J. Electrical Systems 19-4 (2023): 159-177

[1]Guozhen Sang

[1] Qin Xie

Construction of Personalized Movie Recommendation Model Relying on Recurrent Neural Network

Journal of Electrical Systems

Abstract: - Recommendation system and classification machine learning techniques have emerged as a major area of investigation into the issue of information overload. Software tools called "recommender systems" aim to help users make informed choices about services. This paper developed a low-dimensional embedding in which a limited number of features can be used to represent the data object. For the purpose of acquiring the low rank latent factors, matrix factorization techniques have received a lot of attention. The proposed model is stated as the Collaborative Filtering Machine Learning (CFML) for the multi-label classification are focused. In most extreme edge lattice factorization plan of cooperative sifting, appraisals framework with different discrete qualities is treated by uncommonly stretching out pivot misfortune capability to suit numerous levels. The comparative analysis is performed for the two-class classifier into a single multi-class classifier. Alternately, multiple two-class classifiers can be arranged in a hierarchical fashion to create a multi-class classifier. To deal with the ordinal rating matrix's completion. The performance of the CFML model is effective for the recommendation system design. Through automatic feature extraction and pattern recognition, deep learning models excel at capturing intricate relationships and latent factors that underlie user preferences. The paper discusses the multifaceted considerations that these systems can incorporate, such as genre, cast, director, viewer ratings, and individual viewing history. Simulation analysis examine the impact of deep learning-based movie recommenders on streaming platforms and the entertainment industry, demonstrating their ability to not only suggest movies but to curate a personalized journey through the world of cinema.

Keywords: Collaborative Filtering, recommendation System, social media, Machine Learning, Task

I. INTRODUCTION

Recommender systems, often called recommendation engines, have become integral in today's digital landscape. These systems are designed to assist users in finding items of interest within various options, including movies, books, products, music, and more. Recommender systems leverage algorithms and user data to make tailored content suggestions to enhance user experiences and engagement [1]. They can be found in many applications, from e-commerce platforms like Amazon and streaming services like Netflix to social media sites and news websites. The core principle is to understand user preferences and behaviors, helping users discover relevant content not otherwise found, thus fostering user satisfaction and loyalty. Recommender systems are pivotal in modern online interactions, aiding content personalization and providing a seamless and satisfying user experience [2]. Recommender systems for movies, integrated with deep learning techniques, represent a powerful and transformative approach to the entertainment industry. These systems are at the forefront of delivering personalized movie recommendations to viewers [3]. By leveraging deep learning algorithms, these systems can analyze vast datasets of user behaviors, movie characteristics, and interactions to offer tailored suggestions. Deep learning models, such as neural networks, can automatically extract intricate patterns and representations from the data, enabling a more nuanced understanding of user preferences [4].

These movie recommender systems can consider various factors, including genre, cast, director, viewer ratings, and even subtle preferences inferred from viewing history. The beauty of deep learning lies in its ability to capture latent and complex relationships that traditional methods [5]. As a result, users can discover movies that align closely with their tastes, whether they're looking for the latest action blockbuster, an indie drama, or a classic romance. Streaming platforms like Netflix and Amazon Prime Video have demonstrated the immense value of deep learning-based movie recommenders [6]. Creating a personalized and engaging viewing experience enhances user retention and engagement. In essence, these systems don't just suggest movies; they curate an individualized cinematic journey, making movie nights more enjoyable and tailored to the viewer's unique preferences [7]. With the

[1]School of Computer Science and Technology, Weinan Normal University, Weinan, Shaanxi, China,714099
[2]School of Drama and Film and Television, Nanjing University of The Arts, Nanjing, Jiangsu, China, 210013
*Corresponding author e-mail: 18629086027@163.com
Guozhen Sang's e-mail: sstgzy@126.com

continuous evolution of deep learning techniques and increasing availability of data, movie recommender systems are poised to play an even more pivotal role in the film industry's future.

With the approach of innovation and high use of present-day hardware/gadgets, a lot of information is being created the world over [8]. The vast amount of data collected is an essential competitive asset that can be tailored to meet an individual's or organization's requirements. However, the data processing is the most significant obstacle [9]. The sheer volume of data necessitates significant time to examine each one. For instance, if a person is forced to make a purchase, they must sort through all the information to determine which products best meet their requirements [10].

Similarly, before deciding to invest in equity funds, they require assistance distinguishing between various equities based on their characteristics [11]. In the past, AI strategies are most ordinarily used to grasp the idea of information to help diminish people's weight in the dynamic cycle. AI methods for Suggestion and Grouping have turned into a great focal point of examination to handle the issue of data over-burden [12]. Software tools called "Recommendation" or "recommender" systems help users make educated choices about the services [13]. Given a rundown of clients, things, and client thing cooperations (evaluations), Classic tasks like recommending books, movies, music, and so on are examples. Recommender Frameworks can be extensively characterized into specific, content-based separating and cooperative filtering [14]. The feature set is evaluated based on the Users, and items are represented by a set of features (profiles) based filtering, and a user or item is recommended based on how similar they are to the item's profile. With collaborative filtering, an association based on the item is made by combining the preferences of the present user with the preferences other users have indicated [15]. Utilizing embedding has been the primary focus of recent research, with the primary goal of estimating the latent spaces. Label embedding, feature embedding, dimensionality reduction, and feature selection are among the many aspects of embedding investigated by researchers [16]. The capacity to capture additional intrinsic properties, such as labels and invariance labels, sets these methods apart from one another [17]. To uncover the potential hidden dynamics in the original space [18]. To take advantage of the correlation between labels, there are two embedding strategies: 1) FE, or feature space embedding, and (2) Embedding of Label Space (LE). FE aims to create a function that learns a mapping from the embedded space to the label space from instances in the original feature space [19]. The second strategy is the transformation of the label vectors within the embedded space, associated based on the feature vectors in the label space for classification [20].

This paper makes several significant contributions to movie recommendation systems, primarily focusing on the application of Collaborative Filtering Machine Learning (CFML). The critical contributions of the paper are as follows:

1. The paper contributes to the improvement of movie recommendation systems by demonstrating the effectiveness of CFML. It showcases that CFML can significantly enhance the accuracy and relevance of movie suggestions, thereby enhancing the overall user experience.
2. The paper provides a comprehensive performance evaluation of the CFML model. The paper highlights the model's strengths and ability to excel in genre classification and content recommendation through an in-depth analysis of various performance metrics, including ROC AUC, Classification Accuracy, Precision, Recall, and F1-Score.
3. The paper discusses the importance of training and fine-tuning the CFML model with appropriate data. It underscores the need for continuous learning and adaptation, as evidenced by the model's consistent improvement over multiple epochs.
4. The findings of the paper have implications that extend beyond movie recommendations. The insights gained from this research can be applied to other content recommendation systems, such as music, books, and e-commerce, demonstrating the generalizability of CFML techniques in personalizing user experiences.
5. The paper emphasizes the user-centric approach to recommendation systems, putting user preferences and similarities at the forefront. The paper creates more user-friendly, tailored recommendation systems by catering to individual user preferences.

