

## RESEARCH ARTICLE

# A Community-Driven Deep Collaborative Approach for Recommender Systems

SOFIA BOURHIM<sup>ID</sup>, LAMIA BENHIBA, AND M. A. JANATI IDRISSE

Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes (ENSIAS), Mohammed V University, Rabat 10000, Morocco

Corresponding author: Sofia Bourhim (sofia.bourhim@um5s.net.ma)

**ABSTRACT** Recommender systems (RS) are increasingly leveraging the power of graphs to enhance accuracy. However, we stipulate that existing methods don't take into consideration the inherent behavior of communities and the interaction between all the sub-groups of the network.

In this work, we develop a Deep Graph-based Collaborative Filtering recommender system (DGCF), which incorporates the concept of community profiling and leverages the power of Graph Neural Networks. DGCF utilizes multiple graphs to exploit all types of information from the different user interactions. It extracts the overlapping communities from the homophily user-user graph and also integrates the high-order information from the user-item bipartite graph. We conduct experiments and evaluate the DGCF on the MovieLens datasets (ML-100K and ML-1M), and Douban dataset. Our experiments reveal significant improvements over a number of the latest deep learning models for recommender systems. Results also support that DGCF has the potential to render better recommendations as it extracts deep relationships using the community structure.

**INDEX TERMS** Collaborative filtering, graph neural networks, recommender systems, community profiling.

## I. INTRODUCTION

Recommender systems (RS) play a vital role in providing users with a personalized experience in various domains, from e-commerce to social media. RS are at the core of a several online services providers such as Amazon, Netflix, YouTube, etc. RS can be formalized as a link prediction problem: in order to estimate the user preference regarding an item  $i$ , we need to learn the representation of the user  $h_u$  and the item  $h_i$ . Then, we calculate the preference score for user on the item (as a probability) using a score function between both embeddings. The score function can be either a dot product, MLP, etc.

There are majorly three different types of approaches for RS: Content based filtering (CB) [1], Collaborative Filtering (CF) [2], and Hybrid approaches [3].

- Content based filtering compute recommendations by learning from information about items' features rather than using users' interactions and feedback.

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina<sup>ID</sup>.

- Collaborative Filtering RS make recommendations by learning the similarity between users through the user-item historical interactions, either explicit or implicit feedback.
- Hybrid RS combine CF and CB techniques to benefit from their complementary advantages.

Recently, deep learning (DL) has stepped into the world of RS and has shown promising results by outperforming all the traditional techniques. One of the core strengths of deep learning in RS is its ability to capture hidden patterns through the representation of users-items interactions.

Deep learning based recommendation models can be divided into two main categories [4]:

- Recommendations with Neural Blocks: use one specific deep learning technique (Convolutional neural network CNN, Recurrent neural network RNN...).
- Recommendations with Deep Hybrid Models: use a variety of deep learning techniques. They integrate multiple neural building blocks that complement each other.

Interactions in recommender systems can be viewed as a bipartite graph. Graphs are a powerful tool to encode inter-

actions, and combined with DL, they generate deep graph models such as GNNs and their variants. Graph Neural Networks (GNNs) [5] are a type of deep learning architectures designed to extract deep relations on graph data for different applications. GNNs operate directly on the graph structure to accomplish various tasks: nodes classification, link prediction, graph classification, etc.

**Research objective:** In this paper, we build our approach on the novel concept of 'Community profiling'. In sociology, Community profiling is a social research method that entails constructing a full profile of a community's natural needs and resources of the members' active participation, with the purpose of devising and implementing a strategy to address the concerns identified [6]. We define Community Profiling (CP) as the process of building a detailed picture of a target community in a network. CP helps understand the profile of a community by taking into account its interests, values, attitudes and interactions with other communities. Developing a community profile provides a detailed insight on the collective behavior of users' communities. However, deriving a community profile from its members' personal profiles is a challenging task.

In this paper, we propose a novel deep graph recommendation framework based on community profiling and a model-based CF approach (DGCF), with two key novelties:

- Capturing the behavioral similarity between the user and the item by applying message passing on the bipartite graph while aggregating and updating the state of the nodes when learning with GNN.
- Capturing the proximity similarity of users and their belonging to sub-groups by modeling a user-user graph, while learning simultaneously on the bipartite graph to capture the behavioral similarity signal, and then employing an information fusion layer that integrates both information provided from the user-item bipartite graph and the user-user graph.

The core idea is to enhance the RS accuracy by capturing different types of signals on multiple graphs. It means a more refined representations of user nodes. We conduct intensive experimental studies on a real-world dataset. Results demonstrate the superior performance of DGCF compared to the latest state of the art models.

**Paper organization.** The paper starts with Section 2 that reviews the relevant literature. Section 3 introduces the proposed deep learning-based recommender system (DGCF). The results of applying DGCF to the MovieLens and Douban Datasets are shown and discussed in Section 4. Finally, we discuss potential research directions.

## II. LITERATURE REVIEW

Our novel approach integrates concepts from 1) Model-based Collaborative Filtering methods, and 2) Graph-based Recommender System models. Given the extensive scope of our study, we focus on works that are the most relevant to the DGCF approach.

