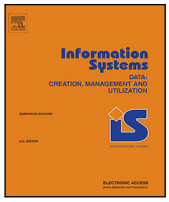
Information Systems 121 (2024) 102334

Contents lists available at ScienceDirect

Information Systems

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



Attention-based multi attribute matrix factorization for enhanced recommendation performance

Dongsoo Jang [a], Qinglong Li [a,*], Chaeyoung Lee [a], Jaekyeong Kim [a,b]

[a] Department of Big Data Analytics, Kyung Hee University, 26, Kyungheedae-ro, Dongdaemun-gu, Seoul, 02447, Korea
[b] School of Management, Kyung Hee University, 26, Kyungheedae-ro, Dongdaemun-gu, Seoul, 02447, Korea

ARTICLE INFO

Keywords:
Auxiliary information
Deep learning
E-commerce platforms
Recommender system
Self-attention mechanism

ABSTRACT

In E-commerce platforms, auxiliary information containing several attributes (e.g., price, quality, and brand) can improve recommendation performance. However, previous studies used a simple combined embedding approach that did not consider the importance of each attribute embedded in the auxiliary information or only used some attributes of the auxiliary information. However, user purchasing behavior can vary significantly depending on the attributes. Thus, we propose multi attribute-based matrix factorization (MAMF), which considers the importance of each attribute embedded in various auxiliary information. MAMF obtains more representative and specific attention features of the user and item using a self-attention mechanism. By acquiring attentive representation, MAMF learns a high-level interaction precisely between users and items. To evaluate the performance of the proposed MAMF, we conducted extensive experiments using three real-world datasets from amazon.com. The experimental results show that MAMF exhibits excellent recommendation performance compared with various baseline models.

1. Introduction

E-commerce platforms provide multiple items to help users purchase those that meet their needs. With the advancement of information and communication technologies, the scale of the e-commerce industry and the number of items and users are rapidly increasing [1]. As a result, the information overload problem that incurs high costs during the purchase decision-making process arises and causes inefficiencies in multiple aspects, such as marketing costs for businesses and user purchasing behavior [2]. In this context, recommender systems have attracted attention as effective solutions for addressing these problems in e-commerce. Through recommender systems, e-commerce platforms can enhance their competitiveness in terms of sales volume and user loyalty, and users can simultaneously receive items that suit their preferences [3]. Consequently, recommender systems have become an integral part of e-commerce.

Matrix factorization (MF) is a popular recommendation methodology with high performance among collaborative filtering (CF) models. It estimates user preferences using the past purchase history as the only source of information. Koren et al. [4] initially proposed MF and decomposed the interaction matrix between users and items to identify the latent factors of users and items, and to estimate user preferences. However, MF models based on linear operations are limited in effectively capturing interactions when the relationships between users and items are complex. To address this, He et al. [5] proposed a novel framework that represents the user and item as latent factor vectors and learns the interaction between the user and item vectors using multi-layer perceptron (MLP) operations to capture nonlinear relationships. Similarly, Deng et al. [6] used representation learning to extract the factors of users and items and performed matching function learning based on these factors to effectively capture complex interaction relationships. Such a deep learning-based recommendation methodology exhibited excellent performance compared to the linear MF methodology but still has inherent limitations. In E-commerce platforms, most users rate only a few items, which results in a very sparse interaction matrix between users and items. Consequently, a recommendation methodology that utilizes a sparse rating matrix as the only information source can face the data sparsity issue, and it can be challenging to estimate user preferences.

In the recommender system field, data sparsity is one of the essential issues and many researchers have reported that using attributes embedded in auxiliary information can relieve data sparsity issues [7]. Therefore, auxiliary information such as item categories or user review text has been widely used to alleviate the data sparsity issue in recommender system studies. Kim et al. [8] proposed a recommendation model that integrates item description information extracted using a

* Corresponding author.
E-mail addresses: zsdf21@khu.ac.kr (D. Jang), leecy@khu.ac.kr (Q. Li), acy98@khu.ac.kr (C. Lee), jaek@khu.ac.kr (J. Kim).

https://doi.org/10.1016/j.is.2023.102334
Received 21 August 2023; Received in revised form 6 December 2023; Accepted 8 December 2023
Available online 9 December 2023

convolutional neural network (CNN) into a traditional MF model to relieve the data sparsity problem. To improve recommendation accuracy, Tuan and Phuong [9] incorporated item content information, such as item title, description, and category, into the recommendation model using a CNN. On the other hand, auxiliary information contains various attributes that can represent users or items such as price and quality. In many previous studies, encoded representations from diverse auxiliary information have been combined without considering the importance of each attribute embedded in the auxiliary information. However, when users select items, their purchasing behaviors can vary depending on the attributes embedded in the auxiliary information. Therefore, such a simple combined embedding approach is ineffective from a recommendation perspective because it cannot consider the importance of each attribute. For example, in the selection of an item, the user may consider the brand a critical attribute but not the color. Simultaneously, another user can prioritize color over other attributes. Wang et al. [10] confirmed that the attributes embedded in items significantly affect user purchasing behavior. Therefore, the recommendation model must consider the importance of each attribute embedded in the auxiliary information to accurately estimate user preferences for items. In this case, attention mechanisms can be applied to the recommendation process to consider the importance of each attribute. The attention mechanism provides an approach for focusing on key features during the input phase of a deep learning model, and its effectiveness has been demonstrated in various fields [11,12]. This study attempted to use self-attention mechanisms to extract more representative and specific attention features embedded in the auxiliary information of the corresponding user and item. At this point, the extracted attention score can be used as a weight for the feature representation to consider the importance of each attribute and consequently obtain specific attention feature representations [13].

This study propose a novel recommendation model called MAMF (multi attribute-based matrix factorization), which effectively captures the interaction between the user and the item by comprehensively using auxiliary information and considering the importance of each attribute embedded in the auxiliary information. MAMF obtains a combined embedding representation corresponding to the user and item. The combined embedding representation is a fusion of multiple attributes embedded in auxiliary information into a multi-embedding representation for each user and item. It then utilizes a self-attention mechanism to identify the importance of each attribute embedded in the auxiliary information. Based on this approach, the MAMF can extract more representative and specific attention features for users and items. In conclusion, the MAMF learns a high level of attentive interaction between users and items to predict user preferences. We used three categories of real-world datasets from amazon.com to effectively evaluate the recommendation performance of the proposed MAMF. The experimental results show that the proposed MAMF outperforms various baseline models. The main contributions of this study are summarized as follows.

