Enhancing Recommender Systems by Fusing Diverse Information Sources through Data Transformation and Feature Selection

Thi-Linh Ho[1*], Anh-Cuong Le[1], and Dinh-Hong Vu[1]
[1] Natural Language Processing and Knowledge Discovery Laboratory,
Faculty of Information and Technology, Ton Duc Thang University,
District 7, Ho Chi Minh City, Vietnam
[e-mail: hothilinh.st@tdtu.edu.vn, leanhcuong@tdtu.edu.vn, vudinhhong@tdtu.edu.vn]
[*]Corresponding author: Thi-Linh Ho

Received March 22, 2023; revised April 26, 2023; accepted May 3, 2023; published May 31, 2023

Abstract

Recommender systems aim to recommend items to users by taking into account their probable interests. This study focuses on creating a model that utilizes multiple sources of information about users and items by employing a multimodality approach. The study addresses the task of how to gather information from different sources (modalities) and transform them into a uniform format, resulting in a multi-modal feature description for users and items. This work also aims to transform and represent the features extracted from different modalities so that the information is in a compatible format for integration and contains important, useful information for the prediction model. To achieve this goal, we propose a novel multi-modal recommendation model, which involves extracting latent features of users and items from a utility matrix using matrix factorization techniques. Various transformation techniques are utilized to extract features from other sources of information such as user reviews, item descriptions, and item categories. We also proposed the use of Principal Component Analysis (PCA) and Feature Selection techniques to reduce the data dimension and extract important features as well as remove noisy features to increase the accuracy of the model. We conducted several different experimental models based on different subsets of modalities on the MovieLens and Amazon sub-category datasets. According to the experimental results, the proposed model significantly enhances the accuracy of recommendations when compared to SVD, which is acknowledged as one of the most effective models for recommender systems. Specifically, the proposed model reduces the RMSE by a range of 4.8% to 21.43% and increases the Precision by a range of 2.07% to 26.49% for the Amazon datasets. Similarly, for the MovieLens dataset, the proposed model reduces the RMSE by 45.61% and increases the Precision by 14.06%. Additionally, the experimental results on both datasets demonstrate that combining information from multiple modalities in the proposed model leads to superior outcomes compared to relying on a single type of information.

http://doi.org/10.3837/tiis.2023.05.006 **ISSN : 1976-7277**

Keywords: Recommender systems, data transformation, multi-modal fusion, recommendation model, deep neural network recommender systems.

1. Introduction

Recommender systems (RS) are designed to determine the preferences of users for given products or services. These systems aim to help products reach potential customers, making them crucial for commerce. Therefore, developing recommendation methods with high accuracy is an interesting and important problem that has attracted a lot of research.

Similar to other studies, we refer to products, services, or anything that users express interest in as "items". The goal is to determine the degree of interest for user-item pairs. Most studies approach the RS problem as a machine learning problem, where a generalized model is built based on known information of users and items to generate the interest degree for an unknown user-item pair.

Traditionally, the RS problem is modeled by representing user preferences for items in a matrix, where rows represent users and columns represent items. Each element at position (i, j) in the matrix contains a real number, typically the rating of user i for item j. **Fig. 1** illustrates an example of the user-item rating matrix, also known as the utility matrix. Note that the values in the matrix are typically integers ranging from 1 to 5. The RS task involves building a model from the known values in the utility matrix to predict values for the empty cells.

	Item1	Item2	Item3	Item4	Item5	…
User1	4	5	3	1		
User2	4					
User3		3	5	2	4	
User4		5			1	
User5	5			2		
…						

Fig. 1. A visual representation of a utility matrix

There are two primary techniques for developing recommendation systems (RS) known as collaborative filtering (CF) and content-based filtering. When trying to predict the rating (r) of an item (i) by a user (u), the CF method utilizes the ratings of users similar to user *u* who have previously rated the item *i*. Two well-known methods used in this approach are Neighborhood-based collaborative Filtering [5, 6] and Matrix Factorization [8, 29-32], which can be constructed using only the information contained in the utility matrix. On the other hand, the content-based approach in such studies [6, 36-39] is based on the simple observation that if a person enjoys item i, they are likely to enjoy similar products. The similarity between items can be calculated by referencing the utility matrix or item profiles found in other sources. Content-based is limited as it only predicts ratings for similar items. Collaborative filtering overcomes this limitation by using the collaboration mechanism to detect a user's preference for an item based on similar users who have rated the item. However, content-based works well when predicting ratings for new items that are similar to past items, especially when using more product description information. Recent studies [6, 7, 12-15, 26] combine both these

approaches using an ensemble learning approach, which combines different individual models to create a combined model that is stronger than the individual models.

There are many sources of information beyond the utility matrix that are useful for recommending items to users, such as the user's profile, the item's description, and reviews about the item. There are also many different methods of data representation and machine learning models that can be used to achieve the goal of recommending items. In general, there are different modalities available for a pair of (u, i) to determine how interested the user *u* may be in the item *i*. Recently, there have been studies following a multimodality approach to the RS problem, such as [11, 16, 18, 28].

Our study also addresses this issue by using and integrating many different sources of information, which we consider as the multimodality of data, to build a unified model for recommendations. Our objective is to utilize diverse information sources including the utility matrix (the rating matrix between user-items), user profiles, item contents, and user reviews for items. We propose a new combined model for solving two problems: first, how to build models based on these sources to extract as much useful information as possible and represent it effectively, and second, how to combine this information in a way that complements and integrates it together to predict ratings. We first use matrix factorization to find latent feature vectors of users and items. Next, we apply various transformation techniques to these vectors, as well as feature vectors of users and items to convert them into a common form. For other modalities such as item's reviews, item's descriptions, and item's categories, we propose representations based on natural language processing techniques. We also apply a method of feature selection for selecting the best feature subset as well as removing noisy features. We then fuse these vectors using a neural network to create a multi-modal feature descriptor that can be applied to recommender systems. The goal is to enrich the information provided to recommender systems and enhance their performance.

Our main contributions include:

(1) Proposing a multi-modal transformation and fusion model that uses a deep neural network to fuse different kinds of features from various sources, including:

- The latent features of users and items, which are obtained using a matrix factorization technique and then converted into lower-dimensional vectors using PCA.
- The item features are extracted from the item dataset and user features using various transformation techniques (such as TF-IDF, LSTM, and BERT), which are transformed into lower-dimensional vectors using PCA.

(2) Conducting experiments to illustrate how our proposed multi-modality transformation and fusion model improves the performance of recommendation models.

