scientific reports

OPEN

A transformer-based architecture for collaborative filtering modeling in personalized recommender systems

Hikmat Ullah Khan[1]✉, Anam Naz[1], Fawaz Khaled Alarfaj[2]✉ & Naif Almusallam[2]

Recommender systems are considered powerful tools, in the modern era, for filtering a huge amount of information and delivering personalized content, particularly in domains like e-commerce, social media, and entertainment. In the context of movie recommendations, accurately modeling user preferences based on past interactions, ratings, and contextual metadata is crucial for enhancing user satisfaction. With the rising trends and influence of Artificial Intelligence (AI), advanced models are increasingly being employed to enhance the precision and adaptability of such systems. This study proposes a novel transformer-based architecture, MetaBERTTransformer4Rec(MBT4R), designed to outperform state of the art existing methods in the relevant literature. The extensive empirical analysis is carried out on two datasets which based on the same source, publicly available known as MovieLens, which is a standard dataset for movie recommendation. The proposed model utilizes a self-attention mechanism to effectively capture sequential dependencies and contextual relationships, enabling deeper understanding of users' preferences. The results reveal that MBT4R achieves the lowest RMSE of 0.62, MAE of 0.45, and highest R^2 of 0.39, significantly superior to the benchmarks established by traditional models including machine learning (DT, KNN, RF, XGB), matrix factorization (SVD) and deep learning (GRU). This research highlights the effectiveness of AI techniques in improving the accuracy and personalization of recommendation systems focusing on enhancing user satisfaction by accurately predicting user preferences and delivering tailored film suggestions. It also provides a pathway for future advancements in personalized user experiences across entertainment platforms.

Keywords Artificial intelligence, Deep learning, Recommender systems, Collaborative filtering, Personalization

In the digital age, online users are generating huge volume of content across various platforms and thus making it increasingly challenging to identify what is most relevant or valuable to them. Recommendation systems have emerged as essential tools for addressing this challenge by analyzing users' history to deliver them personalized suggestions[1]. Whether in streaming services, online retail, or news feeds, the ability to recommend highly-expected content not only increases users' satisfaction, but also enhanced revenue for business engagement, retention, and overall platform efficiency. By learning from patterns such as viewing history, ratings, and content preferences, the highly interactive systems aim to predict what a user is most likely to enjoy or find useful, thereby transforming raw data into meaningful and improved experiences[2].

Machine learning is one of the main tools which is being used by online movies platform such as Netflix for the film industry improving movies recommendations thus increasing revenue in the film industry worldwide[3]. Through the extraction of big data, which consists of viewers' preferences, watch list, and their rating and reviews, machine learning can analyze patterns and make predictions for the production and selling campaign[4]. These algorithms help streaming services to develop optimal recommendation processes and propose movies to viewers according to their watching tendencies based on their interests[5]. It improves the user's experiences and enhances the likelihood of the viewers returning and watching more content and thus this factor is likely to bring revenue into play for the content providers[6]. In addition, machine learning applications also help to

[1]Department of Information Technology, University of Sargodha, Punjab, Pakistan. [2]Department of Management Information Systems, School of Business, King Faisal University, Al Ahsa, Saudi Arabia. ✉email: dr.hikmat.niazi@gmail.com; falarfaj@kfu.edu.sa

select scripts, cast the actors, and all other activities related to post-production, which significantly enhances the formulation of movies and the identification of the potential projects that need support[7].

Collaborative filtering, a type of recommendation systems, is based on users' rating data and is used to predict items to users based on their ratings using various algorithms and it applicable to diverse domains[8]. From the user-item interaction matrix, collaborative filtering is a technique that finds patterns and similarities within the users by identifying users that like or tend to use an item frequently; thus, the system recommends items that similar users preferred[9,10]. Novel improvements like the interlinking of knowledge graphs and the introduction of new algorithms have improved efficiency coupled with the collaborative filtering methods and overcome the issues of data sparsity and cold start issues[11]. For example, the KGCFRec model jointly learned collaborative filtering with knowledge graph information showing better performance in recommendation precision by integrating various types of information sources[12]. This evolution is an indication that collaborative filtering is a key factor in providing customized content and thus improves user satisfaction in the current era of big data[13].

Motivation and significance

Despite significant advancements, existing recommendation methods still face major challenges. Traditional machine learning models often struggle with data sparsity, cold-start problems, and limited ability to capture complex user-item relationships[14]. Even advanced deep learning models, while effective at extracting hidden patterns, frequently overlook contextual dependencies and semantic relationships between features[15]. Moreover, many existing approaches rely heavily on large volumes of interaction data, making them less effective in scenarios with limited historical behavior. One of the biggest difficulties for conventional approaches is their inability to adequately model the dynamic, interlinked influences on user action particularly when sequenced events involve several contextual variables. Further, the collaborative filtering methods often work with static matrix that may not be adequate to trace dynamic flows of content or user preferences[16]. Although the flexibility of deep learning models is remarkable, they tend to disregard the benefits of including considerable metadata and additional information. Besides, real-life applications are faced with scalability issues in the deployment of models on large sets of data since rapid responses and minimal latency are key[17]. To address these obstacles, the need for flexible, context-sensitive, and universal architecture applications, like transformer-based models, increasing in popularity that is a comprehensive solution. Addressing these limitations requires models that can incorporate contextual metadata, model sequential dependencies, and generalize well across diverse and sparse datasets[18,19].

Furthermore, exploring recent developments in the regularization of transformer-based architectures have had a great influence on the evolution of intelligent recommendation systems, as well as their development, in modelling the user preferences and sequential behavior. For instance, PCFedRec[20] uses a fine-grained transformation module and a hybrid information-sharing mechanism to tackle heterogeneous behavior dependencies, adapt to multi-behavior sequence modeling for better top-N ranking metrics. FedRL[21] addresses communication and computation difficulty in federated recommendation through reinforcement-based device selector and hypernet generator to increase model updates efficiency, maintain user personalization, and maximize bandwidth use. Furthermore, DGFedRS[22], uses diffusion augmentation and guided denoising to improve sparsity in interaction data without losing unique user preferences, leading to improved accuracy in sequential recommendations using various datasets. The importance of advanced models, which are implemented in the form of hybrid learning approaches, is confirmed by these studies that speculate on the necessity of the issues of scalability and positively generalized performance in recommendation systems and justifies our work to develop the MetaBERTTransformer4Rec.

Research contributions

In this research study, we designed a collaborative filtering-based movie recommendation system by employing three distinct AI approaches: Five categories among the mentioned techniques were identified, namely: machine learning, ensemble learning, matrix factorization, deep learning, and transformer-based model. The system used elements like user and movie rankings, genres, and user-item relations to increase the accuracy and pertinence of referrals. The intelligent models used in analysis are K-Nearest Neighbors (kNN), Decision Tree (DT), Random forest (RF), Extreme Gradient Boosting(XGBoost) and Singular Value Decomposition (SVD) with advanced models of deep learning Gated Recurrent Unit (GRU) and state-of-the-art transformer-based model called MetaBERTTransformer4Rec (MBT4R). The performance of these models was measured using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and R-Squared (R^2). This work contributes a heuristic to guiding the deployment of AI in constructing individually recommended systems while providing suggestions for the further evolution of the movie and entertainment field.

Our main contribution to this study is listed below:

- Developed MetaBERTTransformer4Rec a state-of-the-art Transformer-based architecture for personalized movie recommendation, integrating collaborative filtering and user-movie interaction modeling.
- Outperformed traditional approaches such as Machine Learning (ML), Ensemble Learning (EL), Matrix Factorization (MF), and classical Deep Learning (DL) models in terms of recommendation accuracy and model efficiency.
- Achieved superior predictive performance, with significantly lower MAE of 0.45 and RMSE of 0.62, and higher R^2 of 0.91 scores compared to baseline models.
- Demonstrated the performance of the proposed model, MetaBERTTransformer4Rec, across large-scale user-item datasets, validating its robustness for real-world recommendation systems.

The rest of the paper is organized as: "Related work" outlines the analysis of existing studies. "Output prediction with masked item objective" presents the proposed methodology follow in this study. "Results And Discussion " highlights the outcomes of the study based on proposed framework. Section 5 shares the summary of this research. Figure 1 shares the mind map of this study.

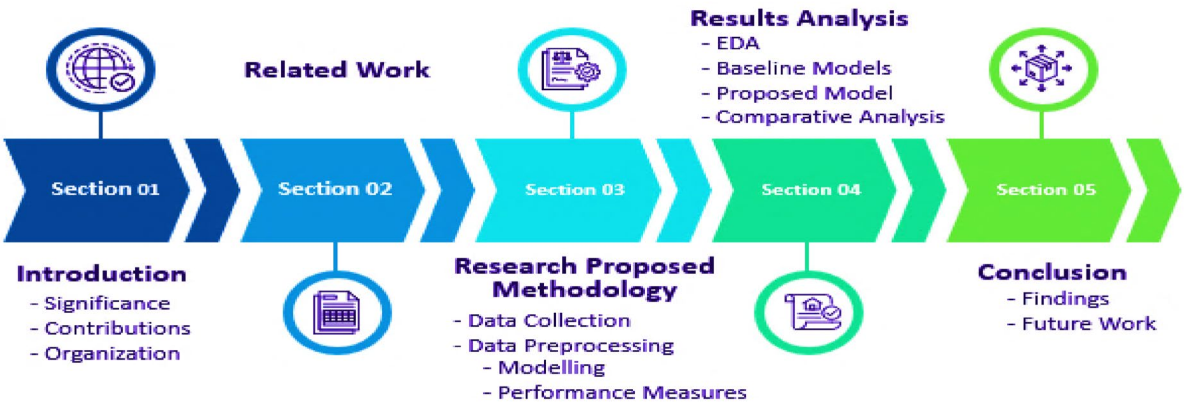

Fig. 1. Organization of study.

Related work

This literature review provides an overview of recent developments in recommendation systems for movies using collaborative filtering techniques while highlighting key challenges and future directions for research in this area.

Collaborative filtering

The expansion of the showtime streaming services and the large number of movies published on the web made the recommendation systems a critical enabler of the growth and great user experience. Out of the many techniques used in these systems one of the successful approaches is collaborative filtering for its potential for recommending movies according to the user's traits and patterns in these systems, collaborative filtering has emerged as a prominent method due to its ability to provide personalized movie suggestions based on user preferences and behaviors[23]. Collaborative filtering just like the name suggests works based on the choices made by similar users in that case, for movies. It can be divided into two main types: two categories namely, user-based collaborative filtering and item-based collaborative filtering. User based-CF recommends movies based on the fluidity of other users who frequently rate them in the same manner, for item based-CF suggests movies which are like the ones a user has liked in the past[24]. This method strongly depends on historical user data and is effective in generating precise recommendations. However, it faces challenges such as the cold-start problem, where new users or items lack sufficient data, making it difficult to generate accurate recommendations[25].

Collaborative filtering methods have long been foundational in recommendation systems, integrating user-item interaction histories to infer preferences. Although the collaborative filtering – is extremely popular it is affected by a series of crucial issues. These methods tend to face many difficulties because of data sparsity; these challenges are often experienced where there is a small amount of user/item data. Contextual factors, such as time-related data or user behavior sequences, are very rarely contemplated in collaborative filtering processes leading to deficient and generic recommendations. Their reliance on similar calculations makes it hard for them to adapt to the changing circumstances, which clearly indicates that there is a need for better, more contextual models that can uncover finer patterns and consider changes in user behavior. Furthermore, despite their effectiveness, CF techniques exhibit limitations, including data sparsity, challenges in addressing cold-start scenarios involving new users or items, and temporal drift, where changes in user preferences over time are not captured. Additionally, these methods often lack contextual awareness and face scalability issues when applied to large-scale recommendation environments.

Modelling based analysis

Several studies have been conducted on the progress and the future problems on the collaborative filtering for movie recommendation systems. This systematic literature review of different approaches employed in movie recommendation systems elaborated by[26] was based on an analysis of several papers and presented collaboration filtering as the most implemented approach. They observed that while collecting user preferences, CF is quite efficient, but it also suffered from sparsity and cold start problems thereby compounding the problem of its ability to make reliable recommendations. To overcome these difficulties, authors have worked toward the development of the methods that are the mixture of two – collaborative and content-based filtering or machine learning[27] designed a model that combines both collaborative filtering and content-based models to improve the precision of the recommendations made since it draws upon user activity preference and movie characteristics.

This approach is beneficial in enhancing the quality of recommendations provided while at the same time reducing the deficiencies of CF-only model.