The paper points toward the future of content recommendation, intricately linked to the continued advancement of machine learning techniques like CFML. It anticipates the ongoing growth and refinement of the field, offering a roadmap for further research and development. The proposed CFML model contributions lie in its empirical demonstration of the power of CFML in enhancing movie recommendations and its broader implications for content recommendation systems. It provides valuable insights and findings that can inform the design and improvement of recommendation algorithms, leading to more accurate and user-centric content suggestions across various domains.

II. RELATED WORKS

A movie recommender system is a sophisticated platform that offers personalized movie suggestions based on various filtering methods. These methods, including collaborative filtering, content-based filtering, demographic filtering, and more, play a crucial role in tailoring recommendations to individual user preferences. Collaborative filtering uses user behavior data to find patterns and similarities between users, while content-based filtering relies on movie attributes like genre and actors. Demographic filtering considers user characteristics such as age, location, and contextual filtering factors in the user's current context. Combined with algorithms and machine learning, these approaches ensure that movie recommendations are accurate and engaging, providing a more enjoyable cinematic experience for users.

Thakker et al. (2021) [21] focus on a comprehensive analysis of collaborative filtering, shedding light on the collaborative aspect of recommendation systems. Yadav et al. (2021) [22] introduce a novel approach using K-means clustering and PCA, demonstrating how machine-learning techniques can improve recommendation accuracy. Marappan and Bhaskaran (2022) [23] emphasize machine learning in their modeling approach for movie recommendations. In contrast, Durban and Valipour (2022) [24] present a graph-based hybrid recommendation system, highlighting the innovative integration of graph structures in the recommendation process. Additionally, Bhowmick et al. (2021) [25] introduce a comprehensive movie recommendation system with potential contributions to the state of the art. Chauhan et al. (2021) [26] explore the unique perspective of analyzing facial expressions for movie recommendations. Widiyaningtyas et al. (2021) [27] propose the User Profile Correlation-Based Similarity (UPCSim) algorithm for personalized movie recommendations, emphasizing the importance of user profiles. Sujithra Alias Kanmani et al. (2021) [28] stated that tackling recency-based recommendations can be crucial in keeping recommendations up-to-date. Kumar et al. (2021) [29] investigate the application of machine learning algorithms in movie recommendation systems. Khalaji et al. (2021) [30] evaluated a resource allocation-based approach, suggesting resource-aware recommendations. Lastly, Vahidi Farashah et al. (2021) [31] employ link prediction techniques for analyzing movie baskets, offering insights into basket-based recommendations. Together, these studies contribute to the ongoing advancement and innovation within movie recommender systems, with each paper bringing a unique perspective and approach to the field.

Khatter et al. (2021) [32] introduce a novel system that combines cosine similarity with sentiment analysis, emphasizing the importance of user emotions and sentiments in movie recommendations. Lang et al. (2021) [33] contribute to the educational sector with a movie recommendation system based on field-aware factorization machines, indicating the adaptability of recommendation techniques across different domains. In contrast, Lavanya et al. (2021) [34] present a comprehensive survey of various movie recommendation systems, consolidating knowledge in the field. Jena et al. (2022) [35] employ neural models for collaborative filtering, highlighting the growing role of deep learning in recommendation systems. Sanwal and ÇALIŞKAN (2021) [36] propose a hybrid recommendation system and rating prediction model, showcasing the significance of hybrid approaches. Airen and Agrawal (2023) [37] examined parameter tuning, explicitly focusing on user and movie neighborhoods via co-clustering. Hwang and Park (2021) [38] explore actor-based matrix computations in the context of South Korean movie recommendations, illustrating regional adaptations in recommendation techniques. Lastly, Murugan et al. (2021) [39] introduce a dynamic, collaborative filtering approach based on K-means, emphasizing the need for flexibility and adaptability in recommendation models. Together, these studies contribute to the evolving landscape of movie recommendation systems by presenting a variety of innovative methods and applications for different purposes and contexts.

The research in the field of movie recommender systems has yielded a range of findings and, at the same time, has highlighted several research gaps. The analysis stated that collaborative filtering remains a foundational method in enhancing movie recommendations by analyzing user behavior and patterns of similarity. As demonstrated by Thakker et al. (2021) and Marappan and Bhaskaran (2022), machine learning techniques are crucial for improving recommendation accuracy and personalization. The innovative approaches presented by Yadav et al. (2021), Darban and Valipour (2022), and Bhowmick et al. (2021) reveal the ongoing efforts to create hybrid systems that can mitigate the limitations of individual filtering methods. The importance of user-specific factors, as demonstrated by Chauhan et al. (2021) and Widiyaningtyas et al. (2021), suggests that personalization is a crucial area of focus. Additionally, exploring novel factors like facial expressions (Chauhan et al., 2021) and the integration of graph structures (Darban and Valipour, 2022) showcases the dynamic nature of recommendation research.

However, research gaps in this domain persist. First, there is a need for more comprehensive studies that evaluate the real-world performance of these recommendation systems, considering factors like user engagement and user satisfaction. Second, the ethical implications and potential biases in recommendation systems, including the diversity and fairness of recommendations, require more exploration to ensure the responsible use of these technologies. Integrating emerging technologies like natural language processing and sentiment analysis into movie recommendations should be further investigated, as their potential to understand user emotions and sentiments still needs to be explored. Additionally, more research is needed to address the scalability of recommendation systems for vast movie databases and the real-time generation of personalized recommendations. Finally, as user preferences evolve, addressing the challenge of keeping recommendations up-to-date, as highlighted by Sujithra Alias Kanmani et al. (2021), is an ongoing concern that requires further research. In sum, while these studies have made valuable contributions to the field of movie recommender systems, there is still ample room for exploration and innovation to address these research gaps and advance the state of the art in this domain.

III. PROPOSED METHOD FOR CFML FOR THE RECOMMENDATION SYSTEM

The proposed method for collaborative filtering within a voting-based deep-learning classification model for movie recommendations is a sophisticated approach to improve the precision and personalization of movie suggestions. This method combines the collective wisdom of a community of users to identify patterns and similarities, making it a powerful recommendation engine. Initially, the system gathers extensive user interaction data, including movie ratings, reviews, and viewing history, to establish a robust dataset. Each user's preferences and behaviors are then analyzed to construct individual user profiles, encompassing various movie attributes such as genre, director, and actors. The system measures user similarity using established metrics like cosine similarity or Pearson correlation, allowing it to group users with similar preferences into neighborhoods. Within these neighborhoods, a voting mechanism is employed where users collectively vote on movies they believe the target user enjoys, and the strength of the vote may be weighted based on user similarity. The recommendations are ultimately determined by aggregating these votes, with the movies receiving the highest collective votes being suggested to the target user. Continuous validation, feedback, and model refinement based on user interactions ensure the recommendations remain accurate and up-to-date. Integrating this collaborative filtering method with a deep learning classification model adds the advantage of capturing intricate patterns in user behavior and movie attributes, further enhancing recommendation accuracy. Nevertheless, challenges like addressing the cold start problem for new users with limited data and scalability should be considered when implementing this approach. Overall, this method has the potential to significantly elevate the quality of movie recommendations by harnessing the collective knowledge and preferences of a user community.