### A. MODEL-BASED COLLABORATIVE FILTERING METHODS

Model-based CF methods is a subgroup of the collaborative-filtering models in recommender systems. It learns the similarities between users and items by extracting information from the data set to build a model that can generate recommendations. There are common classical approaches for model-based CF such as clustering [7], classification [8], Latent model [9], Markov decision process (MDP [10]), and the most extensively used method Matrix Factorization [11] and its variants NMF [12], PMF [13], and BNMF [14]. However, in recent years, various model-based CF methods have been extensively using deep learning techniques and architectures to capture different types of signals that translate the behavior of the user. Deep learning-based recommendation models are more powerful than traditional approaches [15]. Modeling the non-linearity in data allows deep learning techniques to capture complex and intricate user-item interaction patterns. Additionally, they help automate the feature engineering process and thus reduce the efforts in hand-crafting the features. Existing deep learning-based recommendation models make use of one or more deep learning techniques. Because of the flexibility of deep neural networks, combining different strategies produce more powerful hybrid models [4].

### B. GRAPH-BASED RECOMMENDER SYSTEM MODELS

Graph-based RS is a novel category of CF models. It models users' preferences directly on a Graph structure. The graph can be an abstraction of different objects, such as users, items, and attributes. The use of graphs is a promising direction for building a more effective RS because they effectively capture all types of interactions (non-linear and non-trivial) between all objects. Previous works used random walk approaches on graphs [16], [17]. These methods take as input a graph from the user-item interactions. They start from a node  $j$  and then choose, at random, one of his neighbors and move to it. The same step is repeated  $t$  times until all nodes are processed. The random walk algorithm is used to rank items based on the preferences of the users. However, those methods are less effective and lack model parameters to optimize the objective function. HOP-Rec [18] is also a baseline method that combines the factorization approach and the graph model. It begins by taking a random walk over the user-item graph, then it trains the matrix factorization with BPR to build the recommender model. However, this method only uses high-order connectivity to enhance the training data. Among all deep learning models, Graph Neural Networks (GNNs) [5] are the most dominant technique for recommender systems nowadays. Their ability to learn from graph-structured data enables them to capture different types of interactions between nodes. The intuition behind GNNs is that nodes are naturally defined by their neighbors and connections. GNNs begin by accumulating feature information from the users' neighbors, then combine the aggregated data with the node's current state. The technique is repeated until a stable equilibrium is reached. Later on, various GNN

**TABLE 1. Existing literature reviews in the domain of “Overlapping Community Detection”.**

Paper	Year
Overlapping Community Detection in Networks: The State-of-the-Art and Comparative Study [41]	2013
Overlapping Community Discovery Methods: A Survey [45]	2014
A review on overlapping community detection methodologies [46]	2017
Overlapping Community Detection of Complex Network: A Survey [47]	2019
Recent trends on community detection algorithms: A survey [48]	2020
A Review on Community Detection algorithms and Evaluation Measures in Social Networks [49]	2021
A Survey of Community Detection Approaches: From Statistical Modeling to Deep Learning [50]	2021
A Comprehensive Survey on Community Detection with Deep Learning [27]	2021

derivatives have emerged such as GCN [19] which iteratively aggregates information from neighbors by approximating the first-order eigendecomposition of the graph Laplacian, GraphSage [20], GAT [21], GGNN [22] and many more. Therefore, more recent works have started to apply GNN to recommender systems [23], [24], [25], [26]. LightGCN [23] is a variant of GCN, it only uses the main GCN component which is the neighborhood aggregation for the recommendation task. It learns and extract user and item embeddings from the user-item bipartite graph, with the final embedding being the weighted sum of the embeddings acquired at all levels. The embedding of all layers is averaged using LightGCN. The collaborative signal stored as a high-order connection in the embedding function is used by Neural Graph Collaborative Filtering (NGCF) [24], which incorporates user-item interactions into the GCN model. PinSAGE [25] applies GCN on the item-item graph. It generates the item embedding from the graph structure (global and local) and the feature information of the item. Multi-GCCF [26] incorporates three types of graphs: a user focus perspective with user-user, an interaction focus with user-item and an item focus perspective with item-item, in order to learn the final representation of users and items. The user-item bipartite graph uses two GCN layers, and for the user-user and item-item, it uses one GCN layer.

### C. COMMUNITY DETECTION TECHNIQUES

In network analysis, community detection is a critical task. The goal is to discover and extract sub-structures in a network. Most of the works and efforts are directed toward defining efficient methods for finding disjoint communities in a network. Deep learning approaches for detecting communities have been proposed in numerous works [42], [43], [44]. However, in real-world networks, communities overlap, which means that nodes in the graph belong to many groups. In our novel approach, we aim to define, discover and extract sub-structures of the graph using Community detection. Some researchers have conducted surveys and systematic literature reviews to provide insightful information about overlapping community detection. Table 1 lists the existing literature in this field.

