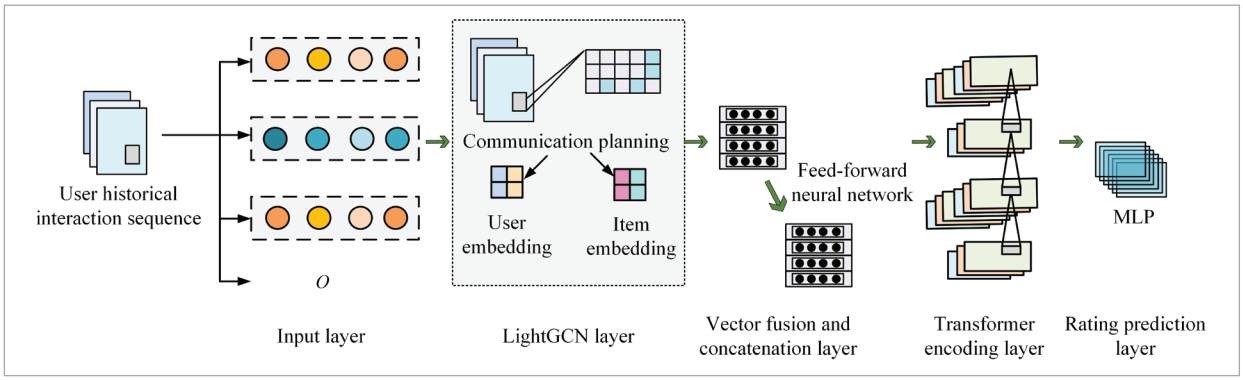
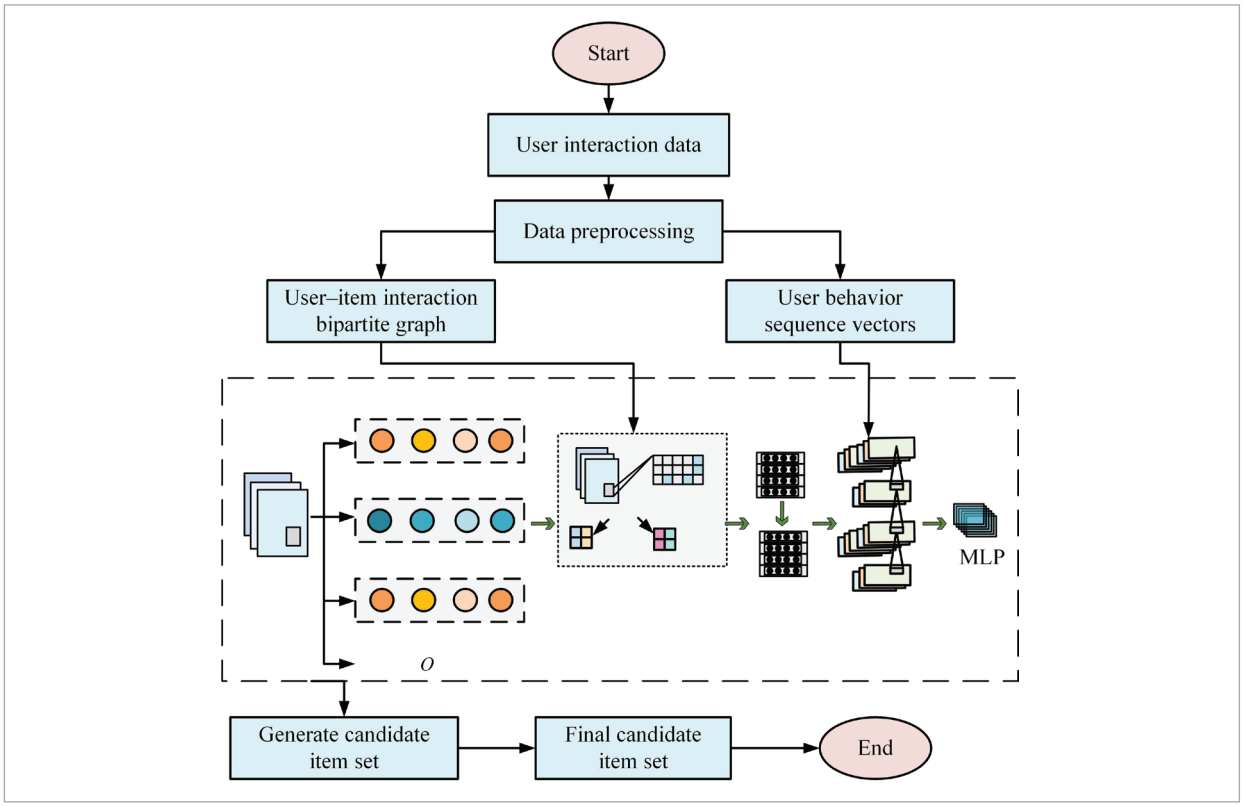


Figure 3
Recall process based on LGTRM.



In Figure 2, LGTRM takes user-item interaction data as input. First, it constructs initial feature representations of users and items in the input layer, and then uses a lightweight GCN layer to propagate high-order neighborhood information through the user-item graph, obtaining embedding vectors for users and items. Subsequently, the representations learned from the graph structure are fused and concatenated with behavioral sequence features, and then fed into the Transformer encoding layer. Multi-head self-attention and an FNN are used to model the temporal dependence and dynamic preferences of user behavior. Finally, in the rating prediction layer, a multi-layer perceptron (MLP) is used to map the fused fea-

tures, outputting the predicted probability of the user for candidate items, thus achieving recommendation decision-making. Based on this, the study further presents a recommendation recall process based on LGTRM from a system application perspective, as shown in Figure 3.

Figure 3 illustrates the recall phase process of LGTRM in a recommender system. Starting with user interaction data, after data preprocessing, user behavior sequence features and user-item interaction bipartite graphs are constructed. Inside the model, the Transformer encoding module models the temporal dependence and dynamic interests of user behavior, while the lightweight LightGCN module aggregates high-order neighborhood information to learn the potential associations between users and items. Finally, candidate items are scored and filtered, and the Top-N items are selected to form a candidate item set, completing the recommendation process in the recall phase.

2.2. Multi-order Feature Interaction Modeling Based on FM and DNN

In the ranking stage of the recommendation system, the candidate products have often been preliminarily screened in the recall stage. At this time, the model needs to further characterize the complex interaction relationship between user features, product attributes and contextual behavioral features in order to achieve refined ranking of candidate products [20]. In view of the problem that users have high purchase frequency, diverse behavior patterns and sparse feature dimensions in the e-commerce FMCG scenario, this study constructs a multi-level feature interaction modeling method based on FM and DNN in the ranking stage to learn the feature relationship at different levels collaboratively. In order to model low-level feature interaction, the study introduces the FM model. FM adds a second-order feature combination term to the general linear model, and its form is shown in Equation (12) [4].

$$Y(x) = v_0 + \sum_{i=1}^{M} \lambda_i x_i + \sum_{i=1}^{M} \sum_{j=i+1}^{M} \lambda_{ij} x_i x_j . \tag{12}$$

In Equation (12), $Y(x)$ represents the model's predicted output for sample x, v_0 is the global bias value, and x_i and x_j are the values of the i-th and the j-th input features. λ_i is the first-order weight parameter corresponding to the i-th feature, λ_{ij} is the weight parameter of the second-order interaction term between the features x_i and x_j, and M is the number of features of the sample. To reduce the scale of the second-order interaction parameters and enhance the generalization ability of the model, FM uses latent vector decomposition to model the second-order feature interaction, and its prediction function is shown in Equation (13) [30].

$$Y_{FM} = v_0 + \sum_{i=1}^{M} \lambda_i x_i + \sum_{i=1}^{M} \sum_{j=i+1}^{M} \left\langle z_i, z_j \right\rangle x_i x_j . \tag{13}$$

In Equation (13), Y_{FM} is the predicted output of the FM, and z_i and z_j are the latent vectors corresponding to the i-th and j-th features. To obtain richer feature interaction information, this study further explores higher-order feature relationships grounded in the FM model. For the transaction sequence data of FMCG on mobile e-commerce platforms, user behavior sequences are first constructed in chronological order, and the embedded representation of the sequence is obtained by using a pre-trained Word2vec model. Then, the representation is input into the DNN for higher-order feature interaction learning [14]. The output of the first layer of the DNN is shown in Equation (14) [22].