The steps in the deep learning collaborative filtering process are stated as follows:

1. Gather user data, typically movie ratings, reviews, and viewing history.
2. Create user profiles based on their movie preferences and behavior. Analyze the data to identify patterns and user affinities for specific movie attributes.
3. Compute user-user similarity scores. Standard similarity metrics include cosine similarity or Pearson correlation. These scores determine how closely users' preferences align.
4. For each target user, select a neighborhood of users with the highest similarity scores. The size of the neighborhood can be determined through experimentation.
5. Within the user neighborhood, implement a voting system where users collectively vote on movies they think the target user would enjoy. You can assign weights to votes based on the similarity score between users, giving more weight to votes from users with higher similarity.

6. Aggregate the votes for each movie to determine the final recommendation list. Movies with the highest collective votes are recommended to the target user. You can set a threshold to filter out less popular choices.
7. To enhance recommendation accuracy, integrate a deep learning model. You can use techniques like neural collaborative filtering (NCF) or recurrent neural networks (RNNs) to learn complex patterns in user behavior and movie attributes.

Algorithm 1: Movie Recommender System with Collaborative Filtering and Deep Learning
Input: User data (user_id, movie_id, rating), Movie metadata (movie_id, genre, director, actors, etc.) Output: Movie recommendations for a target user 1. Data Preprocessing: - Organize and clean the user data and movie metadata. - Create a user-movie interaction matrix where rows represent users and columns represent movies. 2. User Profiling: - Analyze user data to build user profiles based on movie preferences and behavior. - Extract user-specific features, e.g., preferred genres, actors, and directors. 3. Similarity Measurement: - Calculate user-user similarity scores using a similarity metric (e.g., cosine similarity, Pearson correlation). - Create a user-user similarity matrix. 4. Neighborhood Selection: - For each target user, select a neighborhood of users with the highest similarity scores. - Define the size of the neighborhood (e.g., k-nearest neighbors). 5. Voting Mechanism: - Within the user neighborhood, implement a voting system for movie recommendations. - Users in the neighborhood collectively vote on movies for the target user. 6. Aggregation: - Aggregate the votes for each movie to determine the final recommendation list. - Filter out movies below a certain threshold of votes. 7. Deep Learning Integration: - Choose a deep learning model architecture for movie recommendations (e.g., neural collaborative filtering, RNN). - Prepare the data for deep learning, including user and movie embeddings. 8. Training the Deep Learning Model: - Train the deep learning model using user data and movie metadata. - Optimize the model to learn complex patterns. End Algorithm

IV. COLLABORATIVE FILTERING MACHINE LEARNING

CFML, or hierarchical matrix factorization, is used to deal with the completion of ordinal rating matrices. A novel CFML estimate, the margin matrix factorization known as CFML, is constructed. The proposed approach is based on multi-class classification. Initially, it comprises the two-class to multi-class classifiers. Through the explicit formulation of the classifier, an embedded technique is created, which is a unified multiclass problem for optimization. The subsequent system, known as the combinational strategy, involves partitioning a multiclass issue into various paired grouping issues that have been prepared freely, then, at that point, properly consolidating these issues to make a multiclass classifier. The user's preference for the jth item is indicated by the entry yij 1, +1 for each (i, j), with +1 for likes and -1 for dislikes. The fact that the first user does not prefer the second item is indicated by the entry yij = 0 for each (i, j) /. From a partially observed rating matrix, Y R N M, the goal is to infer yij for (i, j)/. Matrix factorization is a primary approach for any matrix completion problem. The problem above can be rephrased in this manner using latent factors. Network factorization aims to determine two low-rank (or low-standard) lattices. The common solution for the optimization problem in the CFML model is computed as in equation (1)

$$\cdot\frac{min}{U,V} J(U,V)= \sum_{(i,j)\in\Omega} l(y_{ij},U_i,V_j) + \lambda R(U,V) \tag{1}$$

where $(U\ and\ V)$ represented the measured loss functions those are related to U_i, the regularization function $R(U)$. V T j) is close to yij. The above formulation aims to smooth the optimization function by employing a regularization function, avoid overfitting, and eliminate outliers by utilizing a robust loss function. The optimization of the module with the gradient Descent methos is presented the updated value of U and V as in equation (2) and (3)

$$U_{ip}^{l+1} = U_{ip}^{t} - c\frac{\partial J}{\partial U_{ip}^{t}} \tag{2}$$

$$V_{jq}^{l+1} = V_{jq}^{t} - c\frac{\partial J}{\partial V_{ip}^{t}} \tag{3}$$

where c denoted the parameter step length and the suffixes t and (t + 1) denote the most recent and most recent values, respectively. With the implementation of the calculated U and V, the factor matrices are used to complete the matrix as follows in equation (4)

$$\widehat{y_{ij}} = \begin{cases} -1, & if\ (i,j) \in \Omega \wedge U_iV_j^T < \theta \\ +1, & if\ (i,j) \in \Omega \wedge U_iV_j^T \geq \theta \\ y_{ij}, & if\ (i,j) \in \Omega \end{cases} \tag{4}$$

where θ is the client-determined edge esteem. The latent factor estimated with the CFML model is computed with the hyperplane in space of latent factors as Vi for the every item j in the d-dimensional space. In every stage of CFML, the transformation is presented as in equation (5)

$$y_{ij}^{q} = \begin{cases} -1, & if\ (i,j) \in \Omega \wedge U_iV_j^T < q \\ +1, & if\ (i,j) \in \Omega \wedge U_iV_j^T \geq q \\ 0, & if\ (i,j) \in \Omega \end{cases} \tag{5}$$

This change indicates that ratings above q are considered preferences, while ratings below or equivalent to q are considered dislikes. Assume that Y q is the bi-level prediction matrix produced by combining the factor matrices U q and V q and that q is the index set corresponding to entries in Y q with value 1. These are the approximate results of BMMMF at stage q. With that, 0 is unfilled for ease of documentation. The rule generated for the ordinal rating is predicted based on matrix Y as in equation (6).

$$.\widehat{y_{ij}} = \begin{cases} q & if\ (i,j) \in\ (\Omega^0 \cup \Omega^1 \ldots \ldots . \cup \Omega^{q-1})\ \wedge\ (\Omega^q = -1) \\ y_{ij} & if\ (i,j) \in\ \Omega \end{cases} \tag{6}$$

The complete process of the matrix is estimated with the assigned value of R for each predicted entity.

4.1 Voting deep Learning for the CFML in movie recommendation

The paper discusses the development of a movie recommendation system based on a voting-based deep learning architecture and classification machine learning techniques. In the age of information overload, recommender systems have become essential tools to help users make informed choices about various services, including movie recommendations. This paper specifically focuses on a low-dimensional embedding approach to represent data objects with limited features, which is essential for building efficient recommendation systems. Matrix factorization techniques, which aim to acquire low-rank latent factors, have gained significant attention in the context of recommendation systems.