CNNs, GANs, and auto-encoders are the three most commonly used deep neural architectures in community detection: [51] used a CNN model to detect communities in topologically incomplete networks, which have some edges missing when compared to real-world networks. Reference [52] included sparse matrix convolution within a CNN

framework to deal with the highly sparse representations associated with adjacency matrices. Furthermore, [54] used auto-encoders to address the matches between the network topology and node attributes by developing a graph regularized auto-encoder approach. In the GNN spectrum, [53] introduced a non-backtracking operator to define the edge adjacency, others [55] used the Markov random field and combined it to an attributed network to detect communities. Despite all the work that have been done, it is still challenging to generate community embeddings instead of creating user embeddings: how to capture the relation between the node in the graph and the community structure (moving from 1-hop or more from nodes in the graph to sub-graphs instead).

### III. METHODOLOGY

This section outlines the main aspects of the methodology used for our DGCF framework, depicted in Figure 1. DGCF contains three main key components. The first layer is a Community Encoding layer (CE) that encodes latent information based on the user-user similarities graph by extracting overlapping sub-structures of the graph. The second is a Bipartite Graph Convolutional Networks encoder (EB-GCN) that generates representations of users and items and captures the collaborative signal in the user-item interaction bipartite graph. The outputs of both layers are federated in an information fusion layer (IF) that aggregates the embeddings from different perspectives.

#### A. COMMUNITY ENCODING LAYER (CE)

In order to identify community profiles, we first need to detect those communities. We thus use a community Encoding layer (CE) to compute the homophily network’s sub-communities from the user-user graph. In the CE layer, we render an affiliation matrix by combining GCN and the Bernoulli–Poisson model. We create a user-user graph  $G$  in addition to the user-item bipartite network by computing jaccard similarity on the rating matrix.

We consider the overlapping community detection problematic as a probabilistic inference task. Detecting communities in this manner entails inferring the unseen affiliations of users to communities from the user-user network  $G$ , using a GCN architecture. We denote the affiliation of users into communities as  $F$ , and  $A$  as the binary adjacency matrix of the undirected unweighted graph  $G$  where  $N$  is the number of user nodes.

$$F := GCN_{\theta}(A) = ReLU(\hat{A}ReLU(\hat{A}W^{(1)})W^{(2)}) \quad (1)$$

where  $\hat{A} = D^{-1/2}\tilde{A}D^{-1/2}$  is an adjacency matrix of the graph  $G$  normalized,  $D$  is the diagonal degree matrix of  $A$ ,  $\tilde{A} = A + I_N$  is the adjacency matrix with self loops with  $N$  is the number of nodes in the graph, and  $W^{(1)}$  and  $W^{(2)}$  are the weights that we optimize using the GCN architecture. In the equation, we use the  $ReLU$  non-linearity to the output layer to ensure non-negativity of the affiliation matrix  $F$ . We aim to minimize the negative log-likelihood by finding the right parameters (weights)  $\theta$  in the GCN model, which is

formulated as:

$$\theta := \operatorname{argmin}_{\theta} \mathcal{L}(\operatorname{GCN}_{\theta}(A)) \quad (2)$$

where  $\mathcal{L}$  represents the negative log-likelihood of the Bernoulli–Poisson model.

$$\mathcal{L}(F) = - \sum_{(u,v) \notin E} \log(1 - \exp(-F_u F_v^T)) + \sum_{(u,v) \in E} F_u F_v^T \quad (3)$$

$F_u$  and  $F_v$  are the row vector of community affiliation  $F$  of node  $u$  and node  $v$  respectively and  $E$  is the set of edges that links nodes in the graph.

In order to optimize the  $F$  matrix, we update the parameters of the neural network architecture by minimizing the negative log-likelihood. In this encoding layer, we use 2-layer of GCN, with a hidden size of 128 and the final layer is the output of the number of communities to detect. We also apply the batch normalization and the dropout with a ratio of 0.5 to avoid overfitting. We take advantage of the distinct relationships conveyed by the two graphs by combining the CE and EB-GCN layers' outputs (user-item graph and user-user graph).

## B. BIPARTITE GRAPH CONVOLUTIONAL NETWORKS ENCODER (EB-GCN)

We propose a Bipartite Graph Convolutional Networks encoding layer (EB-GCN) to address the data sparsity problem in CF by generating two additional embeddings: the user embedding and the item embedding.

The embedding vectors of user  $u$  and item  $i$  are denoted by  $e_u$  and  $e_i$ , respectively, with  $d$  indicating the embedding size.

In this encoding layer, it takes as input the user-item bipartite graph where the set of type nodes ranges between user nodes and item nodes. The key idea is to capture the collaborative signal from all types of interactions in the network, and then learn the final representation of both user and item. For this, we use GNN algorithms on the bipartite graph to their maximum potential.

The EB-GCN layer exploits high-order connectivity from the user-item interaction. It leverages on the message-passing architecture of GNNs to encode the user and item nodes by iteratively aggregating information from the user's neighbors. The high-order propagation is translated as a stacking on  $l$  embedding layers. Each embedding layer encompasses the construction of the message and the aggregation of the message.

