CineFinder: A Movie Recommendation System Using Visual and Textual Deep Features

Mehmet Tuğrul Sarıçiçek[1][a], Rukiye Orman[2][b], Murat Dener[1][c] and Harun Kınacı[3][d]
[1]*Information Security Engineering Department, Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkey*
[2]*Department of Computer Technologies, Vocational School of Technical Sciences, Ankara Yıldırım Beyazıt University, Ankara, Turkey*
[3]*Department of Business Administration, Faculty of Economics and Administrative Sciences, Erciyes University, Kayseri, Turkey*

Keywords: Movie Recommendation System, Deep Learning, Hybrid Recommendation, Visual Features, Textual Features, BERT, RoBERTa, SBert, VGG-16, ImageNet, ResNet, Cosine Similarity, Euclidean Distance, Manhattan Distance, Cold Start Problem, Machine Learning.

Abstract: In recent years, the increasing popularity of digital content platforms has highlighted the need for personalized recommendation systems, particularly in the entertainment industry. Traditional recommendation systems often suffer from limitations such as the "cold start" problem and inadequate personalization due to their reliance on limited user data. To address these challenges, this study proposes CineFinder. This hybrid feature-based movie recommendation system integrates both visual and textual deep features using multiple state-of-the-art pre-trained models. CineFinder extracts visual features from movie posters and backdrops using pre-trained convolutional neural networks—namely VGG-16, ResNet-50, and MobileNet—and captures textual features from movie overviews using pre-trained transformer-based models such as BERT, RoBERTa, and SBERT. These extracted features are fused into a comprehensive hybrid feature vector and utilized for similarity-based recommendations via Cosine similarity, Euclidean distance, and Manhattan distance. The system's performance was evaluated on two datasets created by the authors: the TMDB Dataset, which provides general audience metrics, and the TMDBRatingsMatched Dataset, which incorporates user-specific rating data from MovieLens 20M. Experimental results demonstrate that the proposed approach generates accurate and relevant movie recommendations while mitigating the cold start problem. The findings highlight the effectiveness of integrating multimodal deep learning techniques and leveraging user-driven feedback to enhance recommendation accuracy.

1 INTRODUCTION

The growing volume of digital content and user engagement has created a strong demand for more advanced recommendation systems. The growing volume of digital content and user engagement has created a strong demand for more advanced recommendation systems. While traditional recommendation techniques have been widely applied, they often struggle with personalization challenges and issues like the cold start problem. To overcome these limitations, modern approaches incorporate deep learning techniques and multimodal data sources, enhancing accuracy and personalization in recommendation systems.

In this study, a hybrid movie recommendation system has been developed for cinema enthusiasts, combining text and visual-based features using deep learning methods. This system utilizes pre-trained VGG-16, MobileNet, and ResNet-50 models to extract visual features from movie posters and

[a] https://orcid.org/0009-0004-9317-1112
[b] https://orcid.org/0000-0003-1385-0939
[c] https://orcid.org/0000-0001-5746-6141
[d] https://orcid.org/0000-0002-8572-1143

Sarıçiçek, M. T., Orman, R., Dener, M. and Kınacı, H.
CineFinder: A Movie Recommendation System Using Visual and Textual Deep Features.
DOI: 10.5220/0014385700004848

In *Proceedings of the 2nd International Conference on Advances in Electrical, Electronics, Energy, and Computer Sciences (ICEEECS 2025)*, pages 120-131
ISBN: 978-989-758-783-2

backdrops, and BERT, RoBERTa, and SBert models to extract text features from movie overviews. A hybrid recommendation mechanism combines feature extraction models in all possible combinations, ensuring a comprehensive feature representation. The recommendation process uses three similarity functions: Cosine Similarity, Euclidean Distance, and Manhattan Distance. These similarity functions evaluate the hybrid feature vectors to determine the most relevant movies for a favorite movie, providing a more comprehensive comparison across different feature extraction models.

The developed model has undergone a comprehensive testing process using a large-scale dataset created by the authors via TMDB API and another dataset by matching this dataset with the rating data from the MovieLens 20M dataset. Recommendations generated through these feature extraction combinations and multiple similarity functions aim to mitigate the effects of the cold start problem and enhance user satisfaction.

This study contributes to the literature in the following ways:

- Hybrid Recommendation Approach: Unlike traditional methods focusing solely on textual or visual analysis, this study integrates deep textual and visual features for movie recommendation, providing a more holistic approach.
- Diverse Similarity Metrics: The effectiveness of different feature combinations is evaluated using three distinct similarity metrics: Cosine Similarity, Euclidean Distance, and Manhattan Distance, ensuring a robust comparative analysis.
- New Dataset Contributions: Two new datasets, TMDB Dataset and TMDBRatingsMatched Dataset, have been created to facilitate further research in the field. The TMDB Dataset includes an extensive collection of movies with various attributes, while the TMDBRatingsMatched Dataset incorporates user-specific rating data matched with the MovieLens 20M dataset. These datasets will be publicly available on platforms such as Kaggle and Hugging Face following the publication of this paper, contributing to future research in recommendation systems.
- Cold Start Problem Mitigation: The proposed system effectively solves the cold start problem in movie recommendation by leveraging deep learning-based feature extraction and a hybrid recommendation mechanism.
- Comprehensive Evaluation Framework: The proposed system systematically compares multiple feature extraction models and similarity metrics, demonstrating the impact of different model combinations on recommendation accuracy.

These contributions establish a strong foundation for advancing hybrid recommendation models by integrating multimodal deep learning techniques with similarity-based recommendation methods.

The rest of this paper is organized as follows: Section 2 provides an overview of related work in recommendation systems. Section 3 describes the implementation details of the proposed hybrid recommendation system, including data preprocessing, feature extraction, and similarity measurement methods. Section 4 presents the experimental results and performance evaluation of the system. Finally, Section 5 concludes the study and discusses potential future research directions.