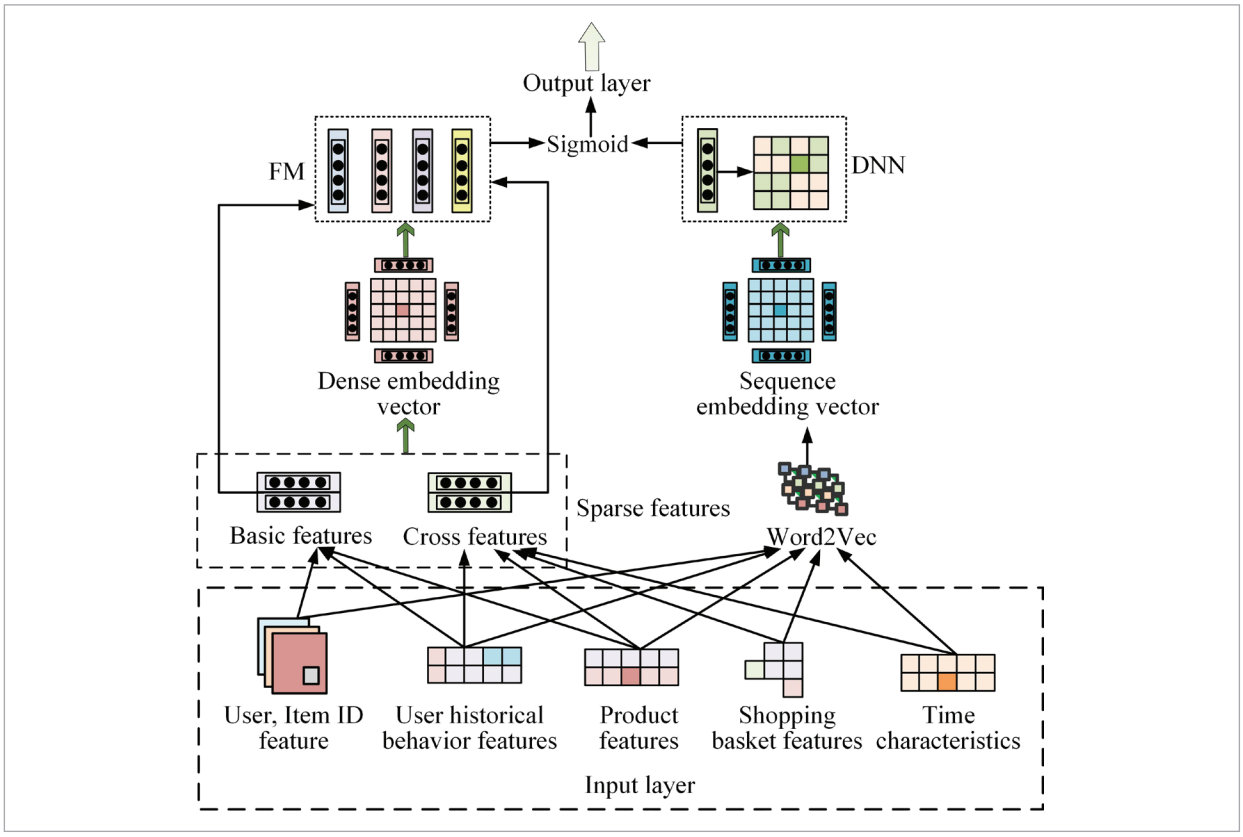
$$Y_{DNN} = \mathrm{ReLU}(\omega^l s^{l-1} + b^l). \tag{14}$$

In Equation (14), Y_{DNN} represents the output of the DNN at layer l, and ω^l and b^l are the weights and biases of layer l, respectively, and s^{l-1} is the input vector of layer l–1. In the output layer, the model predicts the probability of a user clicking or making a purchase, and its calculation form is presented in Equation (15).

$$\hat{Y} = \mathrm{Sigmoid}(Y_{FM} + Y_{DNN}). \tag{15}$$

In Equation (15), $\hat{Y}$ is the final predicted probability of the user clicking or making a purchase. The loss function used during model training is defined as shown in Equation (16).

Figure 4
DNN-FM structure.



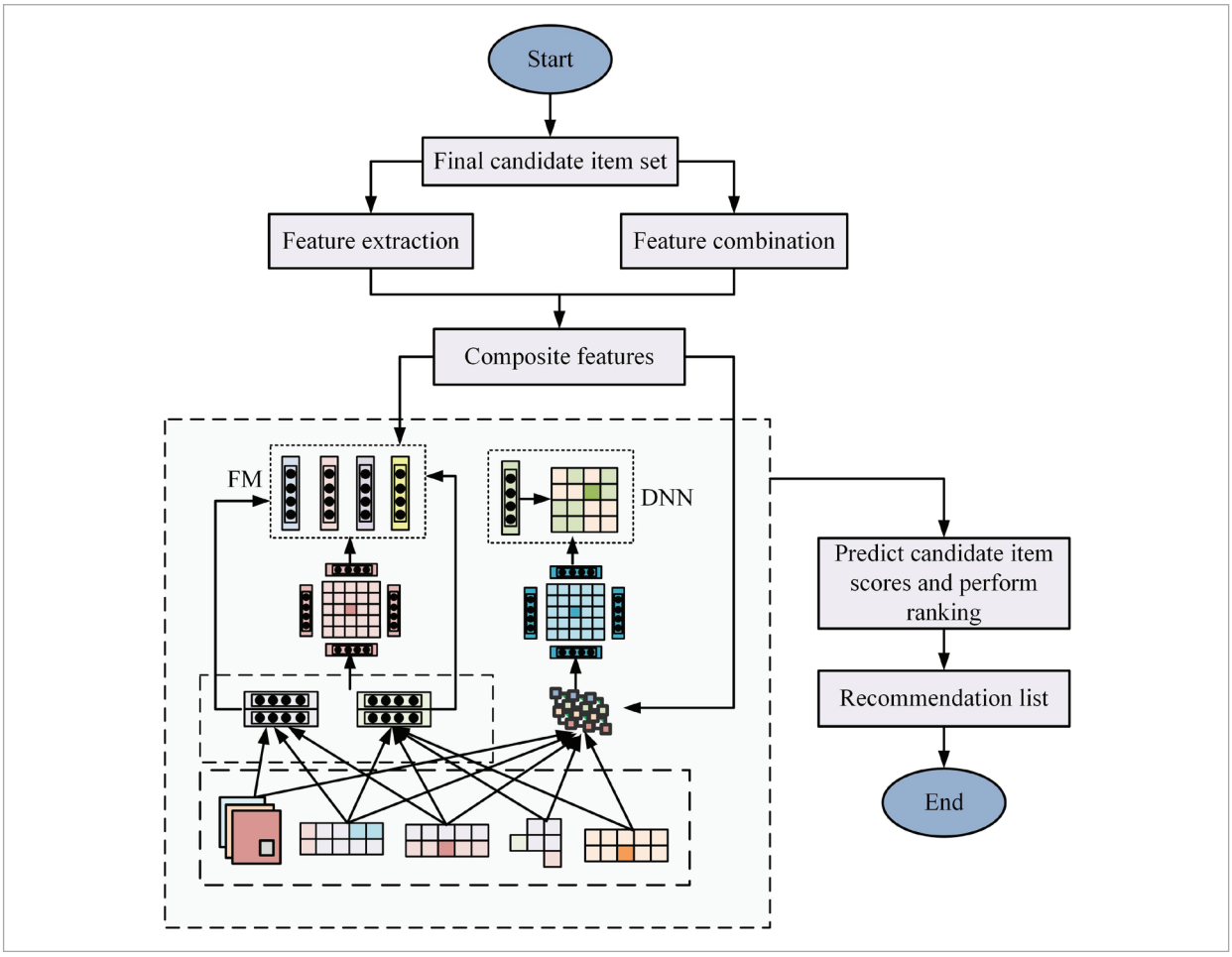
$$L = -\frac{1}{M}\sum_{i=1}^{M}\left[Y_i \log(\hat{Y}_i) + (1-Y_i)\log(1-\hat{Y}_i)\right]. \quad (16)$$

In Equation (16), L represents the overall loss value of the model during training, Y_i represents the true label of the i-th sample, and $\hat{Y}_i$ represents the predicted probability given by the model for the i-th sample. In the ranking stage, in order to learn and integrate the nonlinear relationship between the multi-dimensional features of user purchasing behavior, a DNN-FM model is constructed, the structure of which is shown in Figure 4.