- We propose an MAMF that effectively captures the attentive interaction between the user and an item by considering various types of auxiliary information and their importance. The proposed model provides a method to effectively integrate item and user information and confirms that diverse auxiliary information that can represent users or items more specifically can improve recommendation performance.
- We utilize a self-attention mechanism to identify the importance of each embedded attribute when using various auxiliary information. By considering the importance of each attribute, MAMF can obtain the representative and specific attention features of the user and item.
- This study compared the MAMF with various baseline models using three real-world datasets from amazon.com to evaluate the recommendation performance. The experimental results demonstrate that the proposed MAMF model performs better than the baseline models.

The remainder of this study is organized as follows. Section 2 provides an overview of related studies. Section 3 describes the problems addressed in this study. Section 4 introduces our proposed recommendation model. Section 5 describes the datasets and experiment setting. Section 6 summarizes the experimental results, and Section 7 presents the conclusions of this study.

2. Related studies

2.1. Matrix factorization

A recommender system aims to recommend suitable items to users based on their purchase history. With the continuous growth of the e-commerce industry, the number of items and users has increased, and the recommender system has become important. CF is a representative recommendation method based on neighborhood or model-based approaches [14,15]. The neighborhood-based CF model calculates the similarity between users or items. It provides recommendations by selecting items that users will likely prefer among items they have not purchased using the purchase history of neighbors with high similarity. Although a neighborhood-based CF model can effectively provide recommendations, it faces the issue of exponentially increasing computational complexity when the number of users or items grows [2]. Subsequently, an MF model was proposed as a model-based CF. The MF model utilizes latent factors of user and item and exhibits superior performance to that of the neighborhood-based CF model and has become a popular recommendation model because of its high performance and scalability [16]. For example, Bao et al. [17] proposed a recommendation model that combines latent factors obtained from the rating matrix with topic modeling applied to review texts to improve the recommendation accuracy. Additionally, Zhang et al. [7] searched the impact of useful information on recommendations and applied it to the MF framework. The proposed method captures the interaction between additional field information in the latent space for enhancing the recommendation performance. Liu et al. [18] proposed a recommendation model that utilized heterogeneous information in an MF-based recommendation model, capturing flat and hierarchical information jointly. However, traditional MF methodologies have limitations in reflecting complex interaction relationships between the user and item because they perform linear operations on user-item interactions. These methods hinder the effective capture of complex interaction patterns. To address this issue, a neural collaborative filtering (NCF) model was proposed as an extension of the MF methodology, incorporating the MLP to capture nonlinear interactions [5]. The NCF model represents users and items by latent factor vectors, and captures their interactions nonlinearly through an MLP calculation, achieving superior performance than traditional MF models.

MF has strengths in scalability, implicit modeling, and flexibility compared with the traditional CF model. Consequently, extensive recommendation studies based on MF models have been conducted. This study proposes a novel recommendation model that extends deep learning-based MF models by combining integrated representations of various attributes in auxiliary information. This study aims to effectively incorporate various auxiliary information that can represent users and items into a deep learning-based MF model and analyze the impact of these factors on recommendation performance.

2.2. Auxiliary information-based recommender system

Previous recommendation models using the MF methodology have been developed using various methods. However, MF models that rely only on the rating matrix have inherent limitations related to data sparsity problems [8]. Users of e-commerce platforms usually have a short purchase history for only a few items. Consequently, the rating matrix that represents the relationships between the user and the item becomes sparse, hindering the provision of sufficient information for

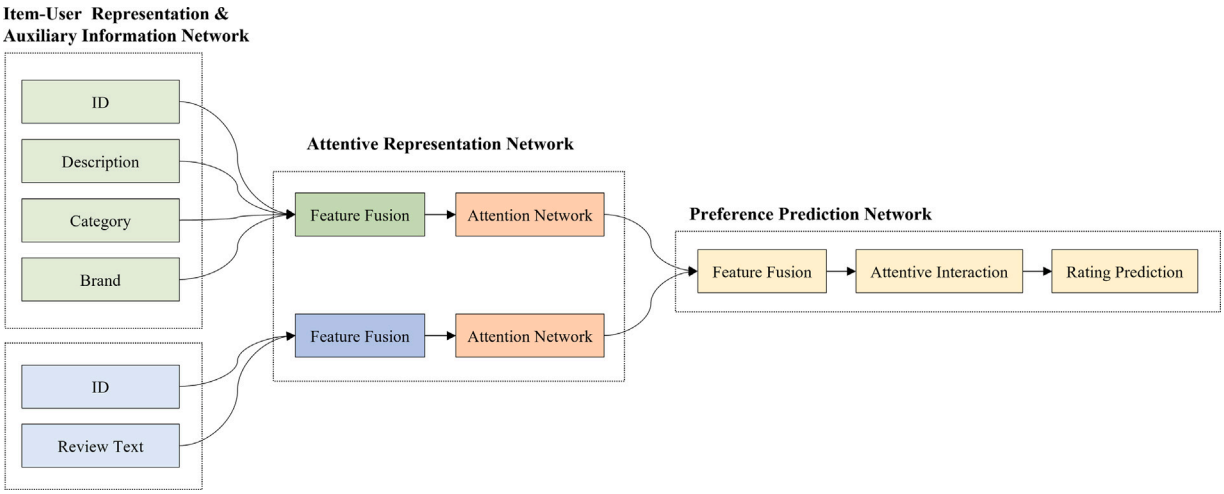


Fig. 1. Architecture of the MAMF.

interaction learning [19]. Thus, reliance on a sparse rating matrix as the only information can constrain recommendation performance owing to an insufficient information source. In this regard, many recommender system researchers have emphasized the use of auxiliary information to address the data sparsity problem [8]. Auxiliary information related to users or items can serve as a useful source of information to capture user preferences effectively. For example, Strub et al. [3] proposed a recommendation model that integrated auxiliary information using an autoencoder to address data sparsity and cold-start problems. They integrated user profiles and item category information to relieve the issues in the recommender system. Likewise, Liu et al. [2] proposed a model incorporating content multiple information into an NCF framework to enhance the recommendation performance. They constructed a hybrid recommendation model based on an autoencoder that incorporates the auxiliary information of the user, such as age, occupation, and gender, and the auxiliary information of the item, such as the movie genre. Tuan and Phuong [9] proposed a recommendation model to overcome the limitations of existing approaches that rely on sparse rating matrix between users and items in e-commerce. They represented item information, such as title, description, and category, using a three-dimensional CNN and incorporated these representations into the recommendation model.

Utilizing auxiliary information related to the user or item can relieve data sparsity problems and improve recommendation performance. However, previous studies employed a simple embedding combined approach and did not consider the importance of each attribute embedded in the auxiliary information. Namely, previous studies used attribute information but assumed each attribute has the same effect on recommendation performance. In practice, each attribute can affect recommendation performance differently when using various auxiliary information. Therefore, the absence of consideration of the importance of each attribute embedded in auxiliary information can constrain the recommendation performance. Therefore, this study aims to capture a high level of interaction between users and items using extended MF models after obtaining an integrated representation of users and items, considering the importance of each attribute.