The remainder of the paper is organized with a structure as follows: Section 2 summarizes recent studies relevant to our work. Section 3 outlines the conceptual and technical background for matrix factorization and feature extraction. In Section 4, we present our proposed multimodal fusion model with corresponding procedures and algorithms. Section 5 shows the experimental results, comparisons, and discussions. Lastly, Section 6 concludes our contribution.

2. Literature Review

Studies in recommender systems have explored various approaches to giving users useful recommendations. There are two commonly used approaches including content based filtering

and collaborative filtering. For instance, Reddy et al. [36] used genre correlation to build a content-based filtering RS that recommends movies based on user preferences. Wahyudi et al. [37] combined hotel categories and city features to recommend the highest rated hotel to users. Singla [38] introduced a movie recommendation framework that finds similarity between movies using publicly available features like plot, rating, country of production, and release year. Ghauth and Abdullah [39] proposed a framework that recommends learning materials based on content similarity and user ratings. In predicting user preferences, Zhao and Shen [40] conducted an empirical study of movie ratings and proposed a preference model that eliminates impracticable predictions.

Deep learning-based approaches have also been explored, with Ahmed [41] proposing a Bi-GRU model architecture for polarity prediction and review rating prediction. For analyzing reviews and predicting user ratings, Gezici [42] used recurrent neural networks. Yang et al. [31] designed a federated collective matrix factorization algorithm that protects user privacy and accurately predicts user preferences. Wang et al. [29] introduced a Visual Recurrent Convolutional Matrix Factorization (VRConvMF) model that uses descriptive texts and posters to extract textual and visual features. Zhang et al. [32] proposed a FeatureMF model that incorporates item features into the matrix factorization framework to enrich item representation. Finally, Xue et al. [30] proposed a new method based on deep matrix factorization using both explicit and implicit information.

Recommendation research has recently seen an increase in studies focused on multimodality-based approaches. In one study, Pádua et al. [16] utilized bag-of-words and TF-IDF models to describe textual data and combined them with descriptors extracted from visual data using a deep convolutional network. They then used autoencoders to create a low-dimensional sparse representation of each video document resulting from the multimodal fusion. In another study, Feng et al. [5] integrated users' local relations and global ratings to improve prediction accuracy and robustness in sparse data. Wang et al. [28] proposed a Fine-grained Multimodal Fusion Network (FMFN) which combines textual and visual factors for detecting fake news. Finally, a deep course recommendation model for multimodal fusion which used the course video, audio, title, and introduction was proposed by Wang and colleagues [18], with multimodal features and using LSTM with attention mechanism. Lei, F. et al. [43] introduced a novel module called Learning the User's Deeper Preferences (LUDP), which employs multi-modal features to create an item-item similarity graph by propagating and aggregating item ID embeddings. Furthermore, a user preference graph is constructed using historical interactions, where multi-modal features are aggregated to represent the user's preferences, and the combination of these graphs enhances user and item representations for a deeper understanding of user preferences through collaborative signals. Mu, Y., & Wu, Y. [44] proposed a personalized multimodal movie RS that employed deep learning algorithms for mining hidden features of movies and users. The extracted features were then used to build a deep-learning network model that predicted movie scores based on input information. An approach proposed by Ren, X. et al. [18] for course recommendation involves utilizing a deep model that combines multimodal feature extraction through Long- and Short-Term Memory (LSTM) networks and Attention mechanism. This model incorporates video, audio, title, and introduction data from courses for multimodal fusion. Additionally, user demographic information, as well as explicit and implicit feedback data, are included to create a comprehensive learner profile.

Recent studies on content-based, user-based, and item-based collaborative filtering models have generally relied on using only item content similarity or similarities among users with similar preferences or items rated by the same user to suggest items to users. For the matrix

factorization-based RS, they mostly use the formula to estimate users' level of preference for items from users' and items' latent feature vectors which are extracted from a utility matrix. Researchers have used neural networks to fuse item feature vectors or/and user rating vectors, then the results have been applied to RSs. Studies conducted recently on multimodal approaches to RSs have refrained from integrating information from both the utility matrix and textual sources, such as item categories, descriptions, and user reviews, due to potential issues of interference and diminished accuracy.

Our research adopts a multimodal approach, which leverages multiple modalities to enhance the strength of input information. We employ various techniques to extract and represent information effectively, based on the unique characteristics of each modality. Instead of using the original feature vectors of users and items, our proposed model utilizes reduced-dimension user and item latent features and feature vectors obtained through PCA. This approach not only helps reduce the dimensionality of the data but also lowers the computational costs of our model. We apply the best k feature selection algorithm to filter out noise for the feature vectors before integrating them, and then feed them as input to a Feedforward Neural Network model for rating prediction. Our proposed model has demonstrated better recommendation performance when compared to both unimodal recommendation models and SVD RSs.

3. Foundational Theory and Feature Extraction

3.1 Matrix Factorization

Matrix factorization is the most effective method used in recommendation systems. The basic idea behind matrix factorization is to decompose a user-item interaction matrix (i.e. utility matrix) into two low-rank matrices representing users and items, respectively. The objective of the matrix factorization technique is to learn latent features (factors) that describe the preferences of users and the attributes of items.

Matrix factorization can handle the sparsity and scalability issues commonly found in recommender systems, where the number of users and items is large and the user-item interaction data is often incomplete. Moreover, matrix factorization can be extended to handle additional information, such as user demographics and item attributes, which can improve the quality of recommendations.

3.1.1 The concept of Matrix factorization:

According to [10] the Matrix Factorization method, using the utility matrix, is responsible for identifying latent features that represent users and items. Suppose that we are given the utility matrix presentating ratings of n users on m items, which is presented as a rating matrix with shape ($n \times m$). The Matrix Factorization method aims to decompose R into matrices P and Q which are considered as thin matrices with shape $n \times f$ and $m \times f$ respectively. Note that f is the number of latent factors which contain important latent information. **Fig. 2** illustrates how the matrix R can be decomposed into matrices P and Q.

$$R \approx P \cdot Q^T \tag{1}$$

The formula to predict each rating r of the pair (user u, item i) is represented as follows:

$$\hat{r}_{u,i} = p_u \cdot q_i^T \tag{2}$$

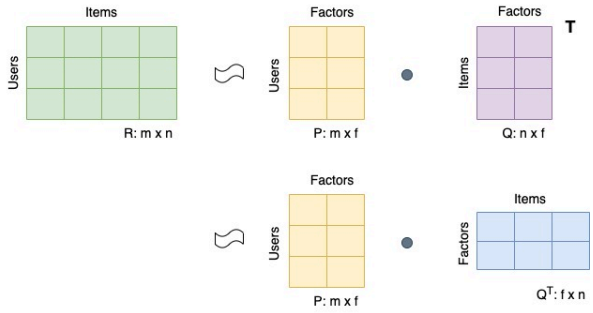

Fig. 2. Decomposition of utility matrix into latent factor matrices

where p_u stands for the user vector and q_i stands for the item vector, both of them have f dimensions. The product of p_u and q_i^T expresses the preference of users on items.

