International Journal of Computing and Digital Systems
2025, VOL. 17, NO. 1, 1–15

http://dx.doi.org/10.12785/ijcds/1571047857

Multimodal Graph-based Recommendation System using Hybrid Filtering Approach

Sorabh Gupta[1], Amit Kumar Bindal[2] and 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*

Received 8 July 2024, Revised 5 January 2025, Accepted 9 January 2025

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 to 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 Autoencoder (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 publicly accessible datasets such as Movielens 1M, TikTok, MovieLens 10M, and MicroVideo 1.7M. Our proposed model demonstrates superior effectiveness compared to the state-of-the-art multimedia recommender systems in various evaluation parameters such as precision, accuracy, recall, Normalized Discounted Cumulative Gain (NDCG), F1-score and RMSE.

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 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 assists consumers in finding new, relevant material. During the initial stages of the business, the primary focus was on developing online shopping recommendation systems [3]. These systems used simple algorithms to analyze customer purchase histories. Powerful recommendation systems that employ machine learning algorithms have become 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 [6]. This sort of recommendation system classifies users into clusters of similar types and recommends to each user based on its cluster's preferences [7]. We divide it into two categories: item-based and user-based CF. Item-based CF compares items for similarity [8]. User-based CF recommends items based on the similarity between users [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]. It uses 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,

E-mail address: sorabhg2003@gmail.com, bindalamit@gmail.com, devendraacad@gmail.com

which led to the creation of hybrid recommendation systems [13][14]. These systems incorporate multiple recommendation methods to mitigate their shortcomings and optimize their strengths. A hybrid system might use CF to find items or users that are similar and then 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 addressing data sparsity and the cold start problem due to their inherent traits. We can combine different approaches to integrate 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 interactions.

In the digital world we live in now, we can access all kinds of information, like text, images, movies, and music. For example, users who shop online can watch videos and read 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. 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. However, the assimilation is challenging in the absence of significant data, such as text, images, video, and music. Combining data from multiple modalities efficiently requires complex algorithms and considerable computational cost [16]. Multimodal recommendation systems are scarce, complicating the issue. Problems with real-time applications are becoming harder to solve [17]. The complexity of multimodal data is increasing. We must handle multimodal datasets without overloading performance, ensuring that the load matches the capacity.

The proposed model has broad applications beyond multimedia recommendations. It can enhance personalized shopping experiences in e-commerce, assist healthcare providers in tailoring treatment plans, recommend personalized learning materials in education, and improve content discovery on social media. Its adaptable architecture using multimodal data can significantly impact various industries, improving user satisfaction and decision-making. The study aims to design and evaluate a hybrid recommender model, MGRS-HFA, using multiple datasets. The experimental results demonstrate that the MGRS-HFA model outperforms various baseline models. This study's primary contributions are:

- The study utilizes deep learning to capture and utilize multiple data modalities for individualized suggestions. It combines graph-based collaborative filtering with cluster-based content-based filtering.
- The study investigates the effectiveness of graph structures in representing connections between users and items for collaborative filtering. It also explores clustering techniques to enhance content-based filtering and improve recommendation accuracy, especially for complex user-item interactions.
- The study conducts extensive experiments using four datasets. The experimental results provide new insights into the potential of the MGRS-HFA model.

The rest of the study is structured as follows: Section 2 reviews related works and highlights gaps in existing approaches. Section 3 discusses the design and construction of the proposed MGRS-HFA model, including its architecture and how it uses both graph-based collaborative filtering and cluster-based content-based filtering. Section 4 outlines the datasets, evaluation metrics, and baseline models used in the study and presents a detailed discussion of the experimental results. Finally, Section 5 concludes the study.

2. Related Work

The predecessors of today's recommendation systems relied on clear interactions between the user and an item. More complex systems that utilize multimodal data and advanced machine learning have replaced these. 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 can hinder user engagement with items. To address these limitations, integrating CF and CBF methods with graph-based approaches has gained traction. 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 advances in multimodal learning allow feature extraction and integration from text, images, videos, and metadata. For instance, models like RoBERTa [19] for textual data, EfficientNet-V2 [20] for images, and Video Transformer [21] for videos generate rich representations that can be integrated into recommendation engines. These multimodal techniques provide a comprehensive understanding of user preferences and item characteristics, enhancing the ability of recommendation systems to capture complex user-item interactions [22].

User-item feature encoding helps recommendation systems work by converting raw data into model-friendly representations. Earlier techniques, such as matrix factorization approaches, a type of latent feature encoding, were often used to make CF better by using low-dimensional user-item relationship matrix representations [23]. However, neural network-based embeddings, such as graph-based embeddings, have simplified recording complex interactions between users and items by turning high-dimensional data into dense, low-dimensional vectors [24]. Moreover, unsupervised clustering approaches such as hierarchical and k-means are used in recommendation systems to find hidden

user and item data structures [25]. These techniques use features to cluster users and items, to identify communities, and pinpoint similarities in content. Using user and item cluster relationships, cluster similarity-based graphs make facilitate for the system to show relevant items without users having to explicitly interact with the system. Semantic clustering, using NLP, has emerged as a powerful technique to group items or users according to semantic similarity. This method uses contextual information in written descriptions, reviews, and other content. Word embeddings and deep learning models use clustered items of semantic content to improve content-based recommendations. Semantic clustering and collaborative filtering improve suggestion relevancy by integrating content similarity and user behavior patterns.

Graph-based recommendation systems, particularly those using bipartite networks, have become foundational in representing nodes for users and items. This framework structures user-item interactions, such as ratings and clicks. Bipartite graphs enable graph-based algorithms to uncover latent links and enhance recommendation accuracy. Graph Convolutional Networks (GCNs) transmit data across the bipartite graph to detect patterns of higher-order connections [26]. Graph autoencoders can learn the structure of user-item interactions from bipartite networks, improving recommendation performance [27]. Due to their ability to capture complex user-item relationships, graph-based recommendation systems have gained popularity. User-item bipartite graphs express interactions, making graph neural networks (GNNs) for collaborative filtering easier. Graph Convolutional Networks (GCNs) and GraphSAGE are well-known for their ability to combine data from nearby nodes, enhancing embeddings better for both users and items [28]. VGAEs provide a powerful foundation for graph link prediction, helping recommendation systems [29]. They address user-item interaction uncertainty and variability by learning probabilistic distributions over latent variables. VGAEs can understand complex relationships and predict how users and items will interact over time using GCNs in the encoder and a probabilistic decoder. This makes VGAEs well-suited for personalized recommendations.

