Multimodal Graph-based Recommendation System using Hybrid Filtering Approach

Sorabh Gupta[1], Amit Kumar Bindal[2], Devendra Prasad[3]

[1]Ph.D. Research Scholar, Department of CSE, Maharishi Markandeshwar University, Mullana, India
[2]Professor, Department of CSE, Maharishi Markandeshwar University, Mullana, India
[3]Professor, Department of CSE, Panipat Institute of Engineering and Technology, Samalkha, India

E-mail address: [1]sorabhg2003@gmail.com, [2]bindalamit@gmail.com, [3]devendraacad@gmail.com

Abstract: This paper proposes a multimodal graph based recommendation system using a hybrid filtering approach. The proposed approach uses various sources of data and advanced graph based deep learning algorithms to provide more accurate and personalized recommendations to users. Our framework captures user and item attributes using text, images, videos, and metadata. We incorporate these attributes into the graph of user-item interactions using collaborative filtering and content based filtering. Graph convolutional networks (GCNs) help us identify collaborative filtering attributes. The intrinsic characteristics of items can be better understood and utilized with graph-based content based filtering. The proposed model initially classifies related users and items into groups using unsupervised clustering, then refines its recommendations using a cross-attention approach. In addition, we use a Variational Graph Autoendcoder (VGAE) approach that encodes intricate interactions inside a hidden space, hence enabling precise predictions of links. Experimental results show that the proposed model provides more accurate and personalized recommendations than existing models. We conduct comprehensive experiments using the publically accessible datasets of Movielens 1M, TikTok, MovieLens 10M and MicroVideo 1.7M. Our proposed model demonstrates superior effectiveness compared to the state-of-art multimedia recommender systems in various evaluation parameters such as precision, accuracy, recall, Normalized Discounted Cumulative Gain (NDCG), and F1-score.

Keywords: Content, collaborative, hybrid filtering, multimdodal, cluster similarity, graph convolutional network, variational graph autoencoder, link prediction.

1. INTRODUCTION

The existing web services are starting to employ recommendation algorithms more and more frequently [1]. Such algorithms almost always adjust their recommendations to meet the user's requirements. Utilizing these technologies, media streaming platforms and e-commerce sites [2] help users navigate massive information landscapes, which in turn helps consumers find new, relevant material. During the initial stages of the business, the primary focus was on developing online shopping recommendation systems [3]. It used simple algorithms to analyze customer purchase histories. Powerful recommendation systems that employ machine learning algorithms have been increasingly popular in recent years, emerging on a wide range of websites and platforms [4]. To improve the precision and accuracy of their product suggestions, e-commerce businesses are experimenting with recommendation systems. Individualized recommendations for media such as articles, books, songs, and movies are among the many services offered by these systems [5].

There are two main ways that recommender systems sift through data: collaborative filtering (CF) and content-based filtering (CBF). Collaborative Filtering recommends similar users' preferences. This sort of recommendation system classifies users into clusters of similar types and recommends each user based on its cluster's preferences [6, 7]. We divide it into two categories: item and user-based CF. Item-based CF compares items for similarity [8]. User-based CF recommends items based on two users' similarities [9]. Collaborative filtering has some issues; without enough data for new users or items, the cold start problem [10] is a major concern. Collaborative filtering systems often struggle with data sparsity [11]. When ratings are low in relation to users and items, recommendations are less reliable.

CBF matches items to users' tastes based on their contents [12]. Use client profiles, item summaries, and previous purchases to make suggestions. Content-based filtering can propose items after analyzing what users have done and what they like, but it can't offer very distinct items.

Both CF and CBF encounter certain limitations, which led to the creation of hybrid recommendation systems [13][14]. These systems incorporate multiple recommendation methods to enhance their shortcomings and optimize their strengths. A hybrid system might use CF to find items or users that are similar, and then it might use CBF to make suggestions that are specific to each user based on their own traits.

Hybrid recommendation systems can make suggestions more varied and accurate at the same time. They are especially effective at helping with data sparsity and the cold start problem because of the traits they offer. We can combine different approaches to assimilate collaborative and content-based methods. Within these methods are arithmetic mixtures, meta-level models, and feature augmentation. However, these systems primarily focus on integrating text data and user-item interaction. They don't always pay attention to the rich multi-modal data that is available in many applications.

In the digital world we live in now, we can get all kinds of information, like text, pictures, movies, and music. For example, people who shop online can watch videos, as well as write and post visual reviews of products. Social networking site posts allow users to include text, images, and videos, and users also interact with these posts. Using this multimodal data lets us understand user tastes and product qualities better, which could lead to better recommendation algorithms. Unimodal data cannot capture some qualities of an item or human behavior, but multimodal data can. When it comes to showing how someone feels and what they like, audio data can often provide aesthetic and contextual information. Visual information is a beneficial way to show contextual and semantic information, while textual data can show how users feel about a material and its meaning [15].

For personalized recommendations, these systems utilize multimodal data intended to offer insight into the user's preferences. But the assimilation is persistent in the absence of much data ie. text, images, video, and music. Combining data from multiple modalities efficiently needs complex algorithms and plenty of computational cost [16]. Multimodal recommendation systems are scarce, complicating the issue. Problems with real-time applications are becoming harder to solve [17]. This is caused by an increasing multimodal data complexity. We must handle multimodal datasets without overloading performance, ensuring that the load matches the capacity. The study intends to design and evaluate a hybrid recommender that uses multiple data sets and algorithms.

2. RELATED WORKS

The more straightforward antecedents of today's recommendation systems, which were dependent on explicit user-item interactions, have been replaced by more complex systems that combine multimodal data and advanced machine learning. Earlier systems widely used both content-based filtering (CBF) and collaborative filtering (CF), each with its own pros and cons. Sparsity and cold-start issues in CF might hinder user engagement with items.

Integrating CF and CBF methods with graph-based methods overcomes their drawbacks. The approach focuses on relevant feature matrices by dynamically integrating user and item domain information via cross-attention methods. This hybrid technique uses user behavior and item features to make more accurate and personalized recommendations [18]. Recent multimodal learning advances allow feature extraction and integration from text, photos, videos, and metadata [19]. RoBERTa [20], EfficientNet-V2 [21], and Video Transformer [22] may generate rich representations from their modalities. These elements create a comprehensive view of users and products, helping the recommendation engine capture complicated user preferences and item characteristics.

User-item feature encoding helps recommendation systems work by turning raw data into model-friendly representations. CF and CBF uses encoded user-item information to predict user preferences. The matrix factorization approaches, a type of latent feature encoding, have been frequently utilized to improve collaborative filtering using low-dimensional user-item relationship matrix representations [23]. Neural network-based embeddings have improved the capture of complicated user-item interactions by encoding high-dimensional data into dense, low-dimensional vectors [24].

Unsupervised clustering approaches as hierarchical and k-means are used in recommendation systems to find hidden user and item data structures [25]. Features can be used to cluster users and items and identify communities and content similarities. Cluster similarity-based graphs employ user and item cluster relationships to improve the system's capacity to offer relevant items without explicit interactions. Semantic clustering groups things or users by semantic similarity, frequently using NLP. This method uses contextual information in written descriptions, reviews, and other content. Word embeddings and deep learning models to cluster items by semantic content, improving content-based recommendations. Semantic clustering and collaborative filtering improve suggestion relevancy by integrating content similarity and user behavior patterns.

Many recommendation systems are based on bipartite networks, which have nodes for users and objects. This structure models user-item interactions like ratings and clicks. Bipartite graphs enable graph-based algorithms to find latent links and recommend objects. To increase recommendation accuracy, Graph Convolutional Networks (GCNs) transmit data across the bipartite graph in order to detect patterns of higher-order connections [26]. The graph autoencoders can learn the structure of user-item interactions from bipartite networks, improving recommendation performance [27].

Due to their capacity to capture complex user-item relationships, graph-based recommendation systems are popular. User-item bipartite graphs express interactions, making graph neural networks (GNNs) for collaborative filtering easier. Graph Convolutional Networks (GCNs) and GraphSAGE aggregate information from surrounding nodes to improve recommendation embeddings for users and items [28][29].

VGAEs provide a powerful foundation for graph link prediction, helping recommendation systems [30]. VGAEs model user-item interaction uncertainty and variability by learning probabilistic distributions over latent variables.

GCNs in the encoder and a probabilistic decoder allow VGAEs to capture complicated relationships and forecast future user-item interactions, making them suited for tailored recommendations.

Social networks, recommender systems, and biological networks use link prediction, a crucial network analysis problem, to predict future links. Traditional methods include network topology and similarity measurements such common neighbors, Jaccard coefficient, and preferred attachment [31]. Bayesian, stochastic block, and matrix factorization approaches like Singular Value Decomposition (SVD) improve prediction accuracy [32]. Supervised learning methods that treat link prediction as a binary classification task show potential [33]. By capturing complex network patterns, unsupervised methods like node embeddings like DeepWalk and Node2Vec improve prediction [34]. By using graph structures, deep learning developments like GNNs and GCNs have revolutionized link prediction [35][36]. Graph Attention Networks (GATs) dynamically weigh neighbor significance to improve predictions [37]. Model performance is often evaluated using AUC, accuracy, recall, and F1-score [38]. This multidisciplinary approach shows link prediction methods' evolution.