Figure 4, the DNN-FM model takes user, item, and multi-dimensional behavioral features as inputs and maps sparse features into low-dimensional vector representations through feature embedding. On the one hand, it explicitly models the low-order interaction relationships between features in the FM module, and on the other hand, it performs nonlinear combination of sequence embedding vectors and context features in the DNN to learn high-order feature interactions [9]. At the same time, Word2Vec is introduced to perform representation learning on user behavior sequences, and sequence information and static features are input into the deep network together [24]. Finally, the prediction outcomes are output through the Sigmoid activation function to realize the modeling of user behavior or click probability. The ranking process based on DNN-FM is shown in Figure 5.

In Figure 5, the ranking stage takes the candidate product set obtained from the recall as input. First, in the feature engineering stage, features are extracted and combined from the candidate products to form composite features for modeling. Then, the composite features are input into the Word2vec and DNN-FM modules respectively for representation learning

Figure 5
Ranking process based on DNN-FM.



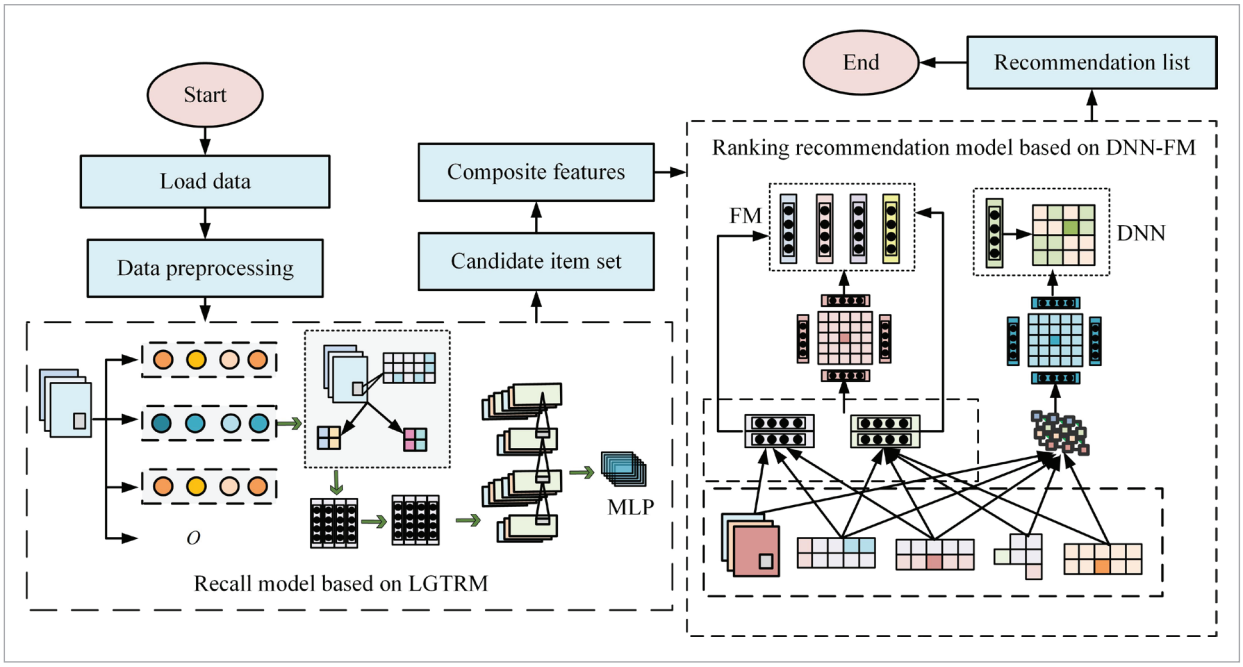
and high-order feature modeling. Based on this, the candidate products are rated and ranked, and finally a recommendation list is generated to complete the recommendation decision in the ranking stage.

2.3. An Optimization Algorithm for E-commerce FMCG Recommendation Based on LGTRM and DNN-FM

Building upon the LGTRM recall model and the DNN-FM ranking model, this study further proposes an e-commerce FMCG recommendation optimization algorithm that integrates LGTRM, DNN, and FM to construct a complete "recall-ranking" integrated recommendation process. This algorithm employs targeted modeling strategies at different stages to efficiently mine users' potential interests and refine the recommendation results, thus balancing recommendation performance and system efficiency. The specific recommendation process is shown in Figure 6.

In Figure 6, the LGTRM-DNN-FM algorithm starts by reading user interaction data. After data preprocessing, it constructs user behavior sequence vectors and a user-item interaction bipartite graph during the recall phase. Based on the LGTRM model, it models user interests to select Top-K items to form a candidate set. Subsequently, the candidate items are ranked, and after feature construction, the DNN-FM model is used to complete prediction and ranking, outputting a personalized recommendation list, thus completing the entire recommendation decision-making process. In addition, the study

Figure 6
Process of recommendation optimization algorithm integrating LGTRM-DNN-FM.



provides new theoretical constructions at the level of fusion architecture. Specifically, this article analyzes the complementarity between the high-order collaborative relationships captured by lightweight graph convolution and the dynamic temporal dependencies modeled by Transformer self attention from the perspective of graph sequential joint representation learning. The former performs isotropic neighborhood aggregation on the user item bipartite graph, which is essentially a low-pass filtering operation that can alleviate the sparsity problem of interactive data. The latter achieves selective attention to key behaviors in the sequence through adaptive weight allocation, and its attention distribution can be explained as an explicit characterization of user interest drift. The combination of the two forms a hierarchical feature extraction framework from static graph structure to dynamic behavior sequence, which is theoretically superior to using any module alone. At the same time, the DNN-FM joint modeling strategy proposed in the ranking stage provides a theoretical basis from the perspective of feature interaction order: FM explicitly models the combination of first-order and second-order features, compensating for the opacity of deep networks in low-level interaction modeling, while DNN automatically learns high-order nonlinear relationships, which together constitute a complete coverage of the feature interaction space.

3. Results

3.1. LGTRM-DNN-FM Algorithm Performance Test

To comprehensively evaluate the effectiveness and stability of the introduced LGTRM-DNN-FM recommendation optimization algorithm in e-commerce FMCG scenarios, this study experimentally validated the algorithm from multiple dimensions, including overall recommendation performance, ranking quality, and model generalization ability. The study selected the Amazon Grocery and Gourmet Food public dataset, which consisted of Amazon product reviews and user behavior records, covering users' purchasing and interaction behaviors in FMCG fields such as food and daily necessities, exhibiting high-frequency consumption and repeat purchase characteristics. This dataset is highly

Table 1
Hardware and software environment.

Category	Configuration item	Specific configuration
Operating system	OS	Ubuntu 20.04.6 LTS (64-bit)
CPU	Processor	Intel Core i9-13900K
Memory	RAM	128 GB DDR5
GPU	Graphics card	NVIDIA RTX 4090 24GB
Storage	Disk	2 TB NVMe SSD
CUDA	CUDA Toolkit	CUDA 11.8
Behavioral sequence	L	50
Batch size	Batch_size	1024
Training rounds	Epochs	200
Optimizer	Optimizer	Adam
Learning rate	Learning rate	0.001
L2 regularization coefficient	L2_reg	0.0001
Number of GCN layers	GCN_layers	3
Embedding dimension	Embedding_dim	64
Number of attention heads	Transformer_heads	4
Dropout rate	Dropout	0.2
Early stopping patience	Early_stopping_patience	10

sparsity, of moderate size, and its scenario closely matches the actual needs of e-commerce FMCG recommendation, thus it is broadly applied in research related to recommendation systems and sequence recommendation. The experimental environment and parameter configuration are shown in Table 1.