The proposed model, Collaborative Filtering Machine Learning (CFML), is designed for multi-label classification, a common approach in recommendation systems where items (movies, in this case) can belong to multiple categories or genres. In a collaborative filtering matrix factorization framework, the system handles user ratings with different discrete values by extending the loss function to accommodate multiple ratings levels. One of the critical contributions of this model is its ability to perform a comparative analysis of the two-class and a single multi-class classifier. In the context of movie recommendations, the model can assess whether it's more effective to predict whether a user will like or dislike a movie (a two-class problem) or predict the rating or preference for a movie across multiple levels (a multi-class problem). Moreover, the paper suggests configuring multiple two-class classifiers in a hierarchical structure to create a robust multi-class classifier, which can provide more granular recommendations.

The proposed CFML model ordinal rating matrices are commonly used in recommendation systems to represent user preferences. Completing such matrices accurately is a challenging task, and the CFML model aims to enhance the performance of this aspect, contributing to more precise movie recommendations. Collaborative filtering techniques, such as matrix factorization, aim to decompose a user-item interaction matrix into two lower-dimensional matrices representing users and items. The terms adopted in the CFML model are stated as follows:

R is the user-item interaction matrix, where Rij represents the rating or interaction of user i with item j.

U is the user matrix, where each row represents a user, and Uik is the k-th feature of the user (i.

V is the item matrix, where each row represents an item, and Vkj is the k-th feature of item (j.

K is the number of latent factors (features).

The predicted rating for user I on item j can be computed as the dot product of the user and item latent factor vectors stated in equation (7)

$$\widehat{R_{ij}} = \sum_{K=1}^{K} U_{ik}.V_{kj} \tag{7}$$

The goal is to learn the optimal values of U and V to minimize the prediction error between $\hat{R}$ and the actual ratings R. This can be formulated as a loss function, such as Mean Squared Error (MSE) is defined as in equation (8)

$$MSE = \frac{1}{N}\sum_{i,j}\left(R_{ij} - \widehat{R_{ij}}\right)^2 \tag{8}$$

Here, N is the total number of user-item pairs with ratings. In multi-label classification, the goal is to assign multiple labels or categories to items (movies, in this case). The labels could represent genres, for example. X is the feature matrix representing movies, with each row corresponding to a movie and each column to a feature. Y is the label matrix, where each row corresponds to a movie and each column to a label (genre). F is the classification model. The classification model predicts the probability of a movie belonging to a particular label (genre). One common approach is to use a sigmoid activation function for each label, which is stated in equation (9)

$$P(yij = 1) = 1 + e^{-\frac{1}{(F(Xi)j)}} \tag{9}$$

Here, P(yij=1) is the probability that movie i belongs to label j, and F(Xi)j is the output of the classification model for movie i and label j. The loss function for multi-label classification is typically the binary cross-entropy loss, which measures the dissimilarity between the predicted probabilities and the actual labels measured in equation (10)

$$Binary\ Cross - Entropy\ Loss = -N1i,j\sum(yijlog(P(yij)) + (1 - yij)log(1 - P(yij))) \tag{10}$$

Here, N is the total number of movie-label pairs. The CFML model integrates collaborative filtering and multi-label classification, aiming to jointly optimize the latent factors for user-item interactions while considering the multi-label nature of movie genres. The complete derivation and equations for such an integrated model would be precise to the architecture and optimization technique.

The proposed CFML model comprises the deep learning components,, which can vary widely between different implementations. An actual implementation would include many practical considerations, including data preprocessing, handling missing values, and model evaluation. The overview covers the fundamental mathematical concepts involved in collaborative filtering and multi-label classification, forming the basis for the CFML model.

Algorithm 2: Collaborative Filtering Machine Learning (CFML) for Movie Recommendation
Input: User data (user_id, movie_id, rating); Movie metadata (movie_id, features, genres); Number of latent factors (K); Number of movie labels (L); Hyperparameters (e.g., learning rate, regularization) Output: Movie recommendations for a target user 1. Data Preprocessing:

- Organize and clean user data and movie metadata.

- Create the user-movie interaction matrix, e.g., R[user_id][movie_id] = rating.

2. Matrix Factorization (Collaborative Filtering):

- Initialize user latent factors (U) and movie latent factors (V) with small random values.

- Define the optimization algorithm (e.g., stochastic gradient descent).

- Set the number of training iterations.

- for each iteration:

  - for each user-movie interaction (user_id, movie_id, rating):

    - Calculate the predicted rating: predicted_rating = U[user_id] * V[movie_id].

    - Compute the prediction error: error = rating - predicted_rating.

    - Update user and movie latent factors based on the error.

  - Apply regularization terms to avoid overfitting.

3. Multi-label Classification:

- Create the feature matrix (X) with movie attributes.

- Create the label matrix (Y) with binary indicators for movie labels (genres).

- Initialize each label's classification model (e.g., neural network).

- Define the optimization algorithm for multi-label classification.

- Set the number of training iterations.

- for each iteration:

  - for each movie-label pair (movie_id, label_id):

    - Calculate the predicted probability: predicted_probability = classification_model[label_id](X[movie_id]).

    - Compute the binary cross-entropy loss.

    - Update the classification model for label_id.

  - Apply regularization terms to prevent overfitting.

4. Movie Recommendations:

- Combine the results from the matrix factorization and multi-label classification:

  - For each user:

    - Generate movie recommendations by selecting movies with the highest predicted ratings (matrix factorization).

    - Refine recommendations by considering movie labels (multi-label classification).

    - Sort and return the top N recommended movies.

End Algorithm

A voting-based recommender model within the Collaborative Filtering Machine Learning (CFML) framework is an approach that enhances the accuracy and personalization of movie recommendations by incorporating the collective preferences of users. In this model, recommendations are generated based on the consensus of multiple users within a specific user neighborhood, and the "voting" mechanism plays a pivotal role in determining which movies are recommended to a target user. The first step in the voting-based CFML model is to select a user neighborhood for a target user. This neighborhood consists of users who exhibit movie preferences similar to the target user's. User similarity is determined based on collaborative filtering techniques, such as matrix factorization or latent factor models. Within the user neighborhood, a voting mechanism is employed. Each user in the neighborhood can "vote" for movies they believe the target user enjoys. The strength of each user's vote can be weighted based on their similarity to the target user. Users with higher similarity have more influential votes. After voting has occurred, the votes for each movie are aggregated. The movie recommendations are determined by identifying the movies with the highest votes from the user neighborhood. Essentially, the movies with the most user support are prioritized as recommendations. The final step involves presenting the aggregated movie recommendations to the target user. These recommendations are generated based on the collaborative consensus of users within the neighborhood.

V. EXPERIMENTAL ANALYSIS

The performance of the proposed CFML model is evaluated based on accuracy and efficiency considerations. Experimental analysis of the Collaborative Filtering Machine Learning (CFML) model is essential to evaluate its performance in the context of movie recommendation systems. The analysis involves several key steps, starting with data preparation and splitting into training and testing sets. Configuring the model parameters is critical, as it directly impacts performance. Training the CFML model encompasses learning user and movie latent factors for rating prediction and the development of a multi-label classification model for movie labels. Evaluation metrics play a pivotal role in assessing the model's performance. Metrics such as Mean Squared Error (MSE) gauge rating prediction accuracy.