3.1.2 Singular value decomposition (SVD):

SVD is an algorithm that can be helpful in building recommender systems. It decomposes a matrix R into a lower-rank approximation of the original matrix R that is simpler to work with. This involves breaking R down into two unitary matrices and a diagonal matrix mathematically, as follows:

$$R = U\sum V^T \tag{3}$$

Where R (with shape $m \times n$) stands for the utility matrix; the matrix U (with shape $m \times r$) stands for singular matrix of the orthogonal left side presenting users and latent feature relationship; the matrix Σ (with shape $r \times r$) stands for diagonal matrix describing the strength for latent features; and the matrix V (with shape $r \times n$) stands for the singular matrix of diagonal right side indicating the how similar of (items, latent features/factors). The goal is to extract latent factors by decreasing the dimension of R.

Let each item be represented by a vector x_i and each user is represented by a vector y_u. The expected rating by a user on an item $\hat{r}_{ui}$ can be calculated by the formula in (4):

$$\hat{r}_{ui} = x_i^T \cdot y_u \tag{4}$$

3.2 Principal Component Analysis (PCA):

PCA [25] is a dimensionality reduction technique that is commonly used in data analysis and machine learning. PCA helps to identify patterns in high-dimensional data by reducing the number of variables while retaining as much information as possible. It is useful for reducing computational complexity, identifying important features or variables, and filtering out noise and irrelevant information from high-dimensional data.

PCA transforms correlated variables into a set of nonlinearly correlated variables which are called principal components (PCs). This transformation determines that the principal components represent the most variation in the data. According to [24] the procedure includes the following steps:

1. Calculate the covariance matrix from the sample data.
2. Compute the eigenvalues and the eigenvectors of a covariance matrix.
3. Generate the PCs by the calculation:

$$p_i = b_{i1}X_1 + b_{i2}X_2 + \ldots + b_{ik}X_k \quad (5)$$

where P_i stands for the i^{th} PC, b_{ik} stands for the weight and X_k stands for the variable.

3.3 Feature Extraction and Representation for Natural Language Texts

Among the information sources used for recommendations, user reviews, item descriptions, and item categories are in natural language text, therefore extracting features from these representations is an important task in our work. We use TF-IDF and Word Embeddings to represent words, where word embeddings are generated from the BERT model [34].

3.3.1 Term Frequency - Inverse Document Frequency:

TF-IDF is a method of measuring the importance of terms in relation to the classification of documents, or measuring the importance of terms in relation to the content of a document. TF-IDF is also a form of the Bag of Words model [4], and it is based on the calculation of term frequency (TF) and \ inverse document frequency (IDF). The formula for TF could be as follows:

$$tf_{t,d} = \frac{n_{t,d}}{\sum_k n_{k,d}} \quad (6)$$

Where $n_{t,d}$ denotes the number of times term t occurs in document d, and $n_{k,d}$ denotes the number of occurrences of every term in document d.

The IDF is calculated as the following formula:

$$idf_t = \log\left(\frac{|D|}{D_t}\right) \quad (7)$$

Where $|D|$ denotes the total number of documents, and $|D_t|$ denotes the number of documents where the term t appears.

3.3.2 Bidirectional Encoder Representations from Transformers (BERT):

BERT [34] is a pre-trained language model which is widely used for many tasks in natural language processing. BERT is highly effective in representing and encoding textual information because it is based on the Transformer architecture, which utilizes a multi-head attention mechanism. Text is first represented by a sequence of tokens and is represented initialized as one-hot vectors. Then it goes through the layers of the Transformer which involves computing the attention of the words with the words around it to generate the word

representation in the next layers. In the BERT model, additional information about position is used. BERT is then learned by masking some words (using MASK) at the input and predicting them in the output layer.

Using BERT as a preprocessing tool has enabled many NLP tasks to achieve state of the art results (at the time BERT was born). In this work, we also use BERT for representing information like review, description. The following basic architecture of BERT is used by us: $BERT_{BASE}$ (L=12, H=768, A=12, Total Parameters=110M) and $BERT_{LARGE}$ (L=24, H=1024, A=16, Total Parameters=340M). In our study, we will use $BERT_{BASE}$ for generating vectors of words.

4. The Proposed Model for Multimodal Fusion

4.1 Our Method for Multimodal Fusion

Multimodal fusion refers to the process of integrating information from multiple sources or modalities such as images, text, audio, video, and other forms of data. The goal of multimodal fusion is to create a more complete and accurate representation of a given situation or event by combining information from different sources. In the context of machine learning and artificial intelligence, multimodal fusion techniques are used to improve the performance of models that deal with complex and heterogeneous data. By combining information from different modalities, these models can better understand the underlying patterns and relationships in the data, and make more accurate predictions or decisions.

Early fusion and late fusion are two common approaches for multimodal fusion. Early fusion, also known as a feature-level approach, involves combining the features or representations of different modalities at the input level. This means that the raw data from each modality is pre-processed and transformed into a common feature space before being fed into the fusion model. Early fusion is typically used when the modalities are highly correlated and provide complementary information that can be easily combined. In contrast, late fusion, also known as a decision-level approach, involves fusing the outputs of individual models that have been trained on each modality separately. In this approach, each modality is processed separately using a specific model, and the outputs are then combined using a fusion mechanism. Late fusion is typically used when the modalities are not highly correlated and provide unique and independent information that cannot be easily combined.

Our proposed model belongs to the early fusion type. The architecture of our proposed multimodal model for the recommendation problem is illustrated in an overview through **Fig. 3.** It includes two main components: the first is modules for extracting and representing features from various information sources of users and items, and the second is a module for integrating these features for the prediction task.