In the domain of link prediction, advancements in graph-based techniques have significantly impacted social networks, biological networks, and recommender systems. Traditional methods use network topology and similarity measures such as Jaccard coefficients, common neighbors, and preferential attachment [30]. Bayesian, stochastic block, and matrix factorization approaches like Singular Value Decomposition (SVD) improve prediction accuracy [31]. Supervised learning methods that use link prediction as a binary classification task show promise by capturing complex network patterns. Unsupervised methods, such as node embeddings, DeepWalk, and Node2Vec, enhance predictions, and by using graph structures, deep learning developments like GNNs and GCNs have revolutionized link prediction [32]. Graph Attention Networks (GATs) dynamically weigh neighbor significance to improve predictions [33]. Model performance is often evaluated using AUC, accuracy, recall, and F1-score [34]. This multidisciplinary approach shows the evolution of link prediction methods.

The hybrid approach that combines CF and CBF with graph-based techniques is better at personalizing items as it considers both user behavior and item attributes. Multimodal data integration incorporates item attributes and user preferences for richer representation. Traditional CF approaches are less attractive to users due to challenges associated with sparsity and cold start. Adding graph-based methods and multimodal data can increase system complexity, making it challenging to handle large-scale data and ensure efficient computing using complex approaches. Our approach dynamically incorporates user and item domain information to address sparsity and cold-start issues, effectively mitigating these shortcomings. Simplifying and managing the system efficiently is crucial for its optimal performance and future scalability. This can be done by better integrating multimodal data and graph-based methods.

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.

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

1) Gather different modalities (text, image, video, and metadata) from various sources.
2) After resizing and normalizing the image, use EfficientNet V2 to extract image features.
3) Preprocess video and extract features using Video Transformer.
4) Split the corpus, remove stop words and punctuation, lemmatize, and tokenize text using RoBERTa.
5) RoBERTa normalizes continuous Prompt Generation metadata variables and encodes categorical information into numerical vectors.
6) The user-item feature encoder integrates all modality features.
7) Construct a user-item bipartite graph.

B. Collaborative Filtering with GraphSAGE

Input: User and Item feature matrix (X_U, X_I)
Output: Processed feature matrices H_U, H_I

1) 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.
2) Construct user and item graphs for each user pair (u_i, u_j) and item pair (i_i, i_j).

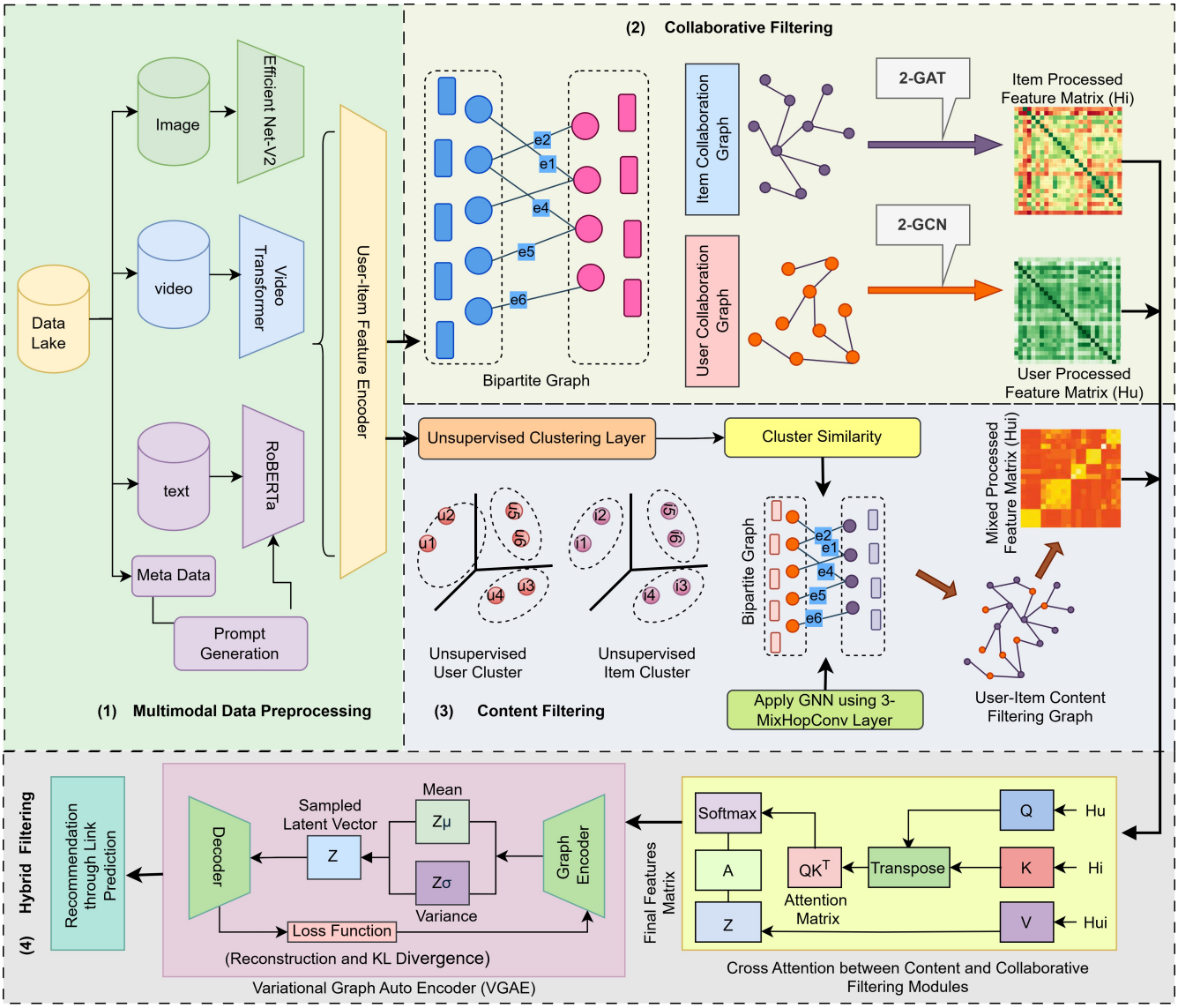


Figure 1. Shows our proposed model MGRS-HFA Framework

3) Apply a two-layer GCN to the user and two-layer GAT to item subgraphs to update feature matrices and fine-tune user and item features.