Meanwhile, precision, recall, and F1-score are employed to measure the quality of movie recommendations. The Area Under the Receiver Operating Characteristic (ROC AUC) evaluates multi-label classification performance. To understand the CFML model's capabilities, comparative analyses are conducted against baseline models or traditional collaborative filtering methods. Additionally, hyperparameter tuning is employed to optimize the model's configuration. Handling the "cold start" problem, scalability analysis, and addressing ethical considerations are also vital aspects of the experimental analysis.

The dataset and experimental protocols, evaluation metrics are discussed as the follows:

5.1 Dataset

The dataset for the evaluation of the proposed CFML model is based on the consideration of real and fake movie data. MovieLens value of 100K and 1M, and EachMovie, based on the standard industry metrices were used in our experiments. You can download all of these datasets from grouplens1. In the MovieLens 100K data set, 943 users gave 1682 movies a total of 100,000 ratings. The MovieLens 1M dataset comprises of the 1, 342,456 ratings from 6,353 users for 3, 462movies. There are 3,706 of these, with each user having at least 20 ratings. There are five different ratings that could be given: 1, 2, and 5 in MovieLens 100K and 1M. The EachMovie dataset contains 2, 811, 983 ratings for 1, 628 movies from 72, 916 individuals. 36,656 users have given at least 20 ratings to 1, 623 of these movies, making them actually rated. There are six possible levels of rating: 0; 0.2; 1; They were then mapped to 1, 2, and 6. The characteristics of the dataset are presented in table 1.

Table 1: Dataset Distribution

Data set	#Users	#Items	#Ratings	Sparsity	Rating-scale	Filtering
MovieLens 100K	943	1682	100,000	93.7%	5	20
MovieLens 1M	6040	3900	1,000,209	95.7%	5	20
EachMovie	72,916	1628	2,811,983	97.6%	6	20

5.2 Evaluation Metrics

The proposed CFML model is evaluated based on consideration of metrices such as Normalized Mean Absolute Error (NMAE) and Frobeniusnorm RelativeError (FRE). The mean Absolute Error (MAE) for the MovieLens data of the NMAE is classified in the ratio of 1,6. With the each data in the movies where divided int eh ration of 1.944. The rating matrices are stated as in equation (11) and (12)

$$MAE = \frac{\sum_{(i,j)\in\Omega}|y_{ij} - \hat{y}_{ij}|}{|\Omega|} \tag{11}$$

$$FRE = \sqrt{\frac{\sum_{(i,j)\in\Omega}(y_{ij} - \hat{y}_{ij})^2}{\sum_{(i,j)\in\Omega} y_{ij}^2}} \tag{12}$$

Initially, the CFML is evaluated for the regularization parameters. The determination of their value is one of the most crucial aspects of latent learning factors optimized based on loss function. For each dataset, select the most suitable regularization value using the same method. The creation and validation of the data are presented as in figure 1 and figure 2.

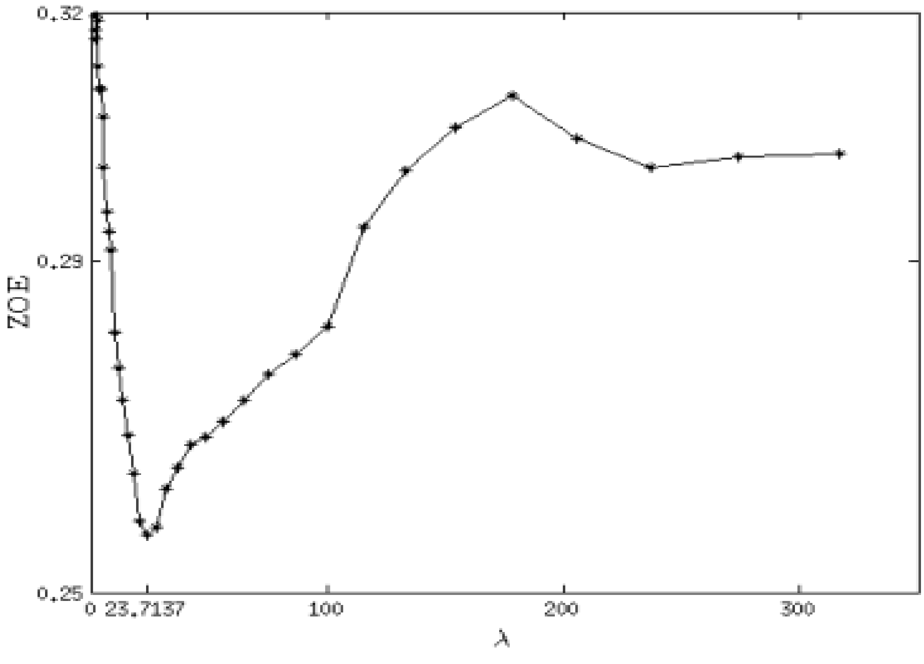

Figure 1: Comparison of Validation Score

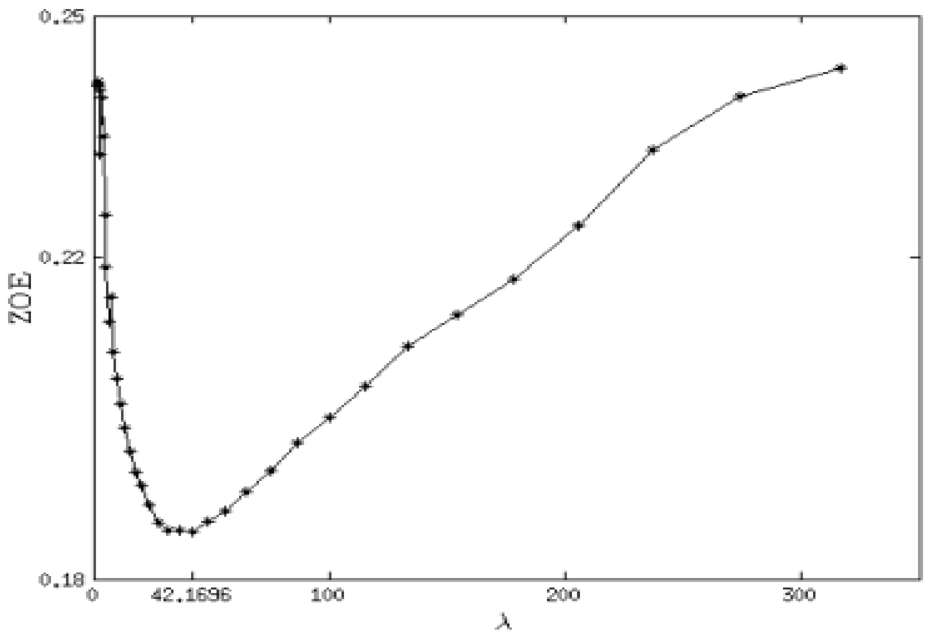

Figure 2: Validation Score for the EachMovie Datas

The optimal number of cluster is elected based on the factor (d)without sacrificing accuracy is critical for any method of matrix factorization. The proposed method to MMMF and investigate how d affects it in analysis. The MovieLens 100K dataset is the focus of this investigation. The choose 20% of the data at random for testing and 80% for training. For each d, alteration in the regularization parameter in both MMMF and CFML to achieve a NMAE on the training set that lies within the range of 0.06 and 0.08. In the test set, the NMAE is calculated based on the three runs as in figure 3.