In terms of data partitioning and training strategy, this study adopts the following settings: the original dataset is randomly partitioned into training set, validation set, and test set in an 8:1:1 ratio, with a fixed random seed of 2024 during partitioning to ensure the reproducibility of experimental results. The validation set is used for early stopping strategy and hyperparameter selection. The monitoring amount for early stopping is the AUC value on the validation set. If there is no improvement in the AUC of the validation set after 10 consecutive training epochs, the training will be stopped and the model parameters will be restored to the optimal epochs. The hyperparameter search adopts a grid search strategy, and the search range includes: learning rates {0.0001, 0.0005, 0.001, 0.005}, L2 regularization coefficients {1e-5, 5e-5, 1e-4, 5e-4}, embedding dimensions {32, 64, 128}, GCN layers {1,2,3}, and attention heads {2,4,8}. Regarding the temporal partitioning method, this article strictly arranges the user behavior sequence in timestamp order when constructing it, with a fixed sequence length of 50. This processing method actually follows the principle of temporal partitioning, ensuring the temporal order of the training, validation, and testing sets and avoiding future information leakage. In the overall architecture of the LGTRM-DNN-FM algorithm, the recall phase and sorting phase adopt independent training strategies, which are sequentially connected on the data stream but do not share parameters. The recall phase takes the us-

er's historical interaction sequence and the user item bipartite graph structure as inputs, with a sequence length of 50 and an item embedding dimension of 64. The lightweight graph convolution module obtains the final embedding representation of users and items through 3-layer neighborhood aggregation. The embeddings of each layer are weighted and fused with equal weights to output a node embedding vector with a dimension of 64. At the same time, the user behavior sequence is embedded in a Transformer encoder for temporal modeling. The encoder includes 4 attention heads, and the hidden layer dimension of the feedforward network is 256. The node embeddings output by the graph convolution are concatenated with the sequence representations output by the Transformer in the feature dimension, resulting in a joint representation vector with a dimension of 128. This vector is mapped to a prediction score through a two-layer feedforward network. The recall phase adopts Bayesian personalized sorting loss as the optimization objective, and optimizes it through paired training of 1024 positive and negative sample pairs per batch. The negative samples are randomly sampled from items that the user has not interacted with, with a sampling ratio of 1:1.

The sorting stage takes the Top-K candidate item set output from the recall stage as input, and combines user features, item features, and contextual features to construct a combined feature vector with a feature dimension of 128. The factorization module explicitly models the interaction between first-order and second-order features, and outputs a low order feature representation with a dimension of 64; The deep neural network module consists of three hidden layers, with 256, 128, and 64 neurons in each layer, used for learning high-order nonlinear feature interactions. The Word2Vec module performs pre training representation learning on user behavior sequences, generating a 64 dimensional sequence embedding vector. This vector is concatenated with the output of the factorization machine and the original dense features before being input into a deep neural network. The output layer of the sorting stage uses the Sigmoid activation function to map the final features to click or purchase probabilities, and trains with cross entropy loss as the optimization objective. The recall phase and sorting phase are trained independently, without joint parameter updates during the training process, in order to reduce system coupling and improve the flexibility of engineering deployment.

To further analyze the contribution of each component module to the overall performance, an ablation experiment was designed and conducted on the complete model LGTRM-DNN-FM. This model consisted of three parts: LGTRM, DNN, and FM. By removing or replacing each of these modules, the role of each component in user interest modeling and recommendation performance improvement could be systematically evaluated. In terms of evaluation metrics, the study selected the area under the receiver operating characteristic curve (AUC) and logarithmic loss (LogLos) as overall predictive performance metrics. AUC evaluates how well a model can differentiate between positive and negative samples, while LogLoss characterizes the fit between the predicted probability and the true label. Furthermore, to evaluate the model's performance in ranking and recommendation quality, the top-50 hit ratio (HR@50) and the NDCG@50 were used as Top-K recommendation evaluation metrics, reflecting the hit coverage and ranking sensitivity of the recommendation results, respectively. The outcomes are presented in Table 2.

From Table 2, it can be seen that the performance of the model without fusion operation, i.e. w/o Fusion, decreased compared to the complete model LGTRM-DNN-FM. The area under the curve of the complete model was 0.902, and w/o Fusion decreased to 0.887. The normalized cumulative loss gain of the complete model was 0.241, while w/o Fusion decreased to 0.223. This is because the core of the fusion operation is to embed the nodes output by lightweight graph convolution and concatenate them with the behavioral sequence features encoded by Transformer, which achieves semantic alignment of two types of heterogeneous information. The area under the curve of w/o LightGCN is 0.885, and the area under the curve of w/o Transformer is 0.889, both of which are lower than the 0.902 of the complete model; In terms of normalized cumulative loss gain index, w/o LightGCN is 0.220 and w/o Transformer is 0.224, both lower than the 0.241 of the complete model. The above results indicate that removing either the graph convolution module or the Transformer module alone leads to performance

degradation, and removing the Transformer has a slightly greater impact on ranking quality indicators than removing the graph convolution. This is consistent with the strong temporal dependence of user behavior sequences in fast-moving consumer goods scenarios. From a theoretical perspective, the interaction between graph convolution and self attention can be explained as a synergistic mechanism of structure and timing. Graph convolution performs neighborhood aggregation on the user item bipartite graph, which is essentially a Laplacian smoothing operation that can propagate collaborative signals along the graph structure to alleviate data sparsity issues. However, the output of graph convolution is a static node representation, which cannot distinguish the intention changes of the same user in different time windows. In contrast, the self attention mechanism of Transformer can dynamically capture the user's interest drift trajectory by calculating the similarity between positions within the sequence. When two types of mechanisms are fused, graph convolution embedding provides a global structural prior for self attention, which clarifies which items are similar to the user's historical preferences in the collaborative filtering sense, thereby guiding attention weights to shift towards items with high collaborative relevance in sparse sequence positions. At the same time, sequence information dynamically

Table 2
Outcomes of ablation experiment.

Method	AUC	LogLoss	HR@50	NDCG@50
Full: LGTRM-DNN-FM	0.902	0.358	0.681	0.241
w/o LGTRM(DNN-FM only)	0.872	0.394	0.631	0.212
w/o Transformer(LightGCN-DNN-FM)	0.889	0.374	0.659	0.224
w/o LightGCN(Transformer-DNN-FM)	0.885	0.379	0.652	0.22
w/o DNN(LGTRM-FM)	0.881	0.381	0.646	0.218
w/o FM(LGTRM-DNN)	0.892	0.368	0.666	0.231
w/o Fusion	0.887	0.376	0.656	0.223

Figure 7
Comparison of AUC results.