2 LITERATURE REVIEW

Recommendation systems have emerged as an essential tool for providing users personalized content in various domains, including e-commerce, entertainment, and healthcare. Traditional recommendation approaches, such as collaborative and content-based filtering, have demonstrated effectiveness but also suffer from inherent limitations, including the cold start problem, data sparsity, and scalability challenges. Recent advancements in deep learning and hybrid recommendation methods have significantly enhanced the accuracy and adaptability of these systems by incorporating multimodal data sources, such as textual and visual features. This section reviews contemporary research efforts to improve recommendation performance through deep learning models, feature extraction techniques, and hybrid recommendation strategies.

Numerous studies have focused on applying recommendation systems in various fields and improving their success. Kumar and Kumar [1] aimed to develop a system tested on music and hotel datasets that could provide recommendations even to users logging in for the first time (Kumar & Kumar, 2022). Iwendi et. Al aimed to develop a product recommendation system that combines item-to-item collaborative filtering with machine learning to provide more accurate recommendations (Iwendi et al., 2021). Ullah et. al aimed to develop a product recommendation system that suggests similar products based on a user's product image (Ullah et al., 2020). Yoon and Choi aimed to develop a recommendation system that suggests customized tourist destinations tailored to specific types of tourists by considering real-time changing factors such as external conditions and distance information

(Yoon & Choi, 2023). Aktas and Ciloglugil aimed to analyze student interactions in an educational recommender system by examining navigation patterns and evaluating the effectiveness of personalized learning material suggestions (Aktas & Ciloglugil, 2024). Abbas et. al aimed to develop a drug recommendation and supply chain management system that includes a drug recommendation module by extracting features from drug reviews (Abbas et al., 2020). Choi et. al aimed to develop a service part recommendation system for service engineers by combining clustering and machine learning methods (Choi et al., 2022). Iwendi et. al aimed to develop an IoMT (Internet of Medical Things)-based patient diet recommendation system (Iwendi et al., 2020). From these works, we can see that recommendation systems can be utilized in many different fields.

Huang et. al aimed to develop a more accurate and efficient recommendation system by combining deep learning and machine learning methods in a hybrid manner. Their study developed a model called DMFL (Deep Metric Factorization Learning), which combines factorization machine and metric learning. The model consists of two parts: feature learning and recommendation generation. The feature learning part of the model developed in the study consists of two parallel deep neural networks that extract static item latent feature vectors and dynamic user latent feature vectors. On the other hand, the recommendation generation part comprises sublayers, including a factorization machine, a deep neural network, and metric learning. According to the paper of the authors, the developed model was trained and tested on open source MovieLens 20M, MovieLens 1M, and BookCrossing datasets and showed higher Recall and AUC (Area Under The Curve) scores than other traditional methods (Huang et al., 2019).

Chen et. al aimed to improve recommendation accuracy for movie recommendation systems by utilizing visual content. Their study focuses on how movie frames and poster visuals can significantly address data sparsity and cold start problems. The authors developed a movie recommendation system called UVMF (Unified Visual Contents Matrix Factorization), which integrates CNN (Convolutional Neural Network) and PMF (Probabilistic Matrix Factorization) for feature extraction, and VGG-16 was used for learning visual features. MovieLens 2011 and IMDB datasets, which include movie metadata, user data, posters, and movie frames, were used to train and test the model. The authors separated the datasets into 70% train and 30% test. RMSE (Root Mean Square Error), precision, and recall metrics were adopted to evaluate the model's performance, and it was stated that the developed model achieved over 70% success (Chen et al., 2018).

Harshvardhan et. al develop a movie recommendation system called UBMTR (Unsupervised Boltzmann Machine-based Time-aware Recommendation System) which combines movie rating data with time information to consider users' past preferences and the time factor. The study investigates the impact of the time information on user preferences and the feasibility of integrating this information into recommendation systems using RBM. Their study created a 3D tensor data structure by combining users' movie rating data with time information. The model was tested on the MovieLens100K dataset, and its performance was measured as 88% based on the RMSE metric and 76% based on the MSE (Mean Squared Error) metric (Harshvardhan et al., 2022).

Wei et. al aimed to develop a hybrid movie recommendation system by combining tags and ratings elements from SMN (Social Movie Networks). The study emphasized that movie recommendation systems should consider social elements such as users, media items, tags, ratings, and tag assignments. It also examined how SMNs incorporate these aspects into their recommendations. The model developed in the study operates based on the tripartite relationships between users, movies, tags, and ratings. Users' interaction with movies is addressed through ratings and tags integrated to model user preferences accurately. The authors developed a model based on SVD (Singular Value Decomposition) and matrix factorization techniques. Multidimensional vectors for users and movies were created based on the social influences of users and tags to model latent factors in users' preferences. By learning these multidimensional vectors, the developed model generates predictions based on users' past interactions, including ratings and tagging behaviors. The developed model was trained and tested with the MovieLens dataset. The model's performance was evaluated using precision and recall metrics on datasets containing different numbers of movies. It was noted that the model's performance improved as the dataset size increased (Wei et al., 2016).

Zhang et. al aimed to develop a real-time personalized movie recommendation system. In their study, the authors used the K-Means algorithm to measure user similarities and grouped them into clusters. For each cluster, they created a virtual opinion leader representing the users in that cluster, thereby reducing the size of the user-movie matrix.

The fundamental principle of the developed model is to cluster users based on their profile features, create a virtual user for each cluster, and then predict ratings between each virtual user and the movies. The authors developed an algorithm named Weighted KM-Slope-VU One to make recommendations. They compared their algorithm with KM-Slope-VU, SVD, and SVD++ by testing it on the MovieLens dataset. In the comparison based on the RMSE metric, the developed model achieved higher accuracy than SVD and SVD++ but fell short of KM-Slope-VU (Zhang et al., 2020).