3. PROPOSED MODEL

As shown in Figure 1, our proposed model uses deep learning and multimodal data preprocessing. The model integrates user and item attributes into the recommendation system, merges data, trains the model, and improves user score prediction over cutting-edge techniques.

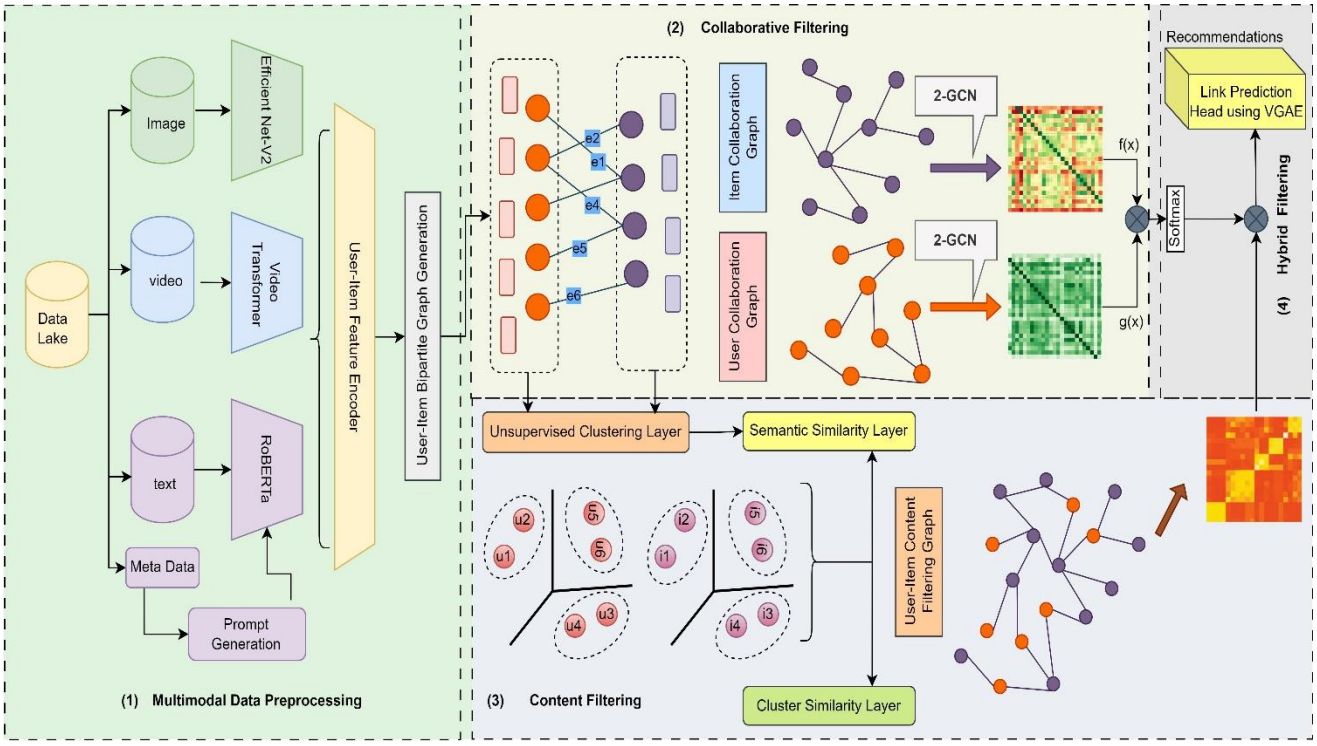


Figure 1 shows our proposed model's framework

3.1 Algorithm: Multimodal Graph-based Recommendation System using Hybrid Filtering Approach (MGRS-HFA)

A. Multimodal Feature Extraction, Fusion and User-Item Bipartite Graph Generation

i) Gather different modalities (text, image, video and metadata) from the various sources.

ii) Resize and normalize image modality and pass through EfficientNet V2 for image features extraction.

iii) Preprocess video and extract features using Video Transformer.

iv) Preprocess the text modality by splitting the corpus, removing stop words and punctuation, and performing lemmatization; tokenize text and generate RoBERTa embeddings from tokenized input.

v) Normalization processes for continuous variables in metadata from Prompt Generation and encoding for categorical variables transform into numerical vectors using RoBERTa.

vi) The user-item feature encoder was used to integrate features from all modalities into a single representation.

vii) Construct a user-item bipartite graph.

B. Collaborative Filtering with GraphSAGE

Input: User and Item feature matrix (X_U, X_I), User and Item adjacency matrix (A_U , A_I)

Output: Processed feature matrices H_U , H_I

i) Calculate user and item similarity for each user pair (u_i, u_j) and item pair (i_i, i_j), and also compute the similarity based coefficient for both pairs.

ii) Construct user and item graphs for each user pair (u_i, u_j) and item pair (i_i, i_j), if the similarity exceeds a threshold value, set the adjacency matrix to 1otherwise to 0.

iii) GCN Processing on Attributed Graphs and Obtaining the Processed Feature Matrix : Apply a Two layer GCN to user and item subgraphs to learn the collaborative filtering and updated adjacency matrix to fine-tune user and item features. Compute the final processed user H_U and item H_I feature matrices.

C. Content Filtering with GraphSAGE

Input: Mixed feature matrix X_{UI}, Mixed adjacency matrix A_{UI}

Output: Processed mixed feature matrix H_{UI}

i) Unsupervised clustering: Apply clustering to both user and item features and Compute cluster centroids of user cluster C_{Uj} and item cluster C_{Ij},

ii) Compute cluster similarity for each user cluster C_{Uj} and item cluster C_{Ik}

iii) for each user cluster C_{Uj} and item cluster C_{Ik}, construct a user-item cluster graph. If the similarity exceeds a threshold value, set the adjacency matrix to 1 otherwise to 0.

iv) Mixed Graph GraphSAGE Processing: Apply GraphSAGE to the mixed feature matrix and adjacency matrix. We refined the mixed features using the updated adjacency matrix. Calculate the final processed mixed feature matrix H_{UI}.

D. Cross-Attention, VGAE, and Recommendation Generation

Input: Processed feature matrices H_U , H_I, H_{UI}

Output: Recommendations

i) Define the query, key, and value matrices and compute the attention weights and cross-attention mechanism's output.

ii) Use the Variational Graph Autoencoder (VGAE) to learn probabilistic distributions over latent variables.

a) The Inference Model (Encoder): Calculate the mean and log variance. Sample Z latent variables from the inferred Gaussian distribution. Calculate the posterior distribution.

b) Decoder: Reconstruct the adjacency matrix by estimating link probabilities between nodes. Using the encoder's latent variables, determine the likelihood of each link's existence.

iii) Calculate Loss Function using reconstruction loss and KL divergence.

iv) Calculate the likelihood of node pairs for Link Prediction.

v) Rank the computed link probabilities to generate recommendations.

3.2 Framework of the Proposed Model

In this section, the overall proposed model consists of four components: multimodal data preprocessing, collaborative filtering, content based filtering, cross-attention, VGAE, and recommendation generation.

A. Multimodal Data Preprocessing

The model encompasses a range of data modalities, including text, image, video, and metadata, for building user and item representations. The feature extraction and multimodal data fusion processes, which are at the core of our recommendation system. The steps proceed as follows:

i) Data Collection and Preprocessing

The model starts with gathering multiple sources of data. In the text features, we utilize user reviews and item descriptions. The image data consists of product images and user-uploaded photos. The video data also includes multimedia content, such as trailers and reviews. Metadata is structured information such as item attributes (e.g., genre, price, category) and user demographics (e.g., age, location). To standardize and validate each data type, a specific preprocessing pipeline is necessary. We split the corpus and removed stop words, punctuation, and lemmatization for text data. The resized and normalized images serve as image-level input data. We divide each video into offset frames, known as keyframes, and extract the features accordingly. Standardizing metadata to ensure consistent records.

ii) Feature Extraction Models

For feature extraction from each data modality, we use specific models:

- Text data: Using a pre-trained RoBERTa model, we derive contextual embeddings. This paradigm provides dense vector representations and captures semantic subtleties.
- Image Data: EfficientNet-V2 extracts high-level features from images. This model's time efficiency and outstanding performance in image categorization tasks persuaded us to select it.
- Video Data: A Video Transformer model processes the video data, capturing the dynamic information within video sequences by analyzing sequential frames to extract temporal properties.
- Metadata: Normalization processes for continuous variables and one-hot encoding for categorical variables

transform metadata characteristics into numerical vectors.

iii) Feature Integration and Encoding

We create an integrated representation of users and items by combining the features retrieved from all modalities. In order to guarantee that the multimodal features are compatible and collectively informative, this integration makes use of alignment and concatenation techniques.