C. Content Filtering with GraphSAGE

Input: User and Item feature matrix (X_U, X_I)
Output: Processed mixed feature matrix H_{UI}

1) Apply unsupervised clustering to both user and item features and compute cluster centroids of user clusters C_{Uj} and item clusters C_{Ij}.
2) Compute cluster similarity for each user cluster C_{Uj} and item cluster C_{Ik} and construct a cluster similarity-based graph.
3) Apply three MixHopConv layers to the cluster similarity-based graph.
4) Apply GCN to refine node features in the constructed graph.
5) 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

1) Define the query, key, and value matrices and compute the attention weights and cross-attention mechanism's output.
2) Use the Variational Graph Autoencoder (VGAE) to learn probabilistic distributions over latent variables:
   a) **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.

3) Calculate the loss function using reconstruction loss and KL divergence.
4) Calculate the likelihood of node pairs for link prediction.
5) Rank the computed link probabilities to generate recommendations.

3.2 Framework of the Proposed Model

There are four components to the proposed model: preprocessing multimodal data, collaborative filtering, content-based filtering, and hybrid filtering (which includes cross-attention, VGAE, and recommendation generation).

A. Multimodal Data Preprocessing

The model encompasses a range of data modalities, including text, images, videos, and metadata, for building user and item representations. The feature extraction and multimodal data fusion process is the core module of our recommendation system. The steps proceed as follows:

i) Data Collection and Preprocessing

The model starts by gathering multiple sources of data, including text features (e.g., user reviews and item descriptions), image data (e.g., product images and user-uploaded photos), video data (e.g., trailers and reviews), and structured metadata (e.g., item attributes and user demographics). Each data type undergoes specific preprocessing to standardize and validate the information. For text data, preprocessing includes removing stop words, punctuation, and applying lemmatization. Images are resized and normalized for consistency. Videos are divided into keyframes, and features are extracted. Metadata is standardized to ensure uniform records. For categorical metadata, we use the most frequent category or a placeholder value like 'unknown'. To handle missing data (such as user IDs, movie genres, or tags) and clean noisy text (such as movie titles, descriptions, and tags containing special characters or irrelevant text), we use imputation techniques, tokenization, and text normalization. Matrix factorization, such as Singular Value Decomposition (SVD), addresses sparse data, such as user-item interaction ratings from users who haven't rated many movies.

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 (e.g., visuals, text, and other metadata). This integration often results in a feature matrix, where each row represents a user or an item and each column corresponds to a particular feature (such as image attributes, textual content, or other item-specific data). Next, we use a feature encoder to transform the unified features into a fixed-dimensional space suitable for further processing. The encoded feature matrices for users and items are then sent to collaborative and content-based filtering modules, which improve the accuracy of the recommendations by refining and updating the user and item embeddings.

iv) Bipartite Graph of User-Item features

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

A bipartite graph $G = (U, I, E)$ entails two distinct sets of vertices: U (users) and I (items), where E represents the edges between these sets. Each edge $e_{ui} \in E$ connects a user $u \in U$ and an item $i \in I$, indicating some form of interaction or relationship (e.g., purchase, rating). The edge list from user-item interactions constructs the graph, representing each interaction as a connection between a user and an item. The feature matrix, which has features for both users and items, helps set up the attributes of these nodes so that users and items are shown with accurate data.

B. Collaborative Filtering

Collaborative filtering is essential in recommender systems, utilizing user-item interactions to enhance personalized recommendations. We create a user collaboration graph (users with similar item interactions) and an item collaboration graph (items with similar user interactions). A Graph Attention Network (GAT) is used on the item graph, and a Graph Convolutional Network (GCN) is used on the user graph to improve feature representations. The

hybrid module then uses the processed item and user feature matrices for personalized recommendations.

i) User Collaboration Graph

This graph represents relationships between users as shown in Figure 2. 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')ϵU, such as the cosine similarity of user feature vectors as shown in equation (1):

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

where v_u is the feature vector of user u.

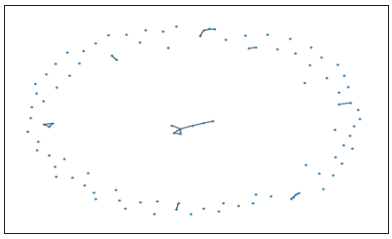


Figure 2. User Collaboration Graph of the MovieLens 1M dataset

ii) Item Collaboration Graph

This graph captures relationships between items as shown in Figure 3. 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')ϵI based on their similarity s(i,i'), as given in equation (2):

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

where v_i is the feature vector of item i.

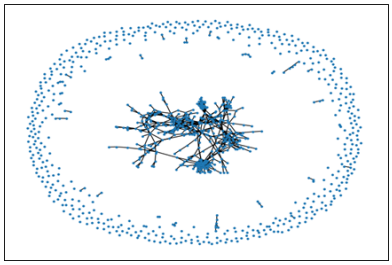


Figure 3. Item Collaboration Graph of the MovieLens 1M dataset

iii) GCN and GAT Processing on Attributed Graphs

We apply two layer GCN and GAT separately to and to learn and refine the node (user or item) representations.

User GCN Architecture

The User GCN architecture processes the user feature matrix and the user graph (represented by edge indexes and weights). It comprises two GCN layers and a fully connected layer. The first layer employs a GCNConv layer to aggregate features from immediate neighbors, followed by ReLU activation and dropout (0.5) to prevent overfitting. The second GCNConv layer continues to aggregate features, including higher-order neighbors, producing the final user embeddings. Finally, a fully connected layer reduces the dimensionality of the embeddings to match the desired feature space for downstream tasks. The propagation rule for a GCN layer is given in equation (3):

$$H^{(I+1)} = \sigma(D^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(I)}W^{(I)}) \tag{3}$$

Item GAT Architecture

The Item GAT architecture processes the item feature matrix and the item graph (represented by an edge index). It includes two GAT layers and a fully connected layer. The first GATConv layer with multiple heads captures attention weights from neighbors, followed by ELU activation and dropout (0.6) for regularization. The second GATConv layer with a single head aggregates information from attention-weighted neighbors, refining item embeddings. The fully connected layer maps the final item embeddings into the desired feature space. The update rule for a GAT layer is expressed in equation (4):

$$h_i' = \sigma\left(\sum_{j \in N(i)} a_{ij} W_i h_j\right) \tag{4}$$

For both GCN and GAT, the learning rate is set at 0.001 to balance convergence speed and stability. The Adam optimizer handles sparse gradients and adaptive learning rates well, which leads to better performance of the recommendation system and better learning of feature representations.

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 the two-GCN and GAT layers. These matrices encapsulate the refined graph structure and node interactions to create high-level user H_U and H_I item representations.

