International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I9P120

Volume 73 Issue 9, 217-228, September 2025

Original Article

Evolutionary Deep Learning Based CNNTWSVM Movie Recommendation System

Nisha Bhalse[1], Ramesh Thakur[2], Archana Thakur[3]

[1]Institute of Engineering & Technology, DAVV, Indore, Institute of Engineering & Science, IPS Academy, Indore, India.
[2]International Institute of Professional Studies, DAVV, Indore, India.
[3]School of CS & IT, DAVV, Indore, India.

[1]Corresponding Author : nishabhalse@gmail.com

Received: 26 March 2025 Revised: 15 September 2025 Accepted: 22 September 2025 Published: 30 September 2025

Abstract - *In this age of digital content, a movie recommendation system is essential for users to recommend highly accurate and efficient personalized movies and services. Over the past decade, researchers have been studying recommendation algorithms, and they have been implemented on various platforms, including e-commerce and movie streaming services. However, traditional recommendation algorithms suffer from cold-start and sparsity problems. The proposed CNNTWSVM movie recommendation system, which integrates the Convolutional Neural Networks (CNN) with Twin Support Vector Machines (TWSVM), was presented in this paper. The experimental performance evaluation of the CNNTWSVM personalized movie recommendation system against traditional algorithms, including Matrix Factorization, Autoencoder, Neural Collaborative Filtering, and standard CNN models, using the 1M MovieLens dataset. The results show the proposed CNNTWSVM model outperforms, achieving the lowest root mean squared and mean absolute errors of 0.805 and 0.63. The proposed CNNTWSVM model minimizes prediction errors effectively. It achieves the highest values of accuracy (95.2%), precision (97.5%), recall (89.0%), normalized discounted cumulative gain (89.0%), diversity (94.7%), and serendipity (93.2%). The proposed CNNTWSVM personalized movie recommendations system recommends accurate and relevant movies and also exposes users to a diverse selection of lesser-known yet engaging movies according to their preferences.*

Keywords - *Collaborative filtering, Twin support vector machines, Convolutional Neural Networks, Recommendation system.*

1. Introduction

In today's digital era, we have easier access to more information than ever before. According to the decision-making theory, the availability of a large volume of information makes the poorest decision rather than people helping them make the right decision and reach the goal [1]. Information overload is a by-product of the mobile and digital revolution, and it is becoming a significant issue. The large amounts of online information are available; then, the information overloading problem is faced by users. This issue creates the problem of anxiety and overwhelm. A recommendation system filters the information and provides personalized suggestions according to the user's interests. These systems resolve the issue of information overload [2]. The first recommendation system appeared in the mid-1990s; since then, it has been the main focus of research [1]. Many websites and platforms use recommendation systems, such as e-commerce websites, YouTube, Spotify, news, movies, etc. [2]. A recommendation system uses two types of data: textual profiles or keywords that contain the user-item interactions and ratings that contain the purchasing behaviours of items and users [3]. Recommendation systems are categorised into Content-Based filtering (CB), Collaborative Filtering (CF) and a hybrid approach. Collaborative filtering is recommending items to users based on the preferences of other similar users that are liked them in past. CB recommends the items to users according to their own profile features that they have liked in past. These models mainly focused on user profiles [4]. A hybrid approach is the combination of the advantages of collaborative and content-based filtering techniques; it provides a more robust and effective recommendation system [5].

Nevertheless, every recommendation technique has its own advantages and limitations. Today, traditional methods like matrix factorization, autoencoder and CNN face data sparsity challenges. As well as class imbalance, lack of decision boundaries, and the problem of information obsolescence and diversity due to the limited recommendation [6, 7]. The matrix factorization suffers from a non-linear relation, auto encoders struggle with over-fitting, and neural collaborative filtering has a scalability issue. A convolutional neural network extracts features, but their primary drawback is that they lack classifiers.

Multilayer neural networks are used to extract complex information from input data. It recommends a more accurate recommendation even if the information is incomplete. Deep learning has been integrated into recommendation systems during the past decade; it has been highly effective and successful on a variety of tasks [8-10].

A CNN is a widely used model. It can be applied to recommendation systems for processing and analyzing the data. CNN is used in recommendation systems for analyzing and extracting the relevant features of movies from the 1 M Movielens datasets. The CNNTWSVM is a novel approach; in this paper, we integrated the Convolutional Neural Networks (CNN) [11] with Twin Support Vector Machines (TWSVM).

The proposed CNNTWSVM model is used for feature extraction, and TWSVM is recognized for building high-performance classifiers [12, 13]. It has improved the personalized recommendation and distinguished between user preferences and recommended movies according to users' preferences in the recommendation system.

In this paper, the CNNTWSVM Movie Recommendation System is an integration of CNN and TWSVM algorithm. By harnessing the feature extraction abilities of CNN and the classification power of TWSVM, this system provides a refined, personalized movie-watching experience, making it a promising innovation of movie recommendation systems.

The key contributions of the paper:

1) Introduce a novel hybrid CNNTWSVM movie recommendation model that integrates CNN with TWSVM. This convergence effectively extracts complex and meaningful features from MovieLens, which are crucial for making accurate recommendations and have the potential to understand user behaviour better.
2) The CNNTWSVM model allows for highly personalized movie recommendations by using movies and user features.
3) The CNNTWSVM model closes the gap between the recommended movies and user satisfaction.
4) Using simple heuristics like recommending popular or trending movies to new users or movies similar to the ones they have interacted with (based on metadata) could be another practical solution. This is more of a stopgap measure, but it can provide some level of personalized recommendation.

The paper structure is as follows: Section 2 contains the literature studies of movie recommendation algorithms. In Section 3, the proposed CNNTWSVM model algorithm and theoretical framework are described in detail. The outcomes of the experiment are presented in Section 4. In Section 5, conclusions are presented.

2. Literature Review

Data drives today's world, and recommendation systems have revolutionised how we discover content of interest, with movie recommendation systems being especially impactful and lucrative. These systems assist billions of users in recommending personalised movies and videos from large databases such as Flipkart, Movielens, YouTube, Netflix, etc. [14]. Over the past two decades, various algorithms have been developed for movie recommendation systems.

Content-based algorithm recommends movies to users that are very similar to those they have already watched [5]. The movies are typically categorized based on similarities such as genre, actors, directors, and other features. These systems recommend those movies that they have not watched yet, and are similar to the users' viewing preferences, like MovieLens. Content-based filtering in movie recommendation systems relies on the user's provided data, such as textual descriptions of movies [15].

The collaborative filtering technique recommends items according to the user's behaviors and preferences. Machine learning based matrix factorization and clustering models are used in model-based Collaborative filtering. These models extract the feature from the complex user feature interaction [16, 17]. RSs employ neural networks that depict the nonlinear relationship between users and items. Deep learning models make use of recurrent and convolutional neural networks [18].

An ensemble learning technique is used to collect and categorize the user-movie matrix according to user-movie interactions. Trained the system and predicted the movies for active users. The limitations of this algorithm are low accuracy, limited applicability, and a lack of novelty, as it ignores user community similarities. A movie tag prediction system was developed to solve this issue, which forecasts pertinent tags for every movie and segment according to predicted tags [19].

The movie recommendation system will not be able to comprehend a user's preferences if they have rated too few films, which will result in inadequately tailored recommendations. New users with low ratings are unlikely to receive reliable recommendations because there are not enough features taken from the current database.