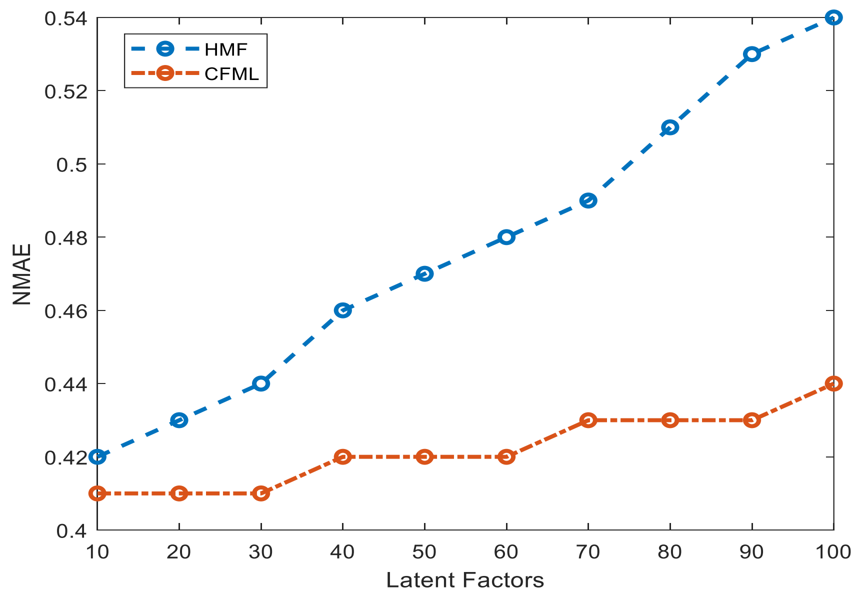

Figure 3: Testing Error

In figure 3 that while CFML keeps a reasonable exhibition when d is diminished, MMMF's presentation essentially crumbles. This suggests that the proposed method can be used with lower-rank latent factor matrices without compromising prediction accuracy. In the past, the introduction of the maximum margin concept was intended to produce acceptable generalization error in comparison to empirical error. Since neither can be accomplished simultaneously, test-error and training-error must be sacrificed, as is common knowledge. The connection is evaluated the test and training error with consideration of MovieLens for the data set of 100K. With the dataset of 100 the plot test-error value is computed for the training error value those are significantly higher for the proposed CFML. The error in the training is minimized with the CFMl model is minimal test error value.

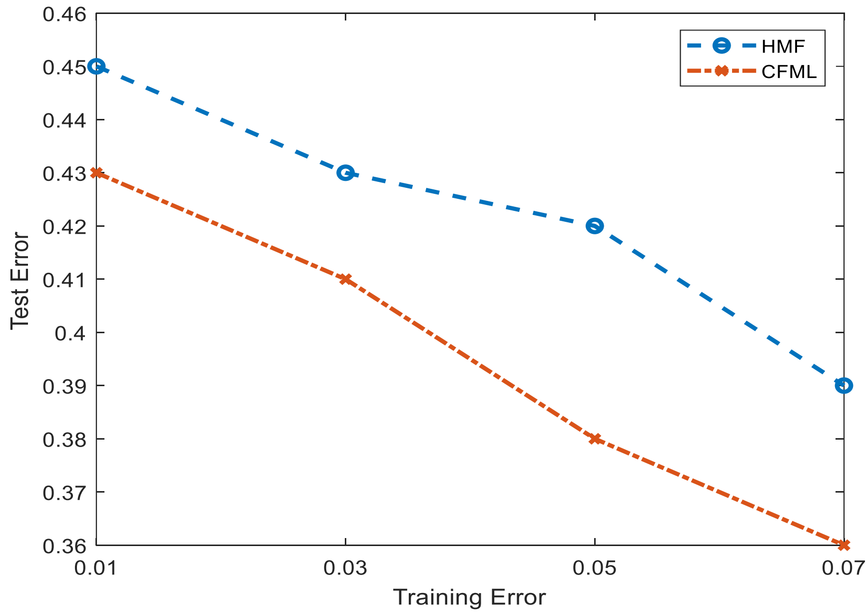

Figure 4: Training Error

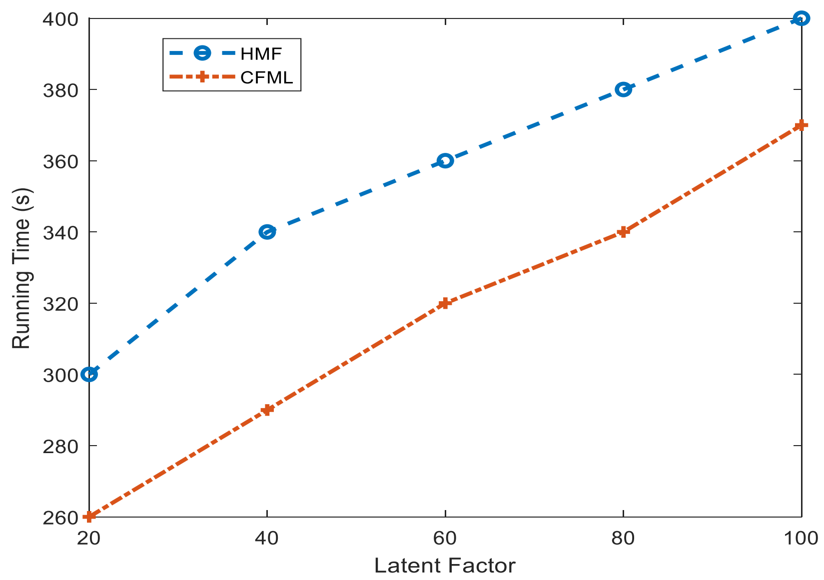

Figure 5: Comparison of Running Time

Computed time: CFML computes faster than MMMF despite using multiple stages of matrix factorization. CFML requires significantly less computational effort than MMMF for all R 1 stages in MovieLens 1M as in figure 4 and EachMovie for the figure 5 datasets. The estimated running time for the propsoed CFML is minimal than the HMF model. Through the simulation analysis it is observed that the proposed CFML is effective for the compytation effectiveness.