While previous studies have explored various recommendation system techniques, they often focus on either textual analysis (e.g., collaborative filtering, SVD), visual analysis (e.g., CNN-based approaches), or user interaction-based methods (e.g., clustering, latent factor models). Unlike these approaches, our study systematically integrates textual and visual deep features to enhance movie recommendation accuracy. Specifically, we employ multiple state-of-the-art pre-trained models—VGG-16, ResNet-50, and MobileNet for visual features, and BERT, RoBERTa, and SBERT for textual features—creating a more robust and diverse hybrid feature representation. Furthermore, unlike many prior works that rely solely on traditional collaborative filtering or matrix factorization, our approach incorporates multiple similarity metrics, including Cosine Similarity, Euclidean Distance, and Manhattan Distance, to provide a more comprehensive evaluation of movie similarities. Additionally, we introduce two new datasets, TMDB Dataset and TMDBRatingsMatched Dataset, specifically created to facilitate multimodal recommendation evaluation, addressing the limitations of existing datasets that often lack rich visual and textual data integration. By combining deep feature extraction with diverse similarity functions and novel dataset contributions, our work offers a more holistic solution to movie recommendation challenges, particularly in mitigating the cold start problem.

3 IMPLEMENTATION OF RECOMMENDATION SYSTEM

This paper uses visual, thematic, and content-based evaluations to develop a recommendation system for movie enthusiasts based on their favorite movies. The developed recommendation system suggests the top 3 similar movies based on the poster, backdrop, and overview features of users' favorite movie.

3.1 Dataset Preparation

The dataset in this study was created by the authors, who developed a Python application and TMDB API (TMDB, 2025). This created dataset, TMDB Dataset, contains 15 features for 48,138 movies. It consists of movies in the TMDB (The Movie Database) released between 1960 and 2024. During dataset construction, movies with missing visual (poster and backdrop) or textual (overview) content were removed, ensuring a more complete dataset. The features included in the created data set are explained in Table 1.

Table 1: Explanation of the features of the TMDB Dataset.

S1. No	Feature Name	Feature Description
1	ID	The release date of the movie
2	Release Date	The release date of the movie
3	Overview	A summary of the movie's plot
4	Genres	The movie's categories or types
5	Production Countries	The countries where the movie was produced
6	Original Language	The primary language of the movie
7	Runtime	The total duration of the movie in minutes
8	Poster File	The image file path of the movie's poster
9	Release Year	The release year of the movie
10	Original Title	The movie's title in its original language
11	Popularity	A numerical value indicating the movie's overall popularity
12	Vote Count	The total number of votes the movie has received
13	Vote Average	The average rating the movie has received based on user votes
14	IMDB ID	The unique ID assigned to the movie on IMDB (Internet Movie Database)
15	Backdrop File	The image file path of the movie's backdrop

The TMDBRatingsMatched Dataset was created by matching the TMDB Dataset with the MovieLens 20M dataset using the TMDB ID field, which is common in both datasets (GroupLens). The movies matching TMDB IDs were retained, and a new field, MovieLens_ID, was added to the dataset to establish this correspondence. This enriched dataset was merged with the ratings data from MovieLens 20M

using MovieLens_ID as the foreign key, enabling user-specific rating information to be incorporated. As a result, the TMDBRatingsMatched Dataset allows recommendation evaluation based on general audience metrics (popularity, average rating) and personalized user ratings. In addition to the features in the TMDB dataset, the features in the new TMDBRatingsMatched Dataset are explained in Table 2.

Table 2: Explanation of the additional features of the TMDBRatingsMatched Dataset.

S1. No	Feature Name	Feature Description
1	MovieLens ID	The unique ID assigned to the movie on the MovieLens
2	User ID	The unique ID assigned to the user for ratings
3	Movie ID	The foreign key ID referenced the MovieLens ID on the ratings
4	Rating	The score given by a user to a movie
5	Timestamp	The Unix timestamp representing the exact time when the rating was given

The authors created two datasets to measure the performance of the developed system. The TMDB dataset is used to assess success based on general audience evaluations, such as average vote similarity, and the TMDBRatingsMatched dataset is used to measure success based on the similarity of ratings from users who have watched both favorite and recommended movies. Figure 1 shows the distribution of movies in the TMDB Dataset, and Figure 2 shows the distribution of movies in the TMDBRatingsMatched Dataset over 10-year periods.

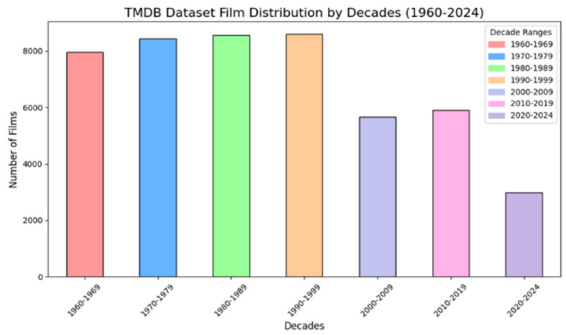

Figure 1: Distribution of Movies in the TMDB Dataset over 10-year periods.

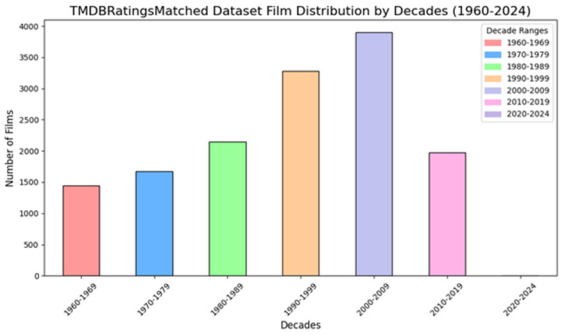

Figure 2: Distribution of Movies in the TMDBRatingsMatched Dataset over 10-year periods

3.2 Feature Extraction and Hybrid Feature Vector Creation

In this developed movie recommendation system, the features of poster and backdrop images, representing the aesthetic and thematic content of the movies, along with the overview features, representing their semantic context, were extracted. These three features were combined to create a comprehensive hybrid feature vector for the movies.

The VGG-16 pre-trained model, consisting of 16 layers and trained on the ImageNet dataset, was chosen to extract visual features from movie posters and backdrop images because it has demonstrated success in visual feature extraction (Baby et al., 2021; Kawaguchi et al., 2019). The VGG-16 model is a CNN with 16 layers trained on the ImageNet Dataset (Suganeshwari et al., 2023). In this study, only the feature map of posters and backdrops was required; therefore, the final layers were removed, and only 13 layers were utilized for visual feature extraction. Before input into the model, images were resized to 224×224 pixels and normalized using the VGG-16 preprocessing function to ensure compatibility with the pre-trained model.