The movie recommendation system provides a poor personalized recommendation if a user has rated very few movies because it is unable to understand their preferences. The new users have limited ratings [17]. In recommendation systems, the CF algorithm predicts a user's preferences based on other similar users' preferences [5]. In which the user's feature and item's feature are used, along with interactions that also occur, such as ratings, reviews and purchases.

There are two categories of CF recommendations: item-based and user-based [20]. The user-based CF model recommends movies by identifying similar user tastes.

An item-based CF model recommends movies by analyzing the user's past history and movie similarities. Using MovieLens dataset ratings, collaborative filtering recommends top-N movies according to similar users' preferences [21]. The recommendation system accuracy is measured through the size of the data and the diversity between users and items. Collaborative filtering algorithms face the problem of sparsity, new users and new items. These challenges are affecting the accuracy and effectiveness of the recommendation system [22].

Many traditional methods of movie recommendation do not recommend accurate and reliable suggestions to new users, due to the limited information available about new users. This challenge is resolved; model-based Collaborative Filtering (CF) methods incorporate advanced deep learning and machine learning techniques. In model-based collaborative filtering techniques, matrix factorization is used to reduce data sparsity [23].

The model-based CF movie recommendation systems use the user-movie rating matrix factorization that integrates the users' and movies' features with lower dimensionality. Model-based recommender systems have various advantages. Training time is less, the recommendation is fast, and it reduces the over-fitting [6].

However, matrix factorization still faces challenges in sparse environments, where overfitting becomes more problematic if data sparsity is increased in the rating matrix [22]. CNN is one type of deep learning technique; it is a good performer in extracting features from complex patterns and hidden features within recommendation datasets, offering promising improvements over traditional CF approaches [23, 24].

A typical movie recommendation system consists of two key components: the encoding and decoding parts. In the encoding phase, it processes raw movie content and user ratings as vectors, extracting hidden features through multiple denoising autoencoders. These features are then used in the decoding phase for model predictions. Studies have shown that deep learning with collaborative filtering techniques can effectively reduce the cold-start problem of recommendation systems [22].

This movie recommendation system demonstrated that deep learning techniques achieve higher efficiency and accuracy. It is also important to recognize that movies contain multiple modalities: text, images, and audio. Multimodal deep learning is particularly effective.

It integrates and processes diverse data types, enhancing feature extraction and improving recommendation performance [25]. The traditional RS models used nearest-neighbor and basic machine learning; advances in AI, especially with CNN, enable effective use of auxiliary data. This study reviews CNN-based RSs, focusing on data utilization, dataset comparisons, performance metrics, and open challenges and research opportunities [26].

The proposed ConvFM model used the basics of the Convolutional Neural Network (CNN). It is merging the extracted feature with Factorization Machines (FM). Using FM from the LibRecommender Library, ConvFM combines user input with hidden features from item metadata, boosting accuracy and addressing data sparsity. By extracting features through CNN and applying them in the FM framework, ConvFM achieves high prediction accuracy. Tests on the MovieLens dataset show ConvFM's superiority over traditional FM models based solely on historical data, effectively overcoming cold-start issues [27, 28].

Deep learning-based multimodal movie recommendation systems are more accurate than traditional methods. It has alleviated the problem of sparsity to a certain degree and recommends accurate personalized recommendations. Combining deep learning with multimodal data analysis further enhances recommendation performance by effectively leveraging diverse data types [29].

More accurately extracting the user-item features matrix using the CNN model. Data sparsity and new user and new item cold start issues with traditional methods. The proposed CNNTWSVM model removes these deficiencies by using the TWSVM algorithm with CNN.TWSVM efficiently balanced the data and more accurately recommended the K movies to the user.

3. Proposed CNNTWSVM Movie Recommendation System

The architecture of the proposed CNNTWSVM model integrates the CNN and TWSVM algorithms in this section. Using CNN, this model extracts features from the users ' item rating matrix from the 1 M MovieLens dataset. The pre-processed data are integrated with the TWSVM algorithm and used to train the model for the recommendation process. Finally, it predicts user movie ratings with greater accuracy, offering significant improvements over traditional algorithms.

3.1. Proposed CNNTWSVM Model Framework

The CNNTWSVM model is designed for pattern recognition and decision boundary properties of CNN &TWSVM. To develop this kind of system, let us break down the equations of each component. Figures 1 and 2 depict the CNN architecture and flowchart of the proposed CNNTWSVM model.