C. Content Filtering

We feed the encoded features into a supervised clustering layer, which groups users and items into clusters based on their feature similarities to integrate content-based filtering into graph structures. This clustering results in user and item clusters that capture the underlying patterns in the data. We construct a bipartite graph using the similarity between these clusters, where nodes represent user and item clusters, and edges indicate similarity relationships. We apply Graph Neural Networks (GNNs), specifically three MixHopConv layers, to this unified graph. These layers

capture complex relationships and dependencies within the graph, refining the feature representations of both users and items. The MixHopConv layers produce processed feature matrices for users and items. The hybrid module integrates these matrices, empowering the system to generate personalized recommendations through comprehensive content-based filtering signals derived from the multimodal data.

i) Unsupervised Clustering

Unsupervised clustering can discover hidden structures within data and increase recommendation accuracy. Using unsupervised learning, we create clustering graphs for users and items. This layer shows user behavior and item quality by combining users and items with similar features. We use k-means to cluster users and items separately during the build process. These clusters help improve recommendations by identifying groups of similar users and items, allowing the system to leverage these patterns for more accurate and relevant suggestions. By understanding these natural groupings, the recommender system can provide more personalized recommendations, improving user satisfaction and engagement. We designate each collection of users and items as U and I. In the dataset, each user uϵU and item iϵI are represented by v_u and v_i.

As shown in equation (5), users in these connections share interests or behaviors.

$$C_U = \{C_{U1}, C_{U2}, \ldots, C_{Uj}\} \tag{5}$$

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

Equation (6) connects clustered items, reflecting their content similarity or co-occurrence patterns.

$$C_I = \{C_{I1}, C_{I2}, \ldots, C_{Ij}\} \tag{6}$$

C_{Ij} represents the j-th user cluster.

ii) Cluster Similarity-based Graph

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 as shown in Figures 4 & 5. 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 (7).

$$C_{Uj} = \frac{1}{|C_{Uj}|} \sum_{u \in CU_j} V_u \tag{7}$$

In a similar way, equation (8) gives the cluster's centroid C_{Ij}:

$$C_{Ij} = \frac{1}{|C_{Ij}|} \sum_{i \in CI_j} V_i \tag{8}$$

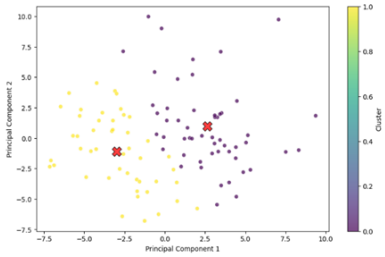


Figure 4. User Clusters of the MovieLens 1M dataset

Use a similarity measure like cosine similarity in equation (9) to compute the similarity among each pair of user-item clusters.

$$s(C_{Uj}, C_{Ij}) = \frac{C_{Uj}, C_{Ij}}{\|C_{Uj}\| \|C_{Ij}\|} \tag{9}$$

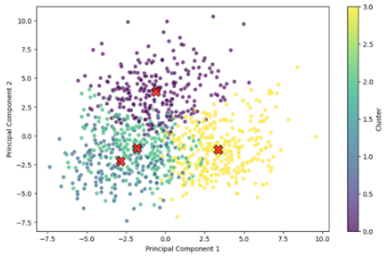


Figure 5. Item Clusters of the MovieLens 1M dataset

The cluster similarity allows for CBF by recommending items from clusters similar to those a user has interacted with earlier. In the context of recommendation systems, clusters group users or items based on shared characteristics or behavior patterns. We quantify the similarity between clusters to ascertain how closely the aggregate preferences of one user cluster align with the aggregate characteristics of an item cluster.

iii) User-Item Content Filtering Graph

To form the user-item content-based graph, we apply three MixHopConv layers to the cluster similarity-based graph. We design the architecture of these layers to capture and refine complex relationships within the graph, thereby enhancing feature representations for both users and items. Each MixHopConv layer extracts features from multiple hops in the graph, specifically leveraging adjacency powers A^0, A^1, and A^2. The input dimensions match the feature dimensions of the graph, and each hop has an output dimension of 60. We ensure regularization and prevent overfitting by applying a dropout rate of 0.7 before the first layer and 0.9 after each MixHopConv layer. We apply batch normalization to stabilize and accelerate training, ensuring the normalization dimension matches the concatenated output size of MixHopConv (3x60=180).

A final linear layer reduces the dimensionality of the

output to a 32-dimensional embedding, making the feature vectors suitable for downstream tasks. The Adam optimizer, with a learning rate of 0.001 and weight decay of 1e4, ensures efficient training by handling sparse gradients and adaptive learning rates. This architecture leverages the strengths of MixHopConv layers to create a robust, content-based graph that enhances the overall recommendation system's performance through refined user-item interactions.

The goal is to construct an integrated graph that blends user and item nodes, leveraging cluster-level similarities. We utilize V as the nodes and E as the cluster-similar edges in this graph. 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 and apply GCN to the constructed graph.

iv) Obtaining the Processed Feature Matrix

The final output feature matrix Hui, represents refined embeddings for users and items. These embeddings incorporate connections from cluster-level interactions and graph convolutions, improving prediction and recommendation tasks.

Graph-based content filtering embeds item features into a graph, creating stronger connections for items with shared attributes. In a movie recommendation system, this involves linking movies by genre, director, and cast, and integrating user preferences to form a bipartite graph. We refine user and item features through multiple hops using GNN, specifically three MixHopConv layers, to produce enhanced feature representations. The final feature matrix improves recommendation accuracy by capturing user-item relationships, item-item similarities, and cross-user interactions. This method provides personalized and relevant recommendations, enhancing overall system performance.

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:

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

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

$$Q = H_U, K = H_I, and\ V = H_{UI} \tag{11}$$

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.

$$AttentionWeight\ (A) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{12}$$

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 preferences 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.

$$Z = AV \tag{13}$$

To improve recommendation accuracy, the cross-attention component dynamically combines and refines user, item, and content-based interaction feature representations. The hyperparameters for the model include 32 feature dimensions for both user and item features. We use the Adam optimizer for optimization, balancing convergence speed and stability with a learning rate of 0.001. We use the sigmoid function for activation, which handles non-linearities and maintains outputs within the range of 0 to 1. The model updates weights and biases dynamically, enhancing interaction predictions for users and items. It uses attention scores to weight and combine processed user and item features from collaborative filtering and mixed-hop features from content filtering based on relevance for each user-item pair. To better understand user-item interactions, the model can personalize recommendations by focusing on the most essential content features that coincide with a user's interests. Overall, the cross-attention technique enhances collaboration and content-based signals to help the recommendation system grasp complex relationships and provide more accurate, personalized options.