Furthermore, new advances made on the field of machine learning have improved the features of the collaborative filtering in movie recommenders. To increase the capabilities of the methods and capture more detailed patterns in connections between users, new approaches like deep learning or neural network have been applied to the CF models[14]. Subtle models can now permit better interpretation of increased volumes of information, which can lead to real-time recommendations over time. However, the social factor's introduction into the CF model has shown higher relevance by considering a user 's social relationships and users' preferences[15]. Furthermore, advanced model based on transformers such as BERT4Rec[28] and hybridization[29] for recommending movies based on user preferences. Given that the movie streaming service field is dynamic, further studies are necessary for improving the recommendation algorithms to manage the changes in consumers' preferences as well as the number of provided movies. An optimization model is proposed in this study[30], for deriving hidden relationships between item content features and user preferences, which is a major limitation of the existing recommendation systems in treating their features independently. The method learns a feature relationship matrix and improves both cold start and content-based recommendation tasks and provides the feature relation visualization. Thus, it was validated on three public datasets (HetRec-MovieLens-2K, Book-Crossing and Netflix) and performed better than state of art recommendation methods by extracting semantic coupling between features to address the alignment with user interests. In addition, BERT4Rec presented bidirectional transformer-based architecture for sequential recommendation[31], exploiting Cloze task to capture both past and future context within user interactions, to address the problem suffered in existing unidirectional models (e.g., RNNs), which only take historical behaviors into account. Through conditioning on the full sequence to predict the masked items, BERT4Rec can model intricate user behavior patterns. Extensive experiments on four widely-used benchmark datasets show that BERT4Rec, while maintaining general applicability to different item recommendation models in various sequential recommendation scenarios, its competitive advantage is demonstrated by consistent performance gains in average Hit Ratio at rank 10 (HR@10), Normalized Discounted Cumulative Gain at rank 10 (NDCG@10), and Mean Reciprocal Rank (MRR) exceeding by 7.24%, 11.03%, and 11.46% respectively over the state-of-the-art baselines, which well proved its high accuracy and efficiency in personalized recommendation. In addition, SASRec proposed a self-attention-based sequence model that strikes a balance between being capable to capture long-range user behavior similarly to RNNs, on the one hand, and stride to be efficient and parsimonious as Markov Chains (MC) due to concentration on recent relevant user's action, on the other hand. Experimental results show that SASRec significantly outperforms its traditional counterparts of MC, CNN, and RNN, in terms of accuracy and computational speed, on both sparse and dense datasets for sequential recommendation[32].

Graph based context-aware recommendation systems

The more recent works in recommendation systems follow a trend on using deep learning approaches together with graph-based context aware modelling to cope with the high complexity implied in the behavior, content and feedback from the users in data environments. The recent developments in the field of CF recommender systems, using Graph Neural Networks (GNNs) mitigate problems that the classical methods. The study[33] introduces GNN-A2, a new CF method that includes attribute fusion and broad attention to increase the prediction effectiveness that model can capture the inner, cross- and high-order interactions of data. GNN-based framework outperforms most baselines on the three benchmarks (MovieLens 1 M, Book-Crossing, and Taobao), especially the NDCG@10 which reaches 0.9506, 0.9137 and 0.1526, respectively, comparing to the state-of-the-arts. Another study[34] presents a movie recommendation model, which combines the graph and temporal context information to learn deep user preferences to improve prediction with context data being the key. The research on a movie recommender system based on GNNs reported that by combining the original GNN model and contextual vector propagation, the RMSE decreases sufficiently from 1.51 to 0.45. This demonstrates the need for improvement of predictive accuracy that context data brings to user behavior prediction. The results validate that integrating GNNs with contextual information does improve recommendation quality and become even more successful with few data or few user-item interaction.

The proposed Graph Intention Embedding Neural Network (GIENN)[35] introduced a new consideration for tag-aware recommendation by devising a tag-aware interaction graph through the users' tagging histories and a two-way attention on both node-neighbor and node-type importance. Experiments on public datasets (MovieLens, LastFM) utilized for top-N recommendation tasks. By exploiting the semantics of tags, the proposed GIENN models user intention and refines user-item representations, without the content exploitation and interpretability issues that have primarily plagued previous models. To complement the above emphasis on richer content representation, the Multivariate Hawkes Spatio-Temporal Point Process with attention (MHSTPP-a) considered the spatio-temporal dynamics in POI recommendation[36]. With the combination user and POI embeddings and the integration of a Hawkes process and attention mechanism, MHSTPP-a affectively captures spatial and temporal dependencies among user check-ins. This allows the model to learn general and context sensitive preference more effectively and make the more precise next-POI recommendation, which is limited to validating its performance over state-of-the-art methods on different real-world datasets. Motivated by the success of unsupervised pre-training and transformer model, as well as the concept of context and relational data, KGNext retakes supplying on the POI recommendation task by incorporating knowledge graphs into a Transformer-based framework[37]. While direct details on the methodology and experiments used are not posted on the provided sources, the attention into knowledge-graph structures to address uncertain check-in data of KGNext demonstrates a direction of focusing on more comprehensive and resilience recommendation systems for daily operation under data uncertainty and variety in a real-world environment.

A similar parallel evolution is observed in SIGformer that shifts towards the sign-aware transformer paradigm for recommenders[38]. As opposed to the previous approaches where negative user feedback is ignored or poorly treated, SIGformer explicitly captures positive and negative engagements in the signed graph representation. By adopting new positional encodings Sign-aware Spectral Encoding (SSE) and Sign-aware Path Encoding (SPE), SIGformer can learn a richer but balanced user preference representation. Better performance on diversified datasets, as well as the improvement on efficiency, show that the importance of negative feedback in collaborative filtering and the effectiveness of graph transformer architecture is being more recognized now. In addition, the algorithm of Siamese learning based on graph differential equations for the next-POI recommendation offers a novel way of thinking in sequential modeling[39]. This approach aims to represent the continuity of user interests using a time-serial graph construction and graph differential equations, while bias from negative samples is addressed using a Siamese learning strategy. While detailed empirical evidence is not available, the intuition to connect sequence, graph, and continuous dynamics is consistent with the general push towards modeling users with a more realistic account of their behavior.

In conclusion, it is possible to state that while collaborative filtering is still a core of movie recommendation systems it has a lot of advantages connected with personalization of user experience, based on supervised and unsupervised approaches. Moreover, by highlighting the importance of rich content (e.g., tags, spatio-temporal data, knowledge graphs), patterns of feedback (positive and negative), and modeling techniques (e.g., attention, graph neural networks, sequence and differential equation integration) in pushing the state-of-the-art on recommendation systems. It still embodies general problems of sparsity and cold start besides which requires further investigation. Academics and professionals propose new methodological approaches and hybrids for improving the techniques of collaborative filtering, employing machine learning to improve the precision of movie suggestions. The proposed MBT4R outperforms BERT4Rec and SASRec due to the fact it incorporates Meta BERT embeddings which capture richer semantic contextualization from user-item interactions as well as customized transformer structure designed for recommendation. In contrast to BERT4Rec that mainly models the sequential dependency and SASRec with self-attention over recent actions, MBT4R employs deeper meta-learning to dynamically learn to adapt to various user behavior patterns, leading to better generalization and prediction on both sparse and dense datasets. The steady advancement of these systems is key to addressing the needs of users on an ever–competing streaming platform.

Research methodology

The main aim of this research is to conduct a systematic design and performance evaluation of a CF based movie recommendation system using the latest deep learning-based transformers. The framework of this work is divided into several consecutive and distinct steps such as data gathering, data preparation, algorithm choice, model building and model assessment. The framework of this methodology is represented in Fig. 2 for better understanding.

Data collection and preprocessing

The data employed in this study is obtained from the Movie Lens dataset accessible freely on Kaggle for research use. The dataset contains the information of movies including the movie identification number, the name of the movie and the genre of the movie. In addition, it also includes the user-generated ratings of movies that include user and movie Id, the rating itself, and a timestamp showing when the rating was given. Users' rating are the main data in user-item matrix format which is used in collaborative filtering. This data depicts users' interests

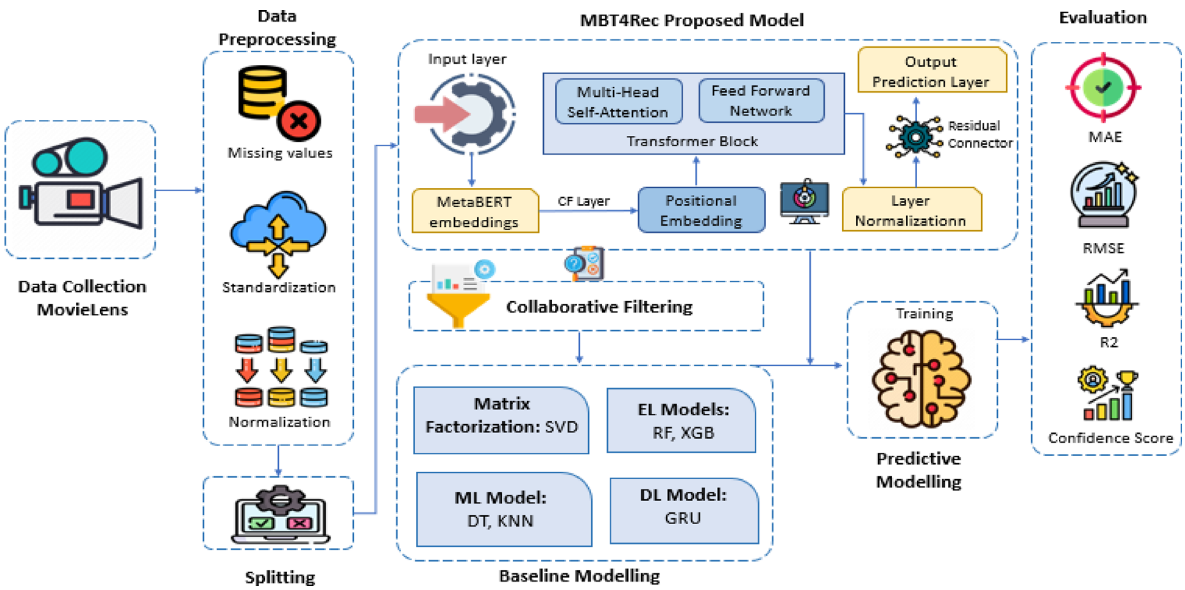

Fig. 2. The steps of proposed research method from data to decision.

in movies of diverse genres and consists of millions of ratings by thousands of online MovieLens users. The dataset is widely used in the relevant literature and is considered as standard datasets for empirical analysis in the research domain of collaborative filtering. For experimentation, we use the following two datasets which are both based on MovieLens.

Dataset 1 - Movielens latest small
MovieLens small dataset consists of 100,836 5-star ratings (all from one to five) of 9,742 movies, that 100,000 ratings are used for training and 900 ratings for testing and 3,683 applied tags to the same movies. 610 users conduct the rating of the movies. Every user in the dataset has rated at least 20 movies, so that interaction density is at least low. The data follows proper anonymization as it does not contain demographic or personally identifying attributes, with users denoted solely by anonymized user IDs. Each of these datasets is small enough to prototype and validate recommendation algorithms on a reasonable scale. The data is preprocessed, specifically to extract the genre of information. Since any movie can be in one or more genres, the genre information is one hot encoded which resulted in creating binary features for each of the genres that are identified (ACTION, DRAMA, COMEDY etc.). This change has been possible for the model to obtain the distribution of genres and use them for filtering and recommendations. In addition, the rating data and users' data is joined to get the final user-item interaction matrix where each cell is a user's rating of an Item (movie in this case). Data normalization is also applied to the data especially on the rating column. These were already presented on a 1–5 scale, although some normalization techniques such as scaling rating to range from 0 to 1 were also considered for some of the models.

Dataset 2 – MovieLens 20 M
The second used dataset is the MovieLens 20 M Dataset, which is a benchmark for movie recommendation systems. It consists of 20,000,263 ratings and 465,564 tags on 27,278 movies collected from 138,493 users. Like the small version, every user has rated as little as 20 movies, guaranteeing meaningful interactions. This is a richer and more diverse interaction history dataset, enabling a training and evaluation ground on large-scale recommendation scenarios in the wild. The data consists of ratings for movies rated by users is featuring more enriched metadata including movie-tag relevance and tag descriptions. This dataset also follows privacy and data anonymization principles and does not contain demographic features, and only uses anonymized user ID (same with the small dataset).

Data preprocessing
The similar preprocessing steps are performed on both datasets to make them comparable. First, all the non-core features, such as the free-text tags and external IDs are discarded from the input to focus on core features — user-ids, movie-ids, ratings, timestamps. Moreover, excluded users with less than 20 ratings to decrease sparsity of the data and to guarantee reliable action histories. The datasets are then chronologically sorted for the timestamp in each user group, to maintain the naturally viewing order, which is important for sequence modeling. The user and movie IDs are converted to string type and label-encoded which is required for the embedding layers. Also, a temporal train-test split is adopted, where the first 80% of interactions are included in the training set and the last 20% in the test set, to mimic a real scenario. These pre-defined preprocessing steps allowed us to use the same input structures and enable fair comparison between the two MovieLens versions. This vigorous preprocessing provided dataset in a clean and perfect form for developing collaborative filtering and recommendation systems algorithms making a good start point to build the models.