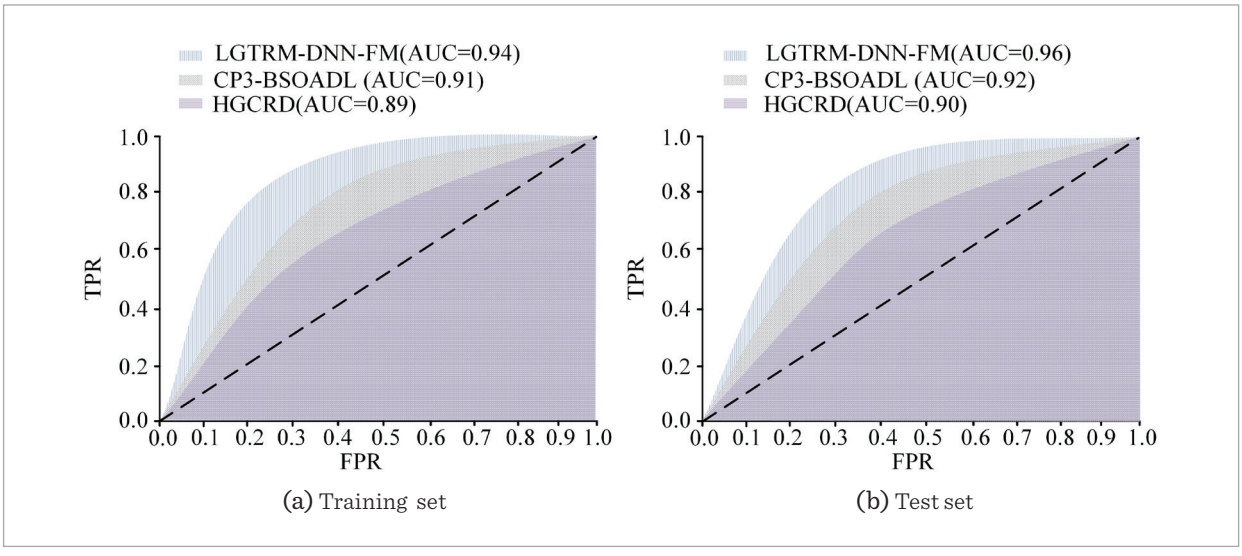
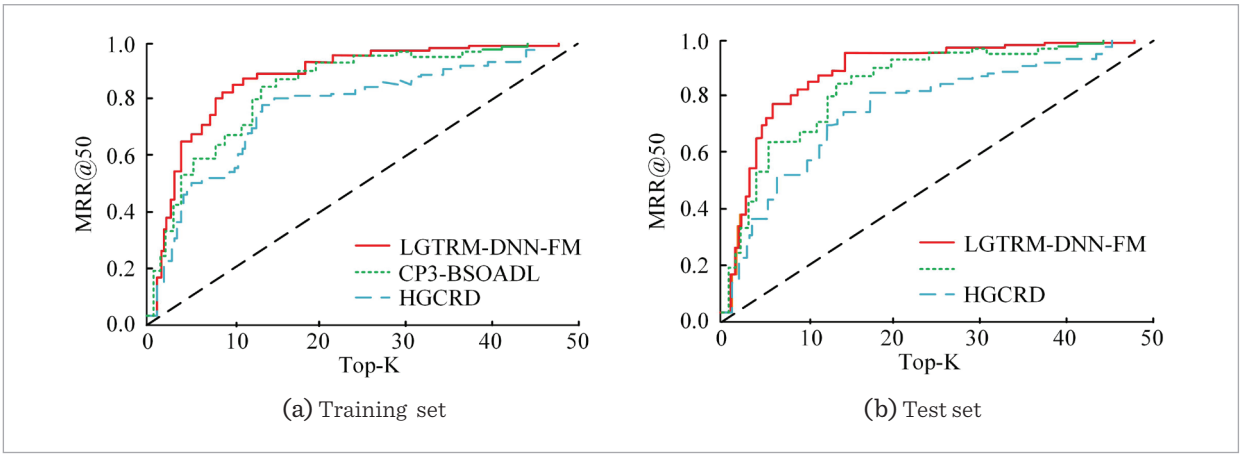


Figure 8
MRR results comparison.



modulates graph convolution embedding through temporal context, avoiding outdated recommendations of static graph structures when user interests shift. From an information theory perspective, fusion operations reduce the conditional entropy of joint representations, as graph information and sequence information encode different orthogonal dimensions of user preferences. When both work together, they provide a more complete depiction of the user's true intention. The above analysis verifies the rationality of the current fusion design from both theoretical and empirical perspectives.

To further confirm the validity and superiority of the introduced LGTRM-DNN-FM algorithm in the e-commerce FMCG recommendation task, the proposed LGTRM-DNN-FM algorithm was compared with Heterogeneous Graph Convolutional Network Recommendation with Adaptive Denoising Training (HGCRD) [11] and Consumer Purchasing Power Prediction via Barracuda Swarm Optimization Algorithm Driven Deep Learning (CP3-BSOADL) [1]. The AUC results of the three algorithms on the dataset are shown in Figure 7.

Figures 7(a) and 7(b) show the AUC outcomes of the three algorithms on the training and test sets, respectively. In Figure 7(a), during the training phase, the ROC curve of LGTRM-DNN-FM consistently ranked at the top, with an AUC of 0.94, demonstrating its strongest ability to distinguish between positive and negative samples. In Figure 7(b), the AUC of LGTRM-DNN-FM further improved to 0.96, and its ROC curve rose more steeply in the low false positive rate region. When the false positive rate was less than 0.2, the true positive rate approached 0.9, demonstrating outstanding early recognition ability. The reason for the difference compared to the AUC values in Table 2 is as follows. The AUC values in Table 2 are the final generalization performance indicators obtained on the test set after selecting the optimal training round based on the early stopping strategy on the validation set in the complete dataset, representing the true performance of the model on unknown data. The AUC value on the training set is used to reflect the model's sufficient learning level from the training data; The AUC value on the validation set comes from the optimal training round selected by the early stopping strategy, reflecting the peak discriminative ability reached by the model on the validation set. This is because it is the peak performance of the model's discriminative ability during the optimal training round, and the user purchasing behavior in the fast-moving consumer goods scenario has clear time dependence and repeat purchasing patterns, which provides a strong discriminative signal for the model, making the AUC reach 0.96 within a reasonable range. These results verified that LGTRM-DNN-FM had stronger discriminative and generalization abilities in FMCG recommendation. The Mean Reciprocal Rank (MRR) results of different algorithms are shown in Figure 8.

Figures 8(a)-(b) show the MRR results of the three algorithms on the training and test sets, respectively.

Figure 9
Comparison of training time.

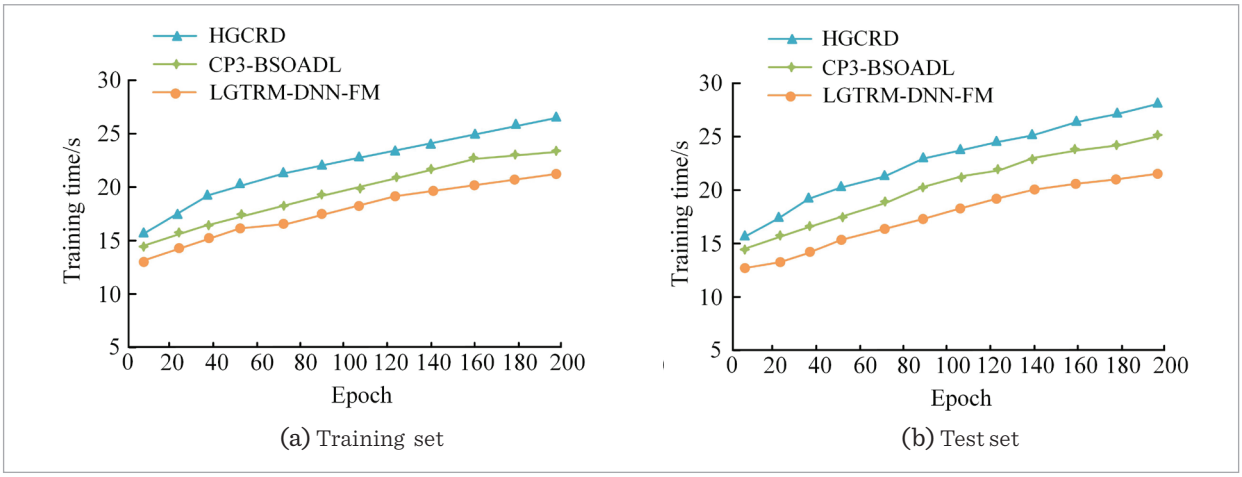
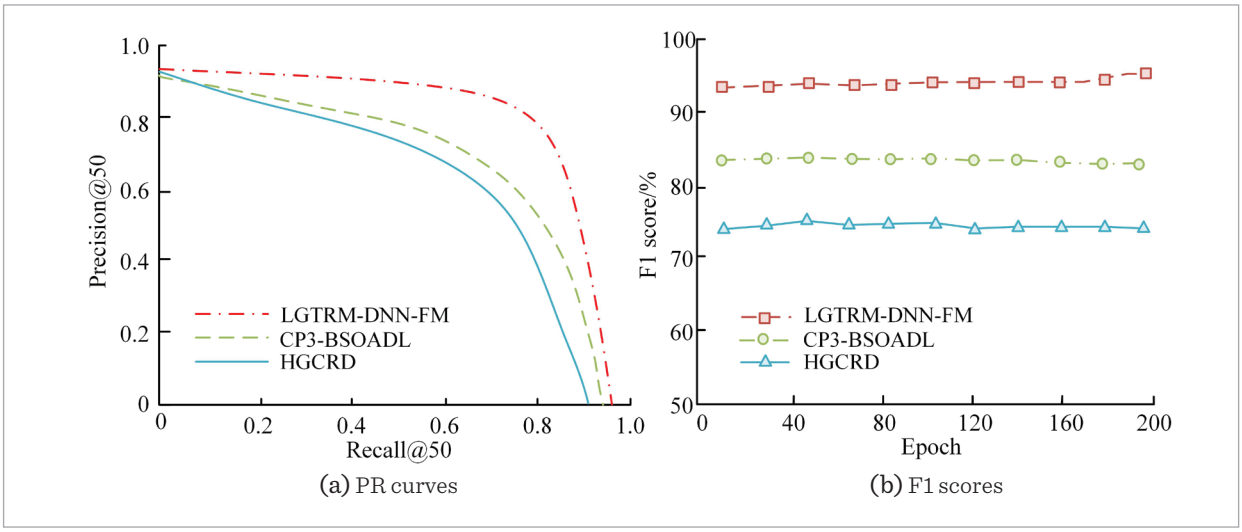


Figure 10
Comparison between PR and F1 score.