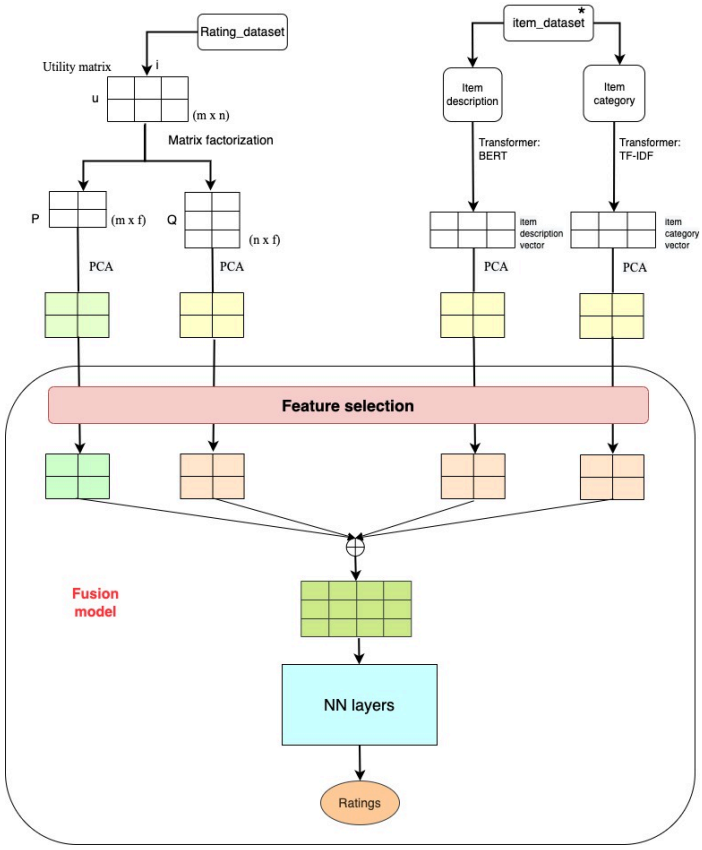

Fig. 3. The proposed transformation and fusion model for recommender system

The general scheme for our recommendation algorithm based on a multimodality approach is:

1) Training stage:
   - Input: given a data set containing multi-modality of a set of users and items.
   - Output: Model for rating
   - Step 1: Building the Feature Extraction Model (named as FEM)
     - Doing the task: Feature extraction and transformation for each modality
     - Return vector of features for each modality. Suppose that we obtain a set of *k* feature vectors denoted by $Vu = \{vu_1, \ldots, vu_k\}$ for a given user, and a set of *j* feature vectors denoted by $Vi = \{vi_1, \ldots, vi_j\}$ for a given item.
   - Step 2: Building the Predicting Fusion Model (named as PFM)
     - Combining feature vectors from Step 1 and feed them all into a Regression model for the task of rating generation (i.e. prediction)
     - Training prediction model
2) Inference stage:
   - Given a pair (user *u*, item *i*)
   - Step 1: Using FEM to return feature vectors for user *u* and feature vectors for item *i*. They include: $Vu = \{vu_1, \ldots, vu_k\}$ and $Vi = \{vi_1, \ldots, vi_j\}$

- Step 2: Combining Vu and Vi then use the result as input features for the PFM for generating the output (i.e. the rating).

4.2 The Feature Extraction Model (FEM)

The component for feature extraction and representation plays an essential role in our proposed model. We construct different kinds of features for representing users and items from four different sources of information, including the utility matrix, user reviews, item descriptions, and item categories as follows:

Rating-Feature: From the given utility matrix, we apply the matrix factorization technique to decompose the matrix into two lower rank matrices that represent the user and item latent features. In particular, from the utility matrix of m users and n items (i.e. the utility matrix has the size of (m x n)) we obtained:

- The matrix (m x f) represents user features, meaning that each user is represented by a vector of f features. Each row in this matrix corresponds to a particular user, which is considered as a modality of this user.
- The matrix (n x f) represents item features, meaning that each item is represented by a vector of f features. Each row in this matrix corresponds to a particular item, which is considered as a modality of this item.

Review-Feature: Given a (user u, item i) pair for which there exists a review written by user u for item i, we use a pre-trained BERT model to generate a vector representation of this text review. We then consider this vector as a modality of user u.

Description-Feature: Similarly, we are given a description of item i, which is also in text form, and we use a pre-trained BERT model to generate a vector representation of this description. We consider this vector as a modality for item i.

Category-Feature: The category of item i contains a list of categories to which the item belongs. Therefore, instead of using BERT, we will use a TF-IDF measure to convert these category features into a vector representation. We also consider this vector as a modality for item i.

In summary, given a pair of (user u, item i), we will generate two feature vectors for user u (considered as two modalities of user u) and three feature vectors for item i (considered as three modalities of item i). These vectors of features will be passed through the PCA algorithm to obtain the final feature representations.

We utilize the PCA technique to reduce the dimensionality of the vectors and extract important representative features. Additionally, we employ an algorithm to identify the best k features for each vector. We will put all these feature vectors to the fusion component for rate prediction of (u, i).

In summary, the algorithm for extracting and transforming to obtain feature vectors of a given user u và item i includes the following steps:

Procedure FEM:

Step 1: Extracting feature kinds including:

- Rating-Feature for user u and item i, denoted by Rat(u) and Rat(i) respectively
- Review-Feature for user u, denoted by Rev(u)
- Description-Feature for item i, denoted by Des(i)
- Category-Feature for item i, denoted by Cat(i)

Step 2: Using PCA to transform feature vectors obtained from Step 1 into the same low dimensional vectors. As the result we obtain feature vectors for user u and item i as follows:

- For user *u*: transform vectors Rat(u) and Rev(u) we obtain corresponding vectors fu_1 and fu_2.
- For item *i*: transform vector Rat(i), Des(i) and Cat(i) we obtain corresponding vectors fi_1, fi_2, fi_3 .

Step 3: Using the *SelectKBest* algorithm from the *sklearn* library to select a subset with *k* best features from transformed feature vectors obtained after Step 2.

- In our proposed model, the SelectKBest algorithm is used to select the top K features in the dataset based on their statistical significance and the scoring function is mutual information. This approach can help improve the accuracy and efficiency of machine learning models and remove irrelevant features.
- We create an instance of the SelectKBest class, specify the scoring function and K, fit the algorithm to our data, and transform it to obtain the selected features.

4.3 The Predicting Fusion Model (PFM)

After identifying and extracting the features of the modalities, we will use a neural network to integrate these information sources to generate a rating that shows the interest of user *u* in item *i*. The algorithm for performing this task is as follows:

Procedure PFM:

Step 1: For each pair (user *u*, item *i*), we put it through the module FEM and get results which are different kinds of features presented as corresponding feature vectors and a subset of selected features:

Feature vectors for user *u*: fu_1, fu_2

Feature vectors for item *i*: fi_1, fi_2, fi_3

Subset of selected features: *fs*

Step 2: Transforming data

After identifying a subset of selected features using Step 1, we fit the algorithm to our data and transform it to obtain the selected features as follows:

- Feature vectors for user *u*: fu_1 and fu_2 are transformed into fu_1_fs and fu_2_fs, respectively.
- Feature vectors for user *i:* fi_1, fi_2, and fi_3 are transformed into fi_1_fs, fi_2_fs, and fi_3_fs, respectively.