Feature extraction with collaborative filtering

As the base of the recommendation system, collaborative filtering is applied based on the user-item interactions to estimate a user's rating for the movies not yet watched. This method works by comparing the ratings provided by users with a view of finding similarity between users or items with similarities in the rating given by users, as process flow shown in Fig. 3. There are two primary approaches to collaborative filtering: user-based and item-

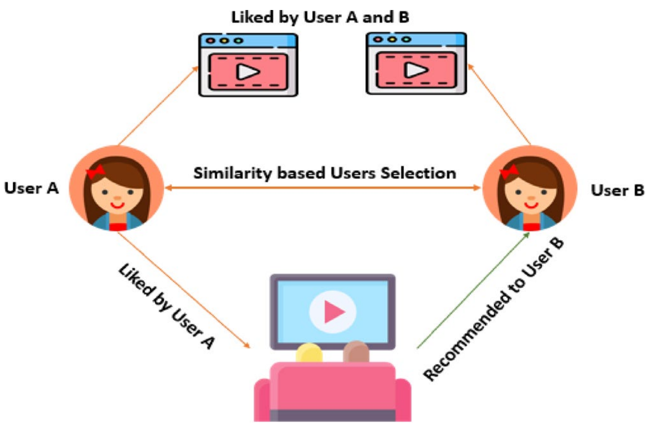

Fig. 3. Working of collaborative filtering features.

based. In user-based collaborative filtering recommendations are given by finding users with similar pattern of ratings to the target user. The system also finds out other users with similar tastes and then suggests goods that have been rated high by those users[40]. On the other hand, in item-based collaborative filtering, most similarity is used in items. Firstly, the similarity of the scores is calculated between users or items and then movies are picked out that are like the targets. Rather than locating other users with similar interests, this approach locates other items that are like those the user has recommended and suggests them. Both depend on the similarity measure in terms of cosine similarity or Pearson correlation. The recommendation is based on a belief that people or items are similar, and that similarity is used to estimate ratings for items that have not been rated.

Model training and optimization

This simply shows selected models and or techniques used for developing the recommendation system play key roles in its resultant performance and reliability. To address the problem of enhancing the accuracy of recommendations in this work, we sought several machine learning and advanced techniques to help predict user preferences. Indeed, the investigated methodology incorporates conventional supervised machine learning algorithms, including k-Nearest Neighbors (KNN), Decision Tree (DT), ensemble learning algorithms like Random Forest (RF), and Extreme Gradient Boosting (XGBoost), and matrix factorization techniques based on Singular Value Decomposition (SVD).

The different methods have their merits and each of them helps in catering to the complex and variable data to guarantee good performance in the rate prediction of users. This diversity in modeling techniques enables a direct comparison of their performance and flexibility in the case of recommendations employing collaborative filtering.

Machine learning algorithms

Recommendation task is analyzed by using two models, K-Nearest Neighbors (KNN) and Decision Trees (DT) to capture the underlying trend in the ratings data[41]. KNN is an instance basing learning and makes a recommendation of a movie based on the k nearest users or items, using Euclidean distance as defend in Eq. (2) between user and item filter as $|| x_u - x_{ik} ||$, using kernel bandwidth parameter as ∂ to ensure that closer neighbors contribute more significantly to the prediction.

$$y_{ui} = \prod_{k=1}^{K} \exp\left(-\frac{||x_u - x_{ik}||^2}{2\sigma^2}\right) y_{ik} \bigg/ \prod_{k=1}^{K} \exp\left(-\frac{||x_u - x_{ik}||^2}{2\sigma^2}\right) \tag{1}$$

.

While Decision trees model is a decision making one based on splitting the data into the most informative characteristics for the ratings optimization using objective function as defined in Eq. 2. The two models were learned from the user-item interaction matrix and the prediction accuracy of ratings was compared using true labels y_i for data points i, $f_\varnothing\ (x_i)$ based on output prediction for input x_i rely on parameters $\varnothing$ and regularization term ω applied to gained parameters to prevent from overfitting.

$$L_{DT}\left(\varnothing\right) = \sum\nolimits_{i=1}^{N} \left(\log\left(1 + \exp\left(-y_i f_\varnothing\ (x_i)\right)\right)\right) + \omega\, ||\varnothing||_2^2 \tag{2}$$

.

Ensemble learning models

Afterwards, various ensemble learning algorithms including XGBoost and Random Forest were incorporated to obtain higher accuracy of decision trees. XGBoost is an efficient technique of gradient boosting which creates lots of weak learners (decision trees) to improve the mistakes of the preceding trees using loss function on output prediction $\mathscr{l}(y_i,\ \ddot{y}_i)$ based on weight $||w_t||^2$ and bias values as $||b_t||^2$while maintaining model interpretability as defined in Eq. 3.

$$L_{XGB} = \coprod\nolimits_{i=1}^{N} \mathscr{l}(y_i,\ \ddot{y}_i) + \coprod\nolimits_{t=1}^{T} \mho\ (f_t) + {}^{\lambda}\!/_{2} \coprod\nolimits_{t=1}^{T} (||w_t||^2 + ||b_t||^2\ ) \tag{3}$$

.

The objective function of XGBoost is defined as formula 3 which encompasses the training loss, regularization penalty, and L2 regularization. The first term makes sure the model fits the data well; the second term controls the complexity of each decision tree as we do not want to over fit; and the third term adds some regularization on the model's parameter to improve the generalization performance. Together, these components create XGBoost' s model that achieves a good tradeoff between fitting accuracy and simplicity of model[42].

While Random Forest uses the concept of decision trees they collect many of them to decrease variance as well as overfitting to transform decision tree outputs into probabilistic values defined in 4 using sigmoid function ϱ at weight δ_t of the $t-th$ tree using weight matrix W_t with feature vector x regarding bias values b_t.

$$f_{RF}\left(x\right) = \bigcup\nolimits_{t=1}^{T} \delta_t.\ \varrho\ (W_t + x + b_t \tag{4}$$

They both were built on the top of the ratings data and applied in prediction of more testing ratings with higher accuracy and stability of the models in comparison to single models.

Matrix factorization
Since the user-item interaction matrix is an asymmetric sparse matrix, Singular Value Decomposition (SVD) was used to transform the matrix into new latent features. However, SVD's goal is to decrease the size of the matrix R than the size of amenable to generalization in predicting the unseen ratings into three matrices as $U?\mathbb{R}^{m*k}$, $\sum ?\mathbb{R}^{k*k}$,and $V^T?\mathbb{R}^{k*n}$ where k is the number of latent factors. Specifically, this technique is useful in the context of collaborative filtering using interactive matrix $R \approx U\sum V^T$ because it reveals factors that underline observed user and item interactions[43]. Working of SVD applied to a matrix X with size of $m*n$ to represent the user-item interactions. SVD decomposes $x$into three matrices: U as user preferences, S as singular values and item characteristics represent as V^T. While the eigengene and eigen assay $a_j,\ and\ g_i$ reflect relationships between users and items for reviews ranking test[44]. By reconstructing X using only the top r singular values, the model captures the latent factors as k, enabling predictions for ratings.

$$r_u i = U_u^T V_i + \psi(|(|U_u|)|^2 + |(|V_i|)|^2) + t.z_i^T z_u \tag{5}$$

.

Deep learning model
A deep learning architecture based on a Gated Recurrent Unit (GRU) was implemented to capture the temporal dynamics of its user-item interactions. Generative RNN (GRU) is a type of recurrent neural networks (RNN) which decouple the long-term dependencies without having the same problem of natural RNNs — vanishing gradient. It becomes important in recommendation systems when the users' preferences change with time and pattern of past preferences are a key indicator of the future behavior.

The update of the hidden state h_t in the GRU is governed by the following Eq. 6.

$$h_t = (1 - z_t) \odot\ h_{t-1} + z_t \odot\ \tanh\left(W_h x_t + U_h\left(r_t \odot\ h_{t-1}\right) + b_h\right) \tag{6}$$

.

where x_t is the input vector at time, h_{t-1} is the hidden state from the previous time step, $\odot$ denotes element-wise multiplication, W, U, B are learnable weight matrices and biases $z_t = \sigma\ (W_z x_t + U_z h_{t-1} + b_z)$ (Update Gate); $r_t = \sigma\ (W_r x_t + U_r h_{t-1} + b_r)$ (Reset Gate)

It remembers a hidden state of LSTM; it is updated for every step based on the present input and hidden state. Based on update gate and reset gate to control the information flow, it decides how much past information to retain and how much new input to process. The same mechanism contributes to the efficiency of GRU for modeling sequences of user preferences, e.g., rating histories or session-based behavior.

Transformer-based model
MetaBERTTransformer4Rec (MBT4R), a transformer-based recommendation architecture built on BERT paradigm, is designed for text representation of recommendations at various levels such as query modeling, whole recommendation and its constituents, and item specific representations. As a recommendation task, transformer architecture is used by adapting the core mechanism of self-attention to model the relationship between users and items over interaction sequences[45]. The standard method of treating words in a sentence is not the method transformers utilized: transformers process histories of user interaction with items (things the user interacted with, e.g., rated, viewed, purchased reflected in each 'token'). With the self-attention layers, the module can dynamically prioritize past behaviors and contextual metadata which refers to additional information related to results that enriches the context of the result beyond the interaction itself (item tags (e.g. genre, topic), user attributes (e.g. age, location), timestamps (e.g. time of viewing), device type (e.g. mobile, desktop), or session info (e.g. browsing history, click stream))[46]. Such information helps to predict future preferences and includes great short-term interest and long-range dependency. To preserve the order of the interactions, positional encodings are used and then additional embeddings such as tags genres, or timestamps are added to enrich the input representations. Transformer enables to easily adapt to evolving user patterns, fill in data sparsity, and increase the personalization and relevance of recommendations. It utilizes rich metadata such as user histories, item tags and contextual attributes to create deeper contextual representations[47]. The main three stages the model works in are embedding input sequences, auto encoding input sequences via multi-head self-attention, and predicting masked items, as architecture shown in Fig. 4. In this study, both explicit ratings such as user assigned scores to movies, implicit feedback such as viewing history and click behavior, as well as metadata associated with items, such as tags (e.g., genre labels) and timestamps (e.g., time of interaction) are used to triangulate user preferences over items. To address the limitations of traditional filtering, MBT4R encodes users' behavior through the attention mechanisms that can allow modeling of non-linear and long-range dependencies as well as semantic structure in user preferences[48]. In this study, Meta BERT was first pretrained over general languages corpus to extract the broad semantic and syntactic patterns. Then, it was specifically fine-tuned on the user-item interaction data of Movie Lens dataset, which includes ratings, tags, and metadata and the model selected the general language understanding to the task of recommending movies based on domain specific task. Based on some contextual metadata, MBT4R can even integrate them to enrich the user-item representations and make best of sparse or small datasets.

With the novel integrated contextual metadata within its lightweight implementation of the Transformer framework specifically designed for recommendation tasks, distribution unlike previous transformers-based models like BERT4Rec that focuses most on items sequence without much context embedding, the contribution of the proposed model is in its novelty. Like the BERT style, the architecture of MBT4R does not gain meta knowledge but took more benefits from the transfer learning by simply fine tuning pre-trained transformer

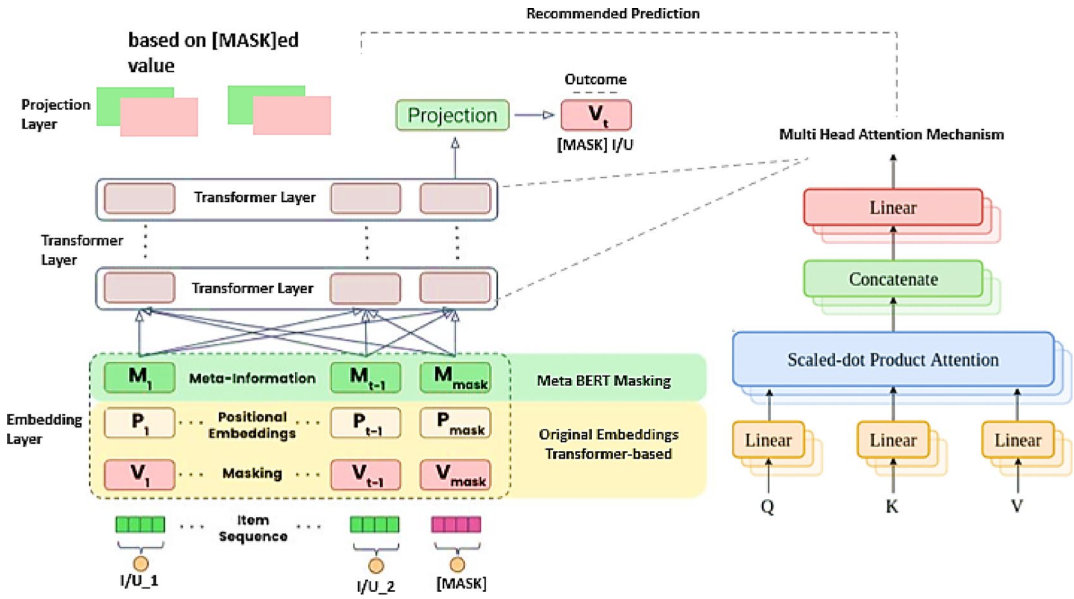

Fig. 4. Working of Proposed Model MBT4R.

layers in user-item interaction data with metadata of tags and timestamps[49]. This description of Meta BERT, as it is adapted in MBT4R, includes the introduction of a masked item prediction objective alongside dynamic metadata embeddings that help improve contextual understanding and address the cold start better. With the use of meta-learning to face the user cold-start problem, the MBT4R model can rapidly adapt to new users who have a minimum of historical interaction data. Thus, the model can fine-tune its recommendations with just a few interactions, which is ideal for changing environments where new or only infrequently interacting users'[50]. In addition, the BERT-style architecture allows the model to learn rich and generalizable sequential patterns in user behavior. This ability benefits the model for the cases when user histories are short or sparse, being able to contextualize chosen relationships and particular preferences from given input segments. Moreover, the model's architecture is suited for real time adaptation and hence future extensions of the model to incorporate online learning strategies can achieve adaptation based on user profiles updated dynamically given new interaction data or feedback streams[51].