In Figure 8(a), LGTRM-DNN-FM consistently performed best, with an MRR close to 0.7 in the Top-5. As K increased to above 20, the MRR improvement of each model slowed down, but LGTRM-DNN-FM still maintained a stable lead, approaching 1.0 near the Top-50, consistently maintaining an advantage of approximately 0.05-0.10. This indicated that the model had a significant advantage in ranking accuracy and top-ranking ability. In Figure 8(b), LGTRM-DNN-FM maintained a significant advantage in the Top-5 to Top-10 region, with the fastest MRR increase, indicating that it could better capture users' core interests. This result reflected the model's good ranking performance and generalization ability. The training time comparison of the three algorithms is shown in Figure 9.

Figures 9(a)-(b) compare the training times of the three algorithms on the training and test sets, respectively. In Figure 9(a), LGTRM-DNN-FM took 14 seconds after 20 rounds, 17 seconds after 100 rounds, and 21 seconds after 200 rounds, saving 5 seconds compared to HGCRD and 1 second compared to CP3-BSOADL, demonstrating that this model had better computational efficiency during

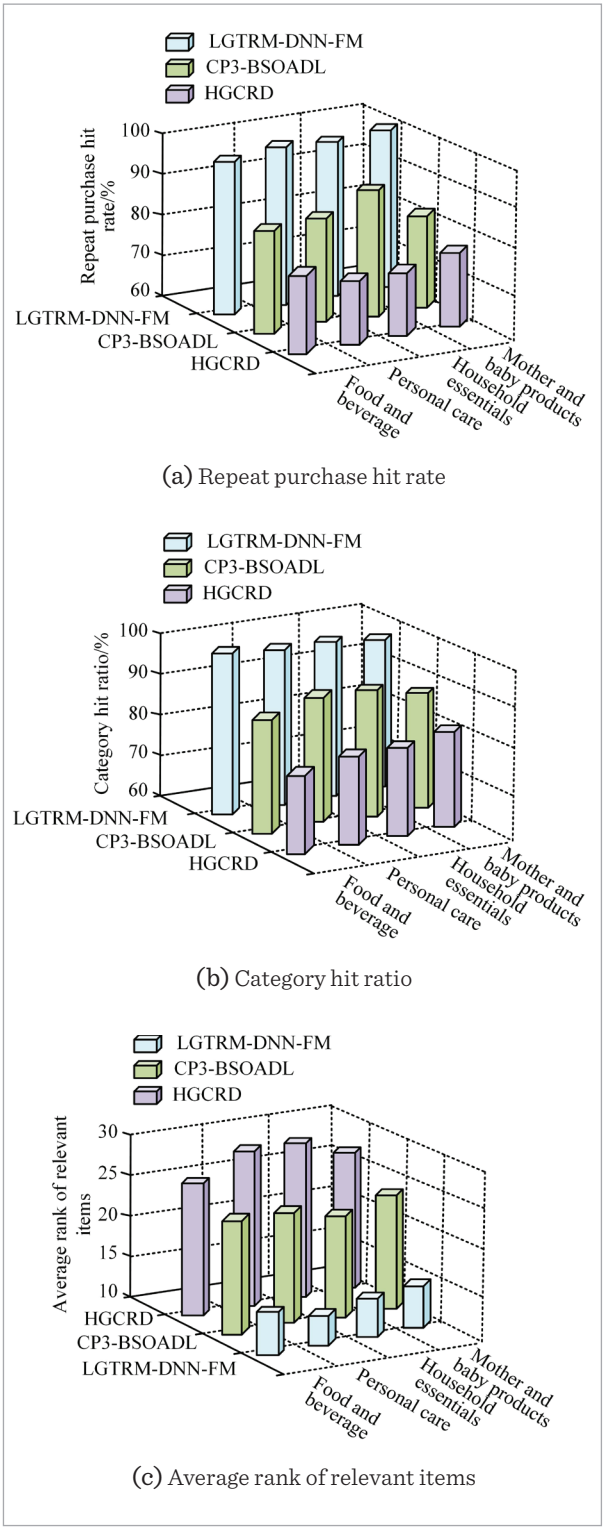
the training phase. In Figure 9(b), the time ranking of the three methods remained unchanged. The results on the training and test sets were consistent, indicating that this time difference stemmed from the computational efficiency of the model itself, rather than the randomness of the data partitioning. In summary, LGTRM-DNN-FM maintained superior recommendation performance while having lower training time overhead, demonstrating good engineering deployability.

3.2. Analysis of the Recommendation Effect of E-commerce FMCG Based on LGTRM-DNN-FM

To further analyze the recommendation quality and application effect of the proposed LGTRM-DNN-FM algorithm in the actual e-commerce FMCG recommendation scenario, this study conducted an in-depth analysis from the aspects of precision and recall balance, classification comprehensive performance, and the stability of recommendation results of different algorithms, to verify its comprehensive trade-off ability between coverage and accuracy. Figure 10 shows the precision-recall (PR) and F1 score of e-commerce FMCG recommendations based on three different algorithms.

Figures 10(a)-(b) compare the PR curves and F1 scores of the three algorithms, respectively. In Figure 10(a), when the recall rate was below 0.6, the precision of LGTRM-DNN-FM remained stable above 0.9. As the recall rate increased to the 0.7-0.8 range, the precision of LGTRM-DNN-FM could still be maintained at about 0.85, while CP3-BSOADL dropped to about 0.7, and HGCRD was below 0.6. LGTRM-DNN-FM had a better balance between coverage breadth and recommendation accuracy. In Figure 10(b), LGTRM-DNN-FM consistently had the highest F1 score, reaching about 93% in the early stages of training and increasing to about 95% around 200 rounds, with small fluctuations and stable convergence throughout. In theory, LGTRM-DNN-FM integrates graph structure information and temporal information, which can more comprehensively characterize user interests. Therefore, achieving an F1 value of 95% on the balance between accuracy and recall does not exceed a reasonable range. In summary, LGTRM-DNN-FM not only outperformed the comparison methods in terms of

Figure 11
Comparison of recommendation effects.



precision and recall, but also demonstrated superior overall classification performance and training stability, further validating the model's effectiveness and practical value in FMCG recommendation tasks. This study focused on FMCG recommendation scenarios in e-commerce, categorizing products into four typical categories: food and beverages, personal care and cleaning products, household goods, and baby products. The recommendation performance of different methods in each category was compared and analyzed, as shown in Figure 11.

Figures 11(a)-(c) compare the repurchase hit rate, category hit rate, and average ranking results of related products for the three algorithms across different FMCG categories. In Figure 11(a), LGTRM-DNN-FM maintained the highest category repurchase hit rate across all four FMCG categories, consistently remaining within the 90%-98% range. Specifically, it achieved 92% and 93% in the food and beverage and personal care/cleaning categories, respectively, and maintained 92% and 94% in the household and baby products categories. In Figure 11(b), LGTRM-DNN-FM maintained the highest category hit rate across all four FMCG categories, consistently remaining within the 90%-98% range. It reached 96% and 94% in the food and beverage and personal care/cleaning categories, respectively, and 95% and 93% in the household and baby products categories. As shown in Figure 11(c), LGTRM-DNN-FM achieved the lowest average ranking across all categories, indicating its ability to prioritize products that users were truly interested in at the top of the recommendation list. In summary, LGTRM-DNN-FM demonstrated stronger accuracy and stability in both specific product repurchase prediction and category-level preference matching, validating its practical application value in FMCG recommendation scenarios.