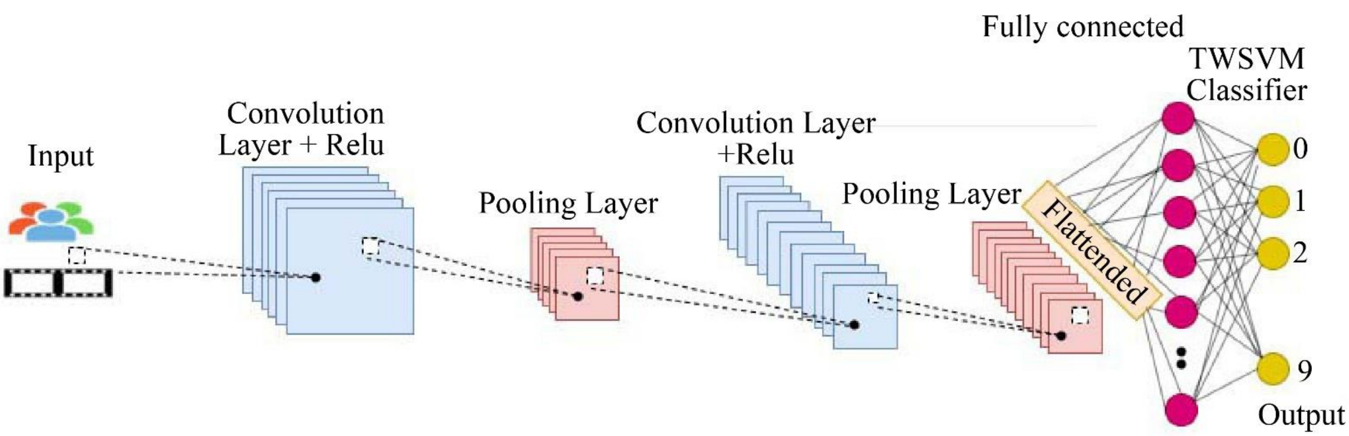


Fig. 1 CNN architecture for proposed CNNTWSVM model

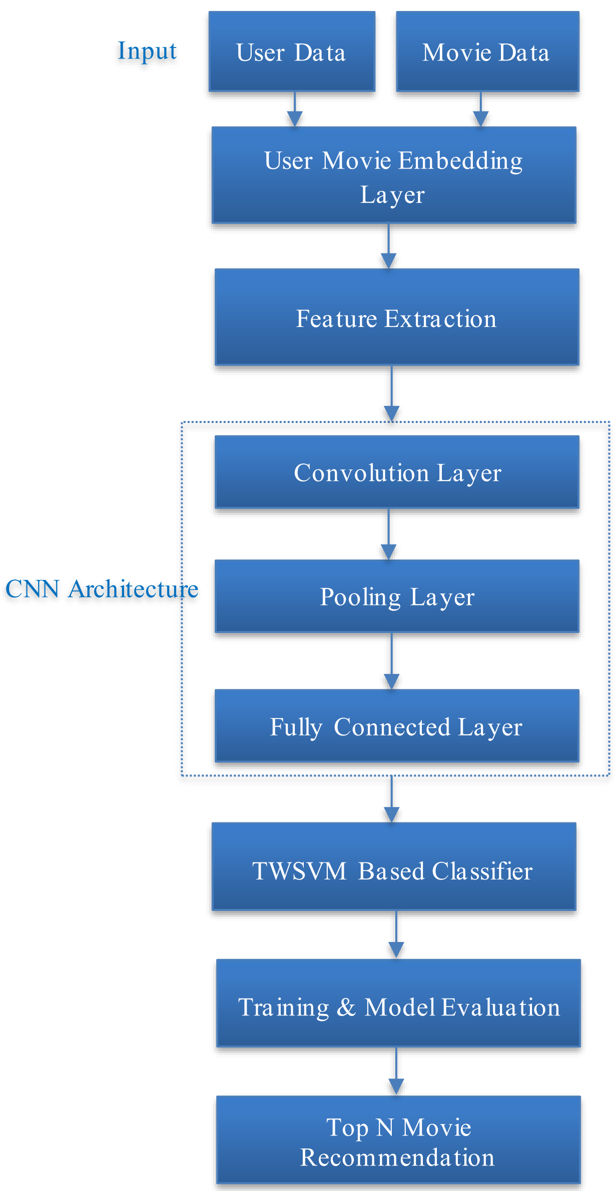


Fig. 2 Flowchart of CNNTWSVM movie recommendation system

Algorithm of CNNTWSVM Recommendation System

Input:

- User-Item data (e.g., user preferences, item features).
- Labels for recommendation (e.g., recommended/not recommended).

Output:

- Recommended items for users.

3.2. Step 1: Data Pre-Processing

3.2.1. Input Data

User and movie data as vectors with embedded features (genres, titles, user ID, movie name, movie ID, ratings).

3.3. Step 2: CNN-based Feature Extraction

The architecture involves four convolutional layers, two max-pooling layers, and a TWSVM classifier.

3.3.1. Initialize CNN Architecture

Set up the layers with a convolutional, pooling, and fully connected layer. This could included

Embedding Layers

- User Embedding: Embedding layer for user IDs, with an embedding size of 50- 100 dimensions.
- Movie Embedding: Embedding layer for movie IDs, also with an embedding size of 50-100 dimensions and using the ReLU activation function.

Convolution Layer Calculation

- Layer 1 and layer 2: 64 filters and 128 filters with a 3x3 kernel size.
- Layer 3 and layer 4: 256 filters with a 5x5 and 3x3 kernel size.

For each convolution layer l:

$$h^{l}_{i,j,k} = f\left(\sum_{m,n} W^{l}_{m,n,k} \cdot x^{l-1}_{i+m,j+n} + b^{l}_{k}\right) \quad (1)$$

Where $h^{l}_{i,j,k}$ Is map the feature at layer l, filters applied on input, $x^{l-1}_{i+m,j+n}$ is input data, and the activation function (ReLU) is f(.).

Pooling Layer Calculation

The pooling size is 2x2 to reduce dimensionality while retaining important features.

Reduce the spatial dimensions using a pooling operation:

$$h^{l}_{i,j,k} = \max_{m,n}(x^{l-1}_{i+m,j+n,k}) \quad (2)$$

Max-pooling helps reduce computational complexity.

Fully Connected Layer

The final pooling layer output is flattened into a 1D feature vector. After the feature maps are flattened, the features h are connected to a fully connected layer to produce feature vectors z:

$$z = W \cdot h + b \quad (3)$$

Where W represents the weight matrix and b represents the bias.

3.4. Step 3: TWSVM-based Classification

The flattened feature vector from the CNN serves as input to the TWSVM. Radial Basis Function (RBF) kernel, optimized for the extracted features. Alternative kernels (linear, polynomial) are tested during hyperparameter tuning. The TWSVM regularization parameter C are (C_1 and C_2) tuned using cross-validation, with values ranging between 0.1 and 100. Twin hyperplanes are optimized to maximize the margin between positive and negative classes for better classification.

3.4.1. Set Up Twin Hyperplanes

The TWSVM trains two classifiers:

First Hyperplane (for Class 1)

$$\min \frac{1}{2}||Aw_1 + e_1 b_1||^2 + C_1 e_2^T \xi \quad (4)$$

Subject to:

$$Bw_1 + e_2 b_1 \geq e_2 - \xi, \xi \geq 0$$

Second Hyperplane (for Class 2)

$$\min \frac{1}{2}||Bw_2 + e_2 b_2||^2 + C_2 e_1^T \eta \quad (5)$$

Subject to:

$$Aw_2 + e_1 b_2 \leq -(e_1 - \eta), \eta \geq 0$$

Where A & B are the feature matrix for classes 1,2, slack variable η, and w_1,b_1, w_2,b_2 are the hyperplane parameters.

3.4.2. Optimization

Solve the optimization problems to find the hyperplanes for both classes using quadratic programming techniques.

3.5. Step 4: Training & Prediction for Recommendation

3.5.1. Training

- The CNNTWSVM model is trained with the Adam optimizer.
- TWSVM is trained after CNN training is complete, using the CNN's extracted features.
- Optional joint fine-tuning: The CNN and TWSVM are jointly fine-tuned by backpropagating through both components.

3.5.2. Regularization

- A dropout rate of 0.5 is applied after convolutional layers to prevent overfitting.
- L2 regularization is used in the TWSVM for additional control over complexity.

3.5.3. Prediction

1. Feature Extraction for New User-Item Pairs: For a new user-item pair, extract the features using the trained CNN model.
2. Classification using TWSVM: Input the extracted feature vector into the trained TWSVM model. The model will predict the class (Recommended or Not Recommended) by determining which hyperplane the feature vector lies closest to.
3. Assign Recommendation: If the feature vector is closer to the hyperplane corresponding to the "Recommended" class, recommend the item to the user. Otherwise, do not recommend the item.

3.6. Step 5: Evaluation and Optimization

1. Model Evaluation: The CNNTWSVM model uses various evaluation metrics and parameters.
2. Parameter Tuning: Tune hyperparameters. Grid Search/Bayesian Optimization, such as Convolutional filter sizes (e.g., 3x3, 5x5), number of filters per layer, pooling strategies, and kernel parameters for TWSVM (RBF kernel width, regularization parameter C) are tuned using grid search or Bayesian optimization techniques.

The CNNTWSVM movie recommendation system uses the 1M MovieLens datasets. Features are extracted from the users' file and the movies' file. CNNs consist of three main parts: convolution layers to identify data features, pooling layers to condense data, and fully connected layers to combine inputs. With their many hidden layers, CNNs are highly effective at detecting complex patterns in data without needing a predefined mathematical model. The TWSVM can classify and predict user preferences. The CNNTWSVM model recommends new movies to users based on extracted features. It identifies similar features of movies and matches them to the user's preferences, recommending movies with features closest to the active user's.

The proposed CNNTWSVM model is used in embedding layers for both users and movies. User and movie IDs are embedded into vectors of 50 to 100 dimensions to represent each entity efficiently. Next, a sequence of convolutional layers processes the data: the 1^{st} & 2^{nd} layer have 64 &128 filters and a 3×3 kernel for both, followed by a third layer with 256 filters and a 5×5 kernel, all using ReLU activation. For deeper feature extraction, an optional fourth layer contains 256 filters and 3x3 kernels. After every two convolutional layers, max pooling layers (with a 2x2 pooling size) reduce dimensionality while retaining essential information. After these layers, the final output layer is flattened into a 1-D.

These 1-D vectors are fed into the TWSVM, which classifies user-movie pairs as "Recommended" or "Not Recommended." The TWSVM uses an RBF (radial basis function) kernel, optimized to suit the CNN-extracted features, though other kernel types (linear, polynomial) can be tested. Regularization parameters, C1 and C2, are tuned via cross-validation, typically within a range of 0.1 to 100, to enhance performance. Twin margins are calculated to create two hyperplanes, maximizing the margin between the classes for accurate classification.

The training strategy for this architecture involves a trained CNN-TWSVM model with a learning rate of 0.001 and an adam optimizer. If validation loss is maximum, the learning rate is decreased. Once CNN training is complete, the TWSVM is trained using the CNN-generated feature vectors. Optionally, both components can be fine-tuned jointly for further optimization. Dropout (at a rate of 0.5) is applied to mitigate overfitting, and L2 regularization is incorporated within the TWSVM to control complexity. Finally, grid search or Bayesian optimization techniques are used for tuning CNN parameters (such as filter sizes and layer counts) and TWSVM kernel parameters, maximizing the architecture's performance.

4. Experiment & Results

The user-movie rating matrix of MovieLens is used to evaluate the performance of our proposed CNNTWSVM model. We compared our proposed CNNTWSVM movie recommendation system with various traditional models of movie recommendation systems.

4.1. Dataset

The proposed CNNTWSVM movie recommendation system experimental setup involves using CNN to extract features from movie textual data, training TWSVM to classify these features according to user preferences, training the model and then recommending the top N movies to users. The performance is evaluated using various metrics to ensure the recommendations are accurate and relevant.

The real-world MovieLens [30] 1M datasets are used for performance evaluation of the proposed CNNTWSVM movie recommendation system. We are using MovieLens datasets, which contain users' interests and preferences for movies. These preferences are represented as ratings, where users provide ratings for movies on a scale of 0-5 stars that they have already watched. The 1M MovieLens dataset contains 6040 users and 3883 movies, as shown in Table 1. Many movies are not rated in these datasets. The data sparsity for the 1M datasets is 95.359%. The labels marked 1 indicate the interaction of users and the movies.

Table 1. Description of dataset

Dataset	Parameters	Value
MovieLens 1 M	Users	6040
	Movies	3883
	Ratings	1000209

4.2. Experiment Setting

The proposed CNNTWSVM model performance is evaluated with Matrix Factorization (MF), autoencoder (AutoRec), Neural Collaborative Filtering (NCF), and standard CNN models. The various evaluation metrics are compared in Table 2.

Set up of the proposed CNNTWSVM model. The train and test method is used for the evaluation of various recommendation systems. The 1M MovieLens dataset is split into a ratio of 8:2 of training and testing sets. We implemented the CNNTWSVM model using MATLAB R2014a on a system with 8 GB of RAM, an Intel® Core i3-4005U CPU @ 1.70 GHz, and running Windows 10 Pro.

4.3. Results

In the proposed CNNTWSVM model, the MovieLens 1M dataset is used. CNN is extracting the fea ! The M dataset and TWSVM classify these features according to user preferences. The CNNTWSVM movie recommendation system's performance compared to other models across key evaluation metrics . With the lowest RMSE (0.805) and MAE (0.635), CNNTWSVM minimizes prediction errors effectively. It also achieves the highest values for accuracy (95.2%), precision (97.5%), recall (89.0%), and NDCG (89.0%), indicating its exceptional ability to provide accurate and relevant recommendations while ranking items effectively. Compared to baseline models, the proposed CNNTWSVM model consistently outperforms, showcasing its strength in predictive accuracy and recommendation quality. These results affirm our proposed CNNTWSVM movie recommendation system as a traditional method in movie recommendation systems.

Table 2. Performance analysis of baseline method & proposed method

Model	RMSE ↓	MAE ↓	Accuracy (%)	Precision (%)	Recall (%)	NDCG (%)	Diversity (%)	Serendipity (%)
Matrix Factorization (MF)	0.850	0.680	90.7	92.0	85.0	85.0	92	88
Auto encoder (AutoRec)	0.835	0.660	92.5	94.0	86.5	87.0	93.4	89.7
Neural Collaborative Filtering (NCF)	0.825	0.655	94.0	96.0	88.0	88.0	93.8	90.5
CNN (Standard)	0.810	0.640	94.7	97.0	88.5	88.5	94.1	92.1
CNNTWSVM (Proposed Model)	0.805	0.635	95.2	97.5	89.0	89.0	94.7	93.2

Various recommendation model errors are graphically depicted in Figure 3. Lowering the values of RMSE and MAE means better accuracy. Figure 4 shows that the proposed CNNTWSVM model achieves the best results, with the lowest RMSE (0.805) and MAE (0.635), highlighting its superior predictive performance. The proposed CNNTWSVM model recommends relevant movies and also introduces users to a wide array of lesser-known yet appealing movies.

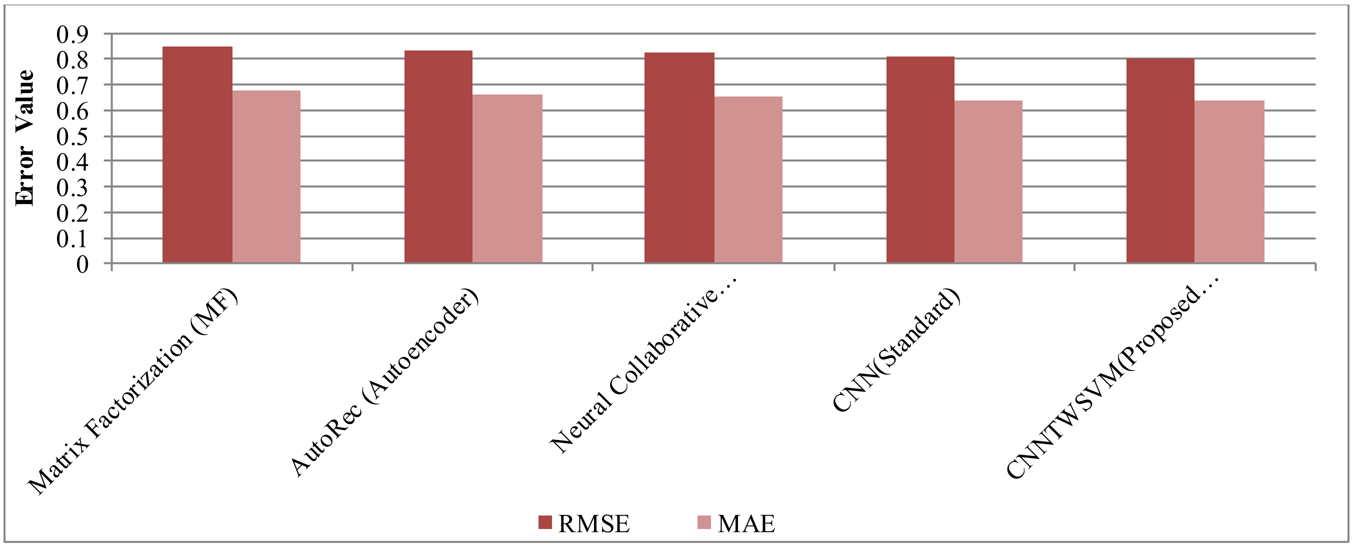
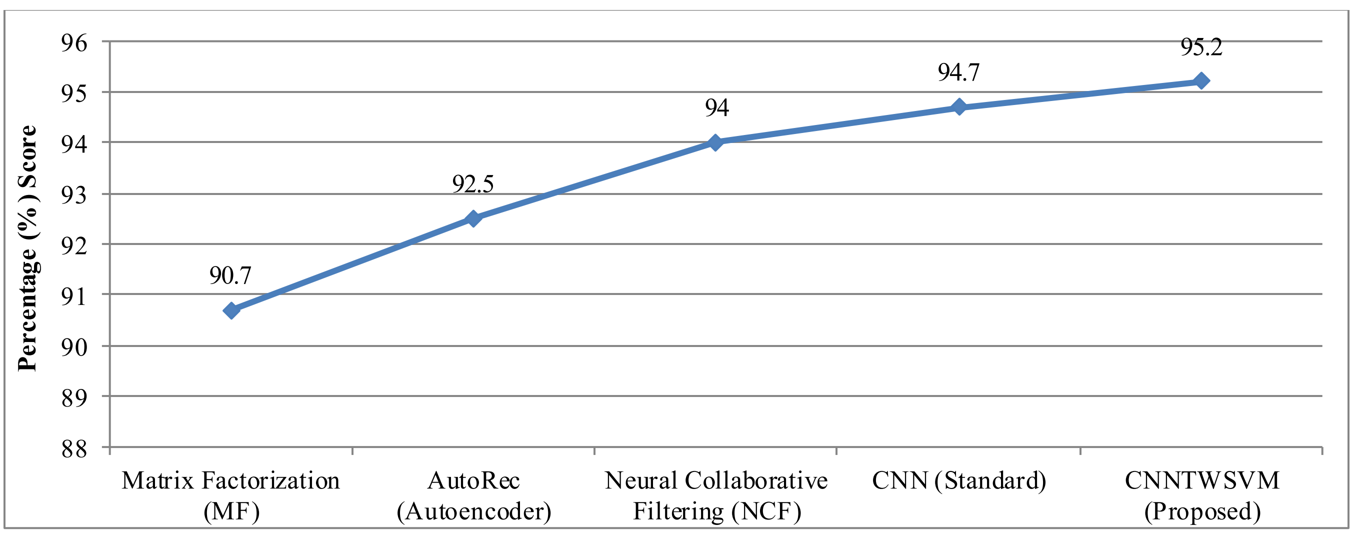
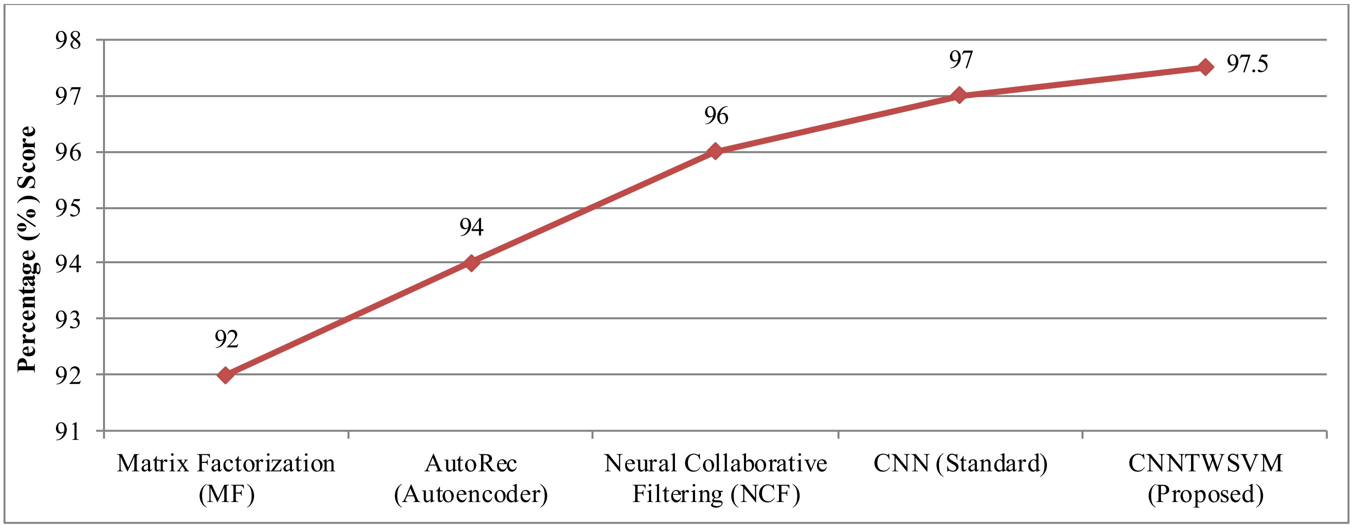


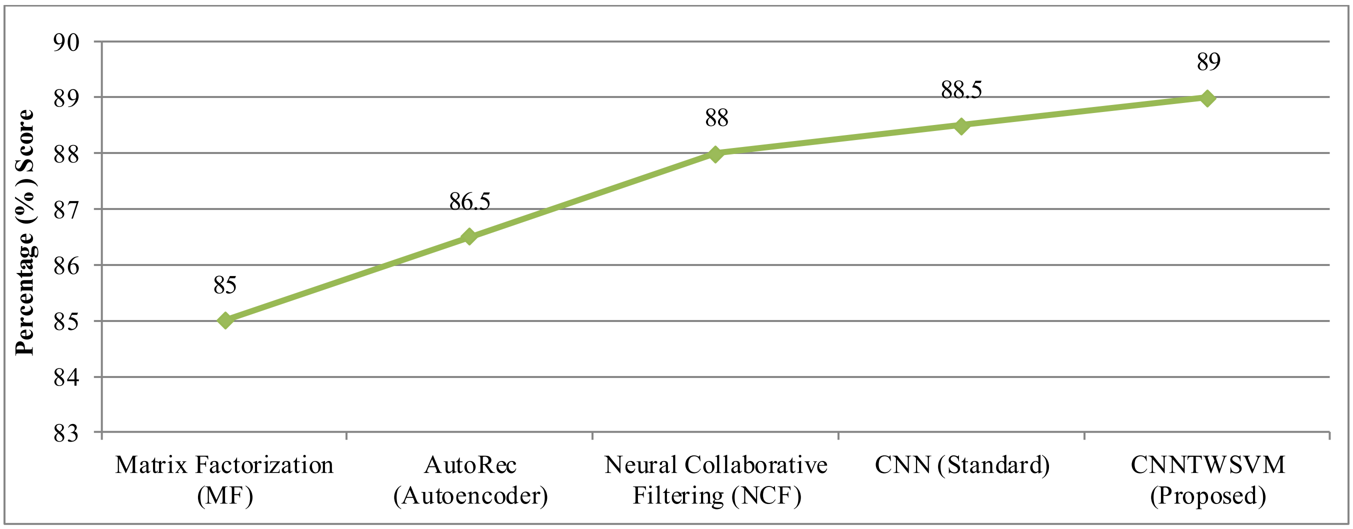
Fig. 3 Error score of various models of the Movie Recommendation Model



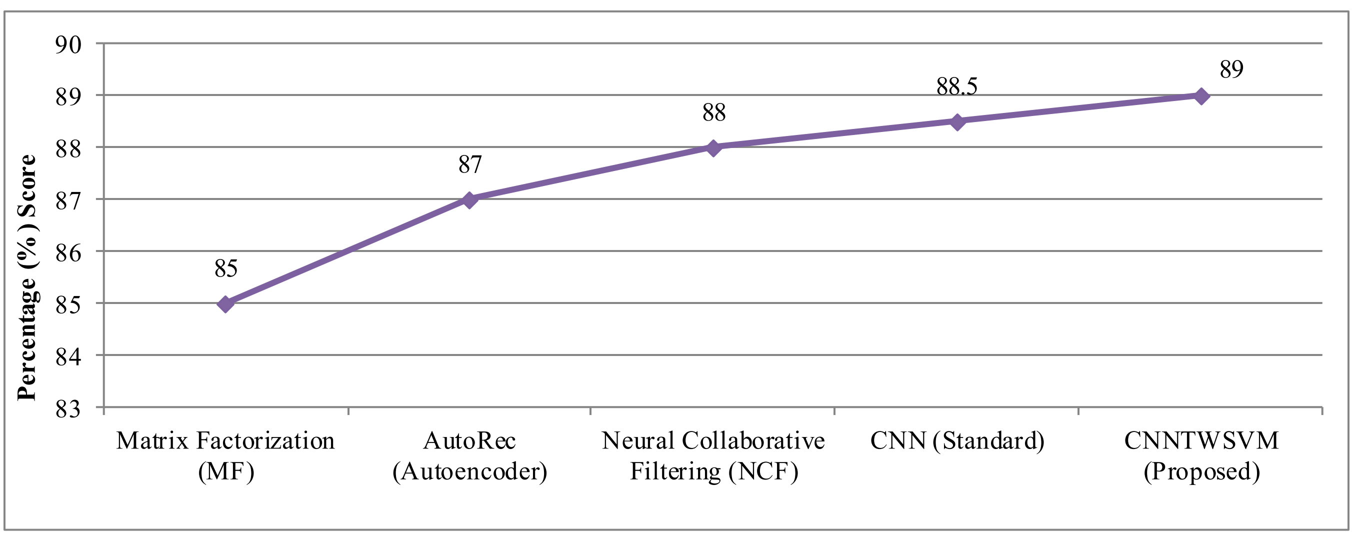
(a) Accuracy (%)



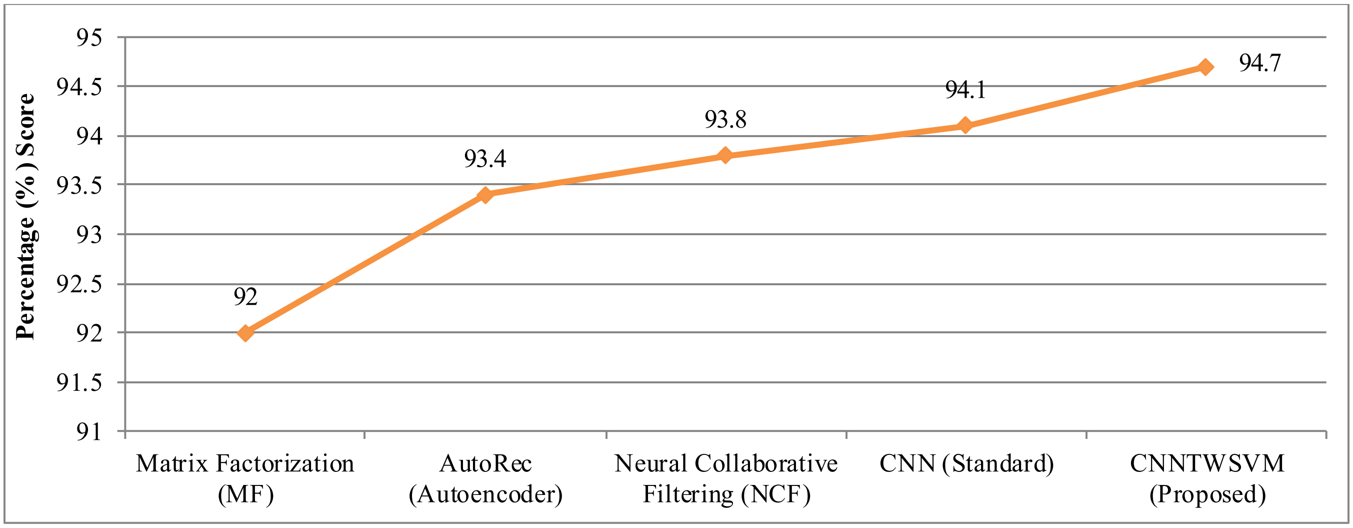
(b) Precision (%)



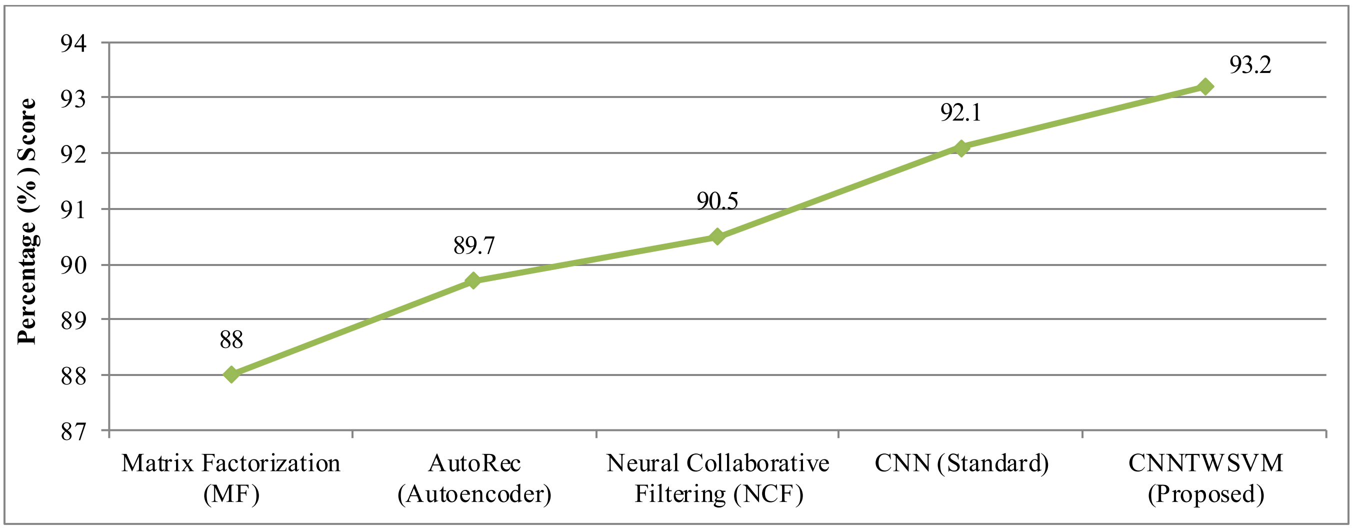
(c) Recall (%)



(d) Normalized discontinued cumulative gain (%)



(e) Diversity (%)



(f) Serendipity (%)