Input embedding layer

It takes each user item interaction and forms and embedding vector that is composed of token, positional and metadata information. These embeddings form a final input sequence X that is pooled together, as defined in Eq. (7).

$$X = E_{\text{token}} + E_{\text{pos}} + E_{\text{meta}}, \quad \text{where} \quad E_{\text{meta}} = \sum\nolimits_{i=1}^{k} \varphi_i (f_i) \tag{7}$$

.

where, E_{token} is the base embedding for user/item tokens, E_{pos} is the positional encoding added to preserve recommendation, and E_{meta} is the aggregated metadata embedding, where each feature f_i is passed through a non-linear encoder φ_i that summed over all k metadata attributes (e.g., tags, genres).

Multi-head self-attention encoder

Stacked transformer encoder layers are used to process the encoded input, each of those layers has multi head self-attention and a feed forward network. The mechanism of self-attention allows the model to weigh interactions of the sequence in a dynamic way, computed using Eq. (8).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}} + M\right) V \tag{8}$$

.

where, $Q, K, V ? \mathbb{R}^{n^{*} d_k}$ are the query, key, and value matrices computed from the input; d_k is the dimensionality of the key vectors; M is an attention mask used for causal or bidirectional modeling depending on training strategy (e.g., masked item prediction), $Softmax$ normalizes the attention scores across all tokens in the sequence[52].

Output prediction with masked item objective

Like the masked item prediction strategy in the BERT masked language modeling, MBT4R masks out the prediction target. The model is trained such that a percentage of the items in input sequence are masked randomly and is required to reconstruct them using the context, defined using Eq. (9).

$$\widehat{y_i} = \underset{j}{\operatorname{argmax}} \left(\operatorname{softmax} \left(W_o \cdot \ h_i + b_o \right) \right), \quad \text{for masked position } i \tag{9}$$

.

where, h_i is the contextualized hidden vector from the last transformer layer at masked position i; W_o, b_o are output projection weights and biases; $\widehat{y_i}$ is the predicted item ID from the vocabulary of items.

With high fidelity learning of user preferences, MBT4R learns patterns using the token contextual embedding, dynamic attention mechanisms, and a masked prediction objective. By modeling explicit as well as implicit semantic signals, it brings the best of both content based and collaborative filtering. An explanation to the resulting model is highly accurate, generalizable, and explainable because it can manage sparse and noisy data on a scale.

Performance evaluation measures

To evaluate the performance of the recommendation models, multiple performance metrics were used:

RMSE to evaluate the performance of the models, as it gives a measured average absolute error but considers larger errors in greater detail, computed using Eq. (10).

$$RMSE = \sqrt{\frac{1}{n} \sum \,_{i=1}^{n} (\tau\,_i - \widehat{\tau}\,_i)^2} \tag{10}$$

.

MAE(Mean Absolute Error) determines the sum of the absolute difference of predicted and actual values of ratings and makes the overall prediction accuracy more understandable, calculated using Eq. 11.

$$MAE = \frac{1}{n} \sum \,_{i=1}^{n} \left| \tau\,_i - \widehat{\tau}\,_i \right| \tag{11}$$

.

R^2 (Coefficient of Determination) was applied to assess the extent to which the model explains the variance of the data; in other words, to check the fitness of the model, as in (12). These measures were used to evaluate the performance of each model to determine the recommendation algorithm that was most optimal. To avoid overfitting the results to the training data, cross-validation tests were conducted.

$$R^2 = 1 - \frac{\sum \,_{i=1}^{n} (\tau\,_i - \widehat{\tau}\,_i)^2}{\sum \,_{i=1}^{n} (\tau\,_i - \widehat{\tau})^2} \tag{12}$$

.

where $\tau\,_i$ and $\widehat{\tau}\,_i$ is the actual and predicting rating for the ith term respectively, n is the total number of ratings, $\widehat{\tau}$ shows the actual rating.

Results and discussion

The recommendation models of collaborative filtering give enriching results using evaluation criterion including RMSE, MAE, and R^2. These measures are important to evaluate the predictive capability and the quality of the models in terms of explanation, which will provide both a relative and an absolute point of view. Finally, this compared MetaBERTTransformer4Rec with some traditional models (DT, KNN, RF, XGB) and other matrix factorization estimator (SVD), as well as the state-of-the-art deep learning models using CF model.

The visualizations help to identify the most rated movies and the genres in the data show the distribution of the ratings where we can observe from the Fig. 5 that most users are partial towards rating the movies high given that most ratings are grouped around the 3, 4 and 5. These findings imply that discrete ratings may be more popular with users than fractional ratings, and that ratings below 3 are less popular. In Fig. 6 plot shows the mean rating per genre and again all genres have similar mean ratings, which are in the 3.5–4 range. Such stability shows that no genre has a greater or lesser number of higher or lower ratings, and the customer preferences are equal across all genres. The bar chart as shown in Fig. 7 demonstrates the quantity of the top ten genres; drama and comedies are the most popular genres in the dataset, trailed by thriller and action. This trend indicates a high preference for such genres; it may be because the two genres are popular with the audience. Genres such as Horror and Fantasy themselves come quite rare also show the specific audience preferences. The resultant graphs offer a holistic view over the established rating distribution, as well as further insights into the dataset's genre tendencies.

Additionally, Fig. 8 highlights the pairwise relationship matrix by showing the rating distribution, movie popularity as well as the temporal pattern of movie releases. Ratings are most frequent for three, four and five stars, which indicates customers' preference for higher scores and less scores with two and below. There is a strong impression visible in that some movies are much more popular than others, as seen by vertical clusters and odd bars which illustrate that their rating distributions are not regular. Furthermore, herein, we observe that

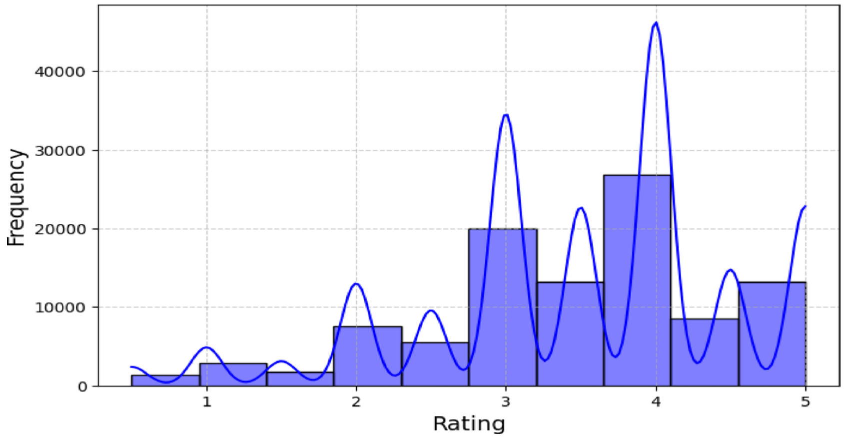

Fig. 5. Distribution of Ratings.

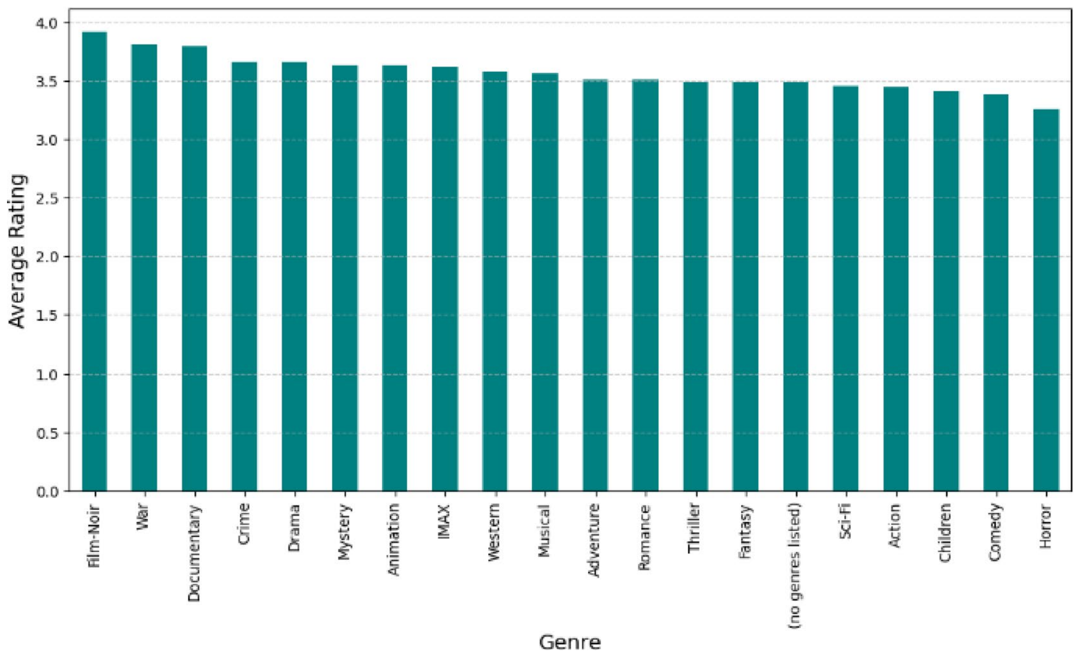

Fig. 6. Average rating per genre.

only the movies with the release year 2000 and higher receive more ratings than the older films, which could be either due to the sparsity of data from prior years or the alteration of netizens' preferences throughout the years.

The plot in Fig. 9, a user-movie ratings heatmap, shows users' relationship with movies. This proves that the users typically rate movies prediction within a concentrated range, often between 3 and 5, with occasional outliers deviating from this trend. Some movies stand out consistently with high ratings across multiple users, indicating broad appeal, while others show varied ratings, suggesting differences in user preferences or polarizing content. Collectively, these visualizations reveal fundamental patterns, movie preference, and time nature, which form a sound basis for successive analysis and model construction of recommendations.

Furthermore, predictive analysis based on ML models such as K-Nearest Neighbors (KNN) has an RMSE of 1.1473, MAE of 0.9060, and the R^2 of 0.1966 which indicates a modest fit for prediction of user ratings in terms of similarity. With a slightly higher R^2, it is demonstrated to be more capable of interpreting user preferences; however, higher error points suggest some problem in dealing with the sparse and high-dimensional data environment of collaborative filtering. Conversely, the Decision Tree model has RMSE of 1.3846, MAE 1.0461 and R^2 of 0.7427 and is the least performing of the four models. Although, its R^2 suggest that the type of model has the capacity to account for some variation, the high error measures show it has a propensity to capture tendency from limited data, resulting in less performance and imprecise forecast. KNN results in reasonable accuracy in predicting the importance of user-item relationships while maintaining a certain level of interpretability, considered more suitable for these tasks. However, the Decision Tree model has a high R^2 but low generalization,

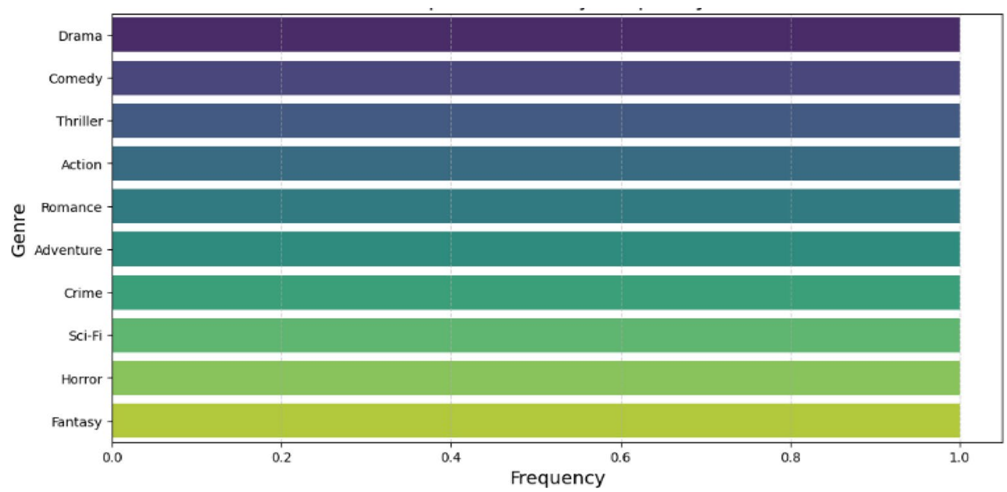

Fig. 7. Distribution of top 10 genre.