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. 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 VGAE encoder processes the user-item graph. Graph convolutional layers are used in this encoder to detect intricate relationships. The inference model aims to learn a probabilistic distribution over the latent variables (node embeddings). We employ GCN layer for computing $\mu = \mathrm{GCN}_{\mu}(X, A)$ and $\log \sigma = \mathrm{GCN}_{\mu}(X, A)$ that shares the weight matrix W_0).

Equation (14 & 15) illustrates how we derive the inference model from the Variational Graph Autoencoder.

$$q(Z|X, A) = \prod_{i=1}^{N} q(z_i \mid X, A) \tag{14}$$

$$q(z_i \mid X, A) = \mathcal{N}(z_i \mid \mu_i, \mathrm{diag}(\sigma_i^2)) \tag{15}$$

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

$$p(A_{uv} = 1 \mid Z) = \sigma(z_u^T z_v) \tag{16}$$

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

$$p(A|Z) = \prod_{i=1}^{N} \prod_{j=1}^{N} p(A_{ij} \mid z_i, z_j) \tag{17}$$

$$p(A_{ij} = 1 \mid z_i, z_j) = \sigma(z_i^T z_j) \tag{18}$$

where σ is the sigmoid function.

As expressed in equation (19), the VGAE loss function is made of two components: the reconstruction loss and the Kullback-Leibler (KL) divergence.

$$L = \mathbb{E}_{q(Z|G)}\left[\log p(A|Z)\right] - \mathrm{KL}(q(Z|G) \parallel p(Z)) \tag{19}$$

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

We implement this module using hyperparameters such as input feature dimensions of 200 and 100, and hidden layers of 100 and 50 dimensions. We use the Adam optimizer for optimization with a learning rate of 0.01. The activation function is ReLU, and the loss function is binary cross-entropy loss with logits. The model undergoes training over 100 epochs, processing batches of data, updating weights, and optimizing attention mechanisms to predict and complete the user-item interaction matrix.

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 [36], MovieLens 10M [38], MicroVideo 1.7M [38] and TikTok [36] as shown in Table I.

- *MovieLens Datasets*

The GroupLens Research Group developed the MovieLens dataset. Researchers have used the MovieLens dataset for movie recommendation research. The original dataset does not contain visual features. We made an effort to gather videos from YouTube, which led to downloading movie trailers and manually verifying their accuracy. The MovieLens 1M dataset contains 1,239,508 ratings from 55,585 users, covering approximately 5,986 movies. It encompasses user demographic data, including age, gender, occupation, and zip code, as well as movie metadata, such

TABLE I. Dataset Statistics and Features

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	10,682	51,001	98.12%	10,380	300
MicroVideo 1.7M	12,737,619	1,704,880	10,986	99.93%	984,983	200

TABLE II. Performance Analysis of the MGRS-HFA with other Collaborative Recommendation Systems

Model	MovieLens 1M				Tiktok			
	Precision	Recall	NDCG	F1-Score	Precision	Recall	NDCG	F1-Score
MGAT [35]	0.1272	0.5412	0.3251	0.2060	0.1251	0.5965	0.3838	0.2068
MGCF [36]	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 III. Performance Analysis of the MGRS-HFA with other Content based Recommendation Systems

Model	MovieLens 10M				MicroVideo 1.7M			
	Precision	Recall	NDCG	F1-Score	Precision	Recall	NDCG	F1-Score
DIEN [37]	0.2820	0.4316	0.6899	0.3411	0.3898	0.0625	0.6892	0.1077
MUIR [38]	0.2917	0.4413	0.6992	0.3512	0.4018	0.0640	0.6978	0.1104
HMCB-GRS [39]	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	**97%**	**6%**	**24%**	**47%**	**64%**	**5%**	**23%**	**34%**

TABLE IV. Performance Analysis for MGRS-HFA with other Recommendation Systems on Accuracy and RMSE

Model	Accuracy				RMSE	
	MLens-1M	Tiktok	MLens-10M	MicroVideo 1.7M	MLens-1M	MLens-10M
HMCB-GRS [39]	-	-	0.3535	0.3559	-	-
FedPerGNN [40]	-	-	-	-	0.8390	0.7930
GHRS [41]	-	-	-	-	0.8380	-
MCGCRS	0.4807	0.5413	-	-	0.6471	-
MGRS-HFA (Proposed)	**0.5182**	**0.5519**	**0.5593**	**0.5295**	**0.8496**	**0.8133**
%Improvement	**8%**	**2%**	**58%**	**49%**	**1%**	**3%**

as titles and genres. The MicroVideo 1.7M dataset, which includes 1.7 million video clips, features such as video ID, publication timestamp, author username, video description, likes, comments, shares, views, and tags. Each dataset comprises comprehensive records of user-item interactions and numerous multimodal features.

- *TikTok Dataset*

TikTok, a popular micro-video sharing platform, published this dataset in a data mining competition. It contains micro-videos with a duration of 3–15 seconds, along with the textual video captions provided by the users. The TikTok dataset comprises 76,085 videos and includes features such as video ID, publication time, country code, author username, video description, music ID, likes, comments, shares, views, hashtags, and audio transcripts.

- *MicroVideo 1.7M Dataset*

This dataset contains 12,737,619 interactions from 10,986 users on 1,704,880 micro-videos. It is openly available at GitHub.

- *Datasets Biases and Ethical Consideration*

All datasets are publicly available and do not infringe upon user privacy. The MovieLens dataset incorporates user demographic data, such as age, gender, and occupation, which could potentially lead to biases if certain demographic groups have excessive representation. Recommendations may prioritize the preferences of specific

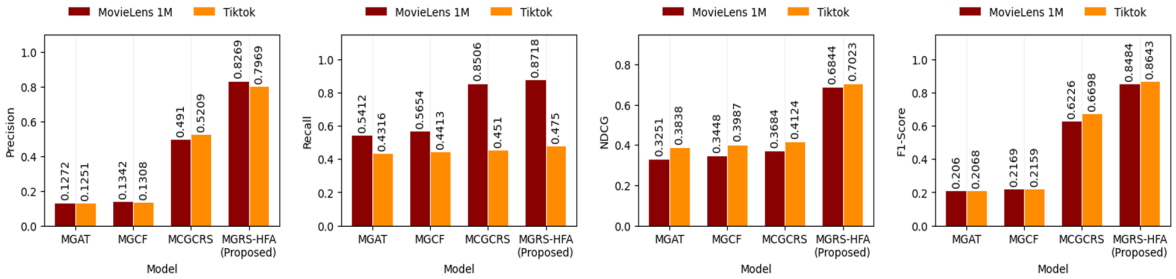


Figure 6. Performance of the MGRS-HFA with other collaborative filtering models on various evaluation metrics