Fig. 4 Various movie recommendation systems' performances on various parameters

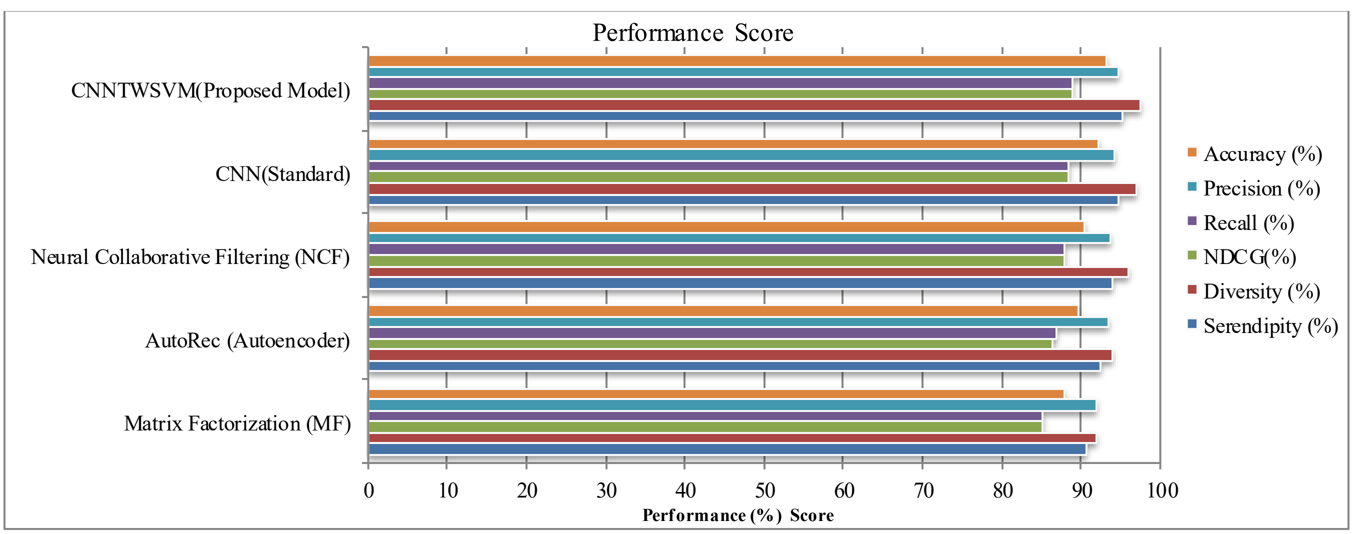


Fig. 5 Comprehensive result of different techniques of the movie recommendation system

The overall performance of the proposed CNNTWSVM model and conventional methods is depicted in Figure 5. When compared to conventional methods like matrix factorization, auto encoders, neural collaborative filtering, and standard CNN models, the proposed CNNTWSVM model overall performs well compared to others. Figure 6 depicts the comparison of the performance of accuracy versus epochs and RMSE versus epochs for the proposed CNNTWSVM model with the baseline model. The proposed CNNTWSVM model has the highest accuracy at 50 epochs as compared to other baseline models. At the same time, the lowest RMSE indicates that it converges more quickly than the others.