The construction of the message of a user-item  $(u,i)$  is defined as  $m_{u \leftarrow i}$ :

$$m_{u \leftarrow i} = h(e_i, e_u, p_{ui}) \quad (4)$$

where  $h(\cdot)$  is the function that encodes the message, where the embeddings of user  $u$  and item  $i$  are inputs, and the coefficient  $p_{ui}$  helps control the decay factor on each propagation and on each edge  $(u,i)$ .

The message encoding function  $h$  is defined as below:

$$m_{u \leftarrow i} = \frac{1}{\sqrt{|N_u||N_i|}} (W_1 e_i + W_2 (e_i \odot e_u)) \quad (5)$$

where  $W_1$  and  $W_2 \in \mathbb{R}^{d' \times d}$  are the trainable weights that extract propagated information and  $d'$  is the transformation size. The  $p_{ui}$  is set as the graph Laplacian norm  $1/\sqrt{|N_u||N_i|}$  where  $N_u$  and  $N_i$  represent the first-hop neighbors of user  $u$  and item  $i$ . After extracting and constructing the message, we aggregate the messages propagated from the user  $u$ 's neighborhood to enhance its representation.

The following represents the definition of the aggregation function:

$$e_u^{(1)} = \operatorname{LeakyReLU}(m_{u \leftarrow u} + \sum_{i \in N_u} m_{u \leftarrow i}) \quad (6)$$

The first element of the aggregation function represents the information retained by user  $u$ . The second element is the aggregation of all the information obtained and captured from his neighborhood. Similarly, we propagate information from adjacent users to derive the representation of the embedding  $e_i^{(1)}$  for item  $i$ .

We optimize the pairwise BPR loss [28], which is extensively used in the recommender systems field, to learn the model parameters. The loss function is as follow:

$$\operatorname{Loss} = \sum_{(u,i,j) \in N} -\ln(\sigma(\hat{y}_{ui} - \hat{y}_{uj})) + \lambda \|\theta\| \quad (7)$$

where  $N$  denotes the set pairwise training data with observed and unobserved interactions. The observed and unobserved user-item interactions are taken into account in the pairwise BPR loss. If we assume that items with which a user interacts are referred to as positive examples (observed), we can assume that items with which the user does not interact are referred to as negative samples (unobserved). The pairwise BPR loss assigns higher prediction values for observed than unobserved samples.  $\sigma$  is the sigmoid function while  $\theta$  incorporates all trainable model parameters that includes the weights, and  $\lambda$  controls the regularization. We also apply dropout techniques, particularly node dropout, to prevent the model from overfitting [28]. This technique is only used in the training of the model.

The EB-GCN layer generates two embeddings, one of the users and one of the items.

## C. INFORMATION FUSION

From the CE and the EB-GCN layers, we output the user embedding  $E_U$ , the item embedding  $E_I$ , and the community affiliation matrix  $F$ . In the information fusion (IF) layer, we summarize all the embeddings and define the fusion method.

We create the community profile  $CP$ , from three different outputs based on the two encoding layers  $E_U \in \mathbb{R}^{n \times d}$ ,  $E_I \in \mathbb{R}^{k \times d}$ , and  $F \in \mathbb{R}^{n \times c}$  where  $n$  is the number of users,  $k$  is the number of items,  $c$  is the number of communities, and  $d$  is the embedding size, as follow:

$$CP = (F^T \cdot E_U) \cdot E_I \quad (8)$$

Furthermore, in order to choose the fittest communities, we select the top two affiliations for each user. The fusion

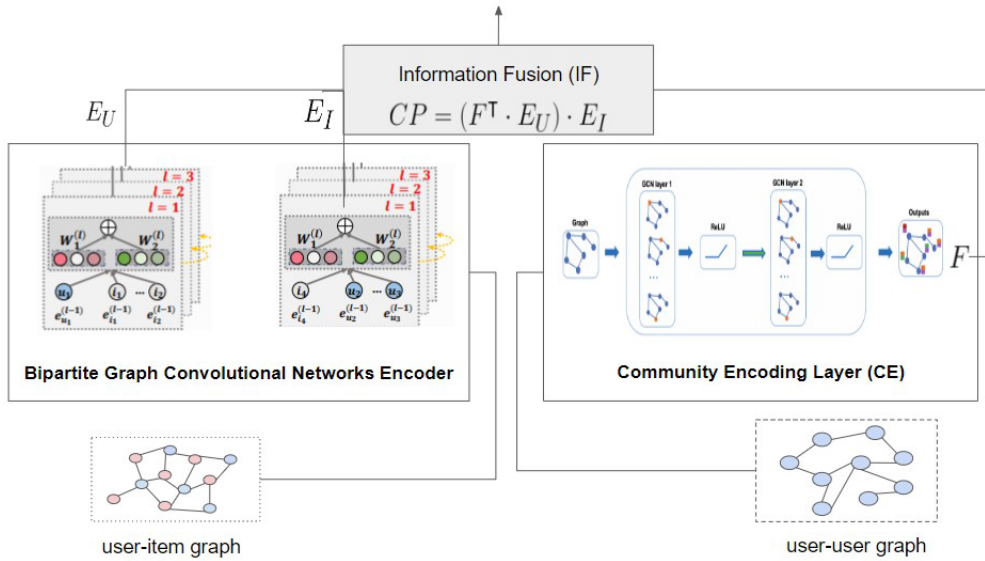


FIGURE 1. The general architecture of DGCF.

formula is helping capture the similarity between the embedding of user  $u$  and item  $i$  while taking into consideration the profile of communities that the user belongs to. As a result, each user’s item recommendations are based on the profile of their top two communities.