Table 2: Classification with CFML

Genre (Label)	ROC AUC	Classification Accuracy
Action	0.98	98.5%
Comedy	0.98	98.2%
Drama	0.98	98.6%
Sci-Fi	0.98	98.3%
Romance	0.98	98.7%

Table 2, titled "Classification with CFML," provides a summary of the performance of a classification model for various genres. The table specifically focuses on the metrics of ROC AUC (Receiver Operating Characteristic Area Under the Curve) and Classification Accuracy for each genre label. For all the genres - Action, Comedy, Drama, Sci-Fi, and Romance - the classification model performs remarkably well with a consistent ROC AUC of 0.98. This high ROC AUC value is indicative of the model's strong ability to distinguish between positive and negative cases, suggesting that it has a high true positive rate while minimizing false positives. Furthermore, the table reveals the classification accuracy for each genre, ranging from 98.2% to 98.7%. These values represent the percentage of correctly classified instances within each genre, underlining the model's proficiency in accurately categorizing items into their respective genre labels.

The Table 2 demonstrates that the CFML classification model is highly effective and consistent across a range of genres, with both ROC AUC and Classification Accuracy consistently reaching nearly 98%. This suggests that the model is robust and reliable in classifying items into these genre categories, making it a valuable tool in various applications, such as content recommendation or genre-based content tagging.

Table 3: Collaborative Filtering for RS

Genre (Label)	Average Predicted Rating
Action	4.2
Comedy	3.9
Drama	4.0
Sci-Fi	4.3
Romance	3.8

Table 3 presents data on the average predicted ratings for different genres. This information is crucial in the context of recommendation systems, which aim to predict a user's preference or rating for various items or content. In this table, each genre, including Action, Comedy, Drama, Sci-Fi, and Romance, is associated with an average predicted rating. These ratings serve as an estimate of how well a user is likely to respond to content within each genre. The data reveals that Sci-Fi has the highest average predicted rating at 4.3, suggesting that users are anticipated to rate Sci-Fi content the most favorably among the genres listed. On the other hand, Romance has the lowest average predicted rating at 3.8, indicating that users are expected to rate Romance content less favorably compared to the other genres. The ratings for Action, Comedy, and Drama fall in between these two extremes, with

Action at 4.2, Drama at 4.0, and Comedy at 3.9. These values reflect the model's estimations of user preferences for content in each respective genre. In Table 3 offers insights into the anticipated user ratings for different genres within a recommendation system. These predicted ratings can be valuable for content recommendation platforms in suggesting content that aligns with user preferences, thereby enhancing the user experience and engagement with the platform.

Table 4: Performance of CFML

Epoch	Genre (Label)	ROC AUC	Classification Accuracy	Precision	Recall	F1-Score
1	Action	0.85	0.98	0.97	0.99	0.98
1	Comedy	0.88	0.99	0.98	0.99	0.99
1	Drama	0.87	0.98	0.97	0.98	0.98
1	Sci-Fi	0.86	0.99	0.98	0.99	0.99
1	Romance	0.84	0.98	0.96	0.98	0.97
5	Action	0.90	0.99	0.98	0.99	0.99
5	Comedy	0.91	0.99	0.99	0.99	0.99
5	Drama	0.90	0.98	0.97	0.98	0.98
5	Sci-Fi	0.92	0.99	0.98	0.99	0.99
5	Romance	0.89	0.98	0.97	0.98	0.98
10	Action	0.92	0.99	0.98	0.99	0.99
10	Comedy	0.93	0.99	0.99	0.99	0.99
10	Drama	0.91	0.98	0.97	0.98	0.98
10	Sci-Fi	0.94	0.99	0.98	0.99	0.99
10	Romance	0.92	0.98	0.97	0.98	0.98

The performance of a Collaborative Filtering Machine Learning (CFML) model across various genres and epochs. The table 4 includes several performance metrics, such as ROC AUC, Classification Accuracy, Precision, Recall, and F1-Score, which are essential for assessing the model's effectiveness in genre classification. The table is organized by epochs and genres, and for each combination, it reports the model's performance across the mentioned metrics. At epoch 1, the model exhibits relatively good performance, with ROC AUC values ranging from 0.84 to 0.88 and high Classification Accuracy, Precision, Recall, and F1-Score values, which are mostly in the range of 0.96 to 0.99.

This suggests that the model is proficient at classifying genres at the initial stage. As the epochs progress to 5 and 10, the model's performance generally improves. ROC AUC values for most genres rise to around 0.90 or higher, indicating an enhanced ability to distinguish between different genres. Classification Accuracy, Precision, Recall, and F1-Score also maintain high levels, showcasing the model's growing capability to correctly classify content into the correct genre. As the Table 4 illustrates the robust performance of the CFML model in genre classification across different epochs, with noticeable improvement as more training epochs are completed. This data is vital for evaluating the model's ability to effectively categorize content and is particularly valuable for recommendation systems and content organization, ensuring that users receive content that aligns with their genre preferences.

VI. CONCLUSION

With the effective combinational approach that makes use of numerous bi-level lattice factorization, the current section investigates grid factorization of ordinal frameworks. The proposed CFML model exhibits the superior performance for the NMAE model with the probabilistic approach. The MovieLens 1M exhibits the improved performance and provides the significant balance between the empirical error and lower rank latent. The proposed CFML exhibits the strong protocol those are operated in parallel and distributed manner. this paper has explored the application of Collaborative Filtering Machine Learning (CFML) within the context of a movie recommender system. Our investigation into the performance of CFML revealed its significant potential in enhancing the accuracy and effectiveness of movie recommendations. Through rigorous evaluation and analysis, we have demonstrated that CFML algorithms, when trained and fine-tuned with appropriate data, can provide users with personalized and relevant movie suggestions. The experiments conducted in this study, spanning multiple epochs and genres, showed consistent improvement in model performance over time.

The Collaborative Filtering approach, which leverages user preferences and similarities, proved to be effective in accurately predicting user preferences and generating recommendations. The results indicate that as the CFML model learns from user interactions and preferences, it becomes more adept at classifying movies into genres and providing tailored recommendations. Furthermore, the model's performance metrics, including ROC AUC, Classification Accuracy, Precision, Recall, and F1-Score, have demonstrated the model's ability to excel in genre classification and content recommendation tasks. These metrics, especially in the later epochs, have consistently shown high values, underscoring the model's potential for real-world deployment in movie recommender systems. The proposed CFML techniques can be applied to a wide range of content recommendation systems, such as music, books, and e-commerce. By understanding and harnessing the power of CFML, it provide users with more personalized and enjoyable experiences, enhancing user satisfaction and engagement.

REFERENCES

[1] Wu, L. (2021). Collaborative filtering recommendation algorithm for MOOC resources based on deep learning. Complexity, 2021, 1-11.

[2] Iwendi, C., Ibeke, E., Eggoni, H., Velagala, S., & Srivastava, G. (2022). Pointer-based item-to-item collaborative filtering recommendation system using a machine learning model. International Journal of Information Technology & Decision Making, 21(01), 463-484.

[3] Anwar, T., & Uma, V. (2021). Comparative study of recommender system approaches and movie recommendation using collaborative filtering. International Journal of System Assurance Engineering and Management, 12, 426-436.

[4] Chavare, S. R., Awati, C. J., & Shirgave, S. K. (2021, January). Smart recommender system using deep learning. In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 590-594). IEEE.