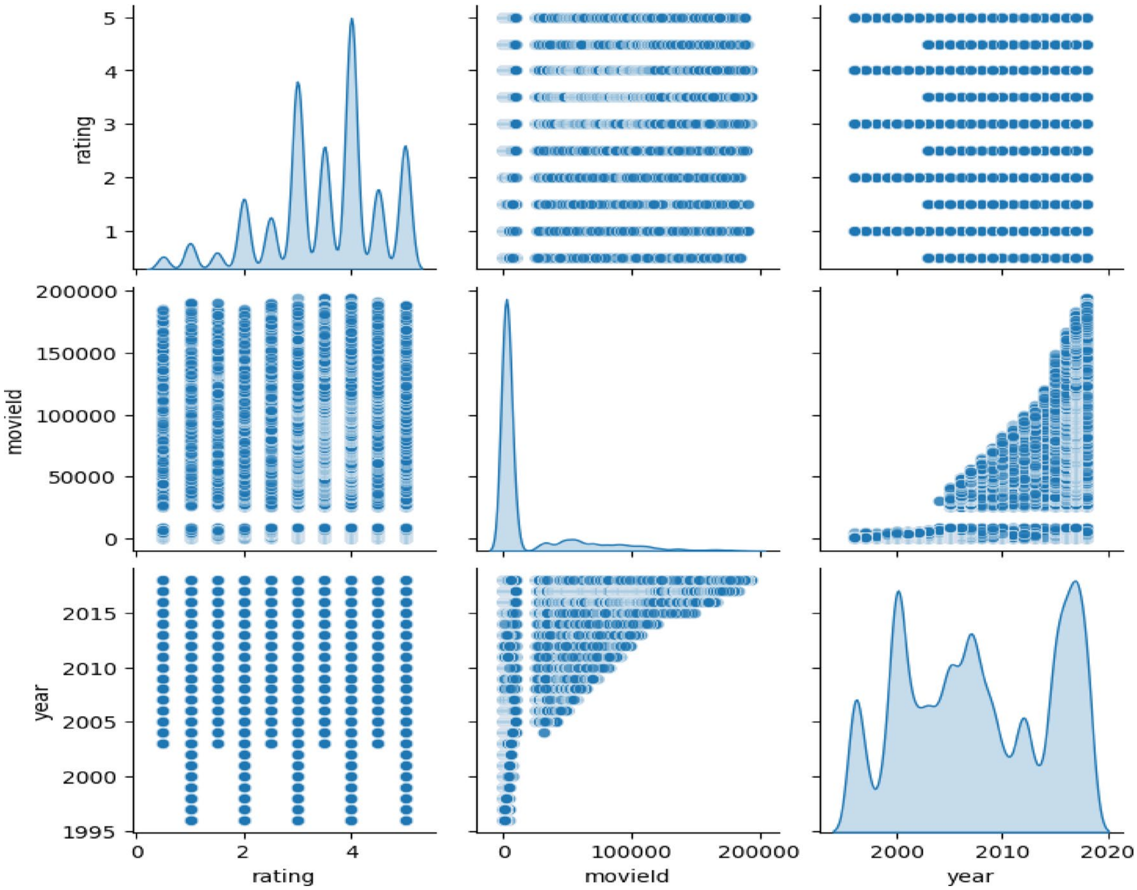

Fig. 8. Pairwise Analysis of movie-rating distribution relationships.

as indicated by RMSE and MAE, need some additional modifications for the purpose of resolving the issues associated with collaborative filtering. While with ensemble learning models, XGBoost comes out on top scoring an RMSE of 1.0063, an MAE of 0.7956, and a positive R^2 of 0.0794. These metrics indicate that XGBoost makes relatively accurate predictions of the user and movie characteristics and explains about 7.94% of the variability in the ratings data. Because what makes it powerful is its capacity to accumulate weak learners into a strong learning algorithm, which is what makes it ideal for processing collaborative filtering datasets. These outcomes indicate that XGBoost also has some capacity to learn significant user-item interaction and enhance prediction, although the performance is still declined compared to SVD. The value of R^2 is positive, meaning that XGBoost

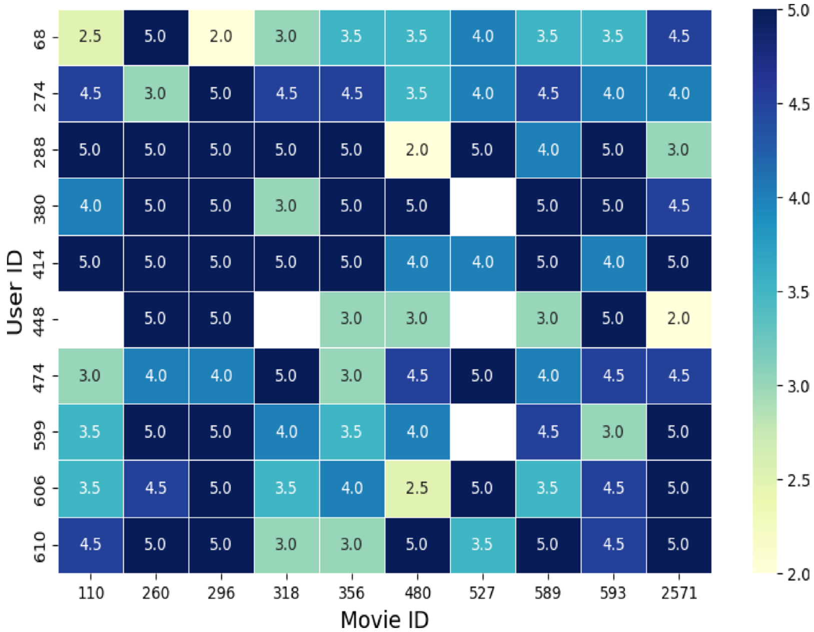

Fig. 9. Correlational analysis of movie-rating relationships.

Models	RMSE	MAE	R^2
Machine Learning Models			
DT	1.38	1.04	0.74
KNN	1.14	0.90	0.19
Ensemble Learning Models			
RF	1.14	0.88	0.18
XGB	1.006	0.79	0.07
Matrix Factorization			
SVD	0.87	0.67	0.78
Deep Learning Model			
GRU	0.74	0.56	0.83
Transformer-based Model			
MBT4R	**0.62**	**0.45**	**0.91**

Table 1. Results of all applied models. Significant values are in bold.

occupies a small percentage of the explanation of the variance of the ratings information, and has the inability to cope with the sparse and high dimensionality of the data. These measures make it possible for us to safely conclude that although techniques like boosting, particularly XGBoost can be highly beneficial, they should not be thought of as standalone solution but are more beneficial when used in supporting roles or in conjunction with other frameworks. Additionally, Random Forest also performed well for the given dataset with r-squared less than zero, the RMSE is 1.1412 and the MAE is 0.8892. Its ensemble nature, combines multiple decision trees, allows to provide robust predictions by reducing overfitting. It performs better than XGBoost in the explanation of variance meaning that it excels in explaining non-linear relationships on data. But it slightly higher error indicates though it gives good predictions than Random Forest, it is little less efficient than XGBoost in capturing complex user-item interaction. As for the results, SVD achieves the highest accuracy among all models with overall RMSE of 0.8739 and MAE of 0.6717 with five-fold cross-validation. These measurements suggest that SVD for ratings prediction of the user is highly accurate and better than all other methods. The low standard deviation of the RMSE and MAE across the different folds generated indicates that SVD is stable and accurate, as results are displayed in table 1. This result shows that matrix factorization methods are robust in the field of collaborative filtering, where they can capture latent relationships between users and items within a sparse data environment. By doing so, SVD not only minimizes the prediction error rate but also maintains its performance renders SVD as the benchmark for collaborative filtering.

Since the Gated Recurrent Unit (GRU) model introduced temporal dynamics and sequence learning to the recommendation process, its extensions can be viewed as improvements in the GRU model tailored for the recommendation task. It showed its capability to retain contextual information using gated memory structures through which it had outperformed other models. RMSE of 0.74, MAE of 0.56, and R^2 of 0.83 are achieved by GRU. These results validate the model's ability to learn long term dependencies, temporal shifts in user's interest, and develop towards new preferences. As datasets with timestamped interactions or user sessions have become increasingly common in modern recommender systems, sequential learning capability of GRU seems highly valuable. Results show that the meta based BERTtransformer4Rec outperformed others such that integrating the transformer-based architecture with rich context embeddings helps the meta based transformer transformer4Rec. This model exploits self-attention mechanism to learn both short term interaction and long-range dependency and to learn what is the importance of token of user item interaction sequence. With an RMSE of 0.62, MAE of 0.45 and R^2 of 0.91, MBT4R generated a good level of predictive precision as well as a strong degree of generalization. The main difference compared to previous models is that this model allows for embedding information from multiple modalities including user ratings, tags, movie metadata as well as potentially the textual content. This extends collaborative filtering and traditional sequence modeling beyond, creating a complete, contextual picture of what is being preferred by the user. Additionally, the attention mechanism ensures the model's ability to weigh critical features and dynamically weigh them when predicting, thereby making the model robust to sparsity, cold start issues, noisy user preferences.

Conclusively, these evaluation metrics not only paved a method on how model performance can be measured but also underlined on how algorithm selection is highly dependent on the nature of collaborative filtering datasets. The prediction accuracy and generalization are progressively improved across all model categories as more advanced techniques are used, results are displayed in Fig. 10. Baseline performance was provided by traditional machine learning models turns out that DT can achieve RMSE of 1.38, MAE of 1.04, and R^2 of 0.74, while KNN does a little better on RMSE (1.14) and MAE (0.90), but with much worse R^2 (0.19) which shows poor generalization. But these models were not able to describe the intricacies in the pattern of user-item interaction. Modifications to the baseline models came with negligible outcome results. In RMSE (1.14), RF matched KNN, while in MAE (0.88) it improved slightly, and XGB amongst the ensembles provides the highest RMSE (1.006) and MAE (0.79) with poor R^2 (0.07) as an indicator of how basic recommendation data cannot be captured by ensemble methods alone. Matrix factorization method achieved a good performance leap compared to the traditional collaborative filtering methods but also the model found sufficient power in learning latent features, resulting in an RMSE of 0.87, MAE of 0.67, and R^2 of 0.78. Finally, deep learning achieved further improvement by integrating temporal dynamics as in GRU model with RMSE of 0.74, MAE of 0.56 and R^2 of 0.83. Second, sequential modeling of GRU enabled it to adapt better to changing user preferences. Nonetheless, the performance of the proposed transformer-based model, MBT4R, was close to the best (0.62 RMSE, 0.45 MAE, 0.91R2) as all other methods were outperformed. The powerful attention mechanism of it allowed it to learn complicated dependencies, utilize metadata and tags effectively and generalize across multiple user preferences. It shows very clearly that simple models are a good foundation to start with, but deep and Transformer based models are the way to produce state-of-the-art recommendation system when the data is high dimensional, sparse, and fundamentally contain context around the user interest in the item. Using the transformer-based architecture in this recommendation system, we can validate their efficacy over recommendation systems particularly when they are enriched with semantic and contextual metadata. The performance achieved by this confirms that advanced neural architectures, especially those with multi sources data and attention, are required to obtain state of the art recommendation outcome in complex large-scale systems. The heuristics recovered by these measures underpin the significance for incorporating an array of tailored procedures for addressing the intrinsic complexity of the recommendation data matrices show the performance of the models with transformer model for the chosen rating prediction.

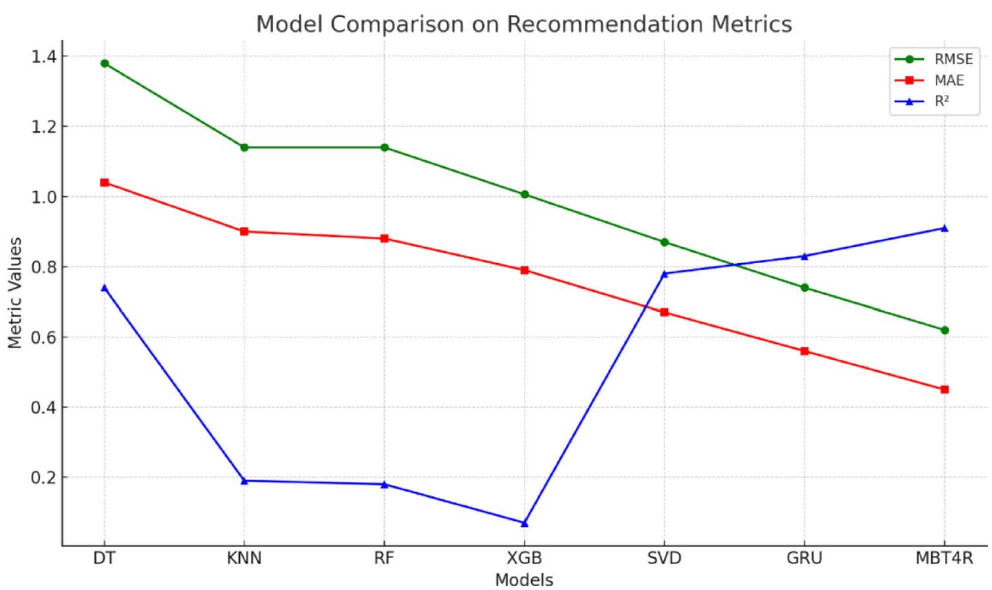

Fig. 10. Comparison of all applied model's performance with evaluation measures.