The ResNet-50 (Residual Network-50) is a convolutional neural network model that was also trained on the ImageNet dataset, and was incorporated to enhance feature extraction by mitigating the vanishing gradient problem through skip connections. ResNet-50 consists of 50 layers, including convolutional layers organized in residual block connections (Kumar et al., 2021; Zhou et al., 2024). In this study, only the convolutional layers were retained, and the fully connected layers were removed to focus on feature extraction. The extracted feature maps were processed using Global Average Pooling (GAP) to generate compact feature representations. Before feature extraction, input images were resized to 224×224 pixels and

normalized using the ResNet-specific preprocessing function to maintain consistency across visual data.

The MobileNet model is a convolutional neural network designed for mobile and resource-constrained devices (Abbass & Ban, 2024; Chawla et al., 2024). It utilizes depthwise separable convolutions to reduce computational complexity while maintaining accuracy. MobileNet is trained on the ImageNet dataset and employs depthwise and point-wise convolutions to optimize efficiency. In this study, only the convolutional layers were retained, and the fully connected layers were removed to focus on feature extraction. The extracted feature maps were processed using Global Average Pooling (GAP) to generate compact feature representations. The input images were resized to 224×224 pixels and normalized using the MobileNet preprocessing function before being passed through the model.

BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based pre-trained deep learning model used for NLP (Natural Language Processing) tasks (Subakti et al., 2022). Instead of analyzing the words sequentially, BERT can analyze the entire text and extract semantic feature representations (Li et al., 2019; Subakti et al., 2022; Yang & Cui, 2021). This study used the BERT model to extract semantically enriched feature vectors from movie overviews. A single feature vector was obtained for each movie overview by averaging the final layer. Before input into the model, each movie overview was first tokenized using the BERT tokenizer, applying padding and truncation to standardize sequence length. The tokenized text was then passed through the model to extract contextualized word embeddings, which were processed using mean pooling to generate a fixed-length feature vector.

The RoBERTa (Robustly Optimized BERT Pretraining Approach) model is an improved version of BERT, designed to enhance training efficiency through dynamic masking, longer training durations, and larger datasets (Fan et al., 2025; Li et al., 2024). Unlike BERT, RoBERTa removes the Next Sentence Prediction (NSP) task and optimizes pretraining strategies, improving performance in various NLP tasks. It employs self-attention mechanisms for efficient sequential data handling. It has been pre-trained on extensive corpora, including BookCorpus and English Wikipedia, with a masked language modeling (MLM) objective (Lak et al., 2024). The RoBERTa-base variant was used in this study, and the final hidden layer representation was averaged to generate a single fixed-length feature vector. Like BERT, the input text was tokenized using the RoBERTa tokenizer, with appropriate padding and truncation applied. The processed tokens were then passed through the model to extract deep semantic representations, which were averaged to produce the final feature vector.

The SBERT (Sentence-BERT) model is an optimized transformer model designed for sentence-level semantic similarity tasks, improving efficiency over traditional BERT-based models by leveraging a Siamese network structure (Chi & Jang, 2024; Ortakci, 2024). SBERT fine-tunes BERT for sentence similarity by employing a pooling operation to generate fixed-size sentence embeddings, making it highly effective for clustering and similarity comparisons (Chi & Jang, 2024; Ortakci, 2024). The model utilizes cosine similarity to measure sentence relationships in a high-dimensional space, reducing computational cost while maintaining performance (Chi & Jang, 2024). The overview text was first tokenized using the SBERT tokenizer, and the resulting tokenized representation was fed into the model. The model's output embeddings were pooled using mean pooling to create a dense, fixed-size vector for each movie overview. In this study, the all-MiniLM-L6-v2 variant was used, which balances computational efficiency and representation quality, and the sentence embeddings were directly extracted to represent movie overviews as dense vectors.

The overview feature vector obtained using BERT, RoBERTa, and SBERT, poster and backdrop feature vectors obtained using VGG-16, ResNet-50, and MobileNet, were concatenated horizontally to form a hybrid vector that encompasses the aesthetic, thematic, and contextual features of the movies. The extracted features from all possible combinations of textual (BERT, RoBERTa, SBERT) and visual (VGG-16, ResNet-50, MobileNet) models were concatenated to create hybrid feature vectors. These hybrid representations were then used to generate movie recommendations based on three different similarity functions: Cosine Similarity, Euclidean Distance, and Manhattan Distance. As a result, recommendations were generated for nine different model combinations for each movie. Since three different similarity measures were applied to each combination, 27 recommendation results were obtained for a single movie. The steps for extracting movie features and creating the hybrid feature vector are explained in Table 3.

Table 3: Steps For Extracting Movies' Features and Creating the Hybrid Feature Vector.

Step	Description
1	Load the pre-trained transformer-based models (BERT, RoBERTa, and SBERT) and their respective tokenizers for text feature extraction.
2	Tokenize the movie overview using each model's tokenizer with appropriate padding and truncation.
3	Pass the tokenized text into each transformer model to obtain the final hidden states.
4	Apply mean (or appropriate) pooling on the hidden states along the token dimension to generate a fixed-length feature vector for each movie overview per model.
5	Load the pre-trained convolutional neural network models (VGG-16, ResNet-50, and MobileNet) with ImageNet weights, configured to exclude their fully connected layers.
6	Resize and preprocess the movie poster and backdrop images to the required input size (e.g., 224x224 pixels).
7	Pass the preprocessed images through each visual model and apply Global Average Pooling on the convolutional feature maps to obtain flattened feature vectors for each image.
8	Normalize all extracted feature vectors (both textual and visual) using a StandardScaler or equivalent to ensure consistent scaling.
9	Concatenate the normalized feature vectors from the multiple textual and visual models to form a comprehensive hybrid feature vector for each movie.

3.3 Recommendation Method

The movie recommendation system leverages multiple similarity functions to identify and suggest movies similar to a given input favorite movie. Three different distance functions are used to evaluate movie similarities: Cosine Similarity, Euclidean Distance, and Manhattan Distance. These metrics allow for a more comprehensive comparison of high-dimensional data, such as the combined feature vectors of movies. Each movie is represented by a combined feature vector, created by concatenating:

1) **Textual features** from the movie overview, extracted using a pre-trained BERT model with mean pooling.