Next, we use a feature encoder to transform the unified features into a fixed-dimensional space suitable for further processing. Typically, this encoder consists of fully linked layers that determine the most important parts of the combined characteristics.

iv) Bipartite Graph of User-Item

The unified and encoded features were used for construct a user-item bipartite graph. User nodes represent the system's individual users. Item nodes represent the items the system offers (products, movies, music, etc.). An edge connects a user and an item node if there is an interaction between the user and item. These interactions can be explicit (e.g., purchases, ratings) or implicit (e.g., browsing history, clicks).

A bipartite graph $G = (U, I, E)$ entails of two distinct sets of vertices: *U* (users) and *I* (items), where *E* represents the edges between these sets. Each edge $e_{ui} \epsilon E$ connects a user $u \epsilon U$ and an item $i \epsilon I$, indicating some form of interaction or relationship (e.g., purchase, rating).

The multimodal feature extraction and data fusion ensures that our recommendation system leverages all available information to make highly accurate and personalized suggestions. By integrating diverse data types and extracting meaningful features, we establish a robust foundation for the subsequent graph-based learning and recommendation processes.

B. Collaborative Filtering

Beyond user-item interactions, we can construct collaboration graphs to capture relationships between users (user collaboration graph) or items (item collaboration graph).

i) User Collaboration Graph

This graph represents relationships between users. The model can create edges based on shared preferences, similar browsing behavior, or social connections. This graph helps identify user communities with similar interests, allowing the system to recommend items popular within those communities.

From the bipartite graph, we extract a user-user graph $G_U = (U, E_U)$ based on feature similarity among users. The edges E_U are defined based on a similarity metric $s(u, u')$ for $u, u' \epsilon U$, such as the cosine similarity of user feature vectors as shown in equation (1):

$$s(u, u') = \frac{v_u, v_{u'}}{\|v_u\| \|v_{u'}\|} \quad (1)$$

where v_u is the feature vector of user *u*.

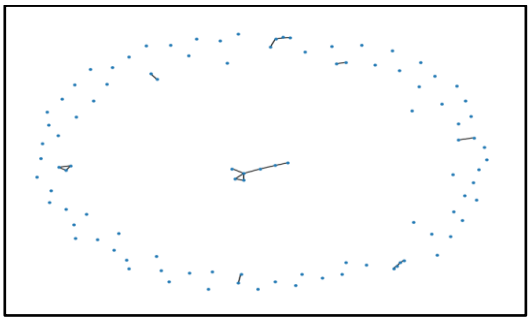


Figure 2 User Collaboration Graph of the MovieLens 1M dataset

ii) Item Collaboration Graph

This graph captures relationships between items. We can generate edges based on item co-purchases, content similarity, or complementary functionalities. This graph helps identify groups of similar items, allowing the system to recommend complementary items or substitutes based on user preferences.

Similarly, we form an item-item graph $G_I = (I, E_I)$ by connecting items $i, i' \epsilon I$ based on their similarity $s(i, i')$ as given in equation (2):

$$s(i, i') = \frac{v_i, v_{i'}}{\|v_i\| \|v_{i'}\|} \quad (2)$$

where $\mathbf{v}_i$ is the feature vector of item *i*.

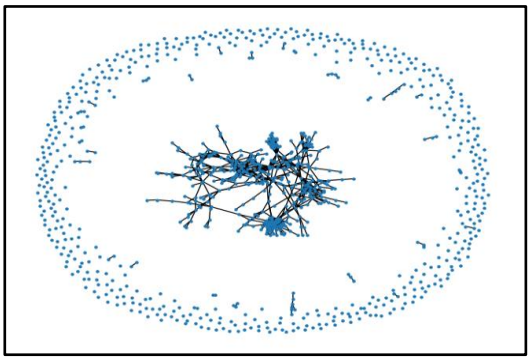


Figure 3 Item Collaboration Graph of the MovieLens 1M dataset

iii) GCN Processing on Attributed Graphs

We apply two layer GCNs separately to G_U and G_I to learn and refine the node (user or item) representations. Equation (3) expresses the propagation rule for a GCN layer.

$$H^{(I+1)} = \sigma(D^{-\frac{1}{2}} \widehat{A} \widehat{D}^{-\frac{1}{2}} H^{(I)} W^{(I)}) \quad (3)$$

iv) Obtaining the Processed Feature Matrix

We obtain the output feature matrices H_U and H_I for users and items, respectively, after processing through two-GCN layers. These matrices encapsulate the refined and graph structure and node interactions to create high-level

user and item representations. Each row in H_U and H_I corresponds to the new feature vector of a user or item, representing their embedded characteristics in the collaborative filtering context.

Various downstream tasks like recommendation, clustering, or classification can use the resulting feature matrices H_U and H_I, enhancing the system's understanding of complex patterns in user-item interactions.

C. Content Filtering

The user-item content filtering network improves the bipartite graph by adding nodes that represent item and user content. Item and user nodes communicate with their content feature nodes via edges.

This enhanced structure allows the system to filter user-item associations by content similarity. The system may connect a user who frequently watches comedies to item nodes that represent additional comedies, even if the user has not directly interacted with them. Content-based filtering uses thematic links between users and items to improve recommendation accuracy.

i) Unsupervised Clustering

Unsupervised clustering can discover hidden structures of data and increase recommendation accuracy. Using unsupervised learning, create clustering graphs for users and items. These graphs show user behavior and item quality by combining users and items with similar features. On build clusters, we use k-means or DBSCAN to cluster users and items differently. We designate each collection of users and items as U and I. In the dataset, each user $u\epsilon U$ and item $i\epsilon I$ are represented by v_u and v_i.

User clustering graphs display users as nodes. As shown in equation (4), edges connect clustered users. Users in these connections share interests or behaviors.

$$C_U = \{C_{U1}, C_{U2}, \dots \dots \dots \dots \dots C_{Uj}\} \quad (4)$$

C_{Uj} represents the *j*-th user cluster.

Nodes in an item clustering network represent items. Equation (5) connects clustered items, reflecting their content similarity or co-occurrence patterns.

$$C_I = \{C_{I1}, C_{I2}, \dots \dots \dots \dots \dots C_{Ik}\} \quad (5)$$

C_{Ik} represents the *j*-th item cluster.

Clustering graphs help the system understand user communities and item links based on raw interaction data. Analyzing these graphs can yield system insights.

ii) Content Filtering using Cluster Similarity-based Graphs

Using user and item clustering graphs, we can create a cluster similarity-based network for content filtering. This graph examines cluster relationships.

Nodes represent the previous stage's user and item clusters. High-similarity edges connect user and item clusters. Content-based feature analysis or common user preferences for cluster elements can measure this similarity. We must calculate user and item similarity after clustering.

First, compute each cluster's centroid. The centroid of a cluster C_{Uj}, can be found using equation (6).

$$C_{Uj} = \frac{1}{|C_{Uj}|} \sum_{u\epsilon UIj} V_u \quad (6)$$

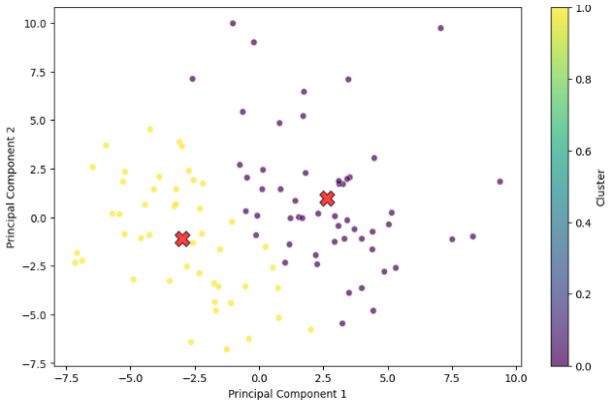


Figure 4 User Clusters of the MovieLens 1M dataset

In a similar way, equation (7) gives the cluster's centroid C_{Ij}:

$$C_{Ij} = \frac{1}{|C_{Ij}|} \sum_{i\epsilon CIj} V_i \quad (7)$$

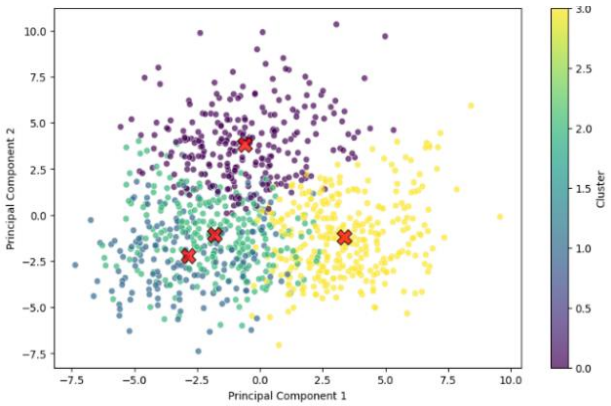


Figure 4 Item Clusters of the MovieLens 1M dataset

Use a similarity measure like cosine similarity in equation (8) to compute the similarity among each pair of user-item clusters.

$$s(C_{Uj}, C_{Ik}) = \frac{C_{Uj}, C_{Ik}}{\|C_{Uj}\| \|C_{Ik}\|} \quad (8)$$

The cluster similarity allows for CBF by recommending items from clusters similar to those a user has interacted with earlier. This approach personalizes recommendations by leveraging the intrinsic properties of items and user behavior patterns learned through unsupervised clustering.

iii) Content Filtering Graph

Create an integrated graph based on cluster-level

similarities that mixes user and item nodes. A graph $G = (V, E)$ using V as the nodes and E as the cluster-similar edges. For each user $u \in C_{Uj}$ and item $i \in C_{Ik}$:

$$(u, i) \epsilon E \quad \text{if} \quad s(C_{Uj}, C_{Ik}) > \text{threshold}$$

We can set this threshold based on a predefined value or derive it from the distribution of similarities.