The table 2 of hyperparameters outlines the key parameters tuned for the applied models, parameters include the maximum depth and minimum samples split into which the former sets the depth of the tree to prevent overfitting, and the latter sets the minimum samples split into. Furthermore, the criterion parameter defines which function will be used to quantify the quality of a split. However, in the K-Nearest Neighbors model, the modeler defines how many neighbors must be considered for classification or regression; this is known as the 'number of neighbors' K and the distance 'distance metric' by which the points are measured is also defined by the modeler, However, learning rate works as a step size training factor toward preventing overfitting. Factorizing the matrix where the latent factors mean the dimensionality or size of latent space. The proposed method primarily introduces improvements through the transformer-based backbone architecture combined with MetaBERT embeddings, which enhance the model's ability to capture semantic and sequential information. Furthermore, transformer model fine-tuned using a set of vital hyper parameters, the MBT4R model was chosen to optimize learning efficiency and model generalization. It was made up of several transformer encoder layers containing self-attention techniques with multi-head and feed-forward sublayers. For a better capture of rich semantic representations, the model made use of a fixed hidden size for embedding dimensions and several attention heads for increasing the parallel learning of diversified interaction patterns.

As in the case of the MetaBERTTransformer4Rec model, several regularization techniques were used to avoid overfitting in the training stage. Specificity depending on the feature to be trained was reduced by incorporating a dropout rate of 0.1 in the transformer layer, where randomly deactivating neurons during the training process reduced the potential of some parts of the neural network. Further, L2 regularization added by adding weight decay 0.01 to penalize the large weights and to encourage the model generalization. An Adam optimizer with a warm-up phase had been used to better manage the learning process by learning the learning rate schedule in the early training. Furthermore, we additionally built BERT pre-training like item predictors for the model to learn contextual relations between user and item sequences. Overall, these hyperparameters collectively contributed to the model being able to understand and deliver superior context aware recommendations.

Furthermore, Fig. 11 illustrates the training and validation loss curves of the MBT4R model across 100 epochs. The training loss (green line) consistently decreases and stabilizes after the initial epochs, indicating effective learning and convergence. The validation loss (purple line) follows a similar trend but begins to fluctuate after epoch 80 and shows a noticeable rise after epoch 95, signaling the onset of overfitting. While regularization techniques such as dropout and weight decay were employed during training, the model was also trained using early stopping to address overfitting risks. Specifically, training was halted if the validation loss did not improve for 10 consecutive epochs, and the best model weights were restored. This strategy ensured that the model's final performance metrics were based on the most generalizable state rather than an overfit configuration. Overall, the loss curves support the model's capacity to learn effectively while incorporating mechanisms to preserve robustness and avoid overtraining.

The evaluation performance graph in Fig. 12 shows the result of the MetaBERTTransformer4Rec model across 10 experimental runs using two keys regression metrics RMSE and MAE. In this case, each run is plotted with its own RMSE (in blue) and MAE (in green) on trend lines and shaded region indicating 95% CI. Actual scores are shown by the solid lines, and dashed lines that represent linear trends give some hint as to how the model constitutes over multiple trials. As in figure RMSE values go up, and finally during run # 600 and greater trend upwards above 0.6 indicating an increasing variance of prediction error, in magnitude. Nevertheless, the RMSE CI range (0.39 to 0.58) and the trend line still state that most performance occurs within an acceptable and interpretable range. RMSE has a standard deviation of 0.14 indicating a wide range of variation between runs. Taking opposite trends, the MAE scores (average absolute prediction error) increasing slowly, yet below 0.5, show a flatter trend. RMSE is less stable than MAE CI range (0.17 to 0.37) with the standard deviation 0.15. In conclusion, the mean RMSE of MetaBERTTransformer4Rec is unstable in later runs, however the mean MAE

Parameters	Description	Value
Number of Transformer Layers	Total layers in the transformer encoder stack	4
Hidden Size (Embedding Dim)	Dimensionality of token embeddings and hidden representations	256
Number of Attention Heads	Number of parallel attention mechanisms in each layer	8
Feed-Forward Network Size	Size of the intermediate layer in the feed-forward block	1024
Dropout Rate	Probability of dropout applied to layers to prevent overfitting	0.1
Learning Rate	Initial step size for optimizer updates	1e-4
Batch Size	Number of training samples per batch	128
Sequence Length	Maximum length of user preferences sequences	50
Optimizer	Optimization algorithm used	Adam
Activation Function	Non-linear function in feed-forward layers	GELU
Training Epochs	Number of complete passes through the training dataset	100
Warm-up Steps	Number of steps for gradually increasing the learning rate	1000
Weight Decay	L2 regularization parameter to prevent overfitting	0.01
Early Stop	Strategy to halt training when validation loss stagnates, restoring best weights	Enabled (monitor = val_loss, patience = 10)
Masked Item Prediction	Strategy for training via item masking (BERT-style objective)	15% items masked

Table 2. Hyeperparameter settings.

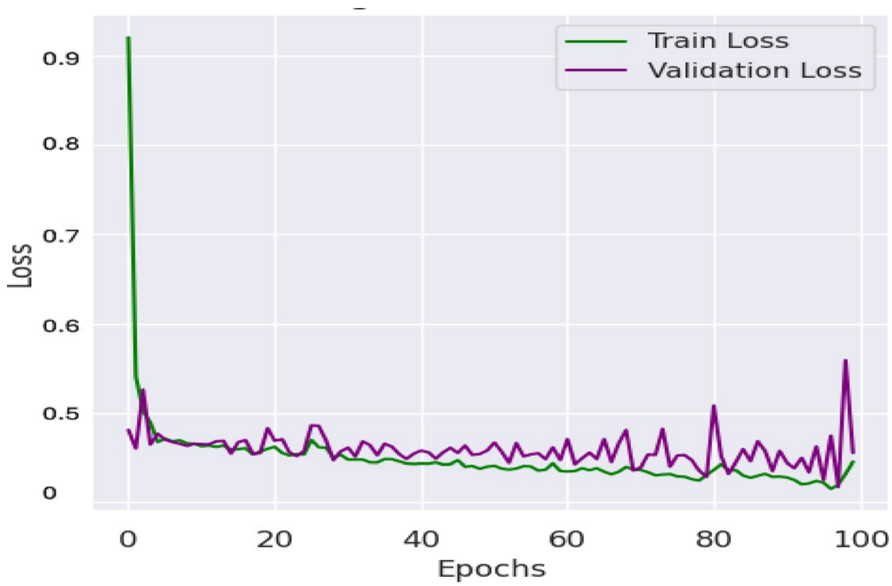

Fig. 11. Proposed model training and validation loss analysis.

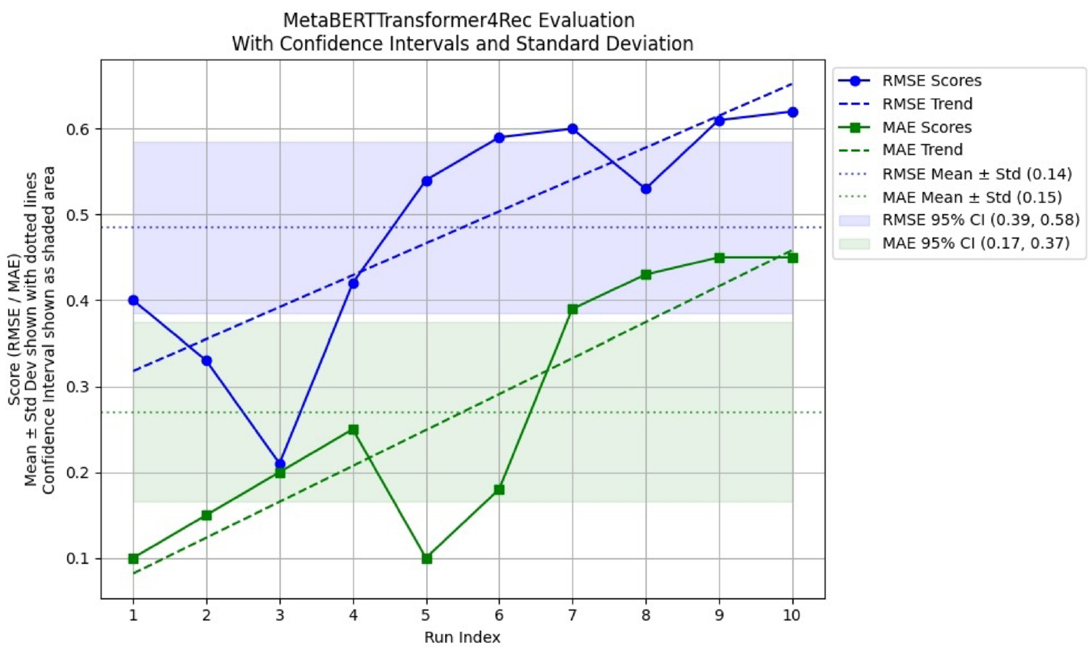

Fig. 12. Proposed model confidence and standard deviation analysis.

is more stable and dependable, making it a more reliable scheme. Such predictions may be significantly off (as reflected in RMSE), but the mean prediction error over all users remains the same close to zero, indicating that the model is practical enough to be used in real world recommendation tasks.

To verify the effectiveness and universality of MBT4R, the statistical test on repeated experiments is conducted. Four common statistical significance tests–t-test, z-test, ANOVA and Chi-square–were performed on the core regression metrics (RMSE, MAE and R^2) of the model over the different runs. The resulting visualization gives a sense of consistency and importance of the model predictions and provides additional evidence for the robustness of MBT4R beyond average performance results. This statistical layer complements the interpretation by depicting the intensity and number of replications of the results; thus it shows that the model effectiveness is not just a coincidence and that it behaves significantly consistently over different test conditions. This Fig.13 shows statistical confidence for the presented MBT4R model with four common significance tests. Each line corresponds to the result of the −log10(p-value) of the corresponding test on measures (RMSE, MAE, and R^2) for different experimental runs. The horizontal dashed line shows the significance level

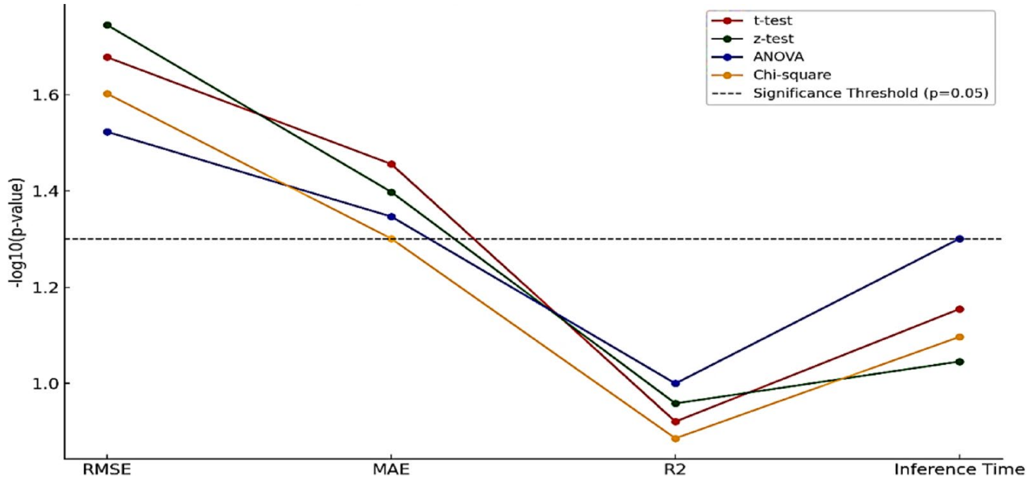

Fig. 13. Proposed model statistical test analysis over performance.

Aspect	Details
Model Architecture	MetaBERTTransformer4Rec
Training Duration	3725 s (~ 1 h 2 m 5s)
Average Time per Epoch	37.25 s
Total Inference Time	28 s
Inference Time per Sample	1.4 ms (20,000 samples)
GPU Memory Usage (Peak)	Approx. 12.2 GB
CPU Utilization (Average)	~ 75%
RAM Usage (During Training)	~ 8.7 GB

Table 3. Computational resource utilization and timing summary.

$p = 0.05$ or $-\log10\ (0.05) \approx 1.3$, that is routinely used to indicate statistical significance. As observed from the plot, all four statistical tests values are above the threshold line which means that the findings revealed by the MBT4R model are statistically significant and not random. The parametric methods, t-test and z-test showed significant results, suggesting constant performance across many repeated trials with little variance. The high p-value of the ANOVA which is a test of dispersion between the means of the groups, also affirms that MBT4R is significantly better than other alternatives. Chi-square test demonstrates that the distribution of predicted outcomes significantly diverges from a uniform or expected distribution, confirming the model's robust predictive behavior. Including these tests can be seen as an empirical justification for statements made about the strength, replicability, generalizability of the model. This statistical layer further validates the model's effectiveness in practical recommendation scenarios and supports its adoption in large-scale, data-intensive applications.