IV. EVALUATION AND RESULTS

The dataset we use to perform our experiments are one of the widely used datasets for recommendation research, which are MovieLens and Douban. We evaluate our model based on:

- The effectiveness of the recommender: whether DGCF achieves high-precision recommendations.
- The performance of the recommender compared to the latest benchmark reference models.

The conducted experiments are mainly focused on the final recommendations. However, in order to understand the different components of DGCF, we also perform a deep evaluation of the quality of communities in the CE component and investigate the viability of including the CE layer in our model.

A. DATASET AND EVALUATION METRICS

1) DATASET

We decide to work with three well known public, real-world dataset in recommendation research, the MovieLens datasets [29], and Douban dataset [56]. The summary for all datasets are shown in Table 2.

The MovieLens datasets is extracted from the MovieLens website, which is a movie recommendation service. The MovieLens-100k dataset consists of  $10^5$  user–movie ratings, 943 users and 1682 movies. The scale of the rating leverage from 1 to 5 (between dislike and like). Each user preference is represented as a tuple of four elements: user, item, rating

TABLE 2. Statistics of the datasets.

Dataset	# Users	# Items	# Interactions	Density
MovieLens 100K	943	1682	100,000	6.30%
MovieLens 1M	6040	3706	1,000,209	4.47%
Douban	3000	3000	136,891	1.52%

and timestamp, whereas the MovieLens-1M dataset consists of 6040 users and 3706 items, which is a larger size dataset.

The Douban dataset is a movie rating service in the mandarian communities, it contains 136, 891 rating for 3000 users and 3000 items.

In order to train the model, the data is divided into a training dataset (70% of the MovieLens dataset) and the test dataset (30% of the data).

2) METRICS

In order to evaluate the quality of recommendations, we use the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE) and Normalized Discounted Cumulative Gain (NDCG). RMSE is the disparity between the real and predicted values is defined as an error. The model’s performance and ability to predict future values improves as errors decrease.

To compute RMSE, we calculate the difference between predicted recommendations and the ground truth recommendations, which is called residual. For each data point, we compute the norm of residual, then the mean of residuals and take the square root of that mean.

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \tag{9}$$

**TABLE 3.** The overall performance comparison of Recommender systems.

Method	MovieLens100K			MovieLens1M			Douban		
	RMSE	MAE	NDCG	RMSE	MAE	NDCG	RMSE	MAE	NDCG
NGCF [24]	0.896	-	-	0.958	-	-	0.7234	0.5698	-
GraphRec [34]	0.904	0.802	-	0.843	0.7815	-	0.7257	0.569	-
IGMC [35]	0.905	0.741	0.899	0.857	0.784	0.903	0.721	0.591	0.939
SparseFC [36]	0.895	0.7	0.905	0.824	0.642	0.928	0.73	0.569	0.941
MG-GAT [37]	0.89	-	-	-	-	-	0.727	-	-
GLocal-K [38]	0.8889	0.695	0.906	0.8227	0.641	0.929	0.7208	0.562	0.943
GHRS [39]	0.887	-	-	0.833	-	-	-	-	-
DGCF	<b>0.804</b>	<b>0.611</b>	<b>0.95</b>	<b>0.81</b>	<b>0.602</b>	<b>0.935</b>	<b>0.701</b>	<b>0.55</b>	<b>0.95</b>

The **MAE** refers to the mean of the absolute values of each prediction error on all instances of the test data set. Prediction error is the difference between the actual value and the predicted value for that instance.

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j| \quad (10)$$

The **NDCG** is the average score that measures the consistency between the ranking of predicted ratings and the ground truth for each user. It is used to assess the accuracy of global and personalized ranking.

Furthermore, we utilize various metrics to assess the quality of the discovered communities. The DGCF accuracy improves as the communities become more refined:

- Coverage: The number of intra-community edges to the total number of edges is the coverage( $C$ ) of a partition  $C$  of a graph [30], [31]. The higher the coverage value, the higher the partition's quality.
- Density [41]: represents the average density of the detected communities, weighted by community size. The density is the ratio of the number of edges in the community to the maximum number of edges it can hold. The greater this metric's value, the better the community detection quality.
- Conductance [32]: represents the average conductance of the detected communities, weighted by community size. The conductance is the ratio between relationships that point outside a community  $C$  and the total number of relationships of  $C$ . The lower the conductance, the more "well-knit" a community is.
- Clustering coefficient [33]: represents the average clustering coefficient of the detected communities, weighted by community size. It measures the number of triangles in a community.