2.3. Self-attention mechanism in the recommender system

Recently, recommender system researchers have begun to emphasize the need to distinguish the importance of attribute information when addressing data sparsity problems. Thus, the attention mechanism has functioned effectively in the recommendation model. Attention mechanism focus on feature representation at the input step and have been widely used in various domains owing to their effectiveness [11,12]. In particular, the self-attention mechanism uses the single input vector and calculates the attention score by calculating the input vector as attention sources [11]. Using the obtained attention score as a weight in the feature representation, the importance of each attribute can be considered to obtain an advanced feature representation. For example, Lv et al. [20] adopted a self-attention mechanism that estimates user preferences for different items based on purchase history. Chen et al. [21] used self-attention mechanism to improve recommendation performance by considering the potential preference of a user for the specific attributes of the item. Ma and Liu [22] adopted a self-attention mechanism to learn the dependencies of user and item review features and consider the weight of different features. Furthermore, Chen et al. [21] used self-attention mechanisms to capture the user's more detailed preference information for the attributes through interactions with each user attribute and overall attributes of the item.

However, such approaches capture the importance of each attribute but do not incorporate the importance of the attribute into user and item representations. Primarily, they only interact with each attribute independently. This study proposes a novel recommendation model that extracts more representative and specific attention features embedded in auxiliary information and incorporates them into the representation of the user and item. Therefore, auxiliary information such as review text, item category, description, and brand for the representation of the detailed aspects of the user or item are utilized. The self-attention mechanism is then utilized to consider the importance of each attribute embedded in the auxiliary information. The proposed model can capture a high-level attentive interaction representation between the user and an item to predict user preferences. The proposed model can capture a high-level attentive interaction for predicting user preferences by the specific representation of the user and item.

3. Problem definition

Fig. 1 illustrates the overall architecture of the proposed MAMF. MAMF consists of four networks: item–user representation, auxiliary information, attentive representation, and preference prediction. Many recommender system studies represent auxiliary information using a simple combined embedding approach without considering the importance of each attribute embedded in the auxiliary information [23,24]. However, this approach can limit the recommendation performance. To address these limitations, we propose a MAMF that combines various types of auxiliary information to effectively represent users or items. Simultaneously, MAMF considers the importance of using a self-attention mechanism to extract more representative and specific attention features embedded in auxiliary information. Subsequently, MAMF utilizes the acquired attentive representation vector of the item

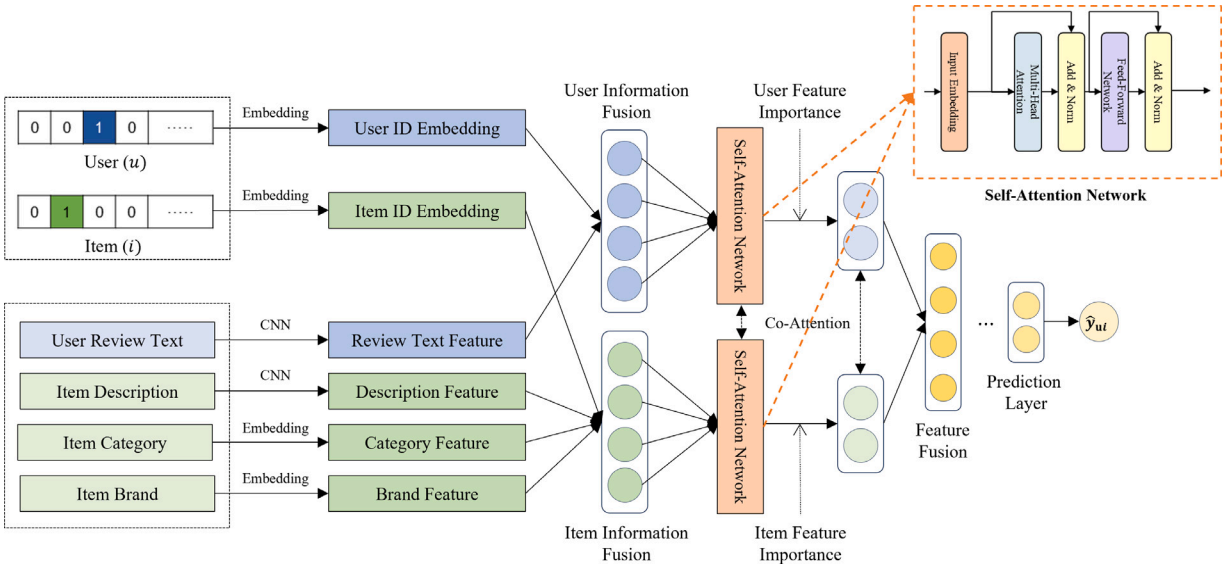


Fig. 2. Framework of the MAMF.

and user to provide rating predictions in the preference prediction network. T indicates the various pieces of information between the user and the item. Each $T = (u, r, i, d, c, b, y)$ is a tuple of user identity (ID) u, user review text r, item ID i, item description d, category c, brand b, and preference rating y. MAMF aims to learn the prediction model F formalized as

$$F(u, r, i, d, c, b; \theta) \rightarrow \hat{y}, \tag{1}$$

where θ and $\hat{y}$ represent the bias and predicted preference rating. The proposed MAMF inputs user ID, user review text, item ID, item description, category, and brand. During the learning process, the model was trained to approximate the predicted preference rating $\hat{y}_{u,i}$ using the actual user preference rating $y_{u,i}$. After the entire learning process, the model outputs the predicted preference ratings of the users for an item.

4. MAMF framework

In this study, we propose MAMF, which effectively utilizes various types of auxiliary information, considering the importance of each attribute embedded in the auxiliary information through a self-attention mechanism. The MAMF consists of four networks, as illustrated in Fig. 2. The item–user representation and auxiliary information networks utilize information such as ID, description, category, brand, and review text to output each representation vector that richly illustrates the detailed preference information of the item and user. The attentive representation and preference prediction networks create attentive representation vectors of items and users and perform nonlinear learning to predict preference ratings. The detailed explanations of each network are provided below.

4.1. Item–user representation network

This network aims to extract each latent factor vector for items and users. The item factor utilizes the item ID and user ID for user factor. The item ID and user ID are transformed into a dense latent representation vector through an embedding layer, as shown in Eq. (2)

$$q_i = Q^T v_i^I;\ p_u = P^T v_u^U, \tag{2}$$

where $Q \in \mathbb{R}^{|I| \times d_i}$ is the embedding matrix of items and $P \in \mathbb{R}^{|U| \times d_u}$ is the embedding matrix of users. $|I|$ and $|U|$ represent the number of items and users, respectively, and d represents the embedding dimension. v_i^I and v_u^U are the one-hot encoded vectors of the ID of item i and user u. Consequently, this network outputs the latent factor vectors q_i and p_u of the item and user, respectively.