Step 3: Combining features

Concatenate fu_1_fs, fu_2_fs, fi_1_fs, fi_2_fs, fi_3_fs and we obtain the final representation of the input:

F = Concat(fu_1_fs, fu_2_fs, fi_1_fs, fi_2_fs, fi_3_fs)

Step 4: Building prediction model

- Designing a Multi-layers Perceptron model for prediction task

- For the training data set containing pairs (user u, item i). Using Step 1, Step 2 and Step 3 to convert them to corresponding feature vectors.
- Training the prediction model

Step 5: Inferencing

- Given an input (user *u*, item *i*) we use Step 1, Step 2 and Step 3 to get corresponding feature vector *F*.
- Input the feature vector F to the prediction model for inferencing the rating for the pair (user *u*, item *i*)

5. Experiments and Results

5.1 The summary of datasets

To demonstrate the effectiveness of the proposed model, we conducted experiments on the MovieLens and the Amazon sub-category datasets. We preprocessed the datasets to select users and items that met our predefined filter criteria. Specifically, for the MovieLens dataset, we selected ratings of items that were rated by 100 or more users and then filtered out the ratings of users who rated 200 or more items.

For the Amazon - Electronics subdataset, we selected ratings of items rated by 100 or more users and filtered out the ratings of users who rated 58 or more items. For the Amazon Toys and Games sub-dataset, we selected the ratings of items that were rated by 30 or more users, and then we filtered out the ratings of users who rated 15 or more items. Lastly, for the Amazon Video Games sub-dataset, we selected the ratings of users who rated 25 or more items. Further details of the datasets are described below.

Amazon dataset: The Amazon dataset used in our experiment is available at https://jmcauley.ucsd.edu/data/amazon/. We use sub-datasets including Electronic, Video Games, and Toys and Games. The two files that we used in the experiments include: (1) The rating file containing overall (rating scores), verified, reviewTime, reviewerID (the identification of reviewers), asin (the identification of items), reviewerName (the name of reviewers), reviewText, summary (the content summary of the reviews), unixReviewTime, vote, image, style; (2) The metadata file of items containing category (listing the categories to which the items belong), tech1 (the technical detail of items), description, fit, title, also_buy, tech2 (the technical detail of items), brand, feature (listing the description of the item features), rank, also_view (listing the related items also viewed), details, main_cat (listing the main item categories), similar_item, date, price, asin (the identification of items), imageURL (links to image file), imageURLHighRes.

MovieLens dataset: The MovieLens dataset used in our experiment is available at https://grouplens.org/datasets/movielens. The two files that we used in the experiment include: (1) The rating file that contains userId as the identification of users, movieId as the identification of movies, rating, timestamp; (2) The metadata file of movies containing movieId as the identification of movies, title, genres.

Selected datasets: **Table 1** presents the details of the MovieLens and the Amazon datasets.

The rating data of MovieLens dataset includes 77764 ratings of 27042 users and 8203 movies with the following properties: the identification of users, the identification of items, rating scores, timestamp, title, genres.

The rating data of Amazon sub-datasets has the properties: overall, verified, reviewTime, the identification of reviewers, the identification of items, the reviewer name, reviewText, summary, unixReviewTime, vote, image, style; item categories, tech1, description, fit, title, also_buy, tech2, brand, feature, rank, also_view, details, the item main categories, similar_item, date, and imageURLHighRe, the item price, imageURL.

- The Electronic dataset includes 85065 ratings of 1042 users and 21200 items.
- The Video Games dataset includes 98769 ratings of 2291 users and 24708 items.
- Finally, the Toys and Games dataset includes 71780 ratings of 5462 users and 3028 items.

Table 1. MovieLens and Amazon dataset summary

Dataset	Dataset sub-category	Number of ratings	Number of users	Number of items
MovieLens dataset		77764	27042	8203
Amazon dataset	**Electronic**	85065	1042	21200
	Toys and Games	71780	5462	3028
	Video Games	98769	2291	24708

5.2 The experiments and analyzing results

5.2.1 Evaluation metrics:

Similar to related studies, we use indicators for the evaluation including Precision and Root Mean Square Error (RMSE). The formula for calculating the precision index is given below:

$$Precision = \frac{tp}{tp + fp} \tag{10}$$

where *tp* is the number of correctly recommended items and *fp* is the number of incorrect recommended items [9].

The formula for calculating RMSE is given as the following [2]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(y_j - \hat{y}_j\right)^2} \tag{11}$$

where y_j denotes the observations (test ratings) and $\hat{y}_j$ stands for predicted ratings.

5.2.2 Experimental setup:

We used 80% of the data for training and 20% for testing. In the training process, we used 20% of the training data for validation purposes.

Table 2. The experimental results on the MovieLens mini dataset

Modality / Modalities	RMSE	Precision
Our proposed model: User latent feature + item latent feature + item features (genres) *[Ratings + genres]*	**1.24**	**54.35%**
Our proposed model: User latent feature + Item latent feature *[Ratings]*	1.32	50.17%
SVD *[Ratings]*	2.28	47.65%

Table 3. The experimental results on the Amazon datasets

Modality / Modalities	RMSE	Precision	RMSE	Precision	RMSE	Precision
	Electronic		Toys and Games		Video Games	
user latent features + item latent features (we called feature R) + item features (category) + item features (description) + user review *[*ratings *+ review +categories + description]*	**1.02**	**82.62%**	0.97	75.96%	1.19	64.34%
Feature R + item features (category) + item features (description) *[Rating, category, description]*	1.04	79.28%	0.91	84.28%	**1.19**	**65.60%**
Feature R + item features (category) *[Rating, category]*	1.05	78.26%	**0.88**	**86.53%**	1.19	65.60%
Feature R + item features (description) *[Rating, description]*	**1.01**	**82.86%**	0.91	84.85%	1.19	64.22%
Feature R *[Ratings]*	1.02	82.43%	0.90	85.54%	1.19	64.85%
SVD *[Ratings]*	1.16	69.55%	1.12	68.41%	1.25	64.27%

Amazon dataset: For extracting features of item description and user reviews, we adopt a "bert-base-uncased" pretrained model where the tokenizer is set up with max length as 100. For the matrix factorization step, we set the number of factors as 50, epochs as 500, the learning rate as 0.0003, and the L2 norm of vectors as 0.04. For PCA, we set the number of components as 30, and set the number of best features for each modality as 15.

Movielens dataset: For the matrix factorization step, we set the number of factors as 50, epochs as 500, the learning rate as 0.0003, and the L2 norm of vectors as 0.04. For PCA, we set the number of components as 20, and set the number of best features for each modality as 15.

In our experiments for both the Amazon and the MovieLens datasets, we adopt a deep neural network for fusing item feature vectors, latent feature vectors of users and items. This network has three layers where the first hidden layer has 64 nodes, the second hidden layer has 32 nodes, and the output layer has 1 node. We also set the batch size as 100, epochs as 50 and used Adam optimizer with the learning rate set to 0.0002 with the activation function as "relu".

5.2.3 Experimental results:

For the purpose of evaluating the performance of our proposed model, we compared the results with the SVD recommendation model which used the same datasets as our proposed multi-modal recommendation model during training and testing.

Table 2 presented the experimental results on the MoviesLen dataset, demonstrating the superiority of our proposed multi-modal recommendation model that combined rating scores and item genres information. Our model achieved an RMSE of 1.24 and Precision of 54.35%, outperforming the SVD recommendation model with RMSE of 2.28 and Precision of 47.65%. Additionally, **Fig. 4** showed that incorporating genre information in our multi-modal model improved its performance compared to both the SVD recommendation model and our model that only used rating information.

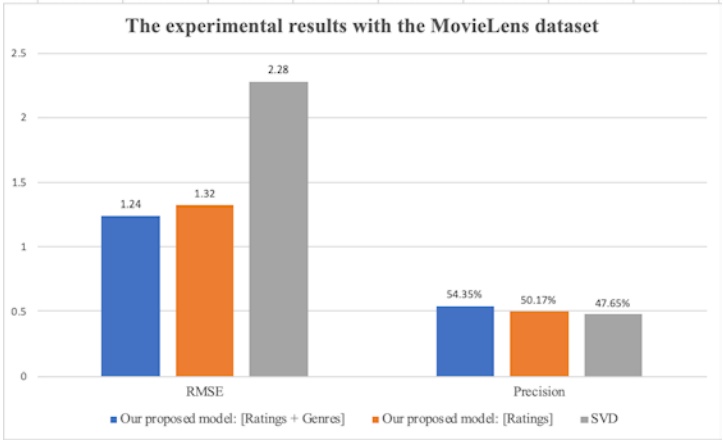

Fig. 4. The experimental results on RMSE and Precision on the Movielens dataset

Table 3 presented the experimental results with the Amazon sub-datasets. For doing experiments on the Amazon - Electronic dataset, our proposed model which fused item descriptions and the rating score information gave the best results with RSME of 1.01 and Precision of 82.86%, that was approximate to our proposed model which fused the rating score, user review, item category and description information with RMSE of 1.02 and Precision of 82.62%. Our proposed model achieved a 12.93% reduction in RMSE and a 19.14% increase in Precision compared to the SVD recommendation model. Furthermore, the experimental findings demonstrated that the combination of category information and rating score information would have a detrimental effect on the performance of our recommender system. Conversely, the graphical representation in **Fig. 5** revealed that our proposed approach, which incorporates data from both the utility matrix and item descriptions, achieved superior results compared to utilizing only the information from the utility matrix.

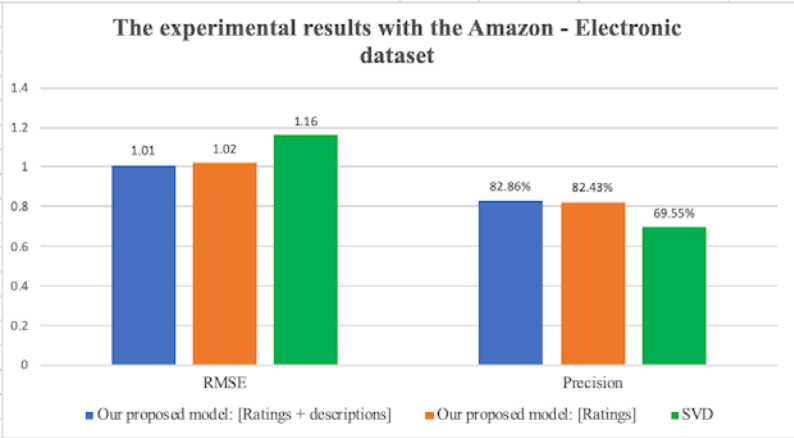

Fig. 5. The experimental results on RMSE and Precision on the Amazon - Electronic dataset

For doing experiments on the Amazon - Toys and Games dataset, our proposed model which fused item category and the rating score information gave the best results with RSME of 0.88 and Precision of 86.53%. In comparison to the SVD recommendation model, our model exhibited a significant reduction of 21.43% in RMSE and a notable increase of 26.49% in Precision. Besides, the results in **Fig. 6** also showed that adding user review information to our recommendation model reduced the performance of the recommender system. We speculated that there was a mismatch between the rating scores and the user reviews.

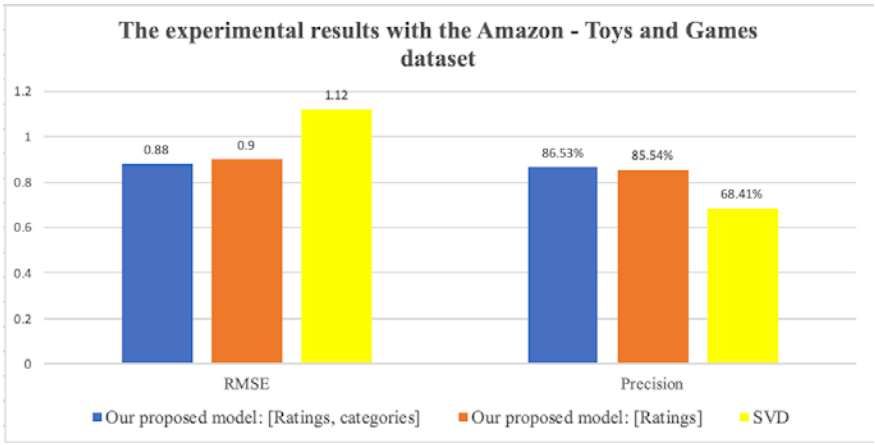

Fig. 6. The experimental results on RMSE and Precision on the Amazon – Toys and Games dataset

For doing experiments on the Amazon - Video Games dataset, our proposed model gave the approximate results about RMSE and Precision index when adding different information in turn. This showed that item category, description information and user reviews do not affect the results of the recommender system.

In summary, from the experimental results above, the proposed multi-modal recommendation model helps to improve the performance of recommender systems compared to the SVD recommendation model which is a famous and popular approach for recommender systems.

6. Conclusion

In our work, we propose a multi-modal recommendation model that transforms and fuses modalities to enrich information for recommendation. These modalities come from various information resources, including the utility matrix, user reviews, item descriptions, and item categories. We first use the matrix factorization technique to extract the user and item latent features from a utility matrix. At the same time, we extract other textual information sources using natural language processing techniques, including BERT and TF-IDF. We also use PCA and a feature selection algorithm to obtain useful features for all modalities. All extracted and selected features are then fed into a neural network for making rating predictions.

We conducted experiments with our proposed model using different feature sets corresponding to different modalities. In summary of the experimental results, our proposed model achieved higher recommendation accuracy compared to SVD, which is known as one of the most effective models for recommender systems. The proposed model exhibited increases in Precision from 2.07% to 26.49%, and in RMSE from 4.8% to 21.43% for the Amazon datasets, as well as increases in Precision by 14.06%, and in RMSE by 45.61% for the MovieLens datasets. Additionally, the experimental results obtained from the MovieLens and Amazon datasets provide evidence that our proposed model, which integrates information from multiple modalities, outperforms using only rating information in terms of recommendation accuracy. Furthermore, the results reveal that modalities may contain noisy information and have different effects for each type of data. Finally, our findings demonstrate that the proposed model is efficient, and the multimodal approach holds promise for addressing the recommendation problem.

Acknowledgement

We would like to express our gratitude to Professor Soundararajan Ezekiel and student Maria Balega from the Department of Mathematics and Computer Science at Indiana University of Pennsylvania, USA, for their valuable discussions, which greatly improved the quality of this paper. They also provided us with significant help in correcting the English writing of the paper.

References

[1] Alolama, Y., “Recommender Systems and Amazon Marketing Bias,” *Thesis, Rochester Inst. of Tech.*, NewYork, USA, 2020.

[2] Christie, D., & Neill, S., “Measuring and Observing the Ocean Renewable Energy Resource,” *Comprehensive Renewable Energy (Second Edition)*, vol. 8, pp. 149-175, 2022. Article (CrossRef Link)

[3] Ebersbach, M., Herms, R., & Eibl, M., “Fusion Methods for ICD10 Code Classification of Death Certificates in Multilingual Corpora,” in *Proc. of CLEF*, September 2017. Article (CrossRef Link)

[4] Eklund, M., “Comparing Feature Extraction Methods and Effects of Pre-Processing Methods for Multi-Label Classification of Textual Data,” *Dissertation, Sch. of Elect. Eng. and Comp. Sci., KTH Royal Int. of Tech.*, Stockholm, Sweden, 2018.

[5] Feng, C., Liang, J., Song, P., & Wang, Z., “A fusion collaborative filtering method for sparse data in recommender systems,” *Information Sciences*, vol. 521, pp. 365–379, 2020. Article (CrossRef Link)

[6] Geetha, G., Safa, M., Fancy, C., & Saranya, D., "A hybrid approach using collaborative filtering and content-based filtering for recommender system," *Journal of Physics: Conference Series*, vol. 1000, pp. 012101, 2018. Article (CrossRef Link)
[7] Ghazanfar, M., & Prugel-Bennett, A, "An improved switching hybrid recommender system using naive bayes classifier and collaborative filtering," in *Proc. of IMECS2010*, March 2010. Article (CrossRef Link)
[8] Hoyer, P. O., "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. 9, 2004. Article (CrossRef Link)
[9] Júnior, A. F., Medeiros, F., & Calado, I., "An Evaluation of Recommendation Algorithms for Tourist Attractions," in *Proc. of ICSEKE 2020*, July 2020. Article (CrossRef Link)
[10] Vijay, K. and Bala, D., "Chapter 11 Recommendation engines," in *Data Science*, 2nd ed., USA: Elsevier, 2019, pp. 343-394. Article (CrossRef Link)
[11] Liu, K., Li, Y., Xu, N., & Natarajan, P., "Learn to combine modalities in multimodal deep learning," 2018. Article (CrossRef Link)
[12] Mulay, A., Sutar, S., Patel, J., Chhabria, A., & Mumbaikar, S., "Job Recommendation System Using Hybrid Filtering," *ITM Web of Conferences*, vol. 44, 2022. Article (CrossRef Link)
[13] Nayak, R., Mirajkar, A., Rokade, J., & Wadhwa, G., "Hybrid Recommendation System For Movies," *International Research Journal of Engineering and Technology*, vol. 5, no. 3, 2018. Article (CrossRef Link)
[14] Okaka, R. A., "A Hybrid Approach for Personalized Recommender System Using Weighted Term Frequency Inverse Document Frequency," Ph. D. Dissertation, Jomo Kenyatta Univ. of Agri. and Tech., Karen, Kenya, 2018.
[15] Otegbade, O., "A Hybridized Recommendation System on Movie Data Using Content-Based and Collaborative Filtering," Ph. D. Dissertation, African Univ. of Sci. and Tech., Abuja F.C.T, Nigeria, 2016.
[16] Conceiç ao, F. L., Pádua, F. L., Lacerda, A., Machado, A. C., & Dalip, D. H., "Multimodal data fusion framework based on autoencoders for top-N recommender systems," *Applied Intelligence*, vol. 49, no. 9, pp. 3267–3282, 2019. Article (CrossRef Link)
[17] Pennington, J., Socher, R., & Manning, C. D., "Glove: Global vectors for word representation," in *Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014. Article (CrossRef Link)
[18] Ren, X., Yang, W., Jiang, X., Jin, G., & Yu, Y., "A Deep Learning Framework for Multimodal Course Recommendation Based on LSTM+ Attention," *Sustainability*, vol. 14, no. 5, pp. 2907, 2022. Article (CrossRef Link)
[19] David E. R., James L. M., "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, MIT Press, 1986, pp. 318-362. Article (CrossRef Link)
[20] Sak, H., Senior, A. W., & Beaufays, F., "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014. [Online]. Available: http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43905.pdf
[21] Shipra, S., "Introduction to Long Short Term Memory (LSTM) Algorithms," *Analytics Vidhya*, 2021. [Online] Available: http://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm. Accessed on: Feb. 20, 2023.
[22] Scheidt, T., & Beel, J, "Time-dependent Evaluation of Recommender Systems," in *Proc. of PERSPECTIVES 2021*, September 2021. Article (CrossRef Link)
[23] Schnabel, T., Labutov, I., Mimno, D., & Joachims, T., "Evaluation methods for unsupervised word embeddings," in *Proc. of the 2015 Conference on Empirical Methods in Natural Language processing*, pp. 298–307, 2015. Article (CrossRef Link)
[24] Shlens, J., "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014. Article (CrossRef Link)
[25] Tharwat, A., "Principal component analysis-a tutorial," *International Journal of Applied Pattern Recognition*, vol. 3, no. 3, pp. 197-240, 2016. Article (CrossRef Link)