## B. BASELINES

To best evaluate the performance of our model, we consider different types of the latest deep learning models for recommender systems in our baseline selection:

- **SparseFC** (2018) [36]: is a neural network model where the weight matrices are reparametrized in terms of low-dimensional vectors using kernel functions.

- **NGCF**(2019) [24]: encodes the collaborative signal using high-order connectivities on a bipartite graph to generate embeddings.
- **GraphRec** (2019) [34]: a factorization model that uses the features from the user-item bipartite interaction graph.
- **IGMC** (2019) [35]: is based on extracting an h-hop enclosing subgraphs, performing a node labeling to define the user type of node and the item node, and then feed it into a GNN model.
- **MG-GAT** (2020) [37]: uses the attention mechanism to aggregate the neighbors' information of the user node and the item node, in order to learn the user/item representations.
- **GLocal-K** (2021) [38]: uses a pre-trained autoencoder with a kernelized weight matrix (2d-RBF kernel) and fine-tuned the auto-encoder with the rating matrix.
- **GHRS** (2022) [39]: It is a graph-based model that uses an autoencoder to extract new features based on combined attributes on a user-user similarity graph.

## C. PARAMETER SETTINGS

Pytorch is used to implement our DGCF model. In the EB-GCN layer, the size of the model's embedding is set to 64. We train our models using the Adam optimizer with default parameters, and the model parameters are initialized using the Xavier initializer. The learning rate is set to  $10^{-3}$ , the L2 normalization coefficient to  $10^{-5}$ , and the dropout ratio to 0.5. Furthermore, we employ an early stopping mechanism, in which the optimization is terminated if the training loss does not improve after 50 epochs.

## D. COMPARISON WITH BASELINES

The performance of our approach is reported in Table 3 when compared to seven baselines on three datasets. It shows that our model deep graph collaborative filtering model significantly outperforms all baselines on all three datasets by a large margin when looking toward rating accuracy (MAE), focusing on the importance of the bad rating prediction (RMSE) and how good the ordering is (NDCG). More precisely, our DGCF outperforms GLocal-K the third best model in ML-100K, best model in ML-1M and Douban, with 0.0849/0.084/0.044, 0.0127/0.039/0.006, 0.0198/0.012/0.007 improvement of RMSE/MAE/NDCG on

**TABLE 4. The overall comparison of quality metrics in detecting communities for the MovieLens 100K dataset.**

Metrics	Ground truth communities	Predicted communities
Coverage	0.885	0.981
Conductance	0.501	0.107
Density	0.134	0.758
Clustering coefficient	0.040	0.604

**TABLE 5. Communities' description for the MovieLens-100K dataset.**

Communities	Number of nodes
C1	6
C2	10
C3	404
C4	6
C5	410
C6	19
C7	20

three explicit datasets ML-100K, ML-1M, and Douban, which has the rating from 1 to 5.

This suggests that exploiting Homophily communities and utilizing multiple graphs to extract different embeddings deliver relevant items in the recommendation list.

Furthermore, to assess and verify the quality of communities extracted by our CE layer, we conduct a comparison of quality metrics between ground truth communities and predicted communities in Table 4 for the MovieLens 100K dataset. We obtain 7 communities detailed in Table 5. The initial Ground Truth communities is a randomized binary assignment of users in communities. Table 4 shows that the detected partitions have lower conductance, higher density and higher clustering coefficient than the Ground Truth communities.

#### E. ON THE EFFECTIVENESS OF THE COMMUNITY ENCODING LAYER (CE)

Our DGCF approach relies on a Deep Learning (DL) architecture to perform all steps, from community detection to community profiling and rendering recommendations. While results show that DGCF outperforms all the latest baseline methods, we set out to investigate the pertinence of our proposed DL-based architecture by trying to answer a main question: Is including a community encoding (CE) layer in a DL architecture really pertinent?

Our initial GCF [40] approach relied on distinct communities detection using classical network analysis metrics. It follows major steps from creating the homophily network using similarity metrics, to identifying communities and their key nodes, and finally computing recommendations to users using the key node's profile. The intuitive next step was to use the same assumptions and upgrade GCF by considering the overlaps between communities. We thus translated the community detection step into a CE layer in DGCF. To examine whether adding a community encoding layer in the DL architecture is relevant, we compared our results with different deep learning models but mainly with the NGCF

approach. NGCF is a DL-based approach that computes recommendations on a user-item bipartite graph. It generates embedding for each type of node: item and user, and feeds it into predictions. Our DGCF approach goes further by fusing information from the EB-GCN and CE layers. The NGCF only captures the similarity signal based on the behavior of the user towards items, while DGCF captures more signals mainly the community behavioral signal, which reflects the impact of the sub-group (in analogy to the impact of belonging to communities (or societies) on individuals in real life). The results in table 4 illustrate how including the community detection step improves overall performance and confirm our hypothesis that it is beneficial in adding contextual and topological information.

DGCF proves that using GCN layer for bipartite graphs doesn't capture all signals in the graph. It actually captures the signal of the similarity outside the properties of sub-groups in the graph. In other words, it doesn't take into consideration the local properties of communities while learning. Thus, integrating the CE layer is important, mainly for recommender systems, to capture the local and global properties of the network.