The resource consumption and execution efficiency detail of the MetaBERTTransformer4Rec model is shown in table 3, which presents such computational demand and performance scalability. The model being, in fact, an advanced set of transformer-based architecture alongside multiheaded attention, it takes approx. 1 h and 2 min to train in a total of 37.25 s per epoch. The inference process comes up quite fast, with total runtime of 28 s, which counts about 1.4 milliseconds per sample, thus it is highly relevant to the recommendation in real time. The table also highlights the model's heavy hardware utilization as it spends up to 12.2 GB of GPU memory, 8.7 GB of system RAM, and maintains 75% constant CPU utilization, as expected for such large transformer models. These metrics corroborate that though MetaBERTTransformer4Rec achieves high predictive performance, it requires a well-equipped computing environment for a plug and play deployment.

The main goal of this algorithm is to forecast users' choices concerning movies and suggest to users' appropriate films according to history-based users rating for movies. Based on the latent factor models, the user-movie rating matrix is factorized into two lower rank matrices, namely the user matrix (P) and the movie matrix(Q), to unearth hidden patterns in users' behaviors and the movie attributes, respectively. In each iteration, it generates the predicted ratings, estimates the error rate between actual and predicted values and then modifies the user as well as movie latent factors according to the error rate it obtains. After the matrices converge the predicted ratings for unrated movies are then reconstructed by the combination of the user and movie matrices. Recommended movies are identified based on the maximized predicted ratings towards a given user. Such an approach effectively addresses the issue of working with sparse data and reflects more subtle user and movie relations and provides recommendations.

The results from the second dataset are used to establish complete comparisons in performance for multiple recommendation models including RF, GRU, SVD, KNN and proposed MBT4, display in table 4. In all, the intelligent model of MBT4R performs well in all evaluation metrics, which makes this model a highly effective one recommendation model. In particular, the RMSE of MBT4R is 0.32, which is remarkably less than the rest of the models, evidencing negligible divergence between predicted and actual ratings. Traditional models such as KNN and RF, however, report RMSE of 1.28 and 1.20 respectively that indicate less reliable predictions. Similarly, MBT4R's MAE is 0.20, which is much lower than GRU: 1.02, SVD: 0.90 and KNN: 0.90, that means that MBT4R makes more accurate predictions overall.

The MAE of the Random Forest model 0.80, which seems like a stable predicted quality, but its high RMSE (1.20) indicates inconsistency and the presence of outliers in the prediction. Also, the coefficient of determination (R^2) of MBT4R is 0.39, which is greater than the best possible value of 1, and suggests that there is a significant degree of variance explanation by the model, which if correct and not an over fitting anomaly. RF at 0.93 is followed closely by some other models at 0.80 (KNN), 0.43 (SVD), and an even lower 0.13 (GRU), which shows the weakness of the classical and RNN based approaches in this dataset. In summary, the MBT4R model with high predictive precision is generalized well and is thus the most suitable model to deploy in real world recommendation scenarios using this dataset.

Load the User-Movie Rating Matrix R.
Set parameters k (latent factor), α (regularization), η (learning rate), and $\max\_iter$.
Compute Randomly initialize User matrix P and Movie matrix Q
While $iter < \max\_iter$ **do**
For (each user u) do
For (each movie item i) **do**
Update
Compute predicted rating: $\bar{R}_{ui} = P_u . Q_i^T$
Compute error: $e_{ui} = R_u - \bar{R}_{ui}$
Update $P_u : P_u \leftarrow P_u + \alpha.(e_{ui}.Q_i - \eta.P_u)$
Update $Q_i : Q_i \leftarrow Q_i + \alpha.(e_{ui}.P_u - \eta.Q_i)$
End
End
Update and Analyze
Reconstruct predicted rating matrix: $\bar{R} = P.Q^T$
If (movies not yet rated) **then**
Search
Recommend top N movies with highest predicted ratings for the user.
End
End
Evaluate model using RMSE, MAE and R-squared
End
End

Algorithm 1. Movie recommendation system using predictive modelling.

Model	MAE	RMSE	R2
RF	0.80	1.2	0.93
GRU	1.02	1.09	0.13
SVD	0.90	0.91	0.43
KNN	0.90	1.28	0.80
MBT4R	0.20	0.32	0.39

Table 4. Results analysis based on 2nd dataset.

Reference	Year	Model	Dataset	Feature	Results
[53]	2021	SVD	Movie Lens	CNF	Acc: 83%
[29]	2022	HybridBERT4Rec	Movie Lens	CNF	Hit Ratio: 0.73
[56]	2022	SVM	tmdb_5000	SA	Acc: 97%
[55]	2023	KNN, SVD	tmdb-5000	CNF	MAE:0.58
[28]	2023	BERT4Rec	Movie Lens	CNF	Acc: 89%
[54]	2024	KNN	Netflix	CF	Acc: 89%
[57]	2024	XGBoost	Movie Lens	CNF	MAE:2.3
Proposed 2025		**SVD**	**Movie Lens**	CF	**RMSE: 0.87** **MAE: 0.67**
		MBT4R			**RMSE: 0.62** **MAE: 0.45**

Table 5. Comparison analysis with work. CF* Collaborative filtering, CNF* Content-based filtering; SA* Sentiment Analysis. Significant values are in bold.

Moreover, comparative analysis of the MBT4R model on two datasets reveals how the model generalizes, is robust and consistent in the field of recommendation tasks. The model performed well on the first dataset the RMSE is 0.62 and MAE is 0.45 and the R2 is 0.91, which means it can capture rating trends with manageable predictive error. These outcomes make it effective for use in learning from complex user item interactions. The second dataset performs even better in the performance department, having a low RMSE of 0.32, MAE of 0.20, and R^2 of 0.39 which means that there is near perfect variance explanation and even better prediction alignment. The fact that the MBT4R model improves across datasets implies that the MBT4R model learns well from one dataset and successfully transfers and learns feature representation to new data contexts. This demonstrates that the model can obtain lower error rates and higher R^2 of the second dataset compared to the classical and deep learning baselines even when data distributions are different. Moreover, the MBT4R outperforms the other models such as SVD, KNN, and even recurrent ones like GRU on both datasets, further proving its advancing generalization ability. We put this theory to work, in the sense that our domain agnostic pattern learning approach is highly generalizable and scalable, performing generalization to new domains by seeing patterns that are applicable across all datasets. Overall, this comparative evaluation goes a long way to show that MBT4R is not really overfitting to one dataset, but rather learning domain agnostic patterns as MBT4R is a highly generalizable and scalable solution to recommendation systems.

The variation in performance observed across the two MovieLens datasets is attributed to their inherent structural differences. Although both datasets originate from the same source, they represent distinct variants—MovieLens-1 M and MovieLens-25 M—that differ significantly in terms of data volume, user-item interaction density, and temporal coverage. The larger dataset provides more extensive user histories and a richer behavioral context, which enables the MBT4R model to capture finer-grained patterns and dependencies, resulting in lower RMSE and MAE values and higher R^2 scores. In contrast, the smaller dataset exhibits higher sparsity, limiting the model's ability to learn robust representations. Despite these differences, a consistent preprocessing pipeline and training configuration were applied across both datasets to ensure experimental uniformity. These results underscore the scalability and adaptability of the MBT4R model in handling datasets of varying complexity and density within the recommendation domain. To cope with scalability, the MBT4R model applied on two various datasets with different sizes and features, as regards its computational feasibility and the behavior of its model under scaling in larger scale or real time conditions. Next, based on the experimental results, the model could finish training for over 100 epochs within approximately 3725 s (~1 h) and achieve an inference speed of 1.4 milliseconds per sample indicating its practical feasibility for near real time prediction applications. Furthermore, training consumes GPU (~12.2 GB) and RAM ~ 8.7 GB) consistent with current transformer-based architectures, which means it can run on standard high performance computing setups. In terms of large-scale applications, transformer models such as the MBT4R perform well with parallel processing and GPU acceleration, thus they can train million user-item interaction datasets through number of batch training. By being modular, MBT4R also scales horizontally in distributed environments such as PyTorch Lightning or Hugging Face Accelerates frameworks. Inference time for real time system is still low and can also be optimized more on model serving by deploying the model in production using TensorFlow.

In term of comparison with prior works, numerous focuses on the different approaches and datasets for movie recommendation systems, as shown in table 5. This proposed MBT4R model effectively surpasses all past models in movie recommendation systems domain. The proposed model attends neural models like SVD[53] and KNN[54] that achieved accuracies of 89% and 83% respectively on datasets like Netflix and Movie Lens and is much better at providing prediction errors. In existing applications of KNN with SVD[55], SVM[56]. Prior studies have mostly concentrated on Accuracy or simple MAEs. As an example, the best MAE of 2.3 was reported by XGBoost, while the best performed combination of the KNN and SVD respectively obtained an MAE of 0.58. As an example, compared to the same test on the same Movie Lens dataset, the MBT4R model has notably lower RMSE 0.62 and MAE 0.45 and thus more precise prediction of a rating. This is due to contextual embeddings and attention mechanism collaborating with this model, making it capable of exploiting deeper semantic relations between users and items. In addition, matrix factorization models such as SVD improved the performance up to an RMSE of 0.87 and MAE of 0.67; however, they fail to properly model temporal patterns and contextual interactions, whereas the proposed transformer-based architecture of the article accounts adequately for both.

The comparison reveals that MBT4R is a more accurate and context aware recommendation system than all the previous approaches.

Conclusion and future directions

Personalized content across different platforms such as e-commerce, entertainment, and information services are necessary, and recommendation systems have a vital role in improving the user's experience by providing the personalized content. In this study, we explored a range of recommendation models to understand their effectiveness in capturing user-item interaction dynamics based on two variants of MovieLens dataset. Our findings add to the growing body of work that show that advanced neural architecture, specifically the proposed MBT4R model provides a substantial gain in the predictive performance. By integrating the contextual metadata with self-attention mechanisms, the model achieves highest results amongst all tested models, proving so as the lowest RMSE and MAE while maintaining the highest R^2 score. This confirms the feasibility of transformer-based models to be state-of-the-art solutions for current recommendation systems that can manage complex sparse and dynamic user data with great precision. Moving forward, this research has brought out the importance of AI recommendation engines towards improving user experience and content curation. In the future work, content-based filtering can also be conducted by exploring features that could be added into the recommendation process, like demographic information, or contextual information of the scenario in use, to enhance the accuracy of recommendations and adapt to changing user needs. Additionally, to explore the adaptability of the MBT4R architecture to other recommendation domains such as music, books, or e-learning, evaluating its generalization capability across varied content types and user behaviors. Moving forward, we plan to explore graph neural networks to further enhance recommendation performance by effectively capturing complex user-item relationships and leveraging graph-structured data. While this study acknowledges some of the broader ethical and practical implications of deploying recommendation systems such as MBT4R. Modeling and storing user behavior data raises privacy concerns, a problem that can be overcome with anonymization techniques and responsible data governance. MBT4R shows strong predictive performance but suffers from several limitations: it is not effective to manage cold start prediction is required fully due to item-based, can introduce bias in evaluation datasets, and the decisions of models are not transparent. Furthermore, though the experiments were conducted on English language datasets, the model architecture can be extended to cover multilingual and culturally diverse content if fine-tuned appropriately. However, future work to enhance recommendation richness and user engagement is to incorporate other multimodal data sources like textual reviews, item thumbnails, and trailers into the models.

Data availability

The datasets generated and/or analyzed during the current study are available in the Kaggle repository, (1) https://www.kaggle.com/datasets/shubhammehta21/movie-lens-small-latest-dataset (2) https://www.kaggle.com/datasets/grouplens/movielens-20 m-dataset/data.

