BenchCouncil Transactions on Benchmarks, Standards and Evaluations 4 (2024) 100200

Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-onbenchmarks-standards-and-evaluations/

Full Length Article

# Advanced Deep Learning Models for Improving Movie Rating Predictions: A Benchmarking Study

Manisha Valera [a], Dr. Rahul Mehta [b,*]

[a] Research Scholar, Gujarat Technological University, Ahmedabad, India
[b] Department of Electronics & Communication Engineering, GEC Rajkot, Gujarat, India

ARTICLE INFO

*Keywords:*
Variational autoencoder
Convolutional neural networks (CNN)
Recurrent neural networks (RNN)
Long short-term memory (LSTM)
gated recurrent unit (GRU)
Distilbert

ABSTRACT

Predicting movie ratings very precisely has become a vital aspect of personalized recommendation systems, which requires robust and high-performing models. for evaluating the effectiveness in predicting movie ratings, this study conducts a comprehensive performance analysis of various deep learning architectures, which includes BiLSTM, CNN + LSTM, CNN + GRU, CNN + Attention, CNN, VAE, Simple RNN, GRU + Attention, Transformer Encoder, FNN and ResNet. Here each model's performance is evaluated on movie reviews' dataset, enhanced with sentiment scores and user ratings, by using a range of evaluation metrics: Mean Absolute Error (MAE), $R^2$ score, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Explained Variance. Here the results highlight distinct strengths and weaknesses among the models, in which VAE model consistently delivering superior accuracy, whereas attention-based models prove prominent improvements in interpretability and generalization. This analysis offers important insights into choosing models for movie recommendation systems, which also highlights the balance between prediction accuracy and computational efficiency. The discoveries from this study serve as a benchmark for future developments in movie rating prediction, supporting the researchers and practitioners in augmenting recommendation system performance.

## 1. Introduction

Despite noteworthy advancements in the field of recommendation systems, existing studies in this field still leave several important limitations unaddressed. Though traditional collaborative filtering methods are foundational, they frequently encounter challenges e.g., data sparsity and cold-start problems, which are particularly challenging to manage in movie recommendation systems. as observed in earlier research, models based purely on matrix factorization or basic recurrent architectures often struggle to capture the complex temporal patterns and emotional nuances that significantly influence user preferences. More advanced models which are using GRU and attention mechanisms, like those by Xia et al. [1] and Wang et al. [5], have improved in addressing these issues while considering time-based patterns. They still lack a full integration of sentiment analysis, which is really important for understanding user sentiments and likings toward movies.

In addition, multi-modal approaches that comprise data sources like movie posters and plot summaries, as demonstrated by Xia et al. [2] provide all-inclusive view of content preferences but lack robust sentiment-based personalization, which is crucial for the domains where emotional engagement is critical. Variational Autoencoders (VAEs), which are used effectively for collaborative filtering by Askari et al. [3] and Liang et al. [6], offer another trail for grasping hidden patterns in user interactions. However, these models tend to focus more on interaction data rather than user sentiment, possibly overlooking key insights that could improve recommendation accuracy and relevance. Furthermore, while sentiment-enhanced hybrid models have started to bridge this gap, as in Dang et al. [8], their incorporation remains limited, and the models face challenges in scalability and computational cost.

Existing literature, including Siet et al. [7], explores various architectures like CNNs, RNNs, and clustering-based methods, but lacks a comprehensive comparison across these models, limiting our understanding of their relative performance under a unified framework. Few studies provide a thorough evaluation of these models based on standardized error metrics, making it difficult to determine which approach consistently outperforms the others in terms of robustness, accuracy and recommendation quality.

To address these kinds of gaps, this paper provides a comprehensive comparison of state-of-the-art deep learning models for movie recommendation, including BiLSTM, CNN + LSTM, CNN + GRU, CNN +

* Corresponding author.
*E-mail address:* Rdmehta@hotmail.com (Dr.R. Mehta).

https://doi.org/10.1016/j.tbench.2025.100200
Received 10 December 2024; Received in revised form 23 March 2025; Accepted 11 April 2025
Available online 18 April 2025

Attention, CNN, VAE, Simple RNN, GRU + Attention, Transformer Encoder, FNN and ResNet. Exclusively, this work integrates sentiment analysis to enