# Enhancing neural collaborative filtering using hybrid feature selection for recommendation

**Baboucarr Drammeh** [1,2], **Hui Li** [Corresp. 1]

[1] College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou, China

[2] School of Information Technology and Communication, University of The Gambia (UTG), Banjul, Peace Building, Kanifing, The Gambia

Corresponding Author: Hui Li
Email address: huili.gm@gmail.com

The past decade has seen substantial growth in online transactions. Accordingly, many professionals and researchers utilize deep learning models to design and develop recommender systems to suit the needs of online personal services. These systems can model the interactions between users and items. However, existing approaches focus on either modeling global or local item correlation and rarely consider both cases, thus failing to represent user-item correlation very well. Therefore, this paper proposes a deep collaborative recommendation system based on a convolutional neural network with an outer product matrix and a hybrid feature selection module to capture local and global higher-order interaction between users and items. Moreover, we incorporated the weights of Generalized matrix factorization to optimize the overall network performance and prevent overfitting. Finally, we conducted extensive experiments on two real-world datasets with different sparsity to confirm that our proposed approach outperforms the baseline methods we have used in the experiment.

# Enhancing Neural Collaborative Filtering Using Hybrid Feature Selection for Recommendation

Baboucarr Drammeh1[1,2] Hui li[1]

[1] College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou China

[2] School of Information Technology and Communication, University of The Gambia (UTG), Banjul, Peace Building, Kanifing, The Gambia

Corresponding Author:

Hui Li[1]

College of Computer Science and Technology, Guizhou University, Guiyang, Guizhou Province, 550025, China

Email address: huili.gm@gmail.com

## Abstract

The past decade has seen substantial growth in online transactions. Accordingly, many professionals and researchers utilize deep learning models to design and develop recommender systems to suit the needs of online personal services. These systems can model the interactions between users and items. However, existing approaches focus on either modeling global or local item correlation and rarely consider both cases, thus failing to represent user-item correlation very well. Therefore, this paper proposes a deep collaborative recommendation system based on a convolutional neural network with an outer product matrix and a hybrid feature selection module to capture local and global higher-order interaction between users and items. Moreover, we incorporated the weights of Generalized matrix factorization to optimize the overall network performance and prevent overfitting. Finally, we conducted extensive experiments on two real-world datasets with different sparsity to confirm that our proposed approach outperforms the baseline methods we have used in the experiment.

**Keywords:** Recommender Systems, Outer Product, Convolutions, Embedding, Collaborative Filtering

## 1. Introduction

The large amounts of information generated by online services have recently challenged online users to identify meaningful recommendations. However, recommendation systems can help users find valuable information faster, and they are frequently utilized in services such as e-Commerce(L. C. Chen et al., 2016), social media recommendations(Guy et al., 2010), and online video services(Covington et al., 2016).

Traditional approaches provide recommendations based on the similarities between users and items, which can be categorized into collaborative filtering, content-based, and hybrid recommendation systems (Lu et al., 2015). Collaborative filtering has been researched and heavily used in systems based on personal recommendation. It utilized the possibility of items users may be interested in based on their historical interactions (Breese et al., 1998). Furthermore, content-based uses additional features about users or items to recommend similar items (Çano & Morisio, 2017; Javed et al., 2021). Finally, the hybrid method combines two or more recommendation techniques (Çano & Morisio, 2017).

Most traditional recommendation systems are limited when a large amount of data is to be analyzed (Gasmi et al., 2020), Moreover, they rely on a linear kernel that does not fully represent user-item interaction. Accordingly, researchers have recently adopted Deep learning-based approaches to develop recommender systems (Pan et al., 2020) since they can learn complex nonlinear relationships and handle various data types. Several deep learning-based methods have been proposed due to their ability to deal with high-dimensional features effectively. Collaborative deep learning methods based on matrix factorization are known to provide satisfactory performance (He et al., 2017)(H. Zhang et al., 2016). Matrix factorization represents a given user and item as an embedding and learns their relationship using an inner product between the user and item embeddings. Despite its effectiveness, matrix factorization is limited since it uses an inner product as the interaction function(He et al., 2018), which assumes that the embedding dimensions are independent of each other and perform a multiplication between them, thus limiting the expressiveness of the model.

In this study, we proposed a recommender algorithm using a hybrid feature selection module (HSFM) to capture the useful global and local high-dimensional relationship between users and items. Our proposed approach utilizes convolution to capture the valuable nonlinear relationship between users and items by learning the outer product matrix. To verify the effectiveness of our model, we conducted experiments on two real-world datasets, Movielens and Pinterest. Experimental results demonstrate that our proposed model outperforms the baseline methods significantly. The main contribution of the paper is summarized as follows:

1. The stack interaction map is introduced to increase the input features expressiveness and allow the interaction map to encode more latent signals.
2. To effectively capture the correlations between items, we also leverage a hybrid feature selection module, which uses pointwise convolution and general average pooling to learn both local and global item correlations.
3. We also incorporate Generalized Matrix Factorization (GMF) to constrain the network's weight which optimizes the network performance and prevents overfitting.
4. We conducted an extensive experiment on two publicly available datasets to demonstrate the effectiveness of our proposed model.

The remainder of the paper is structured as follows. First, Section 2 reviews related works, and Section 3 elaborates on our proposed method. Then, the experimental results on the two datasets are reported in Section 4. Finally, we conclude this paper in Section 5.

## 2. Related Works

In this section, traditional recommender systems were examined on how they model the similarity between users and items; second, we also look at deep learning techniques due to their high-quality recommendation performance and better ability to learn the relationship between users and items.

## 2.1 Traditional Recommender Systems

Many recommender algorithms are based on collaborative filtering(Adomavicius & Tuzhilin, 2005), which depends on users' past behavior to make predictions. Collaborative filtering-based recommendations are divided into latent factor methods (Koren et al., 2009) and neighborhood-based methods(Sarwar et al., 2001). Neighborhood-based approaches utilize ratings directly to evaluate new items for users. Such models are based on the similarities between a user to a user or item to an item. The similarity between two items is measured as the probability of users rating those items' similarly, which is usually based on the Pearson correlation. On the other hand, the latent factor models users and items as vectors using the same latent space by reducing the number of hidden factors. Latent factors compare users and items directly, where a user's rating of an item is predicted using the inner product between related latent vectors.

Singular value decomposition(Koren et al., 2009) reduces the number of user-item features to a product of two low-rank matrices. However, one of its drawbacks is the high cost of locating singular value decomposition. Another approach that enhanced the singular value decomposition, SVD++(Koren, 2008), uses implicit and explicit feedback to provide a recommendation and demonstrate improved performance over many matrix factorization models.

Generally, traditional recommender algorithms use a linear kernel that does not better represent the user-item relationship.

## 2.2 Deep Learning Base Recommender System

Deep learning has developed extensively in the past decade and has been implemented in various fields, such as computer vision, speech recognition, and natural language processing(W. Zhang et al., 2018). Deep learning learns features directly from data and performs feature engineering automatically, and it has been studied extensively in recommendation systems. The deep learning-based models have demonstrated significant performance over the traditional recommendation system(Singhal et al., 2017). For instance, Neural collaborative filtering (He et al., 2017) utilizes a multilayer perceptron to model the interaction function as it represents users and items as a low-dimensional vector in latent space.

In another research, deep matrix factorization (DMF) (Xue et al., 2017) utilizes a matrix factorization and a neural network architecture, which uses explicit scores of users and non-preference implicit feedback of items.

The correlation denoising autoencoder (Pan et al., 2020) considers the correlation between users with diverse roles to learn a more robust representation from sparse ratings and social networks. It uses three autoencoders to learn user features taking them as a separate matrix of rating, truster, and trustee. The authors (Liu et al., 2020) couple deep neural networks with matrix factorization and learn the deep global and local item relationship of item content by coupling autoencoder with matrix factorization to join the rating and item content information.

Convolutional neural networks (LeCun et al., 1998) are prevalent in image recognition, and they are generally made up of a convolution layer, pooling, and a fully connected layer. Convolutions are also used in recommender systems to model the interaction map. For instance, the convolutional factorization machine(Xin et al., 2019) is a recommender model that is context aware; it uses self-attention, an embedding layer, and a pooling layer. The authors also used an outer product interaction cube coupled with a 3D convolutional neural network to extract higher-order signals. In another research, the authors of ConvNCF (He et al., 2018)also utilize an outer product to explicitly model pairwise correlation instead of just concatenating or mere element-wise multiplication of the embedding. In addition, they also use a convolutional neural network above the interaction map to learn higher-order correlations.

Convolutional Factorization Machine and ConvNCF use regular convolution, which helps learn local features and does not learn global features well.

Our proposed hybrid feature selection uses deep neural networks to learn both local and global item correlation between users and items. In addition, we incorporate GMF into the model to optimize the overall model performance and prevent overfitting.

## 3. Our proposed methods

### 3.1 Input and Embedding Layer

Given a user ***u*** and item ***i***, $V_u^U$ and $V_i^I$ represent the feature vectors of U and I respectively, and their embeddings can be represented as:

$$p_u = P^T v_u^U,\ q_i = Q^T v_i^I \tag{1}$$

Where $P \in R^{M \times K}$ and $Q \in R^{N \times K}$ represent the embedding matrix for the user and item features, M, N, and K represent the number of users, the number of items, and the embedding size, respectively.

### 3.2 Interaction Map

The outer product was utilized to generate the interaction map since it can learn more information between latent features. For example, the outer product between a user and an item can be defined as:

$$m^t \otimes n^T = m^t n = \begin{pmatrix} m_{d_1} n_{d_1} & \cdots & m_{d_1} n_{d_k} \\ \vdots & \ddots & \vdots \\ m_{d_k} n_{d_1} & \cdots & m_{d_k} n_{d_k} \end{pmatrix} \quad (2)$$

where m and n represent row vectors and denote K-dimensional latent vectors.

If $p_u = m^t$ and $q_i = n^T$ then $p_u$ and $q_i$ are used to obtain the interaction map, and it can be represented as:

$$E(p_u, q_i) = p_{u \otimes} q_i = p_u q_i^T \quad (3)$$

where $E(p_u, q_i)$ represents a $K \times K$ matrix.

Matrix factorization is not robust for modeling user-item correlation because it considers only diagonal elements and performs simple concatenation. However, the outer product is more robust and encodes more latent signals. The interaction map obtained by the outer product has one pair of latent factors, which may not perform well in the 2D convolution. As a result, we stack the interaction map into a **k** number of features concatenated along the dimensions. The latent signal of the interaction map determines the k-dimension features passed as input into the 2D convolution. Therefore, the increase in latent features makes the interaction map encode more relational signals, thus making it more expressive. Furthermore, the higher the number of k, the feature dimension of the input to the convolution also increases, making the model more memory and computationally intensive. However, k values greater than three do not guarantee an increase in the accuracy of the model and occasionally lead to overfitting.

### 3.3 Convolution Module

The stack interaction map that encodes richer latent features is used as the first input to the convolution module. The convolution module is a three-layer convolution that learns the local features between users and items. The first convolution is the input layer, followed by the two hidden layers, which help to learn more meaningful information between the users and items. The input convolution layer takes the input channel of the stack interaction map, an output channel size of 32, and a kernel size of 2, while the hidden layer utilizes input and output channels of 32 with a kernel size of 2. The convolution layer is mathematically represented as:

$$f_l^k(a,b) = \sum_c \sum_{p_u, q_i} E_c(p_u, q_i) * e_l^k(u,v) \quad (4)$$

Where * is the convolution, $f_l^k(a,b)$ represent the element of the feature matrix, $E_c(p_u, q_i)$ represents the element of the input stack interaction map $E_c$ of channel **c**, which is element-wise multiplied by $e_l^k(u,v)$ index of the *k*th convolution kernel $k_l$ of the *l*th layer. The feature map of the kth convolutional operation can be expressed as:

$$F_l^k = [f_l^k(1,1),...,f_l^k(a,b),...,f_l^k(A,B)] \quad (5)$$

$F_l^k$ represent the input feature matrix for the $l$th layers and $k$th neuron, A *and B* represent the total number of rows and columns of the feature matrix, respectively.

A convolution block or network can obtain multiple convolutional layers, dropout, and activation map for extracting meaningful information and adding non-linearity to learned complex patterns. In addition, a dropout layer is introduced to reduce overfitting, which negatively impacts the model prediction.

### 3.4 Hybrid Feature Selection Module (HFSM)

The convolution module with a stacked interaction map which is used as input to our model, is prone to overfitting. However, it can capture relational representation well. The convolutional module introduced in section 3.3 is followed by a dropout layer and ReLU(Agarap, 2018) activation function. Nevertheless, the dropout minimizes overfitting at the expense of removing valuable feature relational representation. Therefore, we suggest the hybrid feature selection module (HSFM) to bridge the gap between overfitting and losing valuable information. The HSFM module takes in two inputs, **y,** and **x**, representing the output before dropout and after the ReLU activation function.

The HFSM aggregates two distinctive features to complement each other. A convenient approach to aligning two distinct feature relationships is to learn their local relationship, which can be obtained using pointwise convolutions. Therefore, the inputs **x** and **y** are summed and passed to two branches. The first branch accesses the global feature using general average pooling (GAP), and the second focuses on the local feature relationship. The two branches have two pointwise convolutions, each followed by binary normalization that minimizes the feature variation and a ReLU non-linearity. Finally, the HFSM combines global and local relationships by applying sigmoid activation on the sum of the two-branch feature, which is expressed as:

$$out = x \otimes s(m) + y \otimes (1 - s(m)) \quad (6)$$

where $s$ represents the sigmoid function and **m** is the summation of the global and local branches.

Generally, hybrid feature selection modules access the global and local relationships from the two distinct branches that complement each other to extract better feature representation without introducing overfitting.

### 3.5 Generalized Matrix Factorization

GMF learns from data without uniform constraints and is more expressive than linear Matrix Factorization. Therefore, we combine the losses of GMF(He et al., 2017) and our proposed model to update the overall model weight, which obtains a better result and further avoids overfitting. GMF uses an element-wise product of a latent vector of users $p_u$ and items $q_i$ , which can be represented as:

$$z^{gmf} = \varphi_1(p_u, q_i) = p_u \otimes q_i \tag{7}$$

The prediction of GMF is also represented as:

$$\grave{p}_{ui} = \sigma\left[h^T z^{gmf}\right] \tag{8}$$

Where $\sigma$ represents the sigmoid given as $\sigma(a) = {}^{1}/\left(1 + e^{-a}\right)$ and h is the weight of the output layer.

### 3.6 Fusion of our Proposed Method and GMF

Fusion in a convolutional network joins two or more features using an element-wise product, element-wise summation, or concatenation. Concatenation provides a better representation of latent features at the expense of computational complexity and memory consumption. Alternatively, features can be fused by combining the losses of different network modules, constraining the model's overall weight by considering the submodules' special functions.

The proposed network uses Bayesian Personalized Ranking (BPR), a pairwise loss function, since it measures the dependency between data points and can measure the complex relationship between data points. It can be represented as follows:

$$\mathcal{L}1 = \sum_{u=1}^{N} \sum_{i \in I_u^+, j \in I_u^-} -log\sigma\left(\hat{x}_{uij}\right) + \lambda\Omega(\Phi) \tag{9}$$

Where $\grave{x}_{uij} = p_u^T q_i - p_u^T q_j, \sigma(x) = {}^{1}/(1 + exp(-x))$ is the sigmoid function and $\lambda\Omega(\Phi)$ is the regularization

On the other hand, the GMF model utilized the log loss function, which is a pointwise loss function, and it is easily computable and differentiable by the optimizer. Pointwise loss is also more flexible since it can be applied in many applications and is robust to outliers and noise in data. It can be expressed as:

$$\mathcal{L}2 = -\sum_{(u,i) \in \mathcal{R}^+ \mathcal{R}^-} [f_{ui} \log f_{ui}) + (1 - f_{ui}) \log (1 - \hat{f}_{ui})] \tag{10}$$

where $R^+$ represents positive a training instance, $R^-$ represents the set of negative training instances and $f_{ui}, \grave{f}_{ui}$ is the prediction and label of the GMF.

$$Loss = \alpha * L1 + \beta * L2 \quad (11)$$

where $L1$ is the BPR loss, $L2$ is the log loss for the GMF, and $\alpha$ and $\beta$ are the weighted coefficients.

The $\mathcal{L}oss$ combined $\mathcal{L}1,\ and\ \mathcal{L}2$ to constrain the weight of the overall proposed method, essentially avoiding overfitting and further improving the recommendation performance. In addition, we added the weighted co-efficient $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ values 0.5 and 0.75, respectively, to tune the impact of the sub-networks losses for the ease of model minimization as the training epochs increase.

### 3.7 Final Prediction Layer

The output of the HFSM is reshaped and flattened using a fully connected layer to facilitate the output prediction. Finally, the result is passed into a sigmoid function to calculate the final prediction score. The array of scores $\grave{f}_{ui}$ and $\grave{y}_{ui}$ represent the prediction scores of GMF and the proposed CNN-based model, respectively.

## 4. Experiments

The subsequent section presents our experiments on two publicly available datasets to answer the following questions:

**RQ1** Does the proposed model outperform the baselines in top k recommendations?

**RQ2** Is the proposed stacking of the interaction map helpful for learning from user-item interaction and improving recommendations?

**RQ3** How do key hyperparameter settings influence the performance of our model?

### 4.1 Experimental *Settings*

#### 4.1.1 Datasets

**Movielens** 1M is a movie rating dataset that contains around 1 million ratings of around 3900 movies by 6040 users in which there are 5-grade ratings, and each user rated at least 20 items. It is a widely used data set for evaluating recommendation performance.

**Pinterest** is an implicit feedback dataset constructed by (Geng et al., 2015) for evaluating content-based image recommendations. It has 55187 users and 9916 items. The original dataset is sparse, but the preprocessed contains at least 20 interactions. Each interaction represents if a user has pinned an image to their board.

#### 4.1.2 Evaluation Protocols

We use leave-one-out evaluation, a popular method for testing the quality of the ranking for the recommendation. For each user,256 unrated items are used as test data. We used the hit ratio (HR) and the normalized discounted cumulative gain (NDCG) as the evaluation matrix. Both metrics were calculated for each user, and the average scores were reported.

The hit ratio represents the relevant time items in the top-n list of an individual user that appear. It can be represented as:

$$HR@n = \frac{hits}{n} \tag{11}$$

Where n represents the number of top n items generated from the methods, a higher value denotes better performance.

NDCG is sensitive to the relevance of higher-ranked items and assigns higher scores to the correct recommendations at a higher rank in the list. NDCG is defined as follows:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \tag{12}$$

where $IDCG_P = \sum_{i=1}^{|REL_P|} \frac{2^{rel_i - 1}}{\log_2 (i+1)}$

$\Re L_p$ = list of useful items and p = position

$$RMSE = \sqrt{\sum_{(u,i) \in \mathcal{R}_{test}} \frac{(r_{u,i} - \hat{r}_{u,i})^2}{|\mathcal{R}_{test}|}} \tag{13}$$

$$MAE = \frac{1}{|\mathcal{R}_{test}|} \sum_{(u,i) \in \mathcal{R}_{test}} |r_{u,i} - \hat{r}_{u,i}| \tag{14}$$

Where $r_{u,i}$ represents the actual rating, $\hat{r}_{u,i}$ represents the prediction and $\mathcal{R}_{test}$ represents the number of ratings in the test set.

#### 4.1.3 Baselines

To justify the effectiveness of our proposed model, we compare it with the following baselines:

- **MLP**(He et al., 2017) is a neural collaborative filtering approach that reduces the matrix of users and items into two submatrices and multiplies them together to learn the interaction function.
- **GMF** (He et al., 2017) uses a scalar product to model the interaction between users and items by reducing their metrics into two summaries.

- **DMF**(Xue et al., 2017) uses matrix factorization coupled with neural network architecture. It also projects users and items into lower-dimensional vectors in latent space.
- **NeuMF**(He et al., 2017) is an item recommendation method that joins hidden layers of GMF and MLP to model the user-item interaction function.
- **ONCF**(He et al., 2018) uses a convolutional neural network with an outer product to model the correlation of user-item correlation; it is an improvement of Matrix Factorization.
- **SDMR** (Tran et al., 2019) utilized deep learning to learn the signed distance between users and items and produce a recommendation based on the learned signed distance. Specifically, signed distance measures the difference or similarity between two items. SDMR combines two signed distance scores internally: signed-distance base perceptron (SDP) and signed distance base memory network (SDM).
- **CoCNN**(M. Chen et al., 2022) CoCNN Joins a co-occurrence pattern and Convolutional Neural network to collaborative filtering with implicit feedback. The authors also designed an embedding structure to capture the link between user-item and item-item. They also proposed a multi-task neural network to share the knowledge of the two tasks.

#### 4.1.4 Parameter Settings

We implemented our proposed model using Pytorch on Nvidia GTX 1080. All models were optimized using Mini-batch Adagrad, and the learning rate is searched between [0.001,0.0001, 0.00001,0.000001, 0.00000001]. The batch size is 256, and the embedding size is 64. ONCF and our proposed model used a channel size of 32. We also use a dropout of 0.2 for our CNN-Based and 0.5 for our CNN-Based+HFSM and CNN-Based +HFSM +GMF model.

### 4.2 Performance Comparison (RQ1)

*Table 1 :*

*compares different models when generating top-k recommendations on two datasets. k ∈{10}. The boldface denotes the persistently increased scores of our proposed networks.*

Table 1 shows a comparison between our proposed model and the baselines that we used in the experiment. The performance evaluation used for the comparison utilized HR@10 and NDCG@10. In addition, for a fair comparison, we trained all the baseline models using BPR loss.

1. Table 1 shows that ONCF has outperformed MLP, GMF, NeuMF, DMF, and SDMR by both HR@10 and NDCG@10 on both datasets.
2. The CNN based on our proposed model has outperformed ONCF by a significant margin in both datasets since it does not lose much information during feature extraction to establish local relationships. In addition, The CNN-based approach utilizes stacking of the interaction map to encode better latent signals and establish a better user-item relationship.
3. The CNN-based approach of our proposed method does not capture the global relationship in the interaction map. Therefore, we introduced the HSFM module for the model to learn global relationships in addition to local ones. This mechanism has also improved performance over the CNN-based in HR and NDCG scores, respectively.
4. The fusion of GMF and our proposed method constrain the overall model weight, thus allowing the model to benefit from sub-networks. This combination obtained the best performance in our proposed methods on both datasets.
5. Since all the networks are trained on the BPRLoss, our proposed model performed better than the baselines on both datasets. Moreover, the proposed method obtained a remarkable performance on not only the CNN-based global and the local feature interaction for high relational modeling but the hybrid of the GMF sub-network, which shows promising results and further constrains the model weight. However, these benefits come at the expense of computation complexity.

*Figure 1 shows the graphical representation of the performance of our best-proposed method compared to the baselines using HR@10 and NDCG@10. (a)(b) and (c)(d) demonstrate the performance on the MovieLens and Pinterest datasets, respectively.*

From the charts in Figure 1, our best-proposed model has significantly outperformed the baselines on both datasets at HR and NDCG evaluation metrics. Furthermore, among the baselines, NeuMF outperformed MLP, but it is entirely defeated by GMF, demonstrating that GMF is a simply designed yet powerful prediction model. On the other hand, NeuMF does not achieve the desired result, which may result from the selected optimizer or the poor performance of the underlying MLP in the sparse datasets.

*Table 2:*

*shows the performance of our proposed model against the baselines on RMSE, MAE, and BPR Loss.*

To further evaluate the efficiency of our model, we use RMSE, MAE, and BPRLOSS and compare them to the baselines. Table 2 shows that our proposed model has effectively reduced the loss more than the baselines, thus indicating a strong minimization ability during the model training.

## 4.3 Efficacy of stacking the interaction map (RQ2)

*Table 3:*

*shows the performance of the stack@n of the interaction map as the input to our simple Cnn-based proposed model. n ∈{1, 2, 3}.*

The stacking of the interaction map increased the latent signal, which allowed our proposed model to establish strong user-item relationships. From Table 3, the experiments demonstrated that the increase in the stacking of the interaction map has a proportionate impact on the effectiveness of the model performance. However, beyond stack@3, the model incurred overfitting, which impaired the model performance.

## 4.4 Research of hyper-parameter (RQ3)

*Figure 2 shows the performance of our best model using different channel sizes on the movielens dataset*

In this section, we investigated how the increased convolutional feature maps (channel size) impact the representation ability of our top-performing proposed model. As a result, we experimented with different channel sizes $c \in \{16, 64, 128\}$. Figure 2 shows the performance of our best model using different channel sizes on the movielens dataset. We can observe that the increased number of channel sizes also improves the model's performance. However, these performance gains come at the expense of increased computational complexity and are time-consuming during training.

Furthermore, the charts show that all the curves increase steadily, and the feature map at channel 128 achieves the best performance. Moreover, channel 64 steadily outperformed channel 16 until the final epochs, where a slight difference in the convergence curve was noticed. These reflect our proposed model's strong expressiveness and generalization since increasing the number of parameters adjust the model performance and does not lead to overfitting.

# 5 Conclusions

In this paper, we proposed a deep learning convolutional-based recommender system for modeling user-item correlation. The proposed method utilized convolution mechanisms for local-global feature selection and combined the generalized matrix factorization (GMF) to establish a more robust user and item relationship model that improved accuracy without overfitting. We conducted a series of experiments with two real-world datasets, and corresponding experimental results demonstrated that our proposed model has a higher recommendation accuracy and surpasses the baselines in the top-k recommendation task. In the future, we plan to explore other forms of deep learning techniques, such as transformers, to integrate better global user-item relationships beyond convolutional techniques.

## Data Availability

Data Availability

The original datasets used in this work are publicly available at:

Movielens datasets: GroupLens Research has collected and made available rating data sets from the MovieLens website (https://grouplens.org/datasets/movielens/). The data sets were collected over various periods of time, depending on the size of the set.

The Pinterest dataset contains over 1 million images associated with users who have "pinned" them. This repo contains the full Pinterest dataset released with the paper "Learning Image and User Features for Recommendation in Social Networks" by Xue Geng et al. in CSV form at: https://drive.google.com/file/d/0B0l8Lmmrs5A_REZXanM3dTN4Y28/view?resourcekey=0-jj1wN8qv3fyaP_6vFpGBEg

The preprocessed dataset used in our research is formatted the same as the paper (He et al., 2017) and is available at: https://github.com/Baboucar/HSFR/tree/master/HSFR/data

The source code used in this work is available at: https://github.com/Baboucar/HSFR/tree/master/HSFR

## Acknowledgments

## References


Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, 17(6), 734–749. https://doi.org/10.1109/TKDE.2005.99

Agarap, A. F. (2018). *Deep learning using Rectified Linear Units (ReLU)*. http://arxiv.org/abs/1803.08375

Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. *ArXiv*.

Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, *21*(6), 1487–1524. https://doi.org/10.3233/IDA-163209

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834–848. https://doi.org/10.48550/arxiv.1606.00915

Chen, M., Ma, T., & Zhou, X. (2022). CoCNN: Co-occurrence CNN for recommendation. *Expert Systems with Applications*, *195*, 116595. https://doi.org/10.1016/J.ESWA.2022.116595

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for youtube recommendations. *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems*, 191–198. https://doi.org/10.1145/2959100.2959190

Gasmi, S., Bouhadada, T., & Benmachiche, A. (2020). Survey on Recommendation Systems. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3447568.3448518

Geng, X., Zhang, H., Bian, J., & Chua, T.-S. (2015). *Learning Image and User Features for Recommendation in Social Networks*.

Guy, I., Zwerdling, N., Ronen, I., Carmel, D., & Uziel, E. (2010). Social media recommendation based on people and tags. *SIGIR 2010 Proceedings - 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 194–201. https://doi.org/10.1145/1835449.1835484

He, X., Du, X., Wang, X., Tian, F., Tang, J., & Chua, T. S. (2018). Outer Product-based Neural Collaborative Filtering. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 2227–2233. https://doi.org/10.48550/arxiv.1808.03912

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. *26th International World Wide Web Conference, WWW 2017*, 173–182. https://doi.org/10.1145/3038912.3052569

Javed, U., Shaukat, K., Hameed, I. A., Iqbal, F., Alam, T. M., & Luo, S. (2021). A Review of Content-Based and Context-Based Recommendation Systems. *International Journal of Emerging Technologies in Learning*, *16*(3), 274–306. https://doi.org/10.3991/IJET.V16I03.18851

Koren, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 426–434. https://doi.org/10.1145/1401890.1401944

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. https://doi.org/10.1109/MC.2009.263

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. https://doi.org/10.1109/5.726791

Liu, H., Liu, H., Ji, Q., Zhao, P., & Wu, X. (2020). Collaborative deep recommendation with global and local item correlations. *Neurocomputing*, *385*, 278–291. https://doi.org/10.1016/j.neucom.2019.12.087

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems*, *74*, 12–32. https://doi.org/10.1016/J.DSS.2015.03.008

Pan, Y., He, F., & Yu, H. (2020). A correlative denoising autoencoder to model social influence for top-N recommender system. *Frontiers of Computer Science*, *14*(3). https://doi.org/10.1007/S11704-019-8123-3

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*, 285–295. https://doi.org/10.1145/371920.372071

Singhal, A., Sinha, P., & Pant, R. (2017). Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works. *International Journal of Computer Applications*, *180*(7), 17–22. https://doi.org/10.5120/ijca2017916055

Tran, T., Liu, X., Lee, K., & Kong, X. (2019). Signed Distance-based Deep Memory Recommender. *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*, 1841–1852. https://doi.org/10.1145/3308558.3313460

Xin, X., Chen, B., He, X., Wang, D., Ding, Y., & Jose, J. M. (2019). CFM: Convolutional factorization machines for context-aware recommendation. *IJCAI International Joint Conference on Artificial Intelligence*, *2019-August*, 3926–3932. https://doi.org/10.24963/IJCAI.2019/545

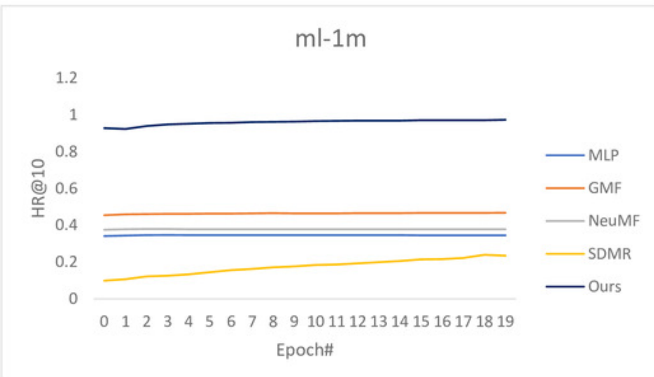
Xue, H. J., Dai, X. Y., Zhang, J., Huang, S., & Chen, J. (2017). Deep matrix factorization models for recommender systems. *IJCAI International Joint Conference on Artificial Intelligence*, *0*, 3203–3209. https://doi.org/10.24963/IJCAI.2017/447

Zhang, H., Shen, F., Liu, W., He, X., Luan, H., & Chua, T. S. (2016). Discrete collaborative filtering. *SIGIR 2016 - Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 325–334. https://doi.org/10.1145/2911451.2911502

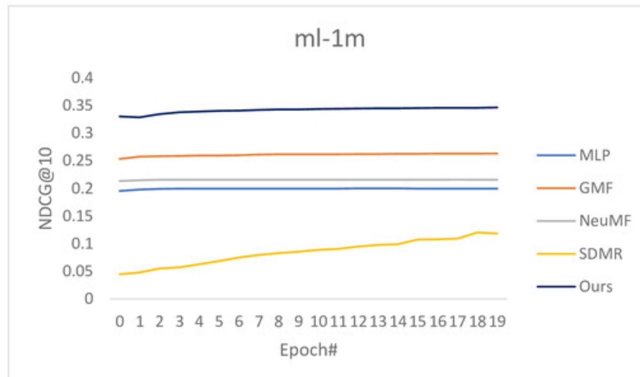
Zhang, W., Yu, X., & He, X. (2018). Learning Bidirectional Temporal Cues for Video-Based Person Re-Identification. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(10), 2768–2776. https://doi.org/10.1109/TCSVT.2017.2718188

# Figure 1

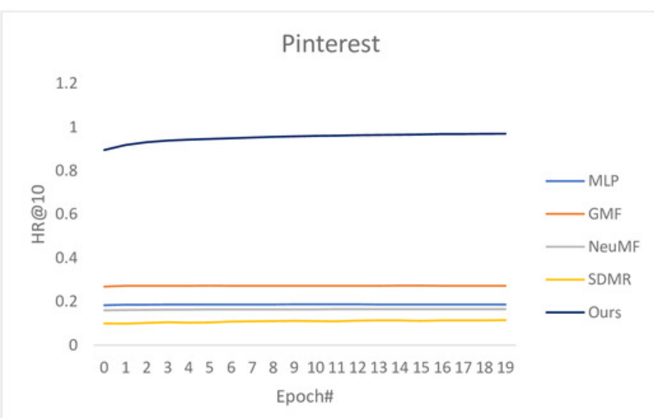
Figure 1 shows the graphical representation of the performance of our best-proposed method compared to the baselines using HR@10 and NDCG@10. (a)(b) and (c)(d) demonstrate the performance on the MovieLens and Pinterest datasets, respectively.



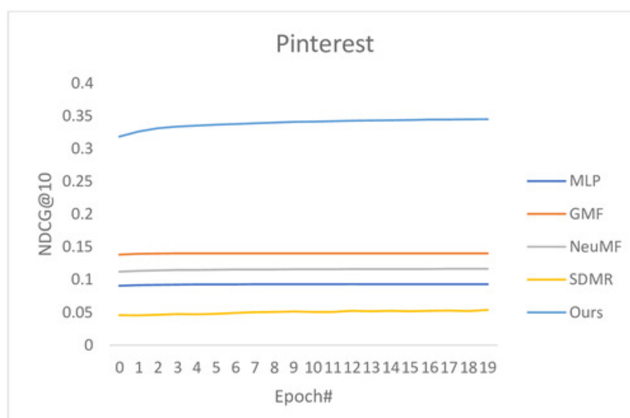

(a) HR on Movielens




(b) NDCG on Movie lens



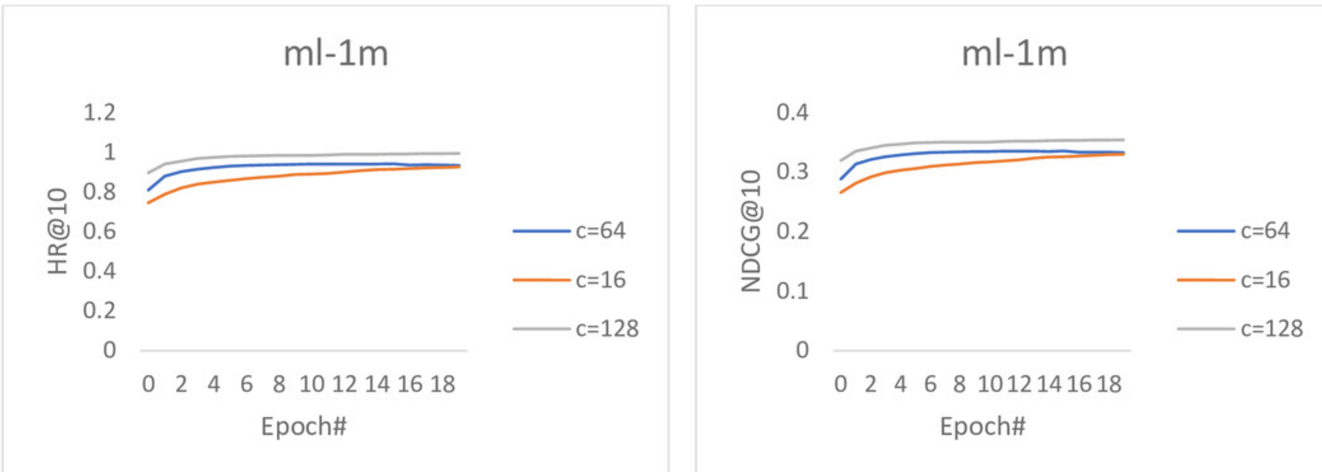

(c) HR on Pinterest




(d) NDCG on Pinterest

# Figure 2

Figure 2 shows the performance of our best model using different channel sizes on the movielens dataset



## Table 1 (on next page)

compares different models when generating top-k recommendations on two datasets. k $\in \{10\}$. The boldface denotes the persistently increased scores of our proposed networks.

**Table 1:** Comparison between different models when generating top-k recommendation on two datasets. k ∈ {10}. Boldface denotes the persistently increased scores of our proposed networks.

| | Movielens | | Pinterest | |
|---|---|---|---|---|
| | HR@K | NDCG@K | HR@K | NDCG@K |
| MLP | 0.3474 | 0.1997 | 0.1874 | 0.0932 |
| GMF | 0.4680 | 0.2633 | 0.2728 | 0.1402 |
| NeuMF | 0.301 | 0.2159 | 0.2267 | 0.1167 |
| SDMR | 0.2397 | 0.1203 | 0.1148 | 0.0537 |
| DMF | 0.2051 | 0.1143 | 0.2057 | 0.0787 |
| ONCF | 0.3874 | 0.2004 | 0.2780 | 0.1350 |
| Cocnn | 0.5796 | 0.3288 | - | - |
| **CNN-Based** | **0.9305** | **0.3314** | **0.9301** | **0.3313** |
| **CNN-Based+HSFM** | **0.9589** | **0.3416** | **0.9472** | **0.3374** |
| **CNN-Based +HSFM +GMF (ours)** | **0.9733** | **0.3467** | **0.9691** | **0.3452** |

## Table 2(on next page)

shows the performance of our proposed model against the baselines on RMSE, MAE, and BPR Loss.

Table 2: show the performance of our proposed model against the baselines on RMSE, MAE, and BPR Loss

| | ML-1M | | |
|---|---|---|---|
| | RMSE | MAE | BPRLOSS |
| MLP | 0.3218 | 0.1543 | 0.1281 |
| GMF | 0.327 | 0.1999 | 0.8028 |
| NeuMF | 0.3181 | 0.1533 | 0.1321 |
| SDMR | 8.705 | 7.6734 | 0.6933 |
| DMF | 0.2565 | 0.1324 | - |
| ONCF | 0.2818 | 0.2358 | 0.0158 |
| Cocnn | 0.5413 | 0.3666 | 0.0861 |
| **CNN-Based +HSFM +GMF (ours)** | **0.0226** | **0.0759** | **0.0023** |

## Table 3 (on next page)

shows the performance of the stack@n of the interaction map as the input to our simple Cnn-based proposed model. $n \in \{1, 2, 3\}$.

**Table 3:** show the performance of the stack@n of the interaction map as the input to our simple Cnn-based proposed model. n ∈ {1, 2, 3}.

| | Movielens | | Pinterest | |
|---|---|---|---|---|
| | HR@K | NDCG@K | HR@K | NDCG@K |
| Stack@1 | 0.9042 | 0.3221 | 0.9198 | 0.3276 |
| Stack@2 | 0.9563 | 0.3407 | 0.9390 | 0.3345 |
| Stack@3 | 0.9767 | 0.3479 | 0.9473 | 0.3374 |