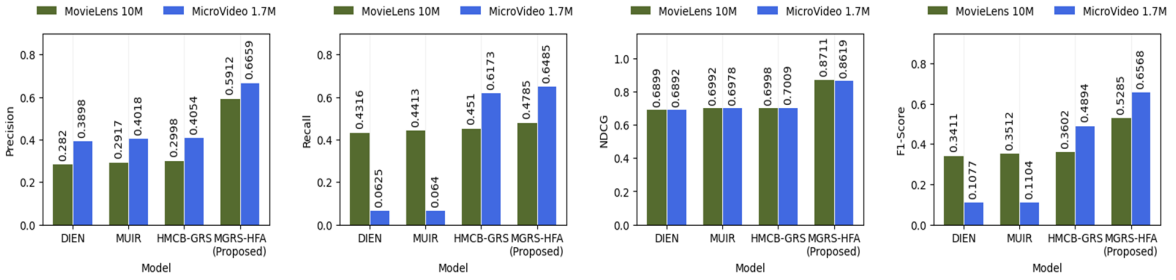


Figure 7. Performance of the MGRS-HFA with other content-based filtering models on various evaluation metrics

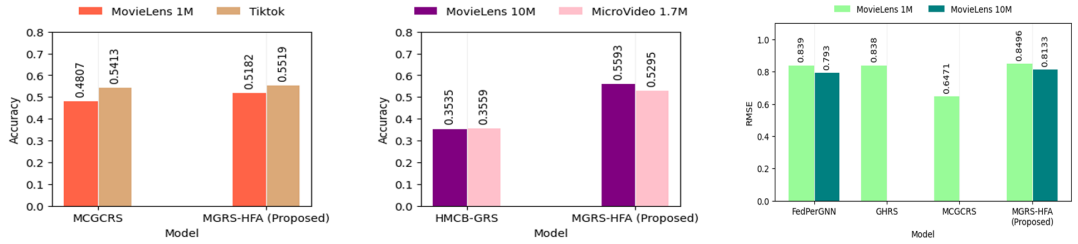


Figure 8. Performance for MGRS-HFA with Content-based and Collaborative Recommendation Systems of Accuracy and RMSE

groups, potentially neglecting the interests of underrepresented communities. The TikTok and MicroVideo datasets include trending videos and user engagement metrics. This may result in a bias favoring the promotion of already popular content, as algorithms tend to emphasize videos with elevated engagement metrics (likes, shares, and views) over niche or less popular content. This may lead to a "rich-get-richer" phenomenon, wherein popular content perpetually prevails in user suggestions. Biases within datasets may lead to skewed recommendations by prioritizing overrepresented demographics, popular products, or highly engaged users while disregarding niche interests and less active users. Balanced and diverse datasets, along with fairness-oriented algorithms, can enhance the accuracy of recommendations.

ii) Baselines

MGAT [35]: User preferences determine gated and attention mechanisms for distinct techniques. This model utilizes comparable attention to determine method relevance.

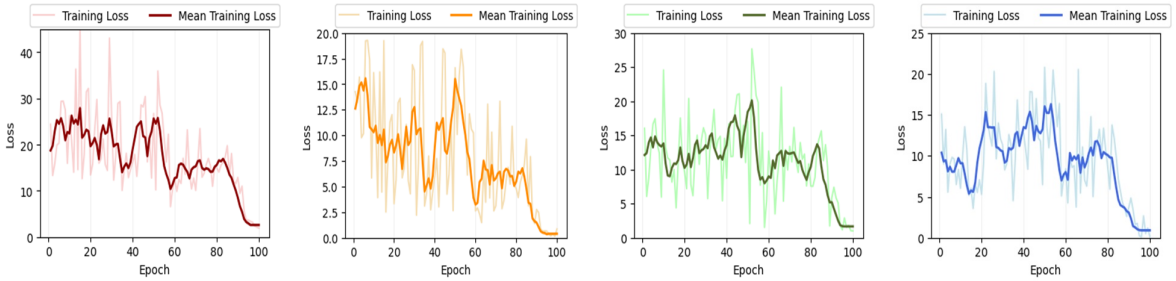
MGCF [36]: Fusion enhances MGCF representation learning. Numerous GCN processes and attention strategies combine multimodal information to improve performance.

MCGCRS: This approach uses multimodal CLIP-guided graphs to predict links between users and items. It uses both adversarial pretraining and Variational Graph Autoencoder (VGAE) techniques to accurately record how users interact with items.

DIEN [37]: It improves DIN by adding a dynamic interest layer to track users' changing interests and eliminating batch normalization.

MUIR [38]: It aims to capture a wide range of user interests by combining several representations for personalized recommendations without using batch normalization.

HMCB-GRS [39]: 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.



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

Figure 9. Training loss by MGRS-HFA over 100 epochs on various datasets

FedPerGNN [40] : This framework offers privacy-preserving personalization. It uses a privacy-preserving model update method to train models on decentralized graphs inferred from local data.

GHRS [41] : This system uses a graph-based model, similarity of ratings, demographic and location information, and autoencoder feature extraction. The method improves performance in cold-start problems.

iii) Evaluation Metrics and Parameter Settings

A random allocation method splits the dataset into three parts, with a ratio of 8:1:1 for training, validation, and testing, respectively. We assess the performance of the top-K using widely recognized metrics such as Precision@K, Recall@K, Accuracy@K, F1-Score@K, RMSE, and NDCG@K. We set a value of K=10 for all models and calculate the mean score value accordingly. Precision measures the proportion of relevant recommendations, recall measures the proportion of recommended items, F1-Score balances precision and recall, NDCG evaluates ranking quality, RMSE measures the difference between predicted and actual ratings, and accuracy measures the proportion of correct recommendations. These metrics help determine the recommender system's accuracy and efficiency in providing users with the right items. The results shown in Tables II-IV highlight the system's performance across these metrics. Adam's optimizer trains the model with randomly initialized parameters using a Gaussian distribution, Sigmoid as the activation function, binary cross-entropy loss, and a learning rate of 0.001.