4.2. Auxiliary information network

To extract specific items and user representations that include attributes embedded in various auxiliary information, this network utilizes item description, category, and brand to represent the detailed aspects of the item. At the same time, user review text is utilized that contains detailed preference information of the user.

Among the auxiliary information, the item description and user review text contain detailed information about the item and rich user preference information in a textual format. For textual information, a CNN is useful for extracting a contextual feature vector that captures the semantic context [24]. Therefore, the CNN was applied to item descriptions and user review texts. First, the description of item i is defined as $D_i = \{w_1, w_2, \ldots, w_n\}$, where w_k represents kth word in item description and n represents the length. To apply CNN operations, item description D_i is processed through a word embedding $f : w_k \rightarrow \mathbb{R}^d$, which is applied for each word to represent a dense vector. After the word embedding process, the item description is represented by a matrix $V \in \mathbb{R}^{n \times d}$, where d is the embedding dimension of each word. Subsequently, a convolution operation is applied to extract a word-level semantic feature vector of the item description through multiple filters. The convolution layer uses jth filter K_j to operate the sliding window as shown in Eq. (3).

$$c_j = \phi(V * K_j + b_j), \tag{3}$$

where $*$ denotes the convolution operator and $K_j \in \mathbb{R}^{t \times d}$ is the filter kernel, where the kernel size is $t \times d$. b_j is the bias and $\varnothing$ is the rectified linear unit (ReLU) used as the activation function. In addition, c_j performs an average pooling operation to retain the main semantic representation and remove noise, as shown in Eq. (4).

$$o_j = average([c_1, c_2, \ldots, c_{n-t+1}]). \tag{4}$$

Multiple filters were applied to consider various semantic features in the item description, as shown in Eq. (5).

$$t = [o_1, o_2, \ldots o_{n_1}]. \tag{5}$$

Through Eq. (5), we can obtain t, which captures the semantic context in the item description.

The user review text indicates $R_{u,i} = \{r_1, r_2, \ldots, r_l\}$, which represents the review text written by user u to item i. r_k and l represent the kth word in review text and length, respectively. To extract the specific

preference information of the user, the review text was processed by the CNN. For this purpose, word embedding $f : r_k \rightarrow \mathbb{R}^d$ is applied for each word in $R_{u,i}$. Then, the user review text is represented by a matrix $M \in \mathbb{R}^{l \times d}$, where d is the embedding dimension. In the word embedding of the review text, a convolution operation is applied to abstract a word-level semantic feature, as shown in Eq. (6):

$$e_i = \phi(M * K_i + b_i), \tag{6}$$

where, $K_i \in \mathbb{R}^{t \times d}$ is ith filter for the convolution layer to operate sliding window, $*$ denotes the convolution operator. $t \times d$, b_i and $\varnothing$ represent the kernel size, bias, and activation function ReLU, respectively. Then, an average pooling operation is performed in e_i to retain the main semantic representation and remove noise in the review text, as shown in Eq. (7).

$$k_j = average([e_1, e_2, \ldots, e_{l-t+1}]). \tag{7}$$

Multiple filters are applied to extract various review semantic features, as shown in Eq. (8).

$$s = [k_1, k_2, \ldots k_{n_1}], \tag{8}$$

where s represents the semantic representation vector containing the user preference information.

Further, the item category $A_i = \{a_1, a_2, \ldots, a_n\}$ represents the categorical information and n represents the number of categories of item I. Simultaneously, the item brand is represented as $B_i = \{b_i\}$. The item categories and brands were highly diverse; therefore, they were represented as sparse matrices. Therefore, one-hot encoding representations transform an item category and brand into a dense representation vector using an embedding layer. This process represents the item category embedding matrix as $E_c \in A^{n \times d}$ and brand embedding matrix as $E_b \in B^{n \times d}$, where n represents the number of items and d represents the embedding dimension. Finally, the item category and brand are represented by e_c and e_b, as shown in Eq. (9).

$$e_c = E_c^T v_i^I; \; e_b = E_b^T v_i^I. \tag{9}$$

4.3. Attentive representation network

Through the previous networks, the extracted item–user latent factor and various auxiliary information representations are combined to form a specific item–user representation:

$$I^c = [q_i; t; e_c; e_b]; \; U^c = [p_u; s], \tag{10}$$

where I^c and U^c represent the combined vector of item and user that includes various attributes and [;] is the concatenation operation. In the item–user representation process, a specific attribute or factor within an attribute may be particularly important; thus, an effective representation can be obtained by considering the importance of the attribute. Therefore, a self-attention mechanism was used to obtain each attentive representation vector that considered the importance of attributes.

$$\begin{aligned} &Q, K, V = XW^Q, XW^K, XW^V, \\ &I^a = softmax\left(\frac{Q_i {K_i}^T}{\sqrt{d_k}}\right) V_i, \\ &U^a = softmax\left(\frac{Q_u {K_u}^T}{\sqrt{d_k}}\right) V_u, \end{aligned} \tag{11}$$

where X represents each representation vector I^c and U^c, respectively. W^Q, W^K and W^V represent weight matrices for each query, key, and value. d_k is the number of scaling dimensions. The attention weights are then acquired using a SoftMax function. Consequently, I^a and U^a represent the attentive representation vector that considers the importance of the attribute embedded in the auxiliary information acquired by element-wise multiplication between each calculated attention weights and V.

In addition, the item–user feature learning process is performed separately, which can help perform interaction operations [25]. The representation proceeds as a feature-learning process through MLP operations and finally obtains the item–user representation vectors I_L^a and U_L^a, reflecting specific attention features in this network.

$$\begin{aligned} &I_1^a = \alpha_1(W_1 I_0^a + b_1), \\ &\vdots \\ &I_L^a = \alpha_L(W_L I_{L-1}^a + b_L), \\ &U_1^a = \alpha_1(W_1 U_0^a + b_1), \\ &\vdots \\ &U_L^a = \alpha_L(W_L U_{L-1}^a + b_L), \end{aligned} \tag{12}$$

where W, b, L, and α denote the weight matrix, biases, number of hidden layers, and activation function ReLU, respectively.