We consider the feature vectors of nodes in the new graph G; v_u for users and v_i for items. Refine the node features, apply GCN to the constructed graph. Equation (9) provides the GCN propagation rule.

$$H^{(I+1)} = \sigma(D^{-\frac{1}{2}}\widehat{A}\widehat{D}^{-\frac{1}{2}}H^{(I)}W^{(I)} \quad (9)$$

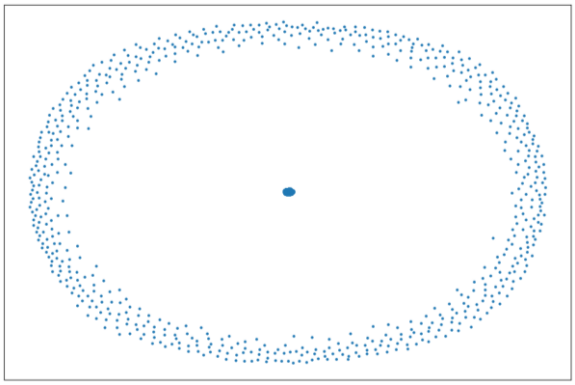


Figure 5 (a) Cluster Similarity of MovieLens 1M Users and Items Network Graph

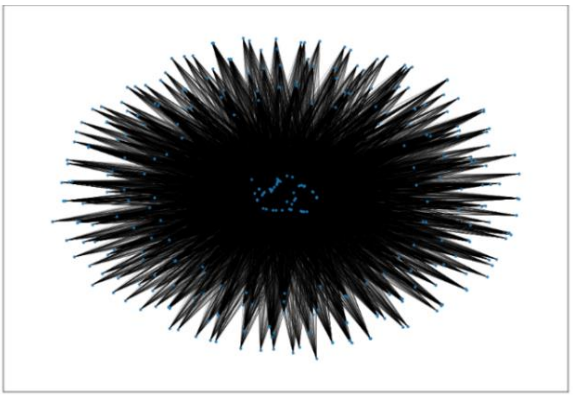


Figure 5 (b) Largest MovieLens IM dataset Network component

iv) Obtaining the Processed Feature Matrix

The final output feature matrix H, after being processed through the GCN layer, represents the refined embeddings for both users and items. These embeddings hold the connections learned from cluster-level interactions and graph convolutions. This makes it possible for tasks like prediction and recommendation to work better later on.

This approach uses unsupervised clustering and graph neural networks to improve content filtering reliability and context.

D. Hybrid Filtering

Most recommendation systems use CF, which uses user-item interaction data, or CBF, which uses item features. However, each method has limitations. CBF may struggle with restricted feature representation, while CF may have sparsity and cold-start issues. Hybrid approaches combine the strengths of both approaches to overcome these constraints. This hybrid filtering approach relies on cross-attention.

i) Cross Attention Mechanism

Dynamically combining user and item information is essential for hybrid filtering. Cross-attention learns the importance of features based on their relevance to the user and the item.

Representing Users and Items: We use embedding layers to convert user and item features to latent representations. These lower-dimensional representations capture user and item traits.

Equation (10) describes the attention mechanism:

$$Z = \text{Attention}\,(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (10)$$

Define the queries (Q), keys (K), and values (V) as follows in equation (11) :

$$Q = H_U, K = H_I \text{ and } V = H_{UI} \quad (11)$$

After addressing cross-attention, the mechanism computes the attention weights. When calculating the model's weights, we can see how much weight each user attribute should have when considering a certain item, and vice versa.

Equation (12) allows us to determine the weight of attention.

$$\text{Attention Weight}\,(A) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \quad (12)$$

Finally, we apply the learned attention weights to the user-item representations. In our recommendation, we use a weighting technique to pay attention to the preference of users and items and qualities.

The cross-attention method allows the model to focus on the relevant regions of the merged feature matrices. H_U , H_I, and H_{UI} to represent the input feature matrices. The cross-attention permits the model to focus on pertinent parts of the feature matrices when combining them. Let H_U , H_I, and H_{UI} be the input feature matrices.

Equation (13) shows the cross-attention process.

$$Z = AV \quad (13)$$

ii) Recommendation through Link Prediction

Predicting user preferences and then suggesting relevant items is the main objective of recommendation systems. For this purpose, link prediction in graphs is a method that has shown some potential. This section explores the Variational Graph Autoencoder (VGAE) within recommendation

systems for potential use in link prediction.

Finding the likelihood that an edge will connect two nodes is one of the primary objectives of graph link prediction. In the past, link prediction relied heavily on either hand-crafted attributes or really basic graph properties. But it's also conceivable that these approaches overlook the complex patterns and relationships in the data pertaining to interactions between items and users.

iii) Variational Graph Autoencoder (VGAE) for Link Prediction

Utilizing deep learning's capabilities, VGAE sidesteps the limitations of conventional methods. VGAE, a subclass of deep learning architectures, specifically handles graph data. This can help you understand how VGAE predicts connections in recommendation systems:

Making an item-based representation of the data is the initial step. Nodes represent users or items, whereas edges indicate interactions between nodes. Engagements include clicks, ratings, and purchases.

The user-item graph is processed by the VGAE encoder. Graph convolutional layers is used in this encoder to detention the intricate relationships.

The inference model aims to learn a probabilistic distribution over the latent variables (node embeddings). We employ two GCN layers for computing $\mu = \mathrm{GCN}_{\mu}(X, A)$ and $\log \sigma = \mathrm{GCN}_{\mu}(X, A)$ that share the weight matrix W_0.

Equation (14.1 & 14.2) illustrates how we derive the inference model from the Variational Graph Autoencoder.

$$q(Z|X, A) = \prod_{i=1}^{N} q(z_i|X, A) \quad (14.1)$$

$$q(z_i|X, A) = \mathcal{N}(z_i|\mu_i, \mathrm{diag}\,(\sigma_i^2)) \quad (14.2)$$

where μ_i and σ_i^2 are the mean and variance obtained from the GCN layers.

The VGAE encoder compresses the user and item representations into a lower-dimensional latent space. This latent space captures the most important features and relationships from the user-item graph.

The equation (15) provides the likelihood of a link between two nodes, u and v, based on the latent representations Z obtained via the VGAE.

$$p = A_{uv} = 1|Z) = \sigma(z_u^T z_v) \quad (15)$$

where nodes *u* and *v* have latent vectors z_u and z_v.

Using latent representations, the decoder reconstructs the original user-item graph. During this process, the VGAE predicts the likelihood of missing edges (i.e., unobserved user-item interactions).

Equation (16.1 & 16.2) shows how the decoder reconstructs the network structure using link prediction based on the encoder function's latent embedding.

$$p(A|Z) = \prod_{i=1}^{N} \prod_{j=1}^{N} p\left(A_{ij}\middle|z_i, z_j\right) \quad (16.1)$$

$$p\left(A_{ij} = 1\middle|z_i, z_j\right) = \sigma(z_i^T z_j) \quad (16.2)$$

where σ is the sigmoid function.

As expressed in equation (17), the VGAE loss function is made of two components: the reconstruction loss and the Kullback-Leibler (KL) divergence.

$$\mathcal{L} = \mathbb{E}_{q(Z|G)}[\log p\,(A|Z)] - \mathrm{KL}(q(Z|G)||p(Z)) \quad (17)$$

where $\mathbb{E}_{q(Z|G)}[\log p\,(A|Z)]$is the reconstruction loss and $\mathrm{KL}(q(Z|G)||p(Z)$ is the KL divergence.

iv) Generating Recommendations

To generate recommendations, compute the probability of new links for each user-item pair. Rank these probabilities to suggest the most likely new links (i.e., recommendations).

4. SIMULATION OF THE PROPOSED MODEL

This section will encompass case studies of MGRS-HFA, experimental scenarios, and performance evaluations.

A. Experimental Setup

i) Datasets

The study employs the framework of description and empirical evaluation on the platforms MovieLens 1M [40], MovieLens 10M [42], MicroVideo 1.7M [42] and TikTok [40] as shown in Table 1. Each dataset comprises comprehensive records of user-item interactions and numerous multimodal features.

TABLE 1. FOUR DATASET STATISTICS

Dataset	Interactions	Items	Users	Sparsity	Visual	Textual
Tiktok	726,065	76,085	36,656	99.99%	128	128
Movielens 1M	1,239,508	5,986	55,485	99.63%	2,048	100
MovieLens 10M	10,216,527	10682	51,001	98.12%	10,380	300
MicroVideo 1.7M	12,737,619	1,704,880	10,986	99.93%	984,983	200

ii) Baselines

MGAT [39]: User preferences determine gated and attention mechanisms for distinct techniques. We utilize comparable attention to determine method relevance.

MGCF [40]: Fusion enhances MGCF representation learning. Numerous GCN processes and attention strategies combine multimodal information to improve performance.

MCGCRS: This approach uses multimodal CLIP-guided graphs to predict links between users and items. It incorporates adversarial pretraining and Variational Graph Autoencoder (VGAE) approaches to accurately capture

complex interactions between users and items.

DIEN [41]: It improves DIN by adding a dynamic interest layer to track users' changing interests and eliminating batch normalization.

MUIR [42]: It aims to capture a wide range of user interests by combining several representations for personalized recommendations without using batch normalization.

HMCB-GRS: This approach for content-based filtering uses a hierarchical fusion, graph-based architecture with GCNs, meta-path-based GNNs, and bipartite graphs to better show how users interact with items, which makes personalized suggestions more accurate.

iii) Evaluation Metrics and Parameter Settings

The dataset is split into two ratio using a random allocation method, with a number ratio of 8:2. The top-K's performance is assess using widely recognized metrics such as Precision@K, Recall@K, Accuracy@K, F1-Score@K and NDCG@K to measure the top-K performance. A value of K=10 is set to all models and mean score value is calculated accordingly. The model is trained using Adam's optimizer, randomly initialized parameters using a Gaussian distribution, Sigmoid as the activation function with a binary cross-entropy loss and 0.001 learning rate. Figures 6–10 depict the outcomes.

TABLE 2. PERFORMANCE ANALYSIS OF THE MGRS-HFA WITH COLLABORATIVE RECOMMENDATION SYSTEMS

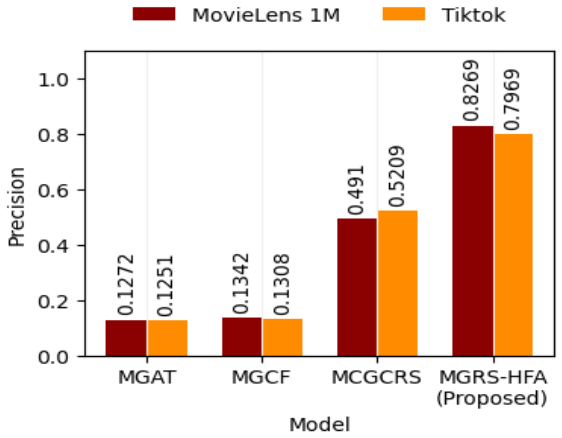
Model	MovieLens 1M				Tiktok			
	Precision	Recall	NDCG	F1-Score	Precision	Recall	NDCG	F1-Score
MGAT	0.1272	0.5412	0.3251	0.2060	0.1251	0.5965	0.3838	0.2068
MGCF	0.1342	0.5654	0.3448	0.2169	0.1308	0.6179	0.3987	0.2159
MCGCRS	0.4910	0.8506	0.3684	0.6226	0.5209	0.9378	0.4124	0.6698
MGRS-HFA (Proposed)	**0.8269**	**0.8718**	**0.6844**	**0.8484**	**0.7969**	**0.9452**	**0.7023**	**0.8643**
%Improvement	**68%**	**2%**	**86%**	**36%**	**53%**	**1%**	**70%**	**29%**

TABLE 3. PERFORMANCE ANALYSIS OF THE MGRS-HFA WITH CONTENT RECOMMENDATION SYSTEMS

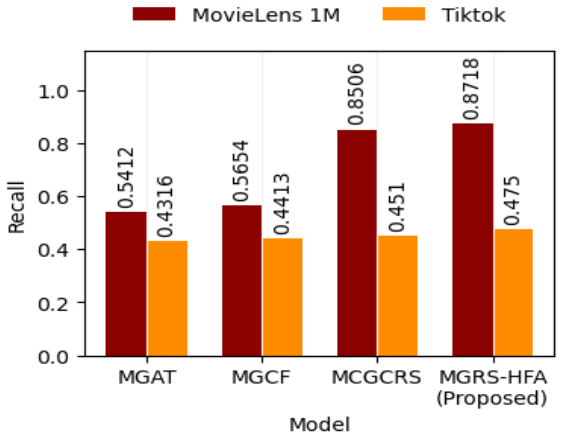
Model	MovieLens-10M				MicroVideo-1.7M			
	Precision	Recall	NDCG	F1-Score	Precision	Recall	NDCG	F1-Score
DIEN [24]	0.2820	0.4316	0.6899	0.3411	0.3898	0.0625	0.6892	0.1077
MUIR [25]	0.2917	0.4413	0.6992	0.3512	0.4018	0.0640	0.6978	0.1104
HMCB-GRS	0.2998	0.4510	0.6998	0.3602	0.4054	0.6173	0.7009	0.4894
MGRS-HFA (Proposed)	**0.5912**	**0.4785**	**0.8711**	**0.5285**	**0.6659**	**0.6485**	**0.8619**	**0.6568**
Improvement in %age	**97%**	**6%**	**24%**	**47%**	**64%**	**5%**	**23%**	**34%**

TABLE 4. PERFORMANCE ANALYSIS FOR MGRS-HFA WITH CONTENT AND COLLABORATIVE RECOMMENDATION SYSTEMS ON ACCURACY

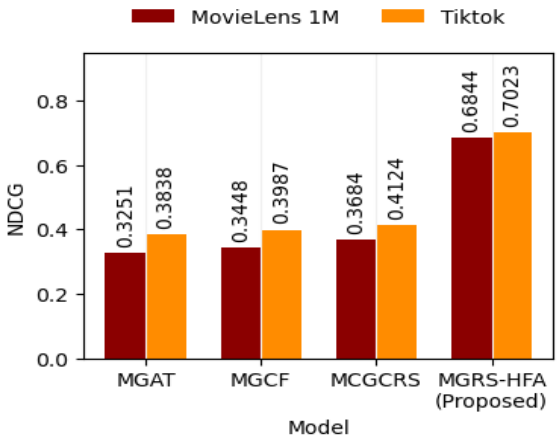
Model	MovieLens 1M	Tiktok	Model	MovieLens-10M	MicroVideo-1.7M
MCGCRS	0.4807	0.5413	HMCB-GRS	0.3535	0.3559
MGRS-HFA (Proposed)	**0.5182**	**0.5519**	**MGRS-HFA (Proposed)**	**0.5593**	**0.5295**
%Improvement	**8%**	**2%**	**Improvement in %age**	**58%**	**49%**



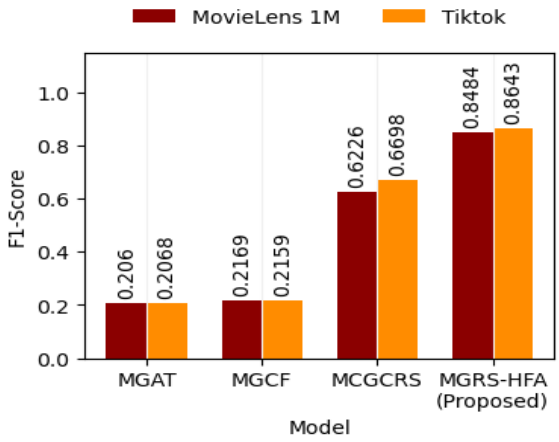
(a) Precision values



(b) Recall Values

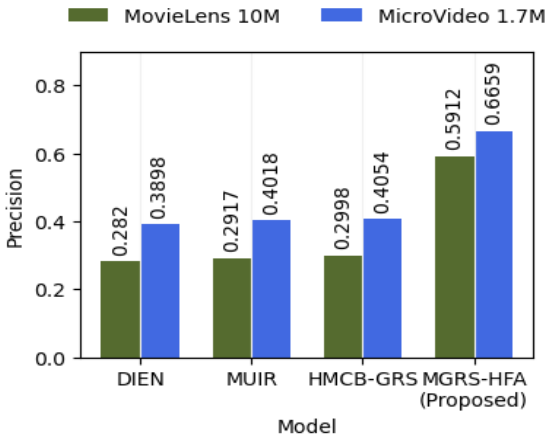


(c) NDCG values

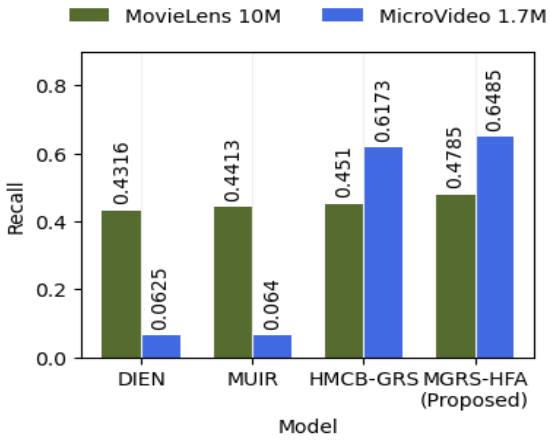


(d) F1-Score values

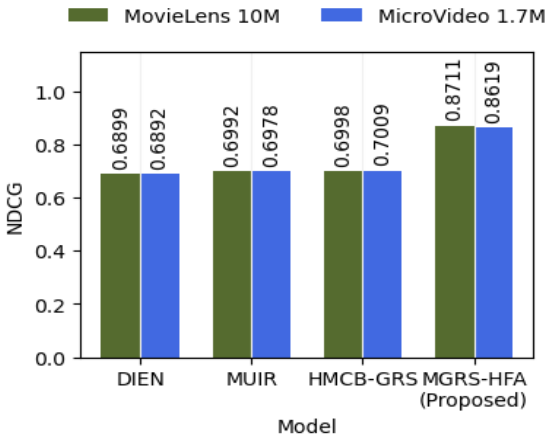
Figure 6 (a)-(d). Performance Analysis of the MGRS-HFA with MGAT, MGCF, and MCGCRS on various evaluation metrics



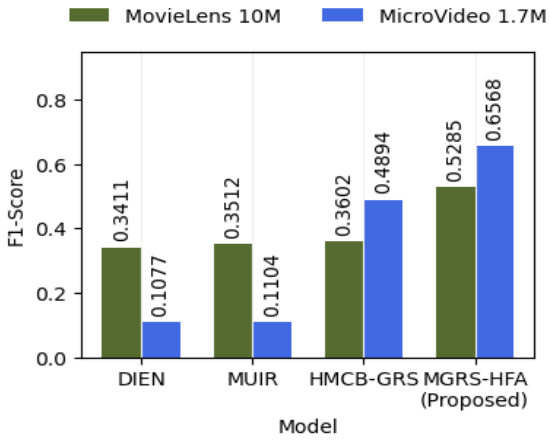
(a) Precision values



(b) Recall values



(c) NDCG values



(d) F1-Score

Figure 7 (a)-(d). Performance Analysis of the MGRS-HFA with DIEN, MUIR, and HMCB-GRS on various evaluation metrics

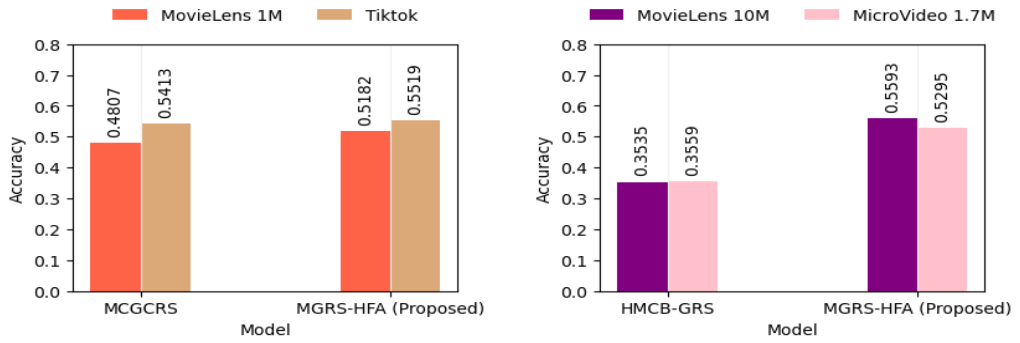
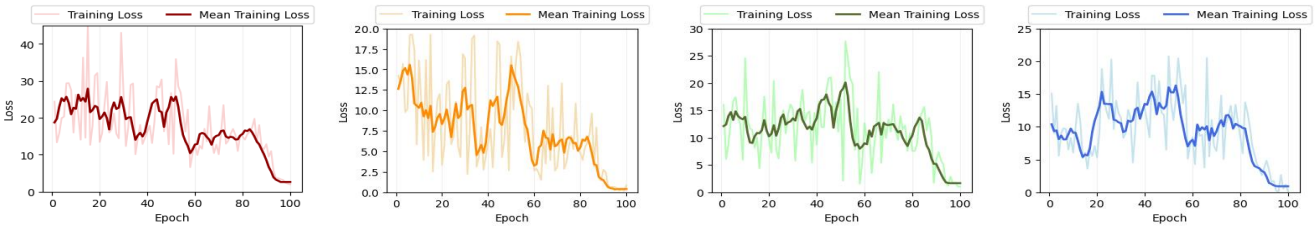
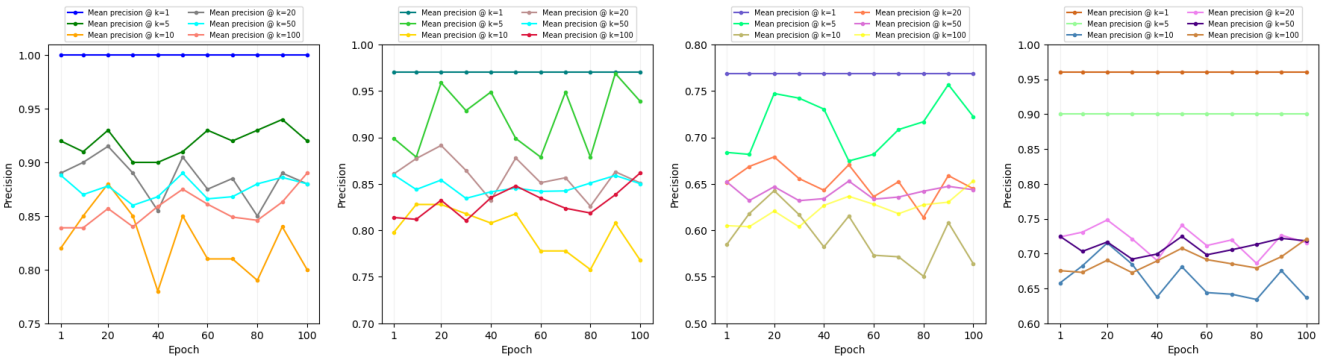


Figure 8. Performance Analysis for MGRS-HFA with Content and Collaborative Recommendation Systems of Accuracy

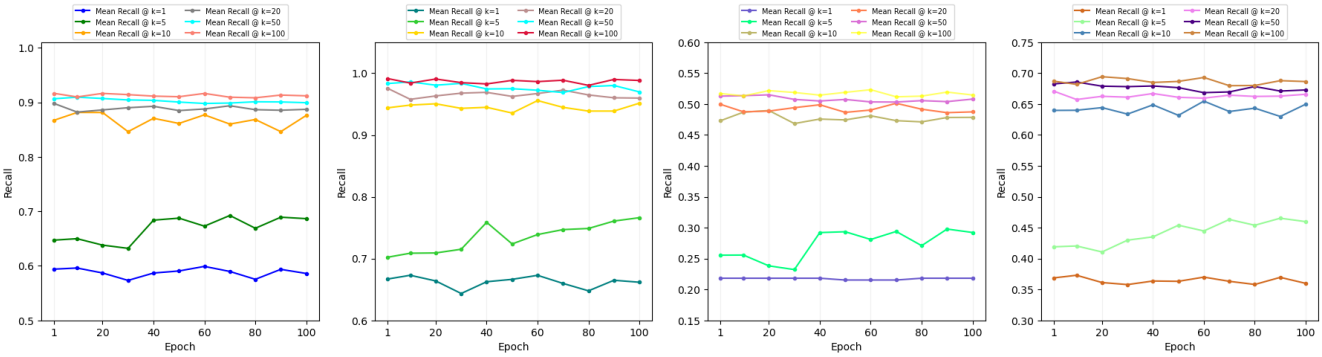


(a) MovieLens 1M (b) Tiktok (c) MovieLens 10M (d) MicroVideo 1.7M

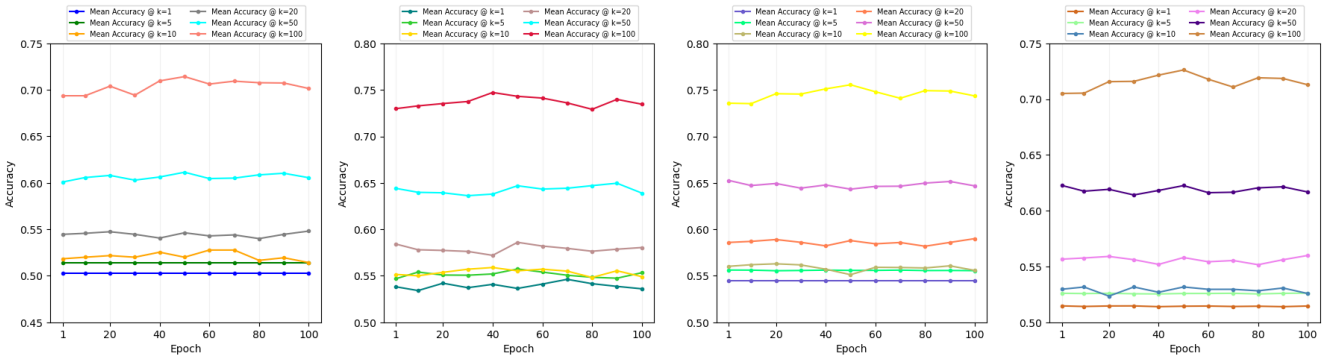
Figure 9 (a)-(d). Training loss by MGRS-HFA over 100 epochs on various datasets



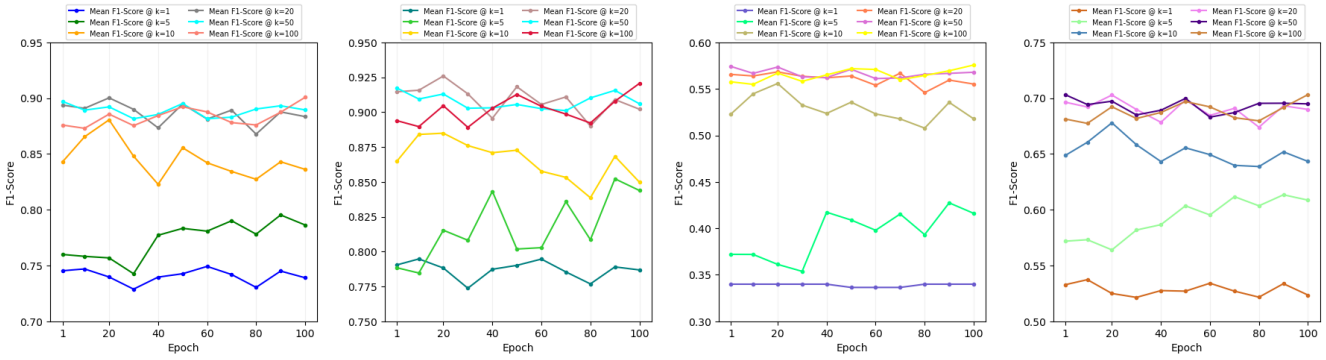
(a) Precision@ different K values



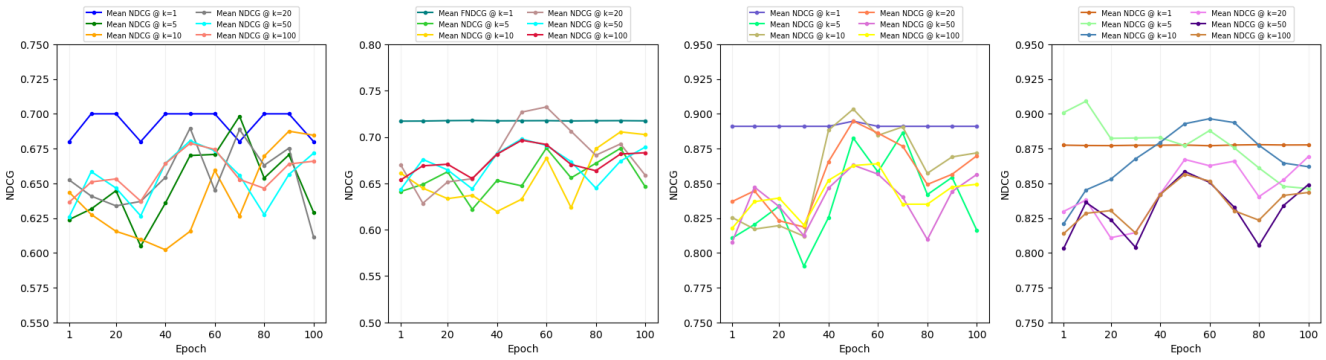
(b) Recall@ different K values



(c) Accuracy@ different K values



(d) F1-Score@ different K values



(e) NDCG@ different K values

Figure 10 (a)-(e). Performance analysis of MGRS-HFA on various datasets (MovieLens 1M, TikTok, MovieLens 10M and MicroVideo 1.7M) for different values of K and Epochs

B. Performance Analysis

The experimental results, shown in Table 2-4 show that MGRS-HFA exhibits outstanding performance on various metrics, viz Precision, Recall, NDCG, Accuracy and F1-Score for different models.

The MGRS-HFA outperforms other collaborative models; for MovieLens 1M, the MGRS-HFA achieves a precision of 0.8269, which is 68% higher than the baseline performance. On TikTok, the MGRS-HFA achieves a precision of 0.7969, marking a 53% improvement. There are modest improvements in recall: 2% for MovieLens 1M and 1% for TikTok. The MGRS-HFA demonstrates substantial enhancements in NDCG: an 86% improvement for MovieLens 1M and 70% for TikTok. Notably, the F1-Score improves by 36% for MovieLens 1M and 29% for TikTok.

The MGRS-HFA outperforms other content-based models. The MGRS-HFA shows a 97% improvement in precision for MovieLens-10M and a 64% improvement for MicroVideo-1.7M. The recall improvements are 6% for MovieLens-10M and 5% for MicroVideo-1.7M. There is a 24% improvement in NDCG performance for MovieLens-10M and a 23% improvement for MicroVideo-1.7M. The F1-Score also sees significant gains: 47% for MovieLens-10M and 34% for MicroVideo-1.7M.

For the MovieLens 1M and TikTok datasets, the MGRS-HFA shows an 8% improvement in accuracy over MCGCRS and a 2% improvement over HMCB-GRS. For the MovieLens-10M and MicroVideo-1.7M datasets, the MGRS-HFA shows a 58% improvement over MCGCRS and a 49% improvement over HMCB-GRS.

The model's precision is highly consistent for smaller k values and shows reasonable performance and variability for larger k values across different datasets and epochs. Each dataset shows unique characteristics in how recall values evolve over epochs, likely due to dataset size, item diversity, and user behavior differences. Across all datasets, as k increases, accuracy tends to improve or stabilize over epochs. Larger values of k (e.g., 50, 100) consistently show more stability or improve accuracy, suggesting that models may benefit from recommending a larger number of items simultaneously. Higher k values tend to stabilize F1-Scores better than lower k values. While some metrics stabilize early on, smaller k values often show more variability and potential for improvement over epochs. Customizing recommendation systems to each dataset is important because each dataset performs radically.

5. CONCLUSION

This paper improves a Multimodal Graph-based Recommendation System using Hybrid Filtering Approach (MGRS-HFA) framework by combining text, image, video, and metadata to generate more relevant recommendations for individual users. Adding GCN-based collaborative filtering and graph-based similarity clustering using content filtering to the model makes it more robust than traditional collaborative filtering and content-based filtering. The model uses cross-attention mechanism and Variational Graph Autoencoder (VGAE) for link prediction to capture complex user-item interactions. Experiments on multiple datasets demonstrate the effectiveness of MGRS-HFA compared to the state-of-the-art. The presented model performs better on various evaluation metrics. Researchers can improve recommendation accuracy by capturing complex user-item interactions using more advanced attention mechanisms and deep learning architectures in the future. The MGRS-HFA model's performance and resilience can be achieved by combining it with additional advanced recommendation approaches like meta-learning and reinforcement learning.

REFERENCES

[1] F. Zhou, B. Luo, T. Hu, Z. Chen and Y. Wen, "A Combinatorial Recommendation System Framework Based on Deep Reinforcement Learning," *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA, 2021, pp. 5733-5740, doi: 10.1109/BigData52589.2021.9671593.

[2] P. M. Alamdari, N. J. Navimipour, M. Hosseinzadeh, A. A. Safaei and A. Darwesh, "A Systematic Study on the Recommender Systems in the E-Commerce," in *IEEE Access*, vol. 8, pp. 115694-115716, 2020, doi: 10.1109/ACCESS.2020.3002803.

[3] T. Omura, K. Suzuki, P. Siriaraya, M. Mittal, Y. Kawai and S. Nakajima, "Ad Recommendation utilizing user behavior in the physical space to represent their latent interest," *2020 IEEE International Conference on Big Data (Big Data)*, Atlanta, GA, USA, 2020, pp. 3143-3146, doi: 10.1109/BigData50022.2020.9377822.