High precision ensures that recommendations are accurate based on user interactions with different data types. Recall is critical in ensuring that the system does not miss recommending relevant items across various modalities, capturing all items of interest. The F1-score helps assess the system's overall ability to both recommend relevant items (precision) and ensure that all relevant items are considered (recall), which is crucial for minimizing false positives and negatives. The NDCG plays a crucial role in evaluating the ranking of recommended items, guaranteeing optimal relevance and rank in the recommendation list, thereby boosting user satisfaction. RMSE is useful for evaluating how well the system predicts user preferences across different modalities and ensuring accurate predictions. Accuracy is an essential metric for evaluating the overall correctness of the system's output across different modalities, leading to higher user satisfaction and engagement. Figures 6-10 depict the outcomes.

iv) Scalability Analysis

We designed the proposed multimodal system to efficiently process diverse data types, particularly large-scale datasets like MovieLens-10M and MicroVideo 1.7M. It uses advanced hardware like the NVIDIA RTX 4060 Tensor Core GPU, ensuring high throughput for deep learning tasks. Table V presents detailed performance in terms of parameters, training time, and memory requirements for various datasets, showcasing the system's efficiency and scalability. The table also reveals that the system can handle smaller datasets (MovieLens 1M, TikTok) more efficiently due to their lower memory usage and processing time, compared to larger ones. As seen in Tables II and III the proposed model shows improved performance on smaller datasets. However, the model's performance decreases when applied to larger datasets due to increased data sparsity, complexity, noise, and computational constraints. These challenges highlight the model's limitations in handling larger datasets, as it requires more sophisticated optimization techniques and better data management strategies.

TABLE V. Parameters, Training Time, and Memory Footprint for various datasets

Dataset	#Param	Time	Memory
MovieLens 1M	12M	17m32s	468 MiB
MovieLens 10M	32M	1h5m13s	876 MiB
Tiktok	3M	7m43s	242 MiB
MicroVideo 1.7M	19M	28m56s	614 MiB

B. Performance Analysis

The experimental results shown in Tables II- IV, show that MGRS-HFA exhibits outstanding performance on various metrics, viz. precision, recall, NDCG, F1-score, accuracy, and RMSE for different models.

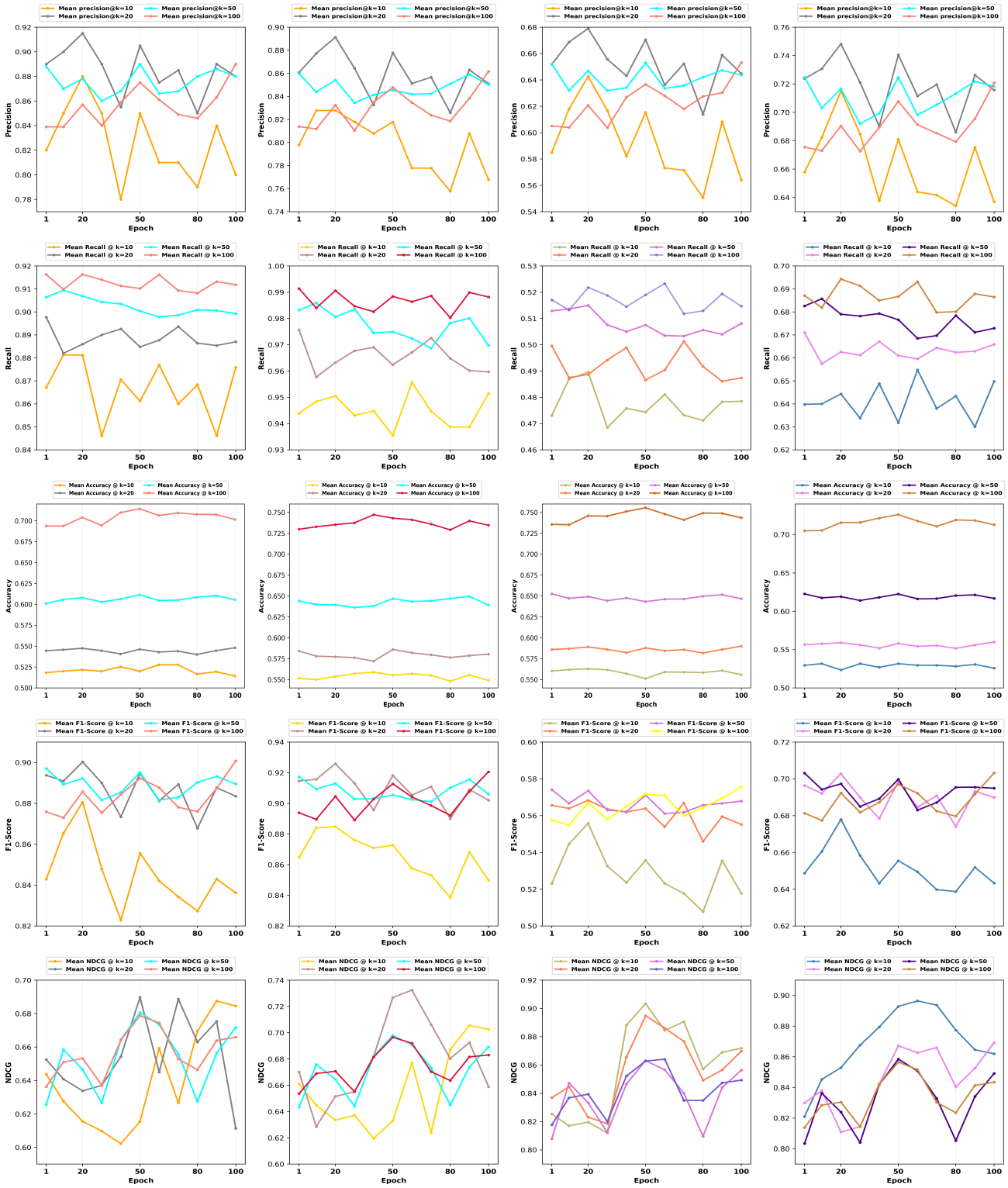


Figure 10. Performance of MGRS-HFA on various datasets (MovieLens 1M (1st image), TikTok (2nd image), MovieLens 10M (3rd image), MicroVideo 1.7M (4th image)) using different evaluation metrics for various values of K and Epochs.

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 improvements 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 MGRS-HFA model outperforms other models, achieving a 1% and 3% improvement using the RMSE metric.

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. It is crucial to tailor recommendation systems to each dataset, as each dataset exhibits unique performance characteristics.

5. Conclusion

The Multimodal Graph-based Recommendation System using Hybrid Filtering Approach (MGRS-HFA) framework improves 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 methods. The model uses a cross-attention mechanism and a 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. However, a notable limitation of the study is the increased computational complexity and resource requirements associated with integrating multimodal data and advanced graph-based methods. This approach faces challenges in scaling to large-scale datasets or real-time applications. Researchers can improve recommendation accuracy in the future by capturing complex user-item interactions using more advanced attention mechanisms and deep learning architectures. Adding more advanced recommendation methods, such as meta-learning and reinforcement learning, to the MGRS-HFA model can enhance its performance and flexibility. Additionally, optimization techniques and distributed computing can significantly improve the system's ability to handle large datasets.