#### V. CONCLUSION AND FUTURE WORK

In this paper, a novel collaborative filtering approach is presented. It produces users' recommendations using community profiling. The model incorporates multiple graphs to integrate any and all contextual information latent in user-item relationships. The proposed model, DGCF, stacks various neural layers to construct users and items embeddings and learn users communities affiliations. It utilizes the learned communities profiles to render accurate recommendations.

In terms of RMSE evaluation metric, experimental results show that our DGCF model outperforms CF baseline models in recommender systems. DGCF, compared to the baselines models, is able to detect sub-groups topologies in a network which helps refine the embedding of the user and the computed recommendations.

For future work, we plan to enhance the DGCF model by:

- Improving the embeddings through adding new attributes and features such as users' geographic features, etc. The idea is to shed more light on possible hidden relationships lying within users features similarities.
- Including the temporal aspect into the DGCF model to handle the dynamic behavior of the user: capturing the time aspect signal is important in order to understand the overall behavior of the user through time, and the impact it has on his purchases (items).
- Investigating the range of applicability of the DGCF model outside of e-retailing into areas like biomedical applications, education, etc.

#### REFERENCES

- [1] M.-L. Wu, C.-H. Chang, and R.-Z. Liu, "Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices," *Expert Syst. Appl.*, vol. 41, no. 6, pp. 2754–2761, May 2014.

- [2] N. Nanas, M. Vavalis, and E. Houstis, "Personalised news and scientific literature aggregation," *Inf. Process. Manag.*, vol. 46, no. 3, pp. 268–283, May 2010.
- [3] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, *Recommender Systems: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 2010.
- [4] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, 2019.
- [5] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.
- [6] T. Blackshaw, *Key Concepts in Community Studies*. London, U.K.: Sage, 2010.
- [7] L. H. Ungar and D. P. Foster, "Clustering methods for collaborative filtering," in *Proc. AAAI Workshop Recommendation Syst.*, vol. 1, Menlo Park, CA, USA, 1998, pp. 114–129.
- [8] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2011.
- [9] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 89–115, Jan. 2004.
- [10] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier, "An MDP-based recommender system," *J. Mach. Learn. Res.*, vol. 6, no. 9, pp. 1265–1295, 2005.
- [11] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [12] L. Zhang, Z. Chen, M. Zheng, and X. He, "Robust non-negative matrix factorization," *Frontiers Electr. Electron. Eng. China*, vol. 6, no. 2, pp. 192–200, 2011.
- [13] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.
- [14] A. Hernando, J. Bobadilla, and F. Ortega, "A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model," *Knowl.-Based Syst.*, vol. 97, pp. 188–202, Apr. 2016.
- [15] B. Liu, Q. Zeng, L. Lu, Y. Li, and F. You, "A survey of recommendation systems based on deep learning," *J. Phys., Conf. Ser.*, vol. 1754, no. 1, Feb. 2021, Art. no. 012148.
- [16] M. Gori, A. Pucci, V. Roma, and I. Siena, "ItemRank: A random-walk based scoring algorithm for recommender engines," in *Proc. IJCAI*, vol. 7, 2007, pp. 2766–2771.
- [17] X. He, M. Gao, M.-Y. Kan, and D. Wang, "BiRank: Towards ranking on bipartite graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 57–71, Jan. 2017.
- [18] J.-H. Yang, C.-M. Chen, C.-J. Wang, and M.-F. Tsai, "HOP-Rec: High-order proximity for implicit recommendation," in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 140–144.
- [19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–14.
- [20] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.
- [21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–12.
- [22] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–20.
- [23] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 639–648.
- [24] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2019, pp. 165–174.
- [25] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 974–983.
- [26] J. Sun, Y. Zhang, C. Ma, M. Coates, H. Guo, R. Tang, and X. He, "Multi-graph convolution collaborative filtering," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2019, pp. 1306–1311.
- [27] X. Su, S. Xue, F. Liu, J. Wu, J. Yang, C. Zhou, W. Hu, C. Paris, S. Nepal, D. Jin, Q. Z. Sheng, and P. S. Yu, "A comprehensive survey on community detection with deep learning," 2021, *arXiv:2105.12584*.
- [28] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2012, pp. 1–10.
- [29] F. M. Harper and J. A. Konstan, "The MovieLens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 1–19, Jan. 2016.
- [30] U. Brandes, M. Gaertler, and D. Wagner, "Experiments on graph clustering algorithms," in *Proc. Eur. Symp. Algorithms*. Berlin, Germany: Springer, 2003, pp. 568–579.
- [31] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.
- [32] J. J. Whang, D. F. Gleich, and I. S. Dhillon, "Overlapping community detection using seed set expansion," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manag.*, 2013, pp. 2099–2108.
- [33] M. C. V. Nascimento, "Community detection in networks via a spectral heuristic based on the clustering coefficient," *Discrete Appl. Math.*, vol. 176, pp. 89–99, Oct. 2014.
- [34] A. Rashed, J. Grabocka, and L. Schmidt-Thieme, "Attribute-aware non-linear co-embeddings of graph features," in *Proc. 13th ACM Conf. Recommender Syst.*, Sep. 2019, pp. 314–321.
- [35] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–14.
- [36] L. Müller, J. Martel, and G. Indiveri, "Kernelized synaptic weight matrices," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, J. Dy and A. Krause, Eds., 2018, pp. 3654–3663.
- [37] A. Ugla, R. Ruiz, X. Dong, and A. Pentland, "Interpretable recommender system with heterogeneous information: A geometric deep learning perspective," *Int. J. Mech. Prod. Eng. Res. Develop.*, vol. 10, no. 3, pp. 2411–2430, 2020.
- [38] S. C. Han, T. Lim, S. Long, B. Burgstaller, and J. Poon, "GLocal-K: Global and local kernels for recommender systems," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 3063–3067.
- [39] Z. Z. Darban and M. H. Valipour, "GHRs: Graph-based hybrid recommendation system with application to movie recommendation," *Expert Syst. Appl.*, vol. 200, Aug. 2022, Art. no. 116850.
- [40] S. Bourhim, L. Benhiba, and M. A. J. Idrissi, "Towards a novel graph-based collaborative filtering approach for recommendation systems," in *Proc. 12th Int. Conf. Intell. Syst., Theories Appl.*, Oct. 2018, pp. 1–6.
- [41] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 1–35, Aug. 2013.
- [42] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning," in *Proc. IJCAI*, vol. 16, 2016, pp. 2252–2258.
- [43] J. Cao, D. Jin, L. Yang, and J. Dang, "Incorporating network structure with node contents for community detection on large networks using deep learning," *Neurocomputing*, vol. 297, pp. 71–81, Jul. 2018.
- [44] S. Cavallari, V. W. Zheng, H. Cai, K. C.-C. Chang, and E. Cambria, "Learning community embedding with community detection and node embedding on graphs," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 377–386.
- [45] A. Amelio and C. Pizzuti, "Overlapping community discovery methods: A survey," in *Social Networks: Analysis and Case Studies*. Vienna, Austria: Springer, 2014, pp. 105–125.
- [46] C. Rashmi and M. M. Kodabagi, "A review on overlapping community detection methodologies," in *Proc. Int. Conf. Smart Technol. Smart Nation (SmartTechCon)*, Aug. 2017, pp. 1296–1300.
- [47] Q. Chen and L. Wei, "Overlapping community detection of complex network: A survey," in *Proc. 20th Int. Conf. Parallel Distrib. Comput., Appl. Technol. (PDCAT)*, Dec. 2019, pp. 513–516.
- [48] S. Gupta and D. P. Singh, "Recent trends on community detection algorithms: A survey," *Mod. Phys. Lett. B*, vol. 34, no. 35, Dec. 2020, Art. no. 2050408.
- [49] P. Ponveni and J. Visumathi, "A review on community detection algorithms and evaluation measures in social networks," in *Proc. 7th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2021, pp. 1892–1899.
- [50] D. Jin, Z. Yu, P. Jiao, S. Pan, D. He, J. Wu, P. S. Yu, and W. Zhang, "A survey of community detection approaches: From statistical modeling to deep learning," 2021, *arXiv:2101.01669*.
- [51] X. Xin, C. Wang, X. Ying, and B. Wang, "Deep community detection in topologically incomplete networks," *Phys. A, Stat. Mech. Appl.*, vol. 469, pp. 342–352, Mar. 2017.
- [52] G. Sperlí, "A deep learning based community detection approach," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 1107–1110.