[4] W. Wang, X. Lin, F. Feng, X. He, and T.-S. Chua, "Generative Recommendation: Towards Next-generation Recommender Paradigm," Apr. 2023, doi: https://doi.org/10.48550/arxiv.2304.03516.

[5] H. Wang, N. Lou and Z. Chao, "A Personalized Movie Recommendation System based on LSTM-CNN," 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 2020, pp. 485-490, doi: 10.1109/MLBDBI51377.2020.00102.

[6] M. Elahi, F. Ricci, and N. Rubens, "A survey of active learning in collaborative filtering recommender systems" Computer Science Review, vol. 20, pp. 29–50, May 2016, doi: https://doi.org/10.1016/j.cosrev.2016.05.002.

[7] S. Gupta, A. K. Bindal and D. Prasad, "Multi-modality Collaborative Recommender Systems: An Overview of Techniques and Evaluation Metrics," 2023 International Conference on Advanced Computing & Communication Technologies (ICACCTech), Banur, India, 2023, pp. 426-433, doi: 10.1109/ICACCTech61146.2023.00076.

[8] Fkih, F., "Similarity measures for Collaborative Filtering-based Recommender Systems: Review and experimental comparison", Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 9, pp. 7645–7669. https://doi.org/10.1016/j.jksuci.2021.09.014

[9] F. Rezaimehr and C. Dadkhah, "A survey of attack detection approaches in collaborative filtering recommender systems," Artificial Intelligence Review, vol. 54, no. 3, pp. 2011–2066, 2021. doi: https://doi.org/10.1007/s10462-020-09898-3.

[10] F. Zhang, T. Gong, V. E. Lee, G. Zhao, C. Rong, and G. Qu, "Fast algorithms to evaluate collaborative filtering recommender systems," Knowledge-Based Systems, vol. 96, pp. 96–103, 2016, doi: https://doi.org/10.1016/j.knosys.2015.12.025.

[11] X. Zhou, D. Lin and T. Ishida, "Evaluating Reputation of Web Services under Rating Scarcity," 2016 IEEE International Conference on Services Computing (SCC), San Francisco, CA, USA, 2016, pp. 211-218, doi: 10.1109/SCC.2016.35.

[12] P. Lops, D. Jannach, C. Musto, T. Bogers, and M. Koolen, "Trends in content-based recommendation," User Modeling and User-Adapted Interaction, vol. 29, no. 2, pp. 239–249, 2019, doi: https://doi.org/10.1007/s11257-019-09231-w.

[13] A. Zagranovskaia and D. Mitura, "Designing hybrid recommender systems," in DEFIN-2021: IV International Scientific and Practical Conference, pp. 1–5, 2021 https://doi.org/10.1145/3487757.3490921.

[14] A. Jamilu Ibrahim, P. Zira, and N. Abdulganiyyi, "Hybrid Recommender for Research Papers and Articles," International Journal of Intelligent Information Systems, vol. 10, no. 2, p. 9, 2021, doi: https://doi.org/10.11648/j.ijiis.20211002.11.

[15] S. Gupta and D. A. Kumar Bindal, "Multi-Modality Recommender Systems: A Review," 2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC), Solan, Himachal Pradesh, India, 2022, pp. 88-93, doi: 10.1109/PDGC56933.2022.10053362.

[16] J. Lund and Y. -K. Ng, "Movie Recommendations Using the Deep Learning Approach," 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 2018, pp. 47-54, doi: 10.1109/IRI.2018.00015.

[17] D. Roy and C. Ding, "Movie Recommendation Using Youtube Movie Trailer Data as the Side Information" In Proceedings of the 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), The Hague, The Netherlands, 7–10 December 2020; pp. 275–279, https://doi.org/10.1109/asonam49781.2020.9381349.

[18] R. Burke, "Hybrid recommender systems: Survey and experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331–370, 2002, doi: https://doi.org/10.1023/a:1021240730564.

[19] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 2, pp. 423–443, 2019, doi: https://doi.org/10.1109/tpami.2018.2798607.

[20] Liu, Y., Ott, M., Goyal, N., et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach", ArXiv abs/1907.11692 (2019), https://doi.org/10.48550/arXiv.1907.11692.

[21] M. Tan and Q. V. Le, “EfficientNetV2: Smaller models and faster training”, Proceedings of the 38th International Conference on Machine Learning, arxiv.org, 2021, doi: https://doi.org/10.48550/arXiv.2104.00298.

[22] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, “Video Transformer Network,” Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, doi: https://doi.org/10.48550/arxiv.2102.00719.

[23] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," in Computer, vol. 42, no. 8, pp. 30-37, 2009, doi: 10.1109/MC.2009.263.

[24] P. Covington, J. Adams, and E. Sargin, “Deep Neural Networks for YouTube Recommendations,” Proceedings of the 10th ACM Conference on Recommender Systems - RecSys ’16, pp. 191–198, 2016, doi: https://doi.org/10.1145/2959100.2959190.

[25] J. E. van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, pp. 373–440, 2020, doi: https://doi.org/10.1007/s10994-019-05855-6.

[26] Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). Graph convolutional neural networks for web-scale recommender systems. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 974-983).

[27] van, T. N. Kipf, and M. Welling, “Graph Convolutional Matrix Completion,” arXiv 2017, doi: https://doi.org/10.48550/arxiv.1706.02263.

[28] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph Convolutional Neural Networks for Web-Scale Recommender Systems,” Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD ’18, 2018, doi: https://doi.org/10.1145/3219819.3219890.

[29] Y. Deng, "Recommender Systems Based on Graph Embedding Techniques: A Review," in IEEE Access, vol. 10, pp. 51587-51633, 2022, doi: 10.1109/ACCESS.2022.3174197.

[30] Nairouz Mrabah, M. Bouguessa, and Riadh Ksantini, “A contrastive variational graph auto-encoder for node clustering,” Pattern recognition, vol. 149, pp. 110209–110209, 2024, doi: https://doi.org/10.1016/j.patcog.2023.110209.

[31] H. Wu, C. Song, Y. Ge, and T. Ge, “Link Prediction on Complex Networks: An Experimental Survey,” Data Science and Engineering, vol. 7, no. 3, pp. 253–278, 2022, doi: https://doi.org/10.1007/s41019-022-00188-2.

[32] A. Zareie and R. Sakellariou, “Similarity-based link prediction in social networks using latent relationships between the users,” Scientific Reports, vol. 10, no. 1, 2020, doi: https://doi.org/10.1038/s41598-020-76799-4.

[33] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” arXiv (Cornell University), 2016, doi: https://doi.org/10.48550/arxiv.1609.02907.

[34] A. Grover and J. Leskovec, “node2vec: Scalable Feature Learning for Networks,” Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855-864, 2019, doi: https://doi.org/10.1145/2939672.2939754.

[35] Van Thuy Hoang, H.-J. Jeon, E.-S. You, Y. Yoon, S. Jung, and O-Joun. Lee, “Graph Representation Learning and Its Applications: A Survey,” Sensors, vol. 23, no. 8, pp. 4168–4168, Apr. 2023, doi: https://doi.org/10.3390/s23084168.

[36] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," Advances in Neural Information Processing Systems, vol. 31, pp. 5165-5175, 2018, doi: https://doi.org/10.48550/arxiv.1802.09691.

[37] Petar Veličković, G. Cucurull, A. Casanova, A. Romero, Pietro Liò, and Y. Bengio, “Graph Attention Networks,” arXiv 2017, doi: https://doi.org/10.48550/arxiv.1710.10903.

[38] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, “Link prediction techniques, applications, and performance: A survey,” Physica A: Statistical Mechanics and its Applications, vol. 553, p. 124289, Sep. 2020, doi: https://doi.org/10.1016/j.physa.2020.124289.

[39] Z. Tao, Y. Wei, X. Wang, X. He, X. Huang, and T.-S. Chua, “MGAT: Multimodal Graph Attention Network for Recommendation,” Information Processing & Management, vol. 57, no. 5, p. 102277, Sep. 2020, doi: https://doi.org/10.1016/j.ipm.2020.102277.

[40] J. Sun, H. Chang, W. Zhao, Y. Yu, L. Yang and X. Huang, "A Multimedia Graph Collaborative Filter," in IEEE Access, vol. 10, pp. 50892-50902, 2022, doi: 10.1109/ACCESS.2022.3174212.

[41] M. Yu, T. Liu, J. Yin, and P. Chai, “Deep Interest Context Network for Click-Through Rate,” Applied Sciences, vol. 12, no. 19, p. 9531, Sep. 2022, doi: https://doi.org/10.3390/app12199531.

[42] X. Chen, D. Liu, Z. Xiong and Z. -J. Zha, "Learning and Fusing Multiple User Interest Representations for Micro-Video and Movie Recommendations," in IEEE Transactions on Multimedia, vol. 23, pp. 484-496, 2021, doi: 10.1109/TMM.2020.2978618.