4.4. Preference prediction network

The item and user representation vectors obtained in previous networks contain various attributes for learning a high level of attentive interaction between the user and item. These representation vectors are used to model the interaction between the user and items and predict the preference rating of the user for an item. To perform the preference prediction, the item and user representation vectors are concatenated as follows:

$$V^c = [I_L^a; U_L^a], \tag{13}$$

where the generated vector V^c is fed into the MLP layer for rating prediction.

$$\begin{aligned} &V_1^c = \alpha_1(W_1 V_0^c + b_1), \\ &\vdots \\ &V_M^a = \alpha_M(W_M V_{M-1}^a + b_M), \end{aligned} \tag{14}$$

where W and b represent the weight matrix and the bias, respectively. The activation function α is ReLU, used for the propagation of the results between hidden layers. The hidden layer is iteratively applied M times to fine-tune the parameters [5]. Moreover, the output of the last hidden layer is considered the input for the final rating prediction layer, and the output of the final layer is the predicted preference rating. The prediction layer is formalized as

$$\hat{y}_{u,i} = f(\mathrm{W}_u V_M^a), \tag{15}$$

where W_u represents the weight matrix of the prediction layer, and $\hat{y}_{u,i}$ denotes the predicted preference rating. The purpose of this network is to predict preference ratings in a specified range of 1–5. Therefore, to predict ratings within a specified range, ratings in the range of 1–5 were adjusted to a range of 0–1 through linear transformation before the training process [26]. Thus, the output function f of the final prediction layer is a sigmoid function with output values ranging from 0–1. Finally, the predicted values were linearly transformed back to the original rating range by applying the linear transformation $\hat{y}_{u,i} \times (y_{\max} - y_{\min}) + y_{\min}$ for performance evaluation. This transformation allowed the predicted values obtained through the sigmoid regression to be positioned within the original rating range of 1–5. The model performance was measured based on the transformed predicted ratings. The proposed MAMF jointly learns the parameters of the four networks and optimizes training. Stochastic gradient descent and backpropagation were used to optimize model training and minimize the error between the predicted and real preference ratings. Additionally, the parameters of the proposed model were fine-tuned using adaptive moment estimation (Adam) for minibatch optimization. The following measures were used to prevent the model from overfitting during the training process. First, dropout was applied to the MLP layers, and the dropout rate was fine-tuned. Second, if the validation loss does not decrease, the learning rate is reduced to further fine-tune the gradient. Third, early stopping was employed to terminate the training if the validation loss did not decrease over five epochs.

Table 1
Statistics of the amazon.com datasets.

Feature	Electronics	Gourmet Food	Video Games
Item	434,817	150,044	43,446
User	729,827	146,843	73,499
Brand	37,755	24,266	5395
Category	6631	1164	2581
Rating	6,866,682	1,340,943	715,277
Density (%)	0.002	0.006	0.023

5. Experiments

We conduct extensive experiments using real-world datasets from three categories collected from amazon.com to evaluate the recommendation performance of the proposed MAMF. We answer four research questions (RQs).

- **RQ 1**: Does the proposed MAMF perform better than the other baseline models?
- **RQ 2:** Does the training time and training loss of the MAMF model represent efficiency?
- **RQ 3**: How do the attention layer and different attribute components contribute to the recommendation performance of MAMF?
- **RQ 4**: How do different hyperparameters affect the recommendation performance of the proposed MAMF?

5.1. Datasets

This study used Electronics, Gourmet Food, and Video Games datasets from amazon.com[1] to validate the proposed MAMF model performance. The amazon.com datasets include numerous purchasing histories and auxiliary information such as user review text and item descriptions. Thus, it has been consistently utilized in recommender system studies [8,27]. Following the strategies used in previous studies, we preprocessed the datasets as follows. First, we removed stop words, whitespace, non-English characters, special characters, and numbers from the text data. Second, we convert the text data to lowercase data and apply tokenization and stop-word removal. Third, we stemmed each word and removed words that appeared three times or fewer, based on frequency, to eliminate noise in the review text. Fourth, we use data from users with a purchase history of at least five items. Table 1 summarizes the detailed statistics of the preprocessed datasets. In the experiments, each dataset was randomly divided into 70% training, 10% validation, and 20% test data.

5.2. Evaluation metrics

The evaluation metrics used to measure the model performance were the mean absolute error (MAE) and root mean square error (RMSE), widely used in recommendation tasks. Many studies have demonstrated that the MAE and RMSE are suitable for effectively measuring prediction accuracy in rating prediction [27]. As defined in Eq. (16), the MAE represents the average of the absolute differences between the actual and predicted ratings, providing intuitive results where all errors have equal weights, regardless of size. The RMSE is a metric used to assign higher weights to larger prediction errors and is calculated by the square root of the mean of the squared errors, as shown in Eq. (17). For both MAE and RMSE, lower values indicated a better model prediction performance.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|, \tag{16}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \tag{17}$$

where N represents the number of predicted ratings and y and $\hat{y}$ represent the actual and predicted ratings, respectively.

5.3. Baseline model

To ensure reliable validation of the proposed MAMF, its performances were compared with those of traditional recommendation models and various baseline models that use diverse types of auxiliary information.

- NCF [5]: A model that addresses the limitation of considering only linear relationships in the traditional MF by incorporating nonlinear learning using a deep learning framework. Thus, it can capture complex relationships between users and items.
- SAFMR [22]: A self-attention-based factorization model using CNN to extract features from review texts to represent each user and item. To consider the importance of various features in review texts, it applies a self-attention mechanism.
- ConvMF [8]: A model that uses item textual information through a CNN to extract item features, and incorporates the extracted features into a probabilistic MF model.
- UCAM [19]: A model that considers unstructured contextual information related to users along with user–item interaction information. It extends the NCF model and uses the extracted features in context vectors using an autoencoder.
- NCTR [24]: A model considering textual information and user–item interaction information for recommendations. It extends the MLP-based recommendation model by incorporating the item description information extracted through a CNN.
- FG-RS [21]: A self-attention-based model that estimates user preference by considering the interaction between all attributes of the user and item. It uses attention mechanisms to capture the user's more detailed preferences between all attributes.
- DFIAM [28]: An attention mechanism factorization model that integrates user, item, and interaction features to construct preference representation. Attention mechanisms are used to capture the importance between features, and user rating is predicted by preference representation.

5.4. Implementation details

To ensure a fair comparison of the overall recommendation models, we tested all of them using the same dataset and in the same environment (128.0 GB RAM and NVIDIA V100 GPU).

All hyperparameters for the proposed MAMF model were determined through experimentation and validation on the training dataset. The batch size was selected from [258, 512, 1024, 2048] and the learning rate was selected from [0.0005, 0.001, 0.005, 0.01]. The embedding size was optimized within [32, 64, 128, and 256]. The dropout rate was adjusted to [0.1, 0.2, 0.3, 0.4] to prevent model overfitting. After optimization, the batch size was set to 2048, the learning rate to 0.001, the embedding size to 64, and the dropout rate to 0.2. The number of convolutional kernels was 300, and the sliding window size was five. The number of epochs is set to 100. Early stopping was employed to prevent overfitting when the validation loss was not reduced for five epochs [14]. To reduce the experimental errors, the experiments were repeated five times, and the performance of each model was obtained by averaging the results.