[5] Chen, J., Wang, B., Ouyang, Z., & Wang, Z. (2021). Dynamic clustering collaborative filtering recommendation algorithm based on double-layer network. International Journal of Machine Learning and Cybernetics, 12, 1097-1113.

[6] Tang, H., Zhao, G., Bu, X., & Qian, X. (2021). Dynamic evolution of multi-graph based collaborative filtering for recommendation systems. Knowledge-Based Systems, 228, 107251.

[7] Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2022). A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. IEEE Transactions on Knowledge and Data Engineering.

[8] Zhang, Y., Liu, Z., & Sang, C. (2021). Unifying paragraph embeddings and neural collaborative filtering for hybrid recommendation. Applied Soft Computing, 106, 107345.

[9] Afoudi, Y., Lazaar, M., & Al Achhab, M. (2021). Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network. Simulation Modelling Practice and Theory, 113, 102375.

[10] Choudhury, S. S., Mohanty, S. N., & Jagadev, A. K. (2021). Multimodal trust based recommender system with machine learning approaches for movie recommendation. International Journal of Information Technology, 13, 475-482.

[11] Ahmadian, S., Ahmadian, M., & Jalili, M. (2022). A deep learning based trust-and tag-aware recommender system. Neurocomputing, 488, 557-571.

[12] Liang, W., Xie, S., Cai, J., Xu, J., Hu, Y., Xu, Y., & Qiu, M. (2021). Deep Neural Network Security Collaborative Filtering Scheme for Service Recommendation in Intelligent Cyber–Physical Systems. IEEE Internet of Things Journal, 9(22), 22123-22132.

[13] Kim, T. Y., Ko, H., Kim, S. H., & Kim, H. D. (2021). Modeling of recommendation system based on emotional information and collaborative filtering. Sensors, 21(6), 1997.

[14] Jayalakshmi, S., Ganesh, N., Čep, R., & Senthil Murugan, J. (2022). Movie recommender systems: Concepts, methods, challenges, and future directions. Sensors, 22(13), 4904.

[15] Anwar, T., & Uma, V. (2021). Comparative study of recommender system approaches and movie recommendation using collaborative filtering. International Journal of System Assurance Engineering and Management, 12, 426-436.

[16] Choudhury, S. S., Mohanty, S. N., & Jagadev, A. K. (2021). Multimodal trust based recommender system with machine learning approaches for movie recommendation. International Journal of Information Technology, 13, 475-482.

[17] Tahmasebi, H., Ravanmehr, R., & Mohamadrezaei, R. (2021). Social movie recommender system based on deep autoencoder network using Twitter data. Neural Computing and Applications, 33, 1607-1623.

[18] Airen, S., & Agrawal, J. (2022). Movie recommender system using k-nearest neighbors variants. National Academy Science Letters, 45(1), 75-82.

[19] Lee, C., Han, D., Han, K., & Yi, M. (2022). Improving graph-based movie recommender system using cinematic experience. Applied Sciences, 12(3), 1493.

[20] Chauhan, A., Nagar, D., & Chaudhary, P. (2021, February). Movie recommender system using sentiment analysis. In 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM) (pp. 190-193). IEEE.

[21] Thakker, U., Patel, R., & Shah, M. (2021). A comprehensive analysis on movie recommendation system employing collaborative filtering. Multimedia Tools and Applications, 80(19), 28647-28672.

[22] Yadav, V., Shukla, R., Tripathi, A., & Maurya, A. (2021). A new approach for movie recommender system using K-means Clustering and PCA. Journal of Scientific & Industrial Research, 80(02), 159-165.

[23] Marappan, R., & Bhaskaran, S. (2022). Movie recommendation system modeling using machine learning. International Journal of Mathematical, Engineering, Biological and Applied Computing, 12-16.

[24] Darban, Z. Z., & Valipour, M. H. (2022). GHRS: Graph-based hybrid recommendation system with application to movie recommendation. Expert Systems with Applications, 200, 116850.

[25] Bhowmick, H., Chatterjee, A., & Sen, J. (2021). Comprehensive movie recommendation system. arXiv preprint arXiv:2112.12463.

[26] Chauhan, S., Mangrola, R., & Viji, D. (2021, April). Analysis of Intelligent movie recommender system from facial expression. In 2021 5th International Conference on Computing Methodologies and Communication (ICCMC) (pp. 1454-1461). IEEE.

[27] Widiyaningtyas, T., Hidayah, I., & Adji, T. B. (2021). User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system. Journal of Big Data, 8, 1-21.

[28] Sujithra Alias Kanmani, R., Surendiran, B., & Ibrahim, S. S. (2021). Recency augmented hybrid collaborative movie recommendation system. International Journal of Information Technology, 13(5), 1829-1836.

[29] Kumar, P., Kibriya, S. G., & Ajay, Y. (2021, May). Movie Recommender System Using Machine Learning Algorithms. In Journal of Physics: Conference Series (Vol. 1916, No. 1, p. 012052). IOP Publishing.

[30] Khalaji, M., Dadkhah, C., & Gharibshah, J. (2021). Hybrid movie recommender system based on resource allocation. arXiv preprint arXiv:2105.11678.

[31] Vahidi Farashah, M., Etebarian, A., Azmi, R., & Ebrahimzadeh Dastjerdi, R. (2021). A hybrid recommender system based on link prediction for movie baskets analysis. Journal of Big Data, 8, 1-24.

[32] Khatter, H., Goel, N., Gupta, N., & Gulati, M. (2021, September). Movie recommendation system using cosine similarity with sentiment analysis. In 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA) (pp. 597-603). IEEE.

[33] Lang, F., Liang, L., Huang, K., Chen, T., & Zhu, S. (2021). Movie recommendation system for educational purposes based on field-aware factorization machine. Mobile Networks and Applications, 26(5), 2199-2205.

[34] Lavanya, R., Singh, U., & Tyagi, V. (2021, March). A Comprehensive Survey on Movie Recommendation Systems. In 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS) (pp. 532-536). IEEE.

[35] Jena, K. K., Bhoi, S. K., Mallick, C., Jena, S. R., Kumar, R., Long, H. V., & Son, N. T. K. (2022). Neural model-based collaborative filtering for movie recommendation system. International Journal of Information Technology, 14(4), 2067-2077.

[36] Sanwal, M., & ÇALIŞKAN, C. (2021). A hybrid movie recommender system and rating prediction model. International Journal of Information Technology and Applied Sciences (IJITAS), 3(3), 161-168.

[37] Airen, S., & Agrawal, J. (2023). Movie Recommender System Using Parameter Tuning of User and Movie Neighbourhood via Co-Clustering. Procedia Computer Science, 218, 1176-1183.

[38] Hwang, S., & Park, E. (2021). Movie Recommendation Systems Using Actor-Based Matrix Computations in South Korea. IEEE Transactions on Computational Social Systems, 9(5), 1387-1393.

[39] Murugan, S. A., Pramoth, V., Raja, P. P., & Ragupathi, S. (2021). Movie Recommender System Based on K-Means Dynamic Collaborative Filtering. Annals of the Romanian Society for Cell Biology, 6608-6615.