To further evaluate the robustness and stability of the proposed LGTRM-DNN-FM algorithm in scenarios with incomplete user-item interaction information, this study constructed experimental environments with 0%, 10%, 20%, and 30% interaction loss by randomly discarding different proportions of interaction edges in the user-item interaction graph. The performance of LGTRM-DNN-FM, CP3-BSOADL, and HGCRD was compared and analyzed in terms of AUC, LogLoss, HR@50, NDCG@50, and MRR@50. The comparison results under different proportions of missing interaction edges are presented in Table 3.

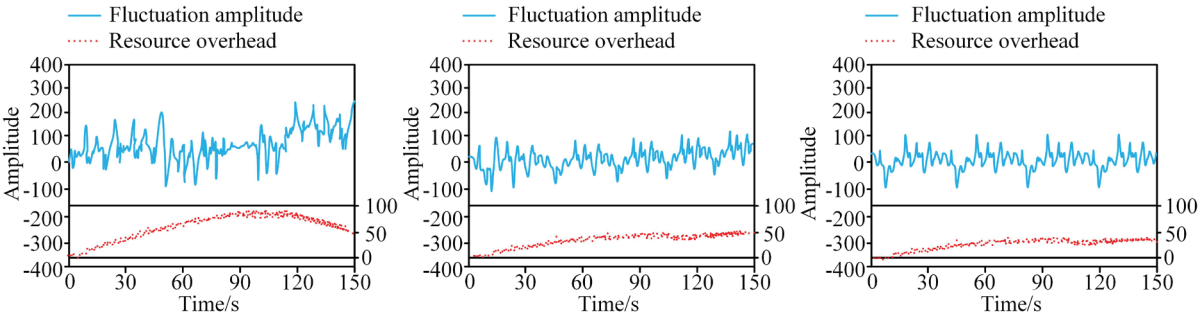
As shown in Table 3, as the proportion of discarded interactive edges gradually increased from 0% to 30%, all three methods showed a continuous decline in metrics such as AUC, HR@50, NDCG@50, and MRR@50, while LogLoss gradually increased, indicating that the lack of interactive information had a significant negative impact on recommendation performance. However, under different discard ratios, LGTRM-DNN-FM consistently performed

Table 3

Comparison under different proportions of missing interaction edges.

Edge dropout	Method	AUC	LogLoss	HR@50	NDCG@50	MRR@50
0%	LGTRM-DNN-FM	0.902	0.358	0.681	0.241	0.425
	CP3-BSOADL	0.889	0.374	0.659	0.224	0.389
	HGCRD	0.885	0.379	0.652	0.22	0.372
10%	LGTRM-DNN-FM	0.895	0.365	0.671	0.235	0.413
	CP3-BSOADL	0.88	0.384	0.646	0.216	0.374
	HGCRD	0.875	0.39	0.639	0.212	0.357
20%	LGTRM-DNN-FM	0.887	0.374	0.659	0.228	0.399
	CP3-BSOADL	0.869	0.396	0.632	0.206	0.356
	HGCRD	0.862	0.404	0.623	0.201	0.338
30%	LGTRM-DNN-FM	0.878	0.384	0.648	0.221	0.385
	CP3-BSOADL	0.856	0.411	0.614	0.193	0.335
	HGCRD	0.847	0.421	0.602	0.187	0.316

Figure 12
Comparison of resource consumption.



best with the smallest performance decline. For example, its AUC dropped from 0.902 to 0.878, a decrease of approximately 2.4%, while CP3-BSOADL and HGCRD decreased by approximately 3.7% and 4.3%, respectively. In terms of ranking metrics, LGTRM-DNN-FM's MRR@50 decreased from 0.425 to 0.385, but was still significantly higher than CP3-BSOADL's 0.335 and HGCRD's 0.316. Under the condition of 30% edge dropout, its HR@50 and NDCG@50 remained at 0.648 and 0.221, respectively, while the comparison method decreased to 0.614/0.193 and 0.602/0.187. Overall, although the lack of interactive edges reduced model performance, the LGTRM-DNN-FM, which integrated graph structure and sequence modeling, was more robust to structural perturbations and could maintain relatively stable recommendation performance under high edge dropout scenarios. The resource consumption of the three algorithms in e-commerce FMCG recommendation is shown in Figure 12.

Figures 12(a)-(c) compare the fluctuation amplitude and resource consumption of the three algorithms. In Figure 12(a), HGCRD exhibited the most severe fluctuation. It oscillated frequently within 150 seconds, with the fluctuation range initially between ±80-120 and later showing multiple peaks close to ±200, indicating a serious lack of stability. Simultaneously, its resource consumption continued to rise and remains high, peaking at approximately 70-80, indicating low efficiency. In Figure 12(b), the fluctuation of CP3-BSOADL showed some improvement: it is generally concentrated in the ±80-100 range, with fewer extreme peaks and a relatively smooth trend, indicating a more stable response. However, it still exhibited continuous oscillations in some areas, and the peak resource consumption was around 60, showing room for optimization. In Figure 12(c), LGTRM-DNN-FM performed best: its fluctuation remained within a very narrow range of ±60-80 throughout the entire process, with a stable trend, indicating strong perturbation suppression and high robustness. Meanwhile, its resource consumption remained consistently low, around 40-50%, significantly lower than the previous two methods. Overall, LGTRM-DNN-FM demonstrated outstanding performance in both stability and efficiency.

4. Conclusion

This study proposed a recommendation optimization algorithm, LGTRM-DNN-FM, which integrated lightweight graph convolution, Transformer, and DNN-FM for e-commerce FMCG recommendations. Its performance was systematically validated on publicly available datasets. Experimental findings showed that the introduced approach significantly outperformed comparative models in prediction accuracy, ranking quality, and robustness. In overall performance evaluation, LGTRM-DNN-FM achieved an AUC of 0.902 and reduces LogLoss to 0.358. In Top-K recommendation metrics, its HR@50 and NDCG@50 reached 0.681 and 0.241, respectively, demonstrating stronger recommendation coverage and ranking accuracy. Robustness analysis showed that even with a 30% drop-

out rate for interactive edges, LGTRM-DNN-FM maintained high AUC and MRR, with a significantly smaller performance degradation than comparative methods, demonstrating its good adaptability to structural perturbations. Furthermore, in recommendation experiments across different FMCG categories, the model maintained a stable advantage in metrics such as repurchase rate, category hit rate, and average ranking, while also exhibiting lower resource consumption and training time, demonstrating good engineering practicality. However, there are still limitations in the research, such as the current model mainly relying on static interaction and behavior sequence information, without explicitly introducing interpretability modules. Future work will explore embedding lightweight interpretable mechanisms without significantly increasing computational overhead, such as using neighborhood aggregation paths in graph convolution to generate recommendation criteria, or combining attention weights to provide users with retrospective analysis of key behavioral features.