For fairness in comparison, we empirically determine the optimal parameter settings for the baseline model. We train models on the training dataset, adjust hyperparameters on the validation dataset, and measure performance on the test dataset. We utilized grid search to

[1] https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews.

Table 2
Performance comparison to baseline models on the amazon.com dataset.

Model	Electronics		Gourmet Food		Video Games	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
NCF	0.779	1.094	0.669	0.947	0.716	1.019
SAFMR	0.774	1.092	0.615	0.905	0.693	0.999
ConvMF	0.769	1.093	0.590	0.898	0.663	0.998
NCTR	0.767	1.093	0.582	0.896	0.658	0.996
UCAM	0.748	1.078	0.565	0.892	0.632	0.985
FGRS	0.735	0.999	0.441	0.752	0.641	0.962
DFIAM	0.420	0.724	0.338	0.609	0.444	0.722
MAMF	0.409	0.698	0.313	0.594	0.430	0.714

Table 3
Training time comparison with baseline models on the amazon.com datasets.

Model	Electronics	Gourmet Food	Video Games
NCF	47.65[a]	25.30	26.77
ConvMF	296.66	118.76	95.09
SAFMR	4864.48	267.63	121.86
NCTR	164.27	112.79	144.23
UCAM	49.70	30.49	43.12
FG-RS	379.89	174.88	171.00
DFIAM	352.89	153.98	139.88
MAMF	422.64	174.47	152.84

[a] The values in the table represent the learning time required for one epoch in seconds.

optimize hyperparameters for each dataset on the validation dataset of all training datasets. The most common parameters include the dimension of the latent vector, learning rate, dropout ratio, and mini-batch size. For CNN-based models (i.e., SAFMR, ConvMF, and NCTR), the parameters of the CNN are set by referring to the author's original paper. This includes the number of kernels, sliding window size, and number of channels. For auxiliary information-based models (i.e., FG-RS and DFIAM), we use review text as user attributes and contextual information such as item description, category, and brand as item attributes, as per the strategy employed in this study. Some parameters not mentioned in the original paper use the same settings as this study.

6. Experimental results and discussion

6.1. Performance comparison to baseline models (RQ 1)

We use three datasets from amazon.com to compare the proposed MAMF recommendation performance with those of various baseline models. Table 2 summarizes the experimental results and shows that the proposed MAMF outperformed the baseline models for all datasets.

The following conclusions were drawn from the experimental results. First, the performance of the NCF model, which relies only on ratings, exhibits low performance compared with other baseline models that include auxiliary information. In other words, the recommendation method using only purchase history has limitations in precisely capturing the preferences of users.

Second, although the ConvMF and SAFMR models outperformed the NCF only learned by purchase history, they exhibited a lower performance than the other baseline models using auxiliary information. These results show that using the online review or item textual information can complement purchase history information from a recommendation perspective and achieve precise interaction learning.

Third, the NCTR and UCAM models performed better than the three models mentioned above. This is because the NCTR and UCAM models incorporate detailed items or user preference information that are valid for recommendation performance, thereby can capture complex user–item interactions. In other words, recommendation performance can be improved when considering the user's specific preference information for items.

Fourth, the FG-RS and DFIAM models perform better than other baseline models. Unlike other baseline models, these models can precisely estimate user preferences by containing various auxiliary information and considering the interaction between the auxiliary information. In addition, since the importance of the auxiliary information is considered by applying the attention mechanism, the user's detailed preference information for the item can be utilized.

Finally, the proposed MAMF exhibited superior performance compared to the baseline models. MAMF outperformed the baseline models for the following reasons: (1) The NCF model captures the interaction between the user and the item using a rating matrix as the only source of information. However, the proposed MAMF enhances the recommendation performance using various feature representations embedded in the auxiliary information. (2) Unlike ConvMF, SAFMR, NCTR, and UCAM, which utilize a single auxiliary information source, the proposed MAMF integrates multiple auxiliary information to understand better user preferences and detail aspects of items. (3) FG-RS and DFIAM capture the importance of each attribute but do not incorporate it into user and item representations, and they only interact with each attribute. On the other side, the proposed MAMF incorporates more representative and specific attention features from auxiliary information into the user and item representation. Thus, it enables more attentive interaction learning between the user and item compared to the baseline models.

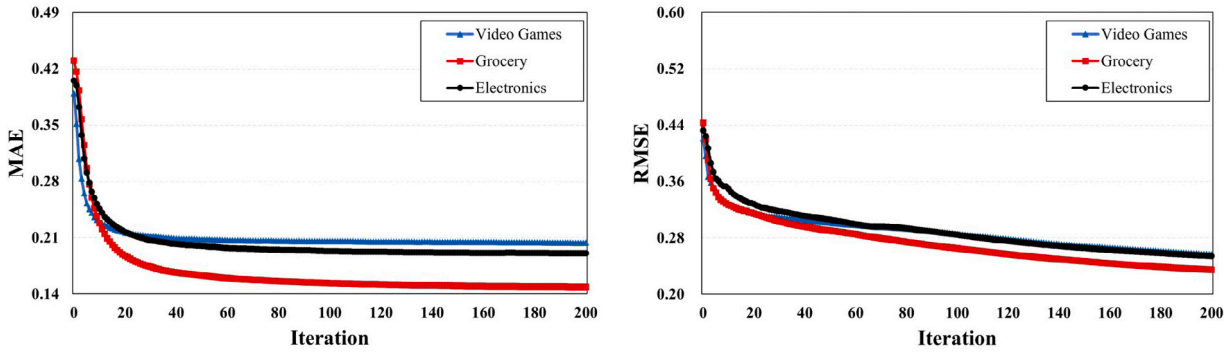
6.2. Training efficiency analysis (RQ 2)

The MAMF model extracts more representative and specific attention features embedded in the auxiliary information and incorporates them into the representation of users and items. However, the MAMF model contains a structure that can increase complexity compared to baseline models. To evaluate the model's efficiency, we compare the training time of each model over the entire training dataset for one epoch. Table 3 summarizes the training times of all models over three datasets. As shown in Table 3, it can be observed that larger datasets require more training time. The SAFMR model integrates review text, which is unstructured data, to generate user and item representations. A high computational cost is essential in utilizing a large amount of review text. Therefore, SAFMR requires the most computational time among all models. Compared to the other models, the proposed MAMF uses various input features and self-attention operations to represent user and item representation precisely, contributing to an increased training burden. As a result, MAMF requires more training time than other baseline models except for SAFMR. In other words, MAMF requires more complex computations and high time costs, but it can significantly improve the recommendation performance. In addition, in the real world, the prediction time for recommendations may be more critical than the training time. Therefore, the time cost of MAMF seems to be at an acceptable level.