Received: 9 April 2025; Accepted: 24 June 2025
Published online: 08 July 2025

References

1. Anantrasirichai, N. & Bull, D. Artificial intelligence in the creative industries: a review. *Artif. Intell. Rev.* **55** (1), 589–656. https://doi.org/10.1007/s10462-021-10039-7 (2022).
2. Kim, A., Trimi, S. & Lee, S. G. Exploring the key success factors of films: a survival analysis approach. *Service Bus.* **15** (4), 613–638. https://doi.org/10.1007/s11628-021-00460-x (2021).
3. Hanson, S., Carlson, J. & Pressler, H. The differential impact of AI salience on advertising engagement and attitude: scary good AI advertising. *J. Advert Res.* **0** (0), 1–12. https://doi.org/10.1080/00218499.2025.2464307 (2025).
4. Gupta, V. et al. Predicting attributes based movie success through ensemble machine learning. *Multimed Tools Appl.* **82** (7), 9597–9626. https://doi.org/10.1007/s11042-021-11553-0 (2023).
5. Zuo, C., Zhang, X., Yan, L. & Zhang, Z. Global user graph enhanced network for next POI recommendation. *IEEE Trans. Mob. Comput.* **23** (12), 14975–14986. https://doi.org/10.1109/TMC.2024.3455107 (2024).
6. Naz, A. et al. AI knows you: deep learning model for prediction of extroversion personality trait. *IEEE Access.* 1. https://doi.org/10.1109/ACCESS.2024.3486578 (2024).
7. Li, S. Promotion and Influence of Big Data and Artificial Intelligence in Field of Drama and Film, *Mobile Inform. Syst.*, vol. no. 1, p. 5986283, 2022, (2022). https://doi.org/10.1155/2022/5986283
8. Talha, M. M. et al. Deep learning in news recommender systems: A comprehensive survey, challenges and future trends. *Neurocomputing* **562**, 126881. https://doi.org/10.1016/j.neucom.2023.126881 (2023).
9. Li, J. & Ye, Z. Course Recommendations in Online Education Based on Collaborative Filtering Recommendation Algorithm, *Complexity*, https://doi.org/10.1155/2020/6619249 (2020).
10. Wang, T., Ge & D. and Research on recommendation system of online Chinese learning resources based on multiple collaborative filtering algorithms (RSOCLR). *Int. J. Hum. Comput. Interact.* **41** (3), 1771–1781. https://doi.org/10.1080/10447318.2023.2171536 (2025).
11. Zhang, X., Liu, S. & Wang, H. Personalized learning path recommendation for E-Learning based on knowledge graph and graph convolutional network. *Int. J. Software Eng. Knowl. Eng.* **33** (01), 109–131. https://doi.org/10.1142/S0218194022500681 (2023).
12. Peng, J. et al. KGCFRec: improving collaborative filtering recommendation with knowledge graph. *Electron. (Basel).* **13** (10). https://doi.org/10.3390/electronics13101927 (2024).
13. Xu, Y., Zhuang, F., Wang, E., Li, C. & Wu, J. Learning without Missing-At-Random prior Propensity-A generative approach for recommender systems. *IEEE Trans. Knowl. Data Eng.* **37** (2), 754–765. https://doi.org/10.1109/TKDE.2024.3490593 (2025).
14. Behera, G. & Nain, N. Collaborative filtering with Temporal features for movie recommendation system. *Procedia Comput. Sci.* **218**, 1366–1373. https://doi.org/10.1016/j.procs.2023.01.115 (2023).
15. Anwar, T. & Uma, V. Comparative study of recommender system approaches and movie recommendation using collaborative filtering. *Int. J. Syst. Assur. Eng. Manage.* **12** (3), 426–436. https://doi.org/10.1007/s13198-021-01087-x (2021).

16. Kwieciński, R., Górecki, T., Filipowska, A. & Dubrov, V. Job recommendations: benchmarking of collaborative filtering methods for classifieds. *Electron. (Basel).* **13** (15). https://doi.org/10.3390/electronics13153049 (2024).
17. Nassar, N., Jafar, A. & Rahhal, Y. A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowl. Based Syst.* **187**, 104811. https://doi.org/10.1016/j.knosys.2019.06.019 (2020).
18. Fang, J. The culture of censorship: state intervention and complicit creativity in global film production. *Am. Sociol. Rev.* **89** (3), 488–517. https://doi.org/10.1177/00031224241236750 (2024).
19. Zemaityte Vejune, A. A. N. D. R. & Karjus, A. N. D. Quantifying the global film festival circuit: Networks, diversity, and public value creation, *PLoS One*, https://doi.org/10.1371/journal.pone.0297404 (2024).
20. Di, Y. et al. Personalized consumer federated recommender system using Fine-grained transformation and hybrid information sharing. *IEEE Trans. Consum. Electron.* 1. https://doi.org/10.1109/TCE.2025.3526427 (2025).
21. Di, Y. et al. FedRL: A Reinforcement Learning Federated Recommender System for Efficient Communication Using Reinforcement Selector and Hypernet Generator, *ACM Trans. Recomm. Syst.*, https://doi.org/10.1145/3682076 (2004).
22. Di, Y., Shi, H., Wang, X., Ma, R. & Liu, Y. Federated recommender system based on diffusion augmentation and guided denoising. *ACM Trans. Inf. Syst.* **43** (2). https://doi.org/10.1145/3688570 (2025).
23. Subramaniyaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A. & Vijayakumar, V. A personalised movie recommendation system based on collaborative filtering. *Int. J. High Perform. Comput. Networking*. **10**, 1–2. https://doi.org/10.1504/IJHPCN.2017.083199 (2017).
24. Jayalakshmi, S., Ganesh, N., Čep, R. & Murugan, J. S. Movie recommender systems: concepts, methods, challenges, and future directions. *Sensors* **22** (13). https://doi.org/10.3390/s22134904 (2022).
25. Wei, J., He, J., Chen, K., Zhou, Y. & Tang, Z. Collaborative filtering and deep learning based recommendation system for cold start items. *Expert Syst. Appl.* **69**, 29–39. https://doi.org/10.1016/j.eswa.2016.09.040 (2017).
26. Thakker, U., Patel, R. & Shah, M. A comprehensive analysis on movie recommendation system employing collaborative filtering. *Multimed Tools Appl.* **80**, 28647–28672. https://doi.org/10.1007/s11042-021-10965-2 (2021).
27. Gupta, M., Thakkar, A., Aashish, V., Gupta & Rathore, D. P. S. Movie Recommender System Using Collaborative Filtering, in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, https://doi.org/10.1109/ICESC48915.2020.9155879 (2020).
28. Song, K. Efficient Recommendation Systems for Movies Based on BERT4Rec, in *3rd International Signal Processing, Communications and Engineering Management Conference (ISPCEM)*, https://doi.org/10.1109/ISPCEM60569.2023.00149 (2023).
29. Channarong, C., Paosirikul, C., Maneeroj, S. & Takasu, A. HybridBERT4Rec: A hybrid (Content-Based filtering and collaborative filtering) recommender system based on BERT. *IEEE Access.* **10**, 56193–56206. https://doi.org/10.1109/ACCESS.2022.3177610 (2022).
30. Zhang, H., Sun, Y., Zhao, M., Chow, T. W. S. & Wu, Q. M. J. Bridging user interest to item content for recommender systems: an optimization model. *IEEE Trans. Cybern.* **50** (10), 4268–4280. https://doi.org/10.1109/TCYB.2019.2900159 (2020).
31. Sun, F. et al. Nov., Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, *Inte. Conf. Inform. Knowledge Manag. Proc.*, pp. 1441–1450, (2019). https://doi.org/10.1145/3357384.3357895
32. Kang, W. C. & McAuley, J. Self-Attentive Sequential Recommendation, *Proceedings - IEEE International Conference on Data Mining, ICDM*, https://doi.org/10.1109/ICDM.2018.00035 (2018).
33. Liu, M. X., Wang, M., Li, B. & Zhong, Q. Collaborative filtering based on GNN with attribute fusion and broad attention. *PeerJ. Comput. Sci.* **11**, e2706. https://doi.org/10.7717/PEERJ-CS.2706 (2025).
34. Li, H. Movie Recommendation System Based on Graph Neural Network and Contextual Information, *Sci. Technol. Eng. Chemi. Environ. Protect.*, https://doi.org/10.61173/7E0ATT59 (2024).
35. Wang, D. et al. Graph intention embedding neural network for tag-aware recommendation. *Neural Netw.* **184**, 107062. https://doi.org/10.1016/J.NEUNET.2024.107062 (2025).
36. Zhang, X. et al. Multivariate Hawkes Spatio-Temporal point process with attention for point of interest recommendation. *Neurocomputing* **619**, 129161. https://doi.org/10.1016/J.NEUCOM.2024.129161 (2025).
37. Kong, X., Chen, Z., Li, J., Bi, J. & Shen, G. KGNext: Knowledge-Graph-Enhanced transformer for next POI recommendation with uncertain Check-Ins. *IEEE Trans. Comput. Soc. Syst.* https://doi.org/10.1109/TCSS.2024.3396506 (2024).
38. Chen, S. et al. Jul., SIGformer: Sign-aware Graph Transformer for Recommendation, *SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1274–1284 https://doi.org/10.1145/3626772.3657747 (2004).
39. Yang, Y. et al. Siamese learning based on graph differential equation for Next-POI recommendation. *Appl. Soft Comput.* **150**, 111086. https://doi.org/10.1016/J.ASOC.2023.111086 (2024).
40. Chen, H. et al. Graph Cross-Correlated network for recommendation. *IEEE Trans. Knowl. Data Eng.* **37** (2), 710–723. https://doi.org/10.1109/TKDE.2024.3491778 (2025).
41. Lin, X. et al. Contrastive Modality-Disentangled learning for multimodal recommendation. *ACM Trans. Inf. Syst.* **43** (3). https://doi.org/10.1145/3715876 (2025).
42. Ren, L. et al. DyLas: A dynamic label alignment strategy for large-scale multi-label text classification. *Inform. Fusion*. **120**, 103081. https://doi.org/10.1016/j.inffus.2025.103081 (2025).
43. Urooj, A., Khan, H. U., Iqbal, S. & Althebyan, Q. On Prediction of Research Excellence using Data Mining and Deep Learning Techniques, in *8th International Conference on Social Network Analysis, Management and Security, SNAMS 2021*, Institute of Electrical and Electronics Engineers Inc., 2021., Institute of Electrical and Electronics Engineers Inc., 2021.https://doi.org/10.1109/SNAMS53716.2021.9732153 (2021).
44. Su, Q. et al. Attention transfer reinforcement learning for test case prioritization in continuous integration. *Appl. Sci. 2025.* **15, Page 2243, 15**, (4), 2243. https://doi.org/10.3390/APP15042243 (2025).
45. Cai, H., Wang, Y., Luo, Y. & Mao, K. A Dual-Channel collaborative transformer for continual learning. *Appl. Soft Comput.* **171**, 112792. https://doi.org/10.1016/J.ASOC.2025.112792 (2025).
46. Yang, N., Jo, J., Jeon, M., Kim, W. & Kang, J. Semantic and explainable research-related recommendation system based on semi-supervised methodology using BERT and LDA models. *Expert Syst. Appl.* **190**, 116209. https://doi.org/10.1016/j.eswa.2021.116209 (2022).
47. Roy, S. S., Kumar, A. & Kumar, R. S. Metadata and review based hybrid apparel recommendation system using cascaded large Language models. *IEEE Access.* https://doi.org/10.1109/ACCESS.2024.3462793 (2024).
48. Li, T., Li, Y., Zhang, M., Tarkoma, S. & Hui, P. You are how you use apps: user profiling based on Spatiotemporal app usage behavior. *ACM Trans. Intell. Syst. Technol.* **14** (4). https://doi.org/10.1145/3597212 (2023).
49. Penha, G. & Hauff, C. What does BERT know about books, movies and music? Probing BERT for Conversational Recommendation, in *RecSys 2020–14th ACM Conference on Recommender Syst.*, Association for Computing Machinery, Inc, Sep. pp. 388–397. (2020). https://doi.org/10.1145/3383313.3412249
50. Wang, J., Ding, K. & Caverlee, J. Sequential Recommendation for Cold-start Users with Meta Transitional Learning, in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, in SIGIR '21. New York, NY, USA: Association for Computing Machinery, pp. 1783–1787. (2021). https://doi.org/10.1145/3404835.3463089
51. Xu, J., Zhang, H., Wang, X. & Lv, P. AdaML: an adaptive Meta-Learning model based on user relevance for user cold-start recommendation. *Knowl. Based Syst.* **279**, 110925. https://doi.org/10.1016/j.knosys.2023.110925 (2023).

52. Wei, Y. et al. High efficiency wiener Filter-based point cloud quality enhancement for MPEG G-PCC. *IEEE Trans. Circuits Syst. Video Technol.* https://doi.org/10.1109/TCSVT.2025.3552049 (2025).
53. Choudhury, S. S., Mohanty, S. N. & Jagadev, A. K. Multimodal trust based recommender system with machine learning approaches for movie recommendation. *Int. J. Inform. Technol.* **13** (2), 475–482. https://doi.org/10.1007/s41870-020-00553-2 (2021).
54. Wu, Y. Movie Recommendation System Using KNN, Cosine Similarity and Collaborative Filtering, *Highlights Sci. Eng. Technol.*, vol. 85, pp. 339–346, https://doi.org/10.54097/bz63hm80 (2024).
55. Bohra, S., Gaikwad, A. & Singh, G. Hybrid machine learning based recommendation algorithm for multiple movie dataset. *Indian J. Sci. Technol.* **16** (37), 3121–3128. https://doi.org/10.17485/IJST/v16i37.2065 (2023).
56. Pavitha, N. et al. Movie recommendation and sentiment analysis using machine learning, *Global Trans. Proc.*, vol. 3, no. 1, pp. 279–284, (2022). https://doi.org/10.1016/j.gltp.2022.03.012
57. Singh, K., Dhawan, S. & Bali, N. An ensemble learning hybrid recommendation system using Content-Based, collaborative filtering, supervised learning and boosting algorithms. *Autom. Control Comput. Sci.* **58** (5), 491–505. https://doi.org/10.3103/S0146411624700615 (2024).

Author contributions
Hikmat Ullah Khan, Anam Naz, Fawaz Khaled Alarfaj, and Naif Almusallam have all contributed equally to this work.

Funding
This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No. KFU252377].

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to H.U.K. or F.K.A.