References

1. Almuqren, L., Alruwais, N., Alhashmi, A. A., Alzahrani, I. R., Salih, N., Assiri, M., Shankar, K. WSN-Assisted Consumer Purchasing Power Prediction via Barracuda Swarm Optimization-Driven Deep Learning for E-Commerce Systems. IEEE Transactions on Consumer Electronics, 2024, 70(1), 1694-1701. https://doi.org/10.1109/TCE.2024.3371249
2. Alqhatani, A., Khan, S. B. IoT-Driven Hybrid Deep Collaborative Transformer with Federated Learning for Personalized E-Commerce Recommendations: An Optimized Approach. Scalable Computing: Practice and Experience, 2024, 25(5), 3408-3426. https://doi.org/10.12694/scpe.v25i5.3007
3. Alshareet, O., Awasthi, A. Light Spectral-Based Graph Convolutional Networks: Optimizing Ranking and Long-Tail Awareness for E-Commerce Recommendation Systems. Knowledge and Information Systems, 2025, 67(11), 10903-10928. https://doi.org/10.1007/s10115-025-02553-9
4. Bahrani, P., Minaei-Bidgoli, B., Keshavarz, M. A. A New Improved KNN-Based Recommender System. The Journal of Supercomputing, 2024, 80(1), 800-834. https://doi.org/10.1007/s11227-023-05447-1
5. Chen, L., Zhou, X., Yang, X., Sellis, T. Guest Editorial Special Issue on Online Recommendation Using AI and Big Data Techniques. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(10), 9809-9811. https://doi.org/10.1109/TKDE.2023.3310728
6. Dong, X., Li, B., Xie, K., Zhao, X., Xiao, R. The Effect of the Product Categories Diversity Recommended on Cross-Buying in Electronic Commerce Platforms: The Moderating Role of User Navigation Heterogeneity. Journal of Consumer Behaviour, 2025, 24(1), 304-331. https://doi.org/10.1002/cb.2393
7. Fu, Z., Cao, B., Liu, S., Peng, Q., Peng, Z., Shi, M., Liu, S. SMAR: Self-Supervised Mobile Application Recommendation Based on Graph Convolutional Networks. International Journal of Web Information Systems, 2024, 20(5), 520-536. https://doi.org/10.1108/IJWIS-06-2024-0178
8. Gao, S., Gao, R., Li, S., Yu, Y., Zhang, A., Zhang, L. Contrastive Learning-Based Personalized Tag Recommendation. Sensors, 2024, 24(18), 6061-6075. https://doi.org/10.3390/s24186061
9. He, Z., Ma, Y., Xiang, J., Zhang, N., Pan, R. A Small Neighborhood Fabric Recommender System Based on User Historical Behavior and Preference. Textile Research Journal, 2025, 95(11-12), 1299-1310. https://doi.org/10.1177/00405175241276794
10. Hong, Y., Rao, R., Li, X., Zhang, J., Dai, X., Zhang, M., Guo, S. Metaverse-Oriented User Preference Recommendation Systems Based on DSD-Transformer. IEEE Internet of Things Journal, 2025, 12(23), 49754-49767. https://doi.org/10.1109/JIOT.2025.3604510
11. Jin, S., Zhang, Y., Li, X., Lu, M. Heterogeneous Graph Convolutional Network for E-Commerce Product Recommendation with Adaptive Denoising Training. IEEE Transactions on Consumer Electronics, 2023, 70(1), 3259-3268. https://doi.org/10.1109/TCE.2023.3314537
12. Khan, I., Zhang, X., Ayyasamy, R. K., Alhashmi, S. M., Rahim, A. Enhancing Classification Algorithm Recommendation in Automated Machine Learning: A Meta-Learning Approach Using Multivariate Sparse Group Lasso. CMES-Computer Modeling in Engineering & Sciences, 2025, 142(2), 1611-1636. https://doi.org/10.32604/cmes.2025.058566
13. Kim, M., Kim, S. An Item Similarity Prediction and Recommendation System Using Aspect-Based Sentiment

Analysis and Graph Neural Networks. Industrial Engineering & Management Systems, 2025, 24(3), 282-294. https://doi.org/10.7232/iems.2025.24.3.282

14. Krouska, A., Troussas, C., Sgouropoulou, C. A Novel Group Recommender System for Domain-Independent Decision Support Customizing a Grouping Genetic Algorithm. User Modeling and User-Adapted Interaction, 2023, 33(5), 1113-1140. https://doi.org/10.1007/s11257-023-09360-3
15. Lai, P.-Y., Dai, Q.-Y., Liao, D.-Z., Yang, Z.-R., Liao, X.-D., Wang, C.-D., Chen, M. MASER: Multi-Order Attention and Semantic-Enhanced Representation Model for Complex Text Recommendation. IEEE Transactions on Emerging Topics in Computational Intelligence, 2024, 9(2), 1743-1755. https://doi.org/10.1109/TETCI.2024.3442872
16. Li, L., Jia, L., Al Otaibi, S. Intelligent Recommendation Algorithm of Consumer Electronics Products with Graph Embedding and Multi-Head Self-Attention in IoE. IEEE Transactions on Consumer Electronics, 2023, 70(1), 1278-1285. https://doi.org/10.1109/TCE.2023.3309978
17. Li, N., Gao, C., Jin, D., Liao, Q. Disentangled Modeling of Social Homophily and Influence for Social Recommendation. IEEE Transactions on Knowledge and Data Engineering, 2022, 35(6), 5738-5751. https://doi.org/10.1109/TKDE.2022.3185388
18. Li, X., Hong, Y., Zhang, M., Xie, X., Zhang, J., Guo, S. Social Group Oriented Recommendation Systems in Wireless Edge-Empowered Metaverse. IEEE Network, 2024, 39(3), 270-276. https://doi.org/10.1109/MNET.2024.3508792
19. Lv, P., Guan, Z., Zhang, Q., Gu, J., Zhang, S., Lv, L., Zhang, M., Li, L., Zhao, C. DSMN: An Improved Recommendation Model for Capturing the Multiplicity and Dynamics of Consumer Interests. IEEE Transactions on Consumer Electronics, 2023, 70(1), 1236-1245. https://doi.org/10.1109/TCE.2023.3277549
20. Ma, X., Zhang, H., Zeng, J., et al. FedKGRec: Privacy-Preserving Federated Knowledge Graph Aware Recommender System. Applied Intelligence, 2024, 54(19), 9028-9044. https://doi.org/10.1007/s10489-024-05634-4
21. Oprea, S. V., Bra, A. Graph-Based User-Centric Recommender System Using Neo4j, Cypher, and Jaccard Similarity in the Field of E-Commerce. IEEE Access, 2025, 13, 135089-135097. https://doi.org/10.1109/ACCESS.2025.3591705
22. Shakoor, D. M., Maihami, V., Maihami, R. A Machine Learning Recommender System Based on Collaborative Filtering Using Gaussian Mixture Model Clustering. Mathematical Methods in the Applied Sciences, 2026, 49(5), 3589-3602. https://doi.org/10.1002/mma.7801
23. Singh, K., Dhawan, S., Gambhir, S. Utilizing Artificial Intelligence to Detect Fraudulent Manipulation in Recommender Systems. Automatic Control and Computer Sciences, 2025, 59(2), 206-218. https://doi.org/10.3103/S0146411625700178
24. Tasgetiren, N., Safak, I., Aktas, M. S. An Event Frequency-Inverse Session Frequency Based Weighting Mechanism for Data Embedding Methodologies. Software: Practice and Experience, 2025, 55(6), 983-1002. https://doi.org/10.1002/spe.3407
25. Tong, G., Li, D., Liu, X. An Improved Model Combining Knowledge Graph and GCN for PLM Knowledge Recommendation. Soft Computing, 2024, 28(6), 5557-5575. https://doi.org/10.1007/s00500-023-09340-0
26. Ullah, I., Adhikari, D., Ali, F., Ali, A., Khan, H., Sharafian, A., Kesavan, S. M., Bai, X. Revolutionizing E-Commerce with Consumer-Driven Energy-Efficient WSNs: A Multi-Characteristics Approach. IEEE Transactions on Consumer Electronics, 2024, 70(4), 6871-6882. https://doi.org/10.1109/TCE.2024.3411606
27. Yue, L., Wang, Q. RETRACTED: Personalized Multi-Attribute Recommendation Method Based on Partial Order Set. Journal of Intelligent & Fuzzy Systems, 2024, 46(4), 8741-8754. https://doi.org/10.3233/JIFS-231724
28. Zhang, Y. E-Commerce Data Mining Analysis Based on User Preferences and Association Rules. Scalable Computing: Practice and Experience, 2024, 25(5), 3515-3522. https://doi.org/10.12694/scpe.v25i5.3041
29. Zheng, Z., Hu, X., Qiu, Z., Cheng, Y., Gao, S., Song, Y., Zhu, H., Xiong, H. Bilateral Multi-Behavior Modeling for Reciprocal Recommendation in Online Recruitment. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(11), 5681-5694. https://doi.org/10.1109/TKDE.2024.3397705
30. Zhou, Y.-Q., Yang, W., Ma, Z.-M., Xu, Z. Weighted Bonferroni Mean-Based Classifiers with Factorization Machines for Binary Classification. Intelligent Data Analysis, 2025, 29(5), 1085-1105. https://doi.org/10.1177/1088467X241301694

This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (http://creativecommons.org/licenses/by/4.0/).