The loss values of the proposed MAMF with respect to the number of iterations during the training process are shown in Fig. 3. In the training process, the loss values showed a limited range owing to the utilization of ratings transformed to a range of 0–1 and the sigmoid as the output function. As the number of iterations increased, MAMF achieved a sharply decreasing loss, and simultaneously, the MAE and RMSE converged to considerably low levels. This indicates that the training process of MAMF is stable and shows that MAMF is capable of effective interaction learning, which enables it to achieve a high level of recommendation performance.

6.3. Model components analysis (RQ 3)

The proposed MAMF utilizes a self-attention mechanism to consider the importance of each attribute embedded in auxiliary information. In the first additional study, we investigated whether the



(a) MAE according to the number of iterations (b) RMSE according to the number of iterations

Fig. 3. Loss value through MAE and RMSE with the increasing number of iterations.

Table 4
Performance comparison with and without an attention layer.

Model	Electronics		Gourmet Food		Video Games	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
MAMF	0.409	0.698	0.313	0.594	0.430	0.714
W/O attention layer	0.423	0.714	0.334	0.606	0.463	0.729
Percentage (%)	3.3	2.2	6.2	2.0	7.1	2.1

self-attention mechanism could effectively improve the recommendation performance of the proposed MAMF. The experimental results are listed in Table 4.

The experimental results showed that the self-attention mechanism led to performance improvements in all datasets, with an average improvement of 5.5% in MAE and 2.1% in RMSE. This indicates that considering the importance of each attribute can enhance the recommendation performance by using more representative and specific attention features of the user and item.

The second additional study compared the impact of different types of auxiliary information on the recommendation performance. MAMF uses a fusion of multiple embedded representations of various auxiliary information. To analyze the impact of each piece of auxiliary information on the recommendation performance, this experiment was conducted by removing each piece of auxiliary information; the results are shown in Table 5.

The experimental results showed that all auxiliary information significantly affected the recommendation performance, but its influence changed depending on the dataset. First, user reviews, which can include user preferences and item-related evaluation information, are the most influential features in all datasets. However, the impact of auxiliary information of the item aspect varies depending on the dataset. Categories, descriptions, and brands strongly influenced recommendation performance for Gourmet Food and Video Games. However, the impact of item descriptions was least effective in Electronics. In summary, various auxiliary information representing items can improve the recommendation performance, but their contribution can differ depending on the dataset.

6.4. Impact of hyperparameters (RQ 4)

In the proposed MAMF, the latent factor vector and feature representation vector are combined and fed into a self-attention operation. In this process, the embedding size of each feature is uniformly set. Namely, the embedding size indicates the representation level of each feature and simultaneously affects the number of attention weights generated by the self-attention operation. The embedding size is a crucial hyperparameter that influences the representation level of the features and self-attention operation in the proposed MAMF. Therefore, this process is essential for determining an appropriate embedding size. In the first hyperparameter test, various embedding sizes are used to compare the recommendation performance of the proposed MAMF.

Table 6 lists the performance of the proposed MAMF for different embedding sizes. The experimental results show that the optimal embedding size is 64 for the Video Games and Gourmet Food datasets and 128 for the electronic dataset. This indicates that when the embedding size is too small, it may be insufficient to completely represent the various attributes. Simultaneously, a lack of representation level can also be unsuitable for self-attention operations [11]. However, if the embedding size is excessively large, overfitting can occur in feature representation, and it can cause decreasing performance. This indicates the need to determine the optimal embedding size based on the dataset type and recommendation model.

The second experiment compared the performance according to the number of hidden layers used in the interaction learning. The proposed MAMF combines user and item attentive representation vectors and feeds them into the hidden layer for rating prediction. During this process, a smaller depth can lead to underfitting problems that cannot provide sufficient learning. However, a large depth can lead to unnecessary overfitting, decreasing the recommendation performance. Therefore, it is important to select an appropriate hidden layer depth for interaction learning. As shown in Table 7, experiments were conducted at various hidden layer depths to investigate this.

The experimental results show a consistent pattern in recommendation performance across all three datasets. In the learning of the interaction between the representation vectors incorporating various attributes, the optimal number of hidden layers was three. This indicates that a lower number of hidden layers hinders the provision of sufficient computation for optimal performance. On the other hand, more than four hidden layers can lead to overfitting and high computational costs, resulting in no additional benefits.

7. Conclusion and future studies

Several studies have proposed the use of diverse auxiliary information to address the data sparsity problem in recommender systems. E-commerce platforms offer various types of auxiliary information such as user reviews and item categories, which can help enhance recommendation performance. From a recommender system perspective, it is essential to consider the importance of each attribute embedded in the auxiliary information in the representation of users and items. Nevertheless, many previous studies have employed a simple combined-embedding approach that does not consider the importance of each attribute, which can be insufficient to effectively represent items or users. To overcome this limitation, we propose a novel recommendation model that combines various types of auxiliary information and considers the importance of each attribute through a self-attention

Table 5
Performance comparison with removal of each auxiliary information.

Dataset	Input attributes	MAE (%)[a]	RMSE (%)	Attributes' importance rank
Electronics	W/O review text	0.746 (45.2%)	1.089 (35.9%)	1
	W/O item category	0.458 (10.7%)	0.730 (4.4%)	2
	W/O item description	0.410 (0.2%)	0.702 (0.6%)	4
	W/O brand	0.428 (4.4%)	0.703 (0.7%)	3
Gourmet Food	W/O review text	0.623 (49.3%)	0.936 (36.5%)	1
	W/O item category	0.351 (10.0%)	0.615 (3.4%)	2
	W/O item description	0.350 (9.7%)	0.610 (2.6%)	3
	W/O brand	0.347 (8.9%)	0.605 (1.8%)	4
Video Games	W/O review text	0.737 (41.7%)	1.032 (30.8%)	1
	W/O item category	0.447 (3.8%)	0.726 (1.7%)	2
	W/O item description	0.437 (1.6%)	0.720 (0.8%)	3
	W/O brand	0.434 (0.9%)	0.719 (0.7%)	4

[a] The percentage represents the rate of performance degradation when information is removed.

Table 6
Performance comparison according to different feature embedding sizes.