References

[1] F. Zhou, B. Luo, T. Hu, Z. Chen, and Y. Wen, "A combinatorial recommendation system framework based on deep reinforcement learning," pp. 5733–5740, 2021.

[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," *IEEE Access*, vol. 8, pp. 115 694–115 716, 2020.

[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," pp. 3143–3146, 2020.

[4] W. Wang, X. Lin, F. Feng, X. He, and T.-S. Chua, "Generative recommendation: Towards next-generation recommender paradigm," *ArXiv*, vol. abs/2304.03516, 2023.

[5] H. Wang, N. Lou, and Z. Chao, "A personalized movie recommendation system based on lstm-cnn," pp. 485–490, 2020.

[6] Q. Liu, J. Hu, Y. Xiao, X. Zhao, J. Gao, W. Wang, Q. Li, and J. Tang, "Multimodal recommender systems: A survey," *ACM Computing Surveys*, vol. 57, no. 2, pp. 1–17, 2024.

[7] A. Fareed, S. Hassan, S. B. Belhaouari, and Z. Halim, "A collaborative filtering recommendation framework utilizing social networks," *Machine Learning with Applications*, vol. 14, p. 100495, 2023.

[8] F. Fkih, "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, 2022.

[9] F. Rezaimehr and C. Dadkhah, "A survey of attack detection approaches in collaborative filtering recommender systems," *Artificial Intelligence Review*, vol. 54, no. 3, p. 2011–2066, 2021.

[10] A. Feitosa, M. Macedo, M. Sibaldo, T. Carvalho, and J. Araujo, "Performance evaluation of collaborative filtering recommender algorithms," pp. 1–6, 2023.

[11] X. Zhou, D. Lin, and T. Ishida, "Evaluating reputation of web services under rating scarcity," pp. 211–218, 2016.

[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, p. 239–249, 2019.

[13] A. Zagranovskaia and D. Mitura, "Designing hybrid recommender systems," pp. 1–5, 2022.

[14] A. J. Ibrahim, P. Zira, and N. Abdulganiyyi, "Hybrid recommender for research papers and articles," *International Journal of Intelligent Information Systems*, vol. 10, no. 2, pp. 9–15, 2021.

[15] E. Jeong, X. Li, A. E. Kwon, S. Park, Q. Li, and J. Kim, "A multimodal recommender system using deep learning techniques combining review texts and images," *Applied Sciences*, vol. 14, no. 20, pp. 1–15, 2024.

[16] Y. Mu and Y. Wu, "Multimodal movie recommendation system using deep learning," *Mathematics*, vol. 11, no. 4, pp. 1–12, 2023.

[17] D. Roy and C. Ding, "Movie recommendation using youtube movie trailer data as the side information," pp. 275–279, 2020.

[18] S. Feng and T. Zhao, "Hybrid recommendation system," pp. 1–5, 2022.

[19] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv*, vol. abs/1907.11692, 2019.

[20] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," *arXiv*, vol. abs/2104.00298, 2021.

[21] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," *arXiv*, vol. abs/2102.00719, 2021.

[22] P. P. Liang, A. Zadeh, and L.-P. Morency, "Foundations trends in multimodal machine learning: Principles, challenges, and open questions," *ACM Computing Surveys*, vol. 56, no. 10, pp. 1–42, 2024.

[23] C. Mao, Z. Wu, Y. Liu, and Z. Shi, "Matrix factorization recommendation algorithm based on attention interaction," *Symmetry*, vol. 16, no. 3, 2024.

[24] K. Yousaf and T. Nawaz, "A deep learning-based approach for inappropriate content detection and classification of youtube videos," *IEEE Access*, vol. 10, pp. 16 283–16 298, 2022.

[25] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, pp. 373–440, 2019.

[26] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, X. He, and Y. Li, "A survey of graph neural networks for recommender systems: Challenges, methods, and directions," *ACM Transactions on Recommender Systems*, vol. 1, no. 1, pp. 1–51, 2023.

[27] C. Su, M. Chen, and X. Xie, "Graph convolutional matrix completion via relation reconstruction," pp. 51–56, 2021.

[28] Y. Deng, "Recommender systems based on graph embedding techniques: A review," *IEEE Access*, vol. 10, pp. 51 587–51 633, 2022.

[29] N. Mrabah, M. Bouguessa, and R. Ksantini, "A contrastive variational graph auto-encoder for node clustering," *Pattern Recognition*, vol. 149, no. C, p. 110209, 2024.

[30] W. Haixia, S. Chunyao, G. Yao, and G. Tingjian, "Link prediction on complex networks: An experimental survey," *Data Science and Engineering*, vol. 7, no. 3, pp. 253–278, 2022.

[31] Z. Ahmad and S. Rizos, "Similarity-based link prediction in social networks using latent relationships between the users," *Scientific Reports*, vol. 10, no. 1, 2020.

[32] V. T. Hoang, H.-J. Jeon, E.-S. You, Y. Yoon, S. Jung, and O.-J. Lee, "Graph representation learning and its applications: A survey," *Sensors*, vol. 23, no. 8, p. 4168, 2023.

[33] A. G. Vrahatis, K. Lazaros, and S. Kotsiantis, "Graph attention networks: A comprehensive review of methods and applications," *Future Internet*, vol. 16, no. 9, p. 318, 2024.

[34] 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, 2020.

[35] 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, 2020.

[36] J. Sun, H. Chang, W. Zhao, Y. Yu, L. Yang, and X. Huang, "A multimedia graph collaborative filter," *IEEE Access*, vol. 10, pp. 50 892–50 902, 2022.

[37] 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, 2022.

[38] X. Chen, D. Liu, Z. Xiong, and Z.-J. Zha, "Learning and fusing multiple user interest representations for micro-video and movie recommendations," *IEEE Transactions on Multimedia*, vol. 23, pp. 484–496, 2021.

[39] S. Gupta, A. K. Bindal, D. Prasad, and N. Raheja, "A hierarchical multi-modal content-based approach to graph-based recommendation system," vol. 1, pp. 394–399, 2024.

[40] C. Wu, F. Wu, Y. Cao, Y. Huang, and X. Xie, "A federated graph neural network framework for privacy-preserving personalization," *Nature Communications*, vol. 13, no. 1, p. 3091, 2022.

[41] Z. Zamanzadeh Darban and M. H. Valipour, "Ghrs: Graph-based hybrid recommendation system with application to movie recommendation," *Expert Systems with Applications*, vol. 200, p. 116850, 2022.