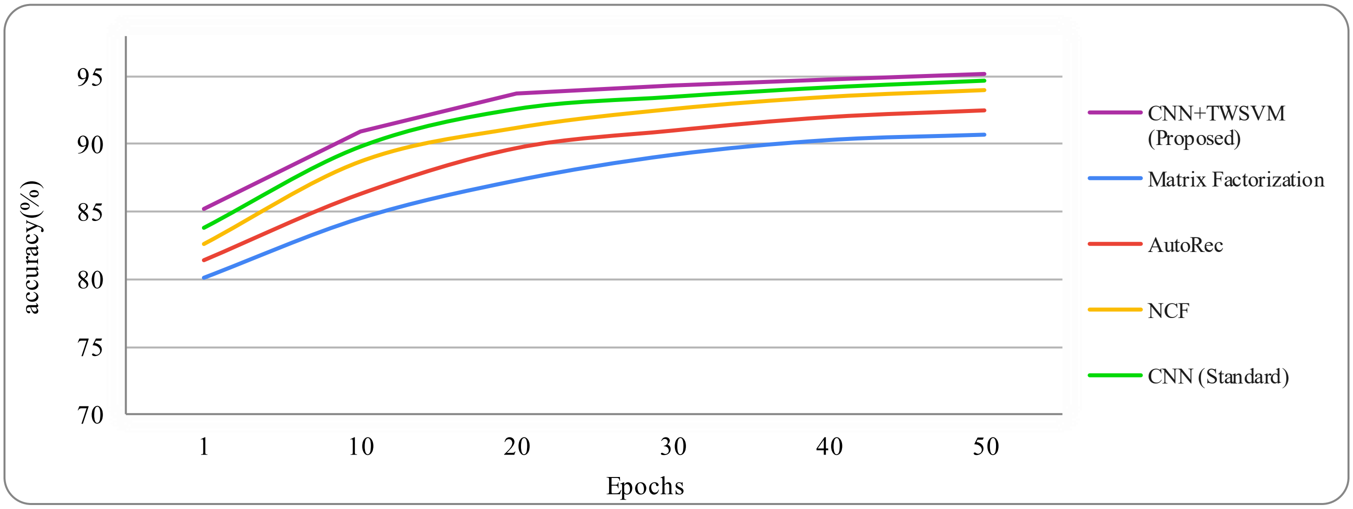


Fig. 6(a) Accuracy versus Epochs

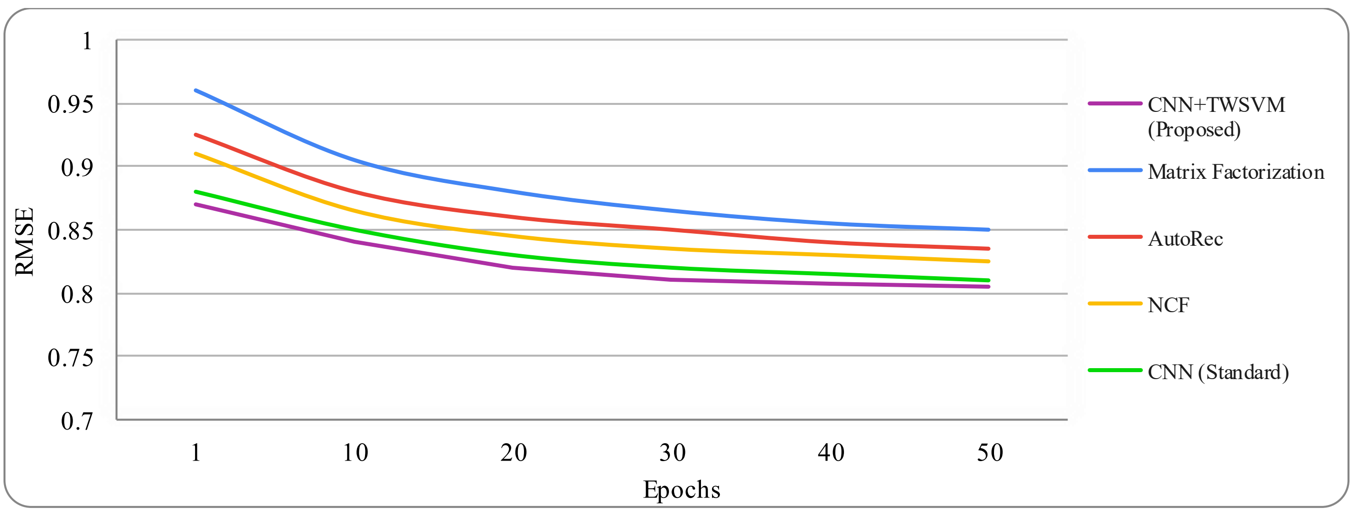


Fig. 6(b) RMSE versus Epochs

5. Conclusion

In the era of digitalization, the main challenge is information overloading in people's daily lives. Recommendation systems are important for assisting users and resolving the problem of information overload. It efficiently filters the prioritizing and recommending relevant information for enabling personalized movies and services for users. In the past decade, the use of personalized recommendation systems has grown significantly in many fields, such as music, movies, articles, and books. A movie recommendation system is a fascinating application. Collaborative filtering is one of the traditional methods that have failed to cover the hidden features from input data, so accurate recommendations are not possible. The proposed CNNTWSVM movie recommendation system incorporates the traditional method of recommendation systems with advanced machine learning models for personalized movie recommendations. CNN (Convolutional Neural Networks) extracts features from input data, such as movie features and user features, and embeds them into utility matrices. The system was trained and tested using the 1M MovieLens datasets. Twin Support Vector Machines (TWSVM) handle the complex decision boundaries with less computational cost and improve classification performance. This system achieved efficient and accurate recommendations according to the user's preferences. The result shows the CNNTWSVM model performs well compared to other traditional algorithms of recommendation systems. It resolves the issues of sparsity and the cold start problem of conventional recommendation systems. The CNNTWSVM model recommends relevant and niche movies to users. The system is adaptable to diverse datasets and the dynamic preferences of users, which highlights its versatility. In conclusion, the CNNTWSVM movie recommendation system shows significant advancement in movie recommendation systems. It improves the user's satisfaction and accuracy of the recommender system and recommends the relevant movies to users. In the future, the recommendation system will integrate contextual data, such as mood or viewing environments, and work out real-time updates.

Acknowledgments

Nisha Bhalse: original draft of research paper, analysis and implementation. Ramesh Thakur: Conceptualization, Analysis and editing of the paper. Archana Thakur: Review and Conceptualization.

References

[1] Yu-Chen Chen, Rong-An Shang, and Chen-Yu Kao, "The Effects of Information Overload on Consumers' Subjective State Towards Buying Decision in the Internet Shopping Environment," *Electronic Commerce Research and Applications*, vol. 8, no. 1, pp. 48-58, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[2] F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh, "Recommendation Systems: Principles, Methods and Evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261-273, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[3] Paul Resnick, and Hal R. Varian, "Recommender Systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[4] Frank Kane, *Building Recommender Systems with Machine Learning and AI*, 2nd ed., Sundog Education, 2018. [Google Scholar] [Publisher Link]

[5] Charu C. Aggarwal, *Recommender Systems: The Textbook*, 1st ed., Springer Cham, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[6] Francesco Ricci, Lior Rokach, and Bracha Shapira, *Recommender Systems: Techniques, Applications, and Challenges,* Recommender Systems Handbook, Springer New York, pp. 1-35, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] G. Adomavicius, and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[8] Sahraoui Dhelim et al., "A Survey on Personality-Aware Recommendation Systems," *Artificial Intelligence Review*, vol. 55, no. 3, pp. 2409-2454, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Urvish Thakker, Ruhi Patel, and Manan Shah, "A Comprehensive Analysis on Movie Recommendation System Employing Collaborative Filtering," *Multimedia Tools and Applications*, vol. 80, no. 19, pp. 28647-28672, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jeffrey Lund, and Yiu-Kai Ng, "Movie Recommendations Using the Deep Learning Approach," *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, Salt Lake City, UT, USA, pp. 47-54, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[11] Snehal R. Chavare, Chetan J. Awati, and Suresh K. Shirgave, "Smart Recommender System Using Deep Learning," *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, pp. 590-594, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Richard Blake, and Katarina Frajtova Michalikova, "Deep Learning-Based Sensing Technologies, Artificial Intelligence-Based Decision-Making Algorithms, and Big Geospatial Data Analytics in Cognitive Internet of Things," *Analysis and Metaphysics*, no. 20, pp. 159-173, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] Laith Alzubaidi et al., "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, vol. 8, no. 1, pp. 1-74, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Divya Tomar, and Sonali Agarwal, "Twin Support Vector Machine: A Review From 2007 to 2014" *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 55-69, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[15] M. Tanveer et al., "Comprehensive Review on Twin Support Vector Machines," *Annals of Operations Research*, vol. 339, no. 3, pp. 1223-1268, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Zhongtai Qin, and Mingjun Zhang, "Towards a Personalized Movie Recommendation System: A Deep Learning Approach," *2021 2nd International Conference on Artificial Intelligence and Information Systems (ICAIIS'21)*, Chongqing, China, pp. 1-5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Stefan Siersdorfer, Jose San Pedro, and Mark Sanderson, "Automatic Video Tagging Using Content Redundancy," *SIGIR '09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, vol. 22, pp. 395-402, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[18] Trang Trinh et al., "Product Collaborative Filtering based Recommendation Systems for Large-Scale E-Commerce," *International Journal of Information Management Data Insights*, vol. 5, no. 1, pp. 1-11, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nisha Sharma, and Mala Dutta, "An Ensemble Movie Recommender System Based on Stacking," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 18, pp. 7264-7273, 2023. [Google Scholar] [Publisher Link]
[20] Wei Fang et al., "Movie Recommendation Algorithm Based on Ensemble Learning," *Intelligent Automation and Soft Computing*, vol. 34, no. 1, pp. 609-622, 2022. [CrossRef] [Google Scholar] [Publisher Link]
[21] Umair Ali Khan et al., "Movie Tags Prediction and Segmentation Using Deep Learning," *IEEE Access*, vol. 8, pp. 6071-6086, 2020. [CrossRef] [Google Scholar] [Publisher Link]
[22] Jingdong Liu, Won-Ho Choi, and Jun Liu, "[Retracted] Personalized Movie Recommendation Method Based on Deep Learning," *Mathematical Problems in Engineering*, vol. 2021, pp. 1-12, 2021. [CrossRef] [Google Scholar] [Publisher Link]
[23] Feng Zhang et al., "Privacy-Aware Smart City: A Case Study in Collaborative Filtering Recommender Systems," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 145-159, 2019. [CrossRef] [Google Scholar] [Publisher Link]
[24] Jian Wei et al., "Collaborative Filtering and Deep Learning Based Recommendation System for Cold Start Items," *Expert Systems with Applications*, vol. 69, pp. 29-39, 2017. [CrossRef] [Google Scholar] [Publisher Link]
[25] Yubing Yan et al., "Transforming Movie Recommendations with Advanced Machine Learning: A Study of NMF, SVD, and K-Means Clustering" *2024 4th International Symposium on Computer Technology and Information Science (ISCTIS)*, Xi'an, China, pp. 178-181, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[26] M. Moeini, A. Broumandnia, and M. Moradi, "A Personalized Web Recommendation System Based on a Weighted user Behavior Profile by Applying Extended Learning Method," *International Journal of Engineering*, vol. 38, no. 4, pp. 871-893, 2025. [CrossRef] [Google Scholar] [Publisher Link]
[27] Paul Covington, Jay Adams, and Emre Sargin, "Deep Neural Networks for YouTube Recommendations," *RecSys '16: Proceedings of the 10th ACM Conference on Recommender Systems*, Boston, Massachusetts, USA, pp. 191-198, 2016. [CrossRef] [Google Scholar] [Publisher Link]
[28] Ronakkumar Patel, Priyank Thakkar, and Vijay Ukani, "CNNRec: Convolutional Neural Network Based Recommender Systems - A Survey," *Engineering Applications of Artificial Intelligence*, vol. 133, 2024. [CrossRef] [Google Scholar] [Publisher Link]
[29] Yongheng Mu, and Yun Wu, "Multimodal Movie Recommendation System Using Deep Learning," *Mathematics*, vol. 11, no. 4, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]
[30] *MovieLens 1M Dataset*, Grouplens, 2015. [Publisher Link]