Size	Electronics		Gourmet Food		Video Games	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
32	0.426	0.704	0.316	0.597	0.438	0.719
64	0.414	0.702	0.313	0.594	0.430	0.714
128	0.409	0.698	0.332	0.596	0.437	0.714
256	0.415	0.701	0.370	0.621	0.452	0.722

Table 7
Performance comparison according to the hidden layer depth.

Depth	Electronics		Gourmet Food		Video Games	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
0	0.430	0.716	0.344	0.612	0.466	0.723
1	0.429	0.716	0.319	0.599	0.449	0.721
2	0.420	0.698	0.319	0.599	0.433	0.714
3	0.409	0.698	0.316	0.594	0.430	0.713
4	0.440	0.701	0.343	0.618	0.433	0.720

mechanism. The superiority of the proposed MAMF was confirmed through comparative experiments with baseline models.

The limitations of this study and future study directions are as follows. First, further discussion on the fusion function between various types of information is required. This study combines auxiliary information through concatenation to specifically represent users and items. However, various methods exist for integrating information, such as addition and gating. Therefore, further study is required to consider the various strategies for feature fusion. Second, to expand the scope of this study, it is necessary to validate the proposed model using datasets from other domains. This study specifically utilized amazon.com dataset from an e-commerce domain, although recommender systems are widely applicable in various fields. Third, performance improvements can be made using a pre-trained text embedding method in the proposed model. This study used textual information, such as item descriptions and user reviews, and a basic embedding model was applied to convert text into dense vectors. However, pre-trained embedding models have been effective in various field, particularly in achieving high performance in natural language processing. Therefore, there is an opportunity to enhance performance improvement by applying pre-trained models, such as FastText or bidirectional encoder representations from transformers.

CRediT authorship contribution statement

Dongsoo Jang: Writing – original draft, Software, Methodology, Conceptualization. **Qinglong Li:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Chaeyoung Lee:** Software, Data curation. **Jaekyeong Kim:** Writing – review & editing, Supervision.

Declaration of competing interest

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

Data availability

I have shared the link to my data.

Acknowledgments

This research was supported by the BK21 FOUR (Fostering Outstanding Universities for Research) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

References

[1] J. Park, X. Li, Q. Li, J. Kim, Impact on recommendation performance of online review helpfulness and consistency, Data Technol. Appl. 57 (2) (2023) 199–221.
[2] Y. Liu, S. Wang, M.S. Khan, J. He, A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering, Big Data Min. Anal. 1 (3) (2018) 211–221.
[3] F. Strub, R. Gaudel, J. Mary, Hybrid recommender system based on autoencoders, in: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, 2016, pp. 11–16.
[4] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 173–182.
[6] Z.-H. Deng, L. Huang, C.-D. Wang, J.-H. Lai, S.Y. Philip, Deepcf: A unified framework of representation learning and matching function learning in recommender system, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 61–68.
[7] Z. Zhang, Y. Liu, Z. Zhang, Field-aware matrix factorization for recommender systems, IEEE Access 6 (2017) 45690–45698.
[8] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, in: Proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 233–240.
[9] T.X. Tuan, T.M. Phuong, 3D convolutional networks for session-based recommendation with content features, in: Proceedings of the 11th ACM Conference on Recommender Systems, 2017, pp. 138–146.
[10] Y. Wang, X. Lu, Y. Tan, Impact of product attributes on customer satisfaction: An analysis of online reviews for washing machines, Electron. Commer. Res. Appl. 29 (2018) 1–11.
[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2017, pp. 5998–6008.
[12] J. Kong, H. Wang, C. Yang, X. Jin, M. Zuo, X. Zhang, A spatial feature-enhanced attention neural network with high-order pooling representation for application in pest and disease recognition, Agriculture 12 (4) (2022) 500.
[13] Z. Niu, G. Zhong, H. Yu, A review on the attention mechanism of deep learning, Neurocomputing 452 (2021) 48–62.

[14] G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, J. Mach. Learn. Res. 10 (2009) 623–656.
[15] P. Symeonidis, A. Nanopoulos, Y. Manolopoulos, MoviExplain: a recommender system with explanations, in: Proceedings of the Third ACM Conference on Recommender Systems, 2009, pp. 317–320.
[16] A. Mnih, R.R. Salakhutdinov, Probabilistic matrix factorization, in: Proceedings of the 20th International Conference on Neural Information Processing Systems, 2007, pp. 1257–1264.
[17] Y. Bao, H. Fang, J. Zhang, Topicmf: Simultaneously exploiting ratings and reviews for recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2014, pp. 2–8.
[18] T. Liu, Z. Wang, J. Tang, S. Yang, G.Y. Huang, Z. Liu, Recommender systems with heterogeneous side information, in: The World Wide Web Conference, 2019, pp. 3027–3033.
[19] M. Unger, A. Tuzhilin, A. Livne, Context-aware recommendations based on deep learning frameworks, ACM Trans. Manage. Inf. Syst. (TMIS) 11 (2) (2020) 1–15.
[20] Y. Lv, Y. Zheng, F. Wei, C. Wang, C. Wang, AICF: Attention-based item collaborative filtering, Adv. Eng. Inform. 44 (2020) 101090.
[21] H. Chen, F. Qian, J. Chen, S. Zhao, Y. Zhang, FG-RS: Capture user fine-grained preferences through attribute information for recommender systems, Neurocomputing 458 (2021) 195–203.
[22] H. Ma, Q. Liu, In-depth recommendation model based on self-attention factorization, KSII Trans Internet Formation Syst. 17 (3) (2023) 721–739.
[23] A.K. Yengikand, M. Meghdadi, S. Ahmadian, DHSIRS: a novel deep hybrid side information-based recommender system, Multimedia Tools Appl. 82 (2023) 34513–34539.
[24] D. Liu, J. Li, B. Du, J. Chang, R. Gao, Y. Wu, A hybrid neural network approach to combine textual information and rating information for item recommendation, Knowl. Inf. Syst. 63 (2021) 621–646.
[25] W. Chen, F. Cai, H. Chen, M.D. Rijke, Joint neural collaborative filtering for recommender systems, ACM Trans. Inf. Syst. (TOIS) 37 (4) (2019) 1–30.
[26] K. Rama, P. Kumar, B. Bhasker, Deep autoencoders for feature learning with embeddings for recommendations: a novel recommender system solution, Neural Comput. Appl. 33 (2021) 14167–14177.
[27] Q. Li, X. Li, B. Lee, J. Kim, A hybrid CNN-based review helpfulness filtering model for improving e-commerce recommendation service, Appl. Sci. 11 (18) (2021) 8613.
[28] Y. Zhou, X. Shen, S. Zhang, D. Yu, G. Xu, DFIAM: deep factorization integrated attention mechanism for smart TV recommendation, World Wide Web 24 (2021) 1465–1481.