- [53] Z. Chen, X. Li, and J. Bruna, "Supervised community detection with line graph neural networks," 2017, *arXiv:1705.08415*.
- [54] J. Cao, D. Jin, and J. Dang, "Autoencoder based community detection with adaptive integration of network topology and node contents," in *Proc. Int. Conf. Knowl. Sci., Eng. Manage.*, 2018, pp. 184–196.
- [55] D. Jin, Z. Liu, W. Li, D. He, and W. Zhang, "Graph convolutional networks meet Markov random fields: Semi-supervised community detection in attribute networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 152–159.
- [56] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.



**LAMIA BENHIBA** is an Associate Professor of computer science with ENSIAS, Mohamed V University, where she teaches graduate and undergraduate courses in business intelligence, data engineering, and graph analytics. Her research interests include deep graph learning to tackle different research problems, such as recommender systems, bioinformatics, and knowledge graphs.



**SOFIA BOURHIM** is currently pursuing the Ph.D. degree with ENSIAS, Mohamed V University. Her primary research interests include artificial intelligence (AI), graph machine learning and its interdisciplinary applications to recommender systems (RS), and AI for social good topics. She is a Microsoft Ph.D. Fellow.



**M. A. JANATI IDRISSE** is a Full Professor of computer science with ENSIAS, Mohamed V University, Morocco. He is the Head of the Business Intelligence and Analytics Program. His research interests include graph theory, business intelligence, decision-aid methods, networks optimization, and project management.

...