2) **Visual features** from poster and backdrop images, obtained using a pre-trained VGG16 model with global average pooling.

Equation (1) shows how cosine similarity is computed between the input movie's feature vector and all other movies in the dataset.

$$\text{Cosine Similarity} = \frac{\boldsymbol{A}\cdot\boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|} \quad (1)$$

Similarly, Equations (2) and (3) give the Euclidean Distance and Manhattan Distance formulas.

$$\text{Euclidean Distance} = \sqrt{\left\{\sum_{\{i=1\}}^{\{n\}}(A_i - B_i)^2\right\}} \quad (2)$$

$$\text{Manhattan Distance} = \sum_{\{i=1\}}^{\{n\}}|A_i - B_i| \quad (3)$$

The similarity scores are calculated using all three metrics for each input favorite movie, and movies are ranked accordingly. Based on the computed similarity scores, the top 5 most similar movies are selected as recommendations.

3.4 Evaluation Metric

In this study, the success of the developed system was evaluated using RMSE, MSE, and precision metrics for the TMDB Dataset and TMDBRatingsMatched Dataset, which the authors created. However, the metrics were calculated using different methods for each dataset.

Since the TMDB Dataset does not contain user-based rating data for the first and larger dataset, the system's performance was measured by comparing the average rating of the recommended movies with that of the favorite movie. This evaluation was conducted using RMSE, MSE, and precision metrics.

For the second and smaller dataset, the TMDBRatingsMatched Dataset, which includes user-based rating data, the system's performance was evaluated using RMSE, MSE, and precision metrics by comparing the ratings given by ordinary users who have watched both the favorite movie and the recommended movies.

4 EXPERIMENTAL RESULTS

Experiments were performed on a system equipped with an 11th-generation Intel Core i5 processor, 16GB RAM, and an NVIDIA RTX 3050 Ti GPU with 4GB GDDR6 VRAM running on Windows 11.

Three favorite movies were randomly selected from the TMDBRatingsMatched Dataset to test the developed system, and recommendations were generated for these movies. The testing was conducted on the entire TMDB Dataset and the entire TMDBRatingsMatched Dataset. For each selected movie, recommendations were made using all

possible feature extraction model combinations, which include:

1) **Visual feature extraction models**: VGG-16, ResNet-50, and MobileNet.

2) **Textual feature extraction models**: BERT, RoBERTa, and SBERT.

Each visual and textual model was systematically combined to generate different hybrid feature representations (e.g., VGG-16 + BERT, ResNet-50 + RoBERTa, MobileNet + SBERT, etc.). These feature representations were then used to compute similarity scores using three functions: **Cosine Similarity**, **Euclidean Distance**, and **Manhattan Distance**. For each selected movie, three movie recommendations were generated using the similarity functions, ensuring a comprehensive evaluation of the system. The testing was conducted on the entire TMDB Dataset and the entire TMDBRatingsMatched Dataset.

When the developed system was tested on the TMDBRatingsMatched Dataset, the randomly selected favorite movies were Patrik, Age 1.5, Kung Fu Panda: Secrets of the Furious Five, and Get the Gringo.

1) **For Patrik, Age 1.5**, the top five recommendation results were achieved using the combinations BERT + ResNet + Euclidean, SBERT + MobileNet + Cosine, SBERT + MobileNet + Euclidean, SBERT + MobileNet + Manhattan, and BERT + MobileNet + Euclidean, with the best cases yielding RMSE: 0.00, MSE: 0.00, and Precision: 100.00%.

2) **For Kung Fu Panda: Secrets of the Furious Five**, the highest-performing combinations were RoBERTa + MobileNet + Cosine, BERT + MobileNet + Euclidean, BERT + ResNet + Cosine, BERT + MobileNet + Cosine, and SBERT + VGG-16 + Cosine, with the best result achieved using RoBERTa + MobileNet + Cosine (RMSE: 0.58, MSE: 0.34, Precision: 96.40%).

3) **For Get the Gringo**, the top five combinations were SBERT + VGG-16 + Euclidean, BERT + VGG-16 + Manhattan, SBERT + MobileNet + Cosine, BERT + MobileNet + Cosine, and RoBERTa + ResNet + Cosine, with the best performance obtained using SBERT + VGG-16 + Euclidean (RMSE: 0.99, MSE: 0.98, Precision: 92.00%). The detailed results of the test conducted on the TMDBRatingsMatched Dataset, including the top 5 performing model combinations for each movie, are presented in Table 4.

Table 4: Top 5 Model Combinations per Movie on the TMDBRatingsMatched Dataset.

Favorite Movie	Model Combination	Distance Function	Recommended Movies	RMSE	MSE	Precision
Patrik, Age 1.5	BERT + ResNet	Euclidean	A Night in the Life of Jimmy Reardon (1988) Tristana (1970) Blind Dating (2006)	0.0	0.0	100%
Patrik, Age 1.5	SBERT + MobileNet	Cosine	The Longest Week (2014) The Substance of Fire (1996) Hemingway & Gellhorn (2012)	0.0	0.0	100%
Patrik, Age 1.5	SBERT + MobileNet	Euclidean	Persuasion (2007) The Substance of Fire (1996) Tristana (1970)	0.0	0.0	100%
Patrik, Age 1.5	SBERT + MobileNet	Manhattan	Persuasion (2007) Tristana (1970) The Substance of Fire (1996)	0.0	0.0	100%
Patrik, Age 1.5	BERT + MobileNet	Euclidean	Saraband (2003) The Rebound (2009) The Opportunists (2000)	0.35	0.12	100%
Kung Fu Panda: Secrets of the Furious Five	RoBERTa + MobileNet	Cosine	Kung Fu Panda 2 (2011) Kung Fu Panda (2008) The Lion King 1½ (2004)	0.58	0.34	96.40%
Kung Fu Panda: Secrets of the Furious Five	BERT + MobileNet	Euclidean	Kung Fu Panda 2 (2011) Kung Fu Panda (2008) Blackbeard's Ghost (1968)	0.59	0.35	95.95%
Kung Fu Panda: Secrets of the Furious Five	BERT + ResNet	Cosine	Kung Fu Panda 2 (2011) Kung Fu Panda (2008) Kung Fu Panda Holiday (2010)	0.59	0.35	99.95%