[26] Walek, B., & Fajmon, P., "A hybrid recommender system for an online store using a fuzzy expert system," *Expert Systems with Applications*, vol. 212, pp. 118565, 2023. Article (CrossRef Link)
[27] Wang, H., & Raj, B., "On the origin of deep learning," *arXiv preprint arXiv:1702.07800*, Feb. 2017. Article (CrossRef Link)
[28] Wang, J., Mao, H., & Li, H., "FMFN: Fine-Grained Multimodal Fusion Networks for Fake News Detection," *Applied Sciences*, vol. 12, no. 3, pp. 1093, 2022. Article (CrossRef Link)
[29] Wang, Z., Chen, H., Li, Z., Lin, K., Jiang, N., & Xia, F., "VRConvMF: Visual recurrent convolutional matrix factorization for movie recommendation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 3, pp. 519-529, 2022. Article (CrossRef Link)
[30] Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J., "Deep matrix factorization models for recommender systems," in *Proc. of IJCAI*, Melbourne, Australia, vol. 17, pp. 3203-3209, 2017. Article (CrossRef Link)
[31] Yang, E., Huang, Y., Liang, F., Pan, W., & Ming, Z., "FCMF: Federated collective matrix factorization for heterogeneous collaborative filtering," *Knowledge-Based Systems*, vol. 220, pp. 106946, 2021. Article (CrossRef Link)
[32] Zhang, H., Ganchev, I., Nikolov, N. S., Ji, Z., & O'Droma, M., "FeatureMF: An Item Feature Enriched Matrix Factorization Model for Item Recommendation," *IEEE Access*, vol. 9, pp. 65266-65276, 2021. Article (CrossRef Link)
[33] Zhou, Z., "Amazon Food Review Classification Using Deep Learning and Recommender System," 2016. Article (CrossRef Link)
[34] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K., "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018. Article (CrossRef Link)
[35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I, "Attention is all you need," in *Proc. of 31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017. Article (CrossRef Link)
[36] Reddy, S. R. S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B, "Content-based movie recommendation system using genre correlation," in *Proc. of Smart Intelligent Computing and Applications: Proc. of the Second International Conference on SCI 2018*, Springer, Singapore, vol. 105, pp. 391-397, 2019. Article (CrossRef Link)
[37] Wahyudi, K., Latupapua, J., Chandra, R., & Girsang, A. S, "Hotel content-based recommendation system," *Journal of Physics: Conference Series*, vol. 1485, no. 1, pp. 012017, Mar. 2020. Article (CrossRef Link)
[38] Singla, R., Gupta, S., Gupta, A., & Vishwakarma, D. K, "FLEX: A Content Based Movie Recommender," in *Proc. of 2020 International Conference for Emerging Technology (INCET)*, IEEE, pp. 1-4, Jun. 2020. Article (CrossRef Link)
[39] Ghauth, K. I., & Abdullah, N. A, "Learning materials recommendation using good learners' ratings and content-based filtering," *Educational technology research and development*, vol. 58, pp. 711-727, 2010. Article (CrossRef Link)
[40] Zhao, Y., & Shen, B, "Empirical study of user preferences based on rating data of movies," *PloS one*, vol. 11, no. 3, 2016. Article (CrossRef Link)
[41] Ahmed, B. H., & Ghabayen, A. S, "Review rating prediction framework using deep learning," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 7, pp. 3423-3432, 2022. Article (CrossRef Link)
[42] Gezici, B., Bölücü, N., Tarhan, A., & Can, B, "Neural sentiment analysis of user reviews to predict user ratings," in *Proc. of 2019 4th International Conference on Computer Science and Engineering (UBMK)*, IEEE, pp. 629-634, Sep. 2019. Article (CrossRef Link)
[43] Lei, F., Cao, Z., Yang, Y., Ding, Y., & Zhang, C, "Learning the User's Deeper Preferences for Multi-modal Recommendation Systems," *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 19, no. 3, pp. 1-18, 2023. Article (CrossRef Link)
[44] Mu, Y., & Wu, Y, "Multimodal Movie Recommendation System Using Deep Learning," *Mathematics*, vol. 11, no. 4, pp. 895, 2023. Article (CrossRef Link)

Thi-Linh Ho received her B.S. degree in information technology from Nong Lam University, Ho Chi Minh City, Vietnam in 2009. She received her master degree in Management Information System in 2012 from Ho Chi Minh City University of Technology, Vietnam. She is a PhD at Ton Duc Thang University, Ho Chi Minh City, Vietnam. From 2013 to 2016, she was a lecture at Ho Chi Minh City University of Food Industry, Vietnam. Currently, she is a lecturer at Ho Chi Minh City University of Banking, Vietnam. Her fields of interest consist of working with recommender systems, multi-modal fusion, machine learning, and information systems.

Anh-Cuong Le is currently an associate professor at Ton Duc Thang University (TDTU) in Ho Chi Minh City, Vietnam. His research areas include artificial intelligence, machine learning, and natural language processing. He received his bachelor's and master's degrees in information technology from Hanoi National University in 1998 and 2002, respectively. He then received his doctorate in computer science from the Japan Advanced Institute of Science and Technology (JAIST) in 2007. After graduating with his doctorate, he worked as a lecturer at the University of Engineering and Technology (UET), Vietnam National University, Hanoi until 2015. Since 2016, he has been working at the Faculty of Information Technology at TDTU.

Dinh-Hong Vu received the B.S. degree in information technology from the VNU Ho Chi Minh City University of Science, Ho Chi Minh City, Vietnam, in 2005, and the M.Sc. degree in computer science from the VNU Ho Chi Minh City University of Science, in 2011. He is currently a Researcher with NLP-KD Lab, Ton Duc Thang University, Vietnam. His research interests include natural language processing, machine translation, and text clustering.