Kung Fu Panda: Secrets of the Furious Five	BERT + MobileNet	Cosine	Kung Fu Panda 2 (2011) Kung Fu Panda (2008) Legend of the BoneKnapper Dragon (2010)	0.59	0.35	95.95%
Kung Fu Panda: Secrets of the Furious Five	SBERT + VGG-16	Cosine	Kung Fu Panda 2 (2011) Open Season 3 (2010) Kung Fu Panda (2008)	0.59	0.35	95.95
Get the Gringo	SBERT + VGG-16	Euclidean	Murder in the First (1995) Chained (2012) Murder on a Sunday Morning (2001)	0.99	0.98	92.00%
Get the Gringo	BERT + VGG-16	Manhattan	Birdman of Alcatraz (1962) Chained (2012) Murder in the First (1995)	0.92	0.84	90.31%
Get the Gringo	SBERT + MobileNet	Cosine	Mud (2013) Training Day (2001) The Criminal (1960)	0.93	0.86	86.92%
Get the Gringo	BERT + MobileNet	Cosine	Mud (2013) The Castaway Cowboy (1974) It's Kind of a Funny Story (2010)	0.99	0.98	86.08%
Get the Gringo	RoBERTa + ResNet	Cosine	Coach Carter (2005) Firestorm (1998) I Am David (2003)	0.98	0.96	82.05%

When the developed system was tested on the TMDB Dataset using the same randomly selected favorite movies as in the TMDBRatingsMatched Dataset (Patrik, Age 1.5, Kung Fu Panda: Secrets of the Furious Five, and Get the Gringo), the following results were obtained:

1) **For Patrik, Age 1.5**, the top five recommendation results were achieved using the combinations SBERT + VGG-16 + Manhattan, SBERT + MobileNet + Manhattan, SBERT + MobileNet + Euclidean, BERT + MobileNet + Manhattan, and BERT + MobileNet + Euclidean, with the best result achieved using SBERT + VGG-16 + Manhattan (RMSE: 0.26, MSE: 0.07, Precision: 100.0)

2) **For Kung Fu Panda: Secrets of the Furious Five**, the highest-performing combinations were BERT + ResNet + Manhattan, BERT + MobileNet + Cosine, RoBERTa + MobileNet + Cosine, RoBERTa + ResNet + Euclidean, and BERT + ResNet + Cosine, with the best result achieved using BERT + ResNet + Manhattan (RMSE: 0.18, MSE: 0.03, Precision: 100.00%).

3) **For Get the Gringo**, the top five combinations were BERT + MobileNet + Cosine, SBERT + MobileNet + Cosine, SBERT + ResNet + Manhattan, RoBERTa + VGG-16 + Cosine, and RoBERTa + MobileNet + Cosine, with the best performance obtained using BERT + MobileNet + Cosine (RMSE: 0.42, MSE: 0.17, Precision: 100.00%). The detailed results of the test conducted on the TMDB Dataset, including the top 5 performing model combinations for each movie, are presented in Table 5.

Table 5: Top 5 Model Combinations per Movie on the TMDB Dataset.

Favorite Movie	Model Combination	Distance Function	Recommended Movies	RMSE	MSE	Precision
Patrik, Age 1.5	SBERT + VGG-16	Manhattan	Tristana (1970) A Little Game (1971) Any Day Now (2012)	0.26	0.07	100.0
Patrik, Age 1.5	SBERT + MobileNet	Manhattan	Persuasion (2007) Tristana (1970) Through My Window 3: Looking at You (2024)	0.29	0.08	100.0
Patrik, Age 1.5	SBERT + MobileNet	Euclidean	Switched at Birth (1991), Persuasion (2007) The Little Gangster (1990)	0.5	0.25	100.0

Patrik, Age 1.5	BERT + MobileNet	Manhattan	Switched at Birth (1991) The Elder Son (1975) The Rebound (2009)	0.5	0.25	100.0
Patrik, Age 1.5	BERT + MobileNet	Euclidean	Switched at Birth (1991) The Elder Son (1975) Saraband (2003)	0.55	0.3	100.0
Kung Fu Panda: Secrets of the Furious Five	BERT + ResNet	Manhattan	Lupin the Third: Dragon of Doom (1994) Kung Fu Panda 2 (2011) Animal Crossing: The Movie (2006)	0.18	0.03	100.0
Kung Fu Panda: Secrets of the Furious Five	BERT + MobileNet	Cosine	Kung Fu Panda 2 (2011) Kung Fu Panda: Secrets of the Masters (2011) Kung Fu Panda: Secrets of the Scroll (2016)	0.27	0.07	100.0
Kung Fu Panda: Secrets of the Furious Five	RoBERTa + MobileNet	Cosine	Kung Fu Panda 2 (2011) The Big Trip (2019) Kung Fu Panda (2008)	0.38	0.14	100.0
Kung Fu Panda: Secrets of the Furious Five	RoBERTa + ResNet	Euclidean	Animal Crossing: The Movie (2006) Boniface's Holiday (1965) Tenchi the Movie 2: The Daughter of Darkness (1997)	0.4	0.16	100.0
Kung Fu Panda: Secrets of the Furious Five	BERT + ResNet	Cosine	Kung Fu Panda 2 (2011) Kung Fu Panda (2008) Kung Fu Panda: Secrets of the Masters (2011)	0.41	0.17	100.0
Get the Gringo	BERT + MobileNet	Cosine	Tailcoat for an Idler (1979) Mud (2013) The System (2022)	0.42	0.17	100.0
Get the Gringo	SBERT + MobileNet	Cosine	Mud (2013), One Ranger (2023), Training Day (2001)	0.62	0.38	100.0
Get the Gringo	SBERT + ResNet	Manhattan	Legend (2015) The Boondock Saints (1999) The Price We Pay (2023)	0.63	0.39	100.0
Get the Gringo	ReBERTa + VGG-16	Cosine	Paradise (1991) Tailcoat for an Idler (1979) The Warning (2018)	0.77	0.6	66.67
Get the Gringo	ReBERTa + MobileNet	Cosine	Tailcoat for an Idler (1979) The Diary of Anne Frank (1980) The Warning (2018)	0.79	0.63	66.67

The experimental results presented in Tables 4 and 5 indicate that the CineFinder recommendation system achieves higher accuracy when evaluated on the TMDBRatingsMatched dataset, which incorporates user-specific rating data, compared to the TMDB dataset, which relies on general audience evaluations. For instance, in the case of Patrik, Age 1.5, multiple model combinations achieved near-zero error rates (RMSE and MSE) and 100% precision in the TMDBRatingsMatched dataset. In contrast, the error rates were slightly higher when evaluated on the TMDB dataset. A similar pattern was observed for Kung Fu Panda: Secrets of the Furious Five and Get the Gringo, where the system demonstrated significantly lower error rates in the user-based dataset. These results highlight the importance of leveraging user-driven feedback to enhance recommendation accuracy, as user-specific ratings provide a more reliable foundation for evaluating the relevance of suggested movies compared to general audience metrics such as average vote scores.

Moreover, comparing the best-performing model combinations in the two datasets reveals some key differences. In the TMDBRatingsMatched dataset, the highest accuracy was achieved for Patrik, Age 1.5, using the SBERT + MobileNet combination with all three similarity measures, all yielding RMSE: 0.00, MSE: 0.00, and Precision: 100.00%. However, in the TMDB dataset, the best result for Patrik, Age 1.5, was obtained using SBERT + VGG-16 with Manhattan

distance, achieving an RMSE of 0.26, MSE of 0.07, and 100.00% precision. Similarly, for Kung Fu Panda: Secrets of the Furious Five, the RoBERTa + MobileNet combination with Cosine similarity performed best in the TMDBRatingsMatched dataset (RMSE: 0.58, Precision: 96.40%), while in the TMDB dataset, the BERT + ResNet combination with Manhattan distance achieved the highest accuracy (RMSE: 0.18, Precision: 100.00%). In the case of Get the Gringo, the SBERT + VGG-16 combination with Euclidean distance yielded the best performance in the TMDBRatingsMatched dataset (RMSE: 0.99, Precision: 92.00%), whereas in the TMDB dataset, the BERT + MobileNet combination with Cosine similarity achieved the highest accuracy (RMSE: 0.42, Precision: 100.00%). These results indicate that while certain model combinations consistently performed well, the optimal feature extraction models varied between the datasets. This suggests that user-driven ratings influence which feature extraction approaches yield the most accurate recommendations, emphasizing the need for adaptive model selection strategies based on the data source.

5 CONCLUSION AND FUTURE WORK

This study developed CineFinder—a hybrid movie recommendation system—by integrating visual and textual deep features extracted via multiple state-of-the-art pre-trained models. The system was evaluated using two datasets: the TMDB Dataset, representing general audience metrics, and the TMDBRatingsMatched Dataset, which includes user-specific ratings. The experimental results demonstrate that the system achieves notably higher precision when evaluated with user-based rating data; in many cases, several feature combinations achieved a precision of 100% for the selected favorite movies. This outcome strongly suggests that leveraging user-driven feedback provides a more reliable basis for assessing recommendation accuracy than traditional popularity or average vote metrics.

Despite these promising results, several avenues for future work remain. First, further enhancement of CineFinder could be achieved by incorporating real-time interaction data—such as watch history, implicit feedback, and session-based activities—to adapt to evolving user preferences dynamically. Second, exploring more advanced similarity measures beyond standard cosine similarity, Euclidean distance, and Manhattan distance (for example, neural collaborative filtering or contrastive learning approaches) may further refine the recommendation process. Finally, extending the system into a scalable, web-based, or streaming platform deployment will facilitate real-world testing and validation, ensuring the system can effectively handle large-scale and diverse user interactions.

Overall, while CineFinder successfully demonstrates the benefits of integrating multimodal deep learning techniques for movie recommendation, the insights gained from the current study pave the way for future, more adaptive, robust, and user-centric recommendation systems.

REFERENCES

Abbas, K., Afaq, M., Khan, T. A., & Song, W. C. (2020). A Blockchain and Machine Learning-Based Drug Supply Chain Management and Recommendation System for Smart Pharmaceutical Industry. *ELECTRONICS*, *9*(5). https://doi.org/10.3390/electronics9050852

Abbass, M. A. B., & Ban, Y. S. (2024). MobileNet-Based Architecture for Distracted Human Driver Detection of Autonomous Cars. *ELECTRONICS*, *13*(2). https://doi.org/10.3390/electronics13020365

Aktas, C., & Ciloglugil, B. (2024). *Exploring the Navigation Patterns of Learners on an Educational Recommender System*

Baby, D., Devaraj, S. J., & Raj M. M, A. (2021). *Leukocyte Classification based on Transfer Learning of VGG16 Features by K-Nearest Neighbor Classifier* 2021 3rd International Conference on Signal Processing and Communication (ICPSC),

Chawla, T., Mittal, S., & Azad, H. K. (2024). MobileNet-GRU fusion is used to optimize the diagnosis of yellow vein mosaic virus. *Ecological Informatics*, *81*. https://doi.org/10.1016/j.ecoinf.2024.102548

Chen, X. J., Zhao, P. P., Xu, J. J., Li, Z. X., Zhao, L., Liu, Y. C., Sheng, V. S., & Cui, Z. M. (2018). Exploiting Visual Contents in Posters and Still Frames for Movie Recommendation. *IEEE ACCESS*, *6*, 68874-68881. https://doi.org/10.1109/Access.2018.2879971

Chi, T. Y., & Jang, J. S. R. (2024). WC-SBERT: Zero-Shot Topic Classification Using SBERT and Light Self-Training on Wikipedia Categories. *Acm Transactions on Intelligent Systems and Technology*, *15*(5), 1-18. https://doi.org/10.1145/3678183

Choi, Y.-H., Lee, J., & Yang, J. (2022). Development of a service parts recommendation system using clustering and classification of machine learning. *EXPERT SYSTEMS WITH APPLICATIONS*, *188*. https://doi.org/10.1016/j.eswa.2021.116084

Fan, M., Kong, M., Wang, X., Hao, F., & Zhang, C. (2025). FITE-GAT: Enhancing aspect-level sentiment classification with FT-RoBERTa induced trees and graph attention network. *EXPERT SYSTEMS WITH*

APPLICATIONS, *264*. https://doi.org/10.1016/j.eswa.2024.125890

GroupLens. *MovieLens 20M Dataset*. https://grouplens.org/datasets/movielens/20m/

Harshvardhan, G. M., Gourisaria, M. K., Rautaray, S. S., & Pandey, M. (2022). UBMTR: Unsupervised Boltzmann machine-based time-aware recommendation system. *Journal of King Saud University Computer and Information Sciences*, *34*(8), 6400-6413. https://doi.org/10.1016/j.jksuci.2021.01.017

Huang, Z. H., Yu, C., Ni, J., Liu, H., Zeng, C., & Tang, Y. (2019). An Efficient Hybrid Recommendation Model With Deep Neural Networks. *IEEE ACCESS*, *7*, 137900-137912. https://doi.org/10.1109/Access.2019.2929789

Iwendi, C., Ibeke, E., Eggoni, H., Velagala, S., & Srivastava, G. (2021). Pointer-Based Item-to-Item Collaborative Filtering Recommendation System Using a Machine Learning Model. *International Journal of Information Technology & Decision Making*, *21*(01), 463-484. https://doi.org/10.1142/s0219622021500619

Iwendi, C., Khan, S., Anajemba, J. H., Bashir, A. K., & Noor, F. (2020). Realizing an Efficient IoMT-Assisted Patient Diet Recommendation System Through Machine Learning Model. *IEEE ACCESS*, *8*, 28462-28474. https://doi.org/10.1109/Access.2020.2968537

Kawaguchi, K., Nishimura, H., Wang, Z., Tanaka, H., & Ohta, E. (2019). *Basic investigation of sign language motion classification by feature extraction using pre-trained network models* 2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM),

Kumar, C., & Kumar, M. (2022). User session interaction-based recommendation system using various machine learning techniques. *MULTIMEDIA TOOLS AND APPLICATIONS*, *82*(14), 21279-21309. https://doi.org/10.1007/s11042-022-13993-8

Kumar, R. L., Kakarla, J., Isunuri, B. V., & Singh, M. (2021). Multi-class brain tumor classification using residual network and global average pooling. *MULTIMEDIA TOOLS AND APPLICATIONS*, *80*(9), 13429-13438. https://doi.org/10.1007/s11042-020-10335-4

Lak, A. J., Boostani, R., Alenizi, F. A., Mohammed, A. S., & Fakhrahmad, S. M. (2024). RoBERTa, ResNeXt and BiLSTM with self-attention: The ultimate trio for customer sentiment analysis. *Applied Soft Computing*, *164*. https://doi.org/10.1016/j.asoc.2024.112018

Li, J., Zhang, C., & Jiang, L. L. (2024). Innovative Telecom Fraud Detection: A New Dataset and an Advanced Model with RoBERTa and Dual Loss Functions. *APPLIED SCIENCES-BASEL*, *14*(24). https://doi.org/ARTN 11628
10.3390/app142411628

Li, W., Gao, S., Zhou, H., Huang, Z., Zhang, K., & Li, W. (2019). *The Automatic Text Classification Method Based on BERT and Feature Union* 2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS),

Ortakci, Y. (2024). Revolutionary text clustering: Investigating transfer learning capacity of SBERT models through pooling techniques. *Engineering Science and Technology-an International Journal-Jestech*, *55*. https://doi.org/ARTN 101730
10.1016/j.jestch.2024.101730

Subakti, A., Murfi, H., & Hariadi, N. (2022). The performance of BERT as data representation of text clustering. *J Big Data*, *9*(1), 15. https://doi.org/10.1186/s40537-022-00564-9

Suganeshwari, G., Balakumar, R., Karuppanan, K., Prathiba, S. B., Anbalagan, S., & Raja, G. (2023). DTBV: A Deep Transfer-Based Bone Cancer Diagnosis System Using VGG16 Feature Extraction. *Diagnostics (Basel)*, *13*(4). https://doi.org/10.3390/diagnostics13040757

TMDB. (2025). *API Reference*. https://developer.themoviedb.org/reference/intro/getting-started

Ullah, F., Zhang, B. F., & Khan, R. U. (2020). Image-Based Service Recommendation System: A JPEG-Coefficient RFs Approach. *IEEE ACCESS*, *8*, 3308-3318. https://doi.org/10.1109/Access.2019.2962315

Wei, S. X., Zheng, X. L., Chen, D. R., & Chen, C. C. (2016). A hybrid approach for movie recommendation via tags and ratings. *Electronic Commerce Research and Applications*, *18*, 83-94. https://doi.org/10.1016/j.elerap.2016.01.003

Yang, Y., & Cui, X. (2021). Bert-Enhanced Text Graph Neural Network for Classification. *Entropy (Basel)*, *23*(11). https://doi.org/10.3390/e23111536

Yoon, J., & Choi, C. (2023). Real-Time Context-Aware Recommendation System for Tourism. *Sensors (Basel)*, *23*(7). https://doi.org/10.3390/s23073679

Zhang, J., Wang, Y. F., Yuan, Z. Y., & Jin, Q. (2020). Personalized Real-Time Movie Recommendation System: Practical Prototype and Evaluation. *TSINGHUA SCIENCE AND TECHNOLOGY*, *25*(2), 180-191. https://doi.org/10.26599/Tst.2018.9010118

Zhou, Y., Wang, Z. Q., Zheng, S. R., Zhou, L., Dai, L., Luo, H., Zhang, Z. C., & Sui, M. X. (2024). Optimization of automated garbage recognition model based on ResNet-50 and weakly supervised CNN for sustainable urban development. *Alexandria Engineering Journal*, *108*, 415-427. https://doi.org/10.1016/j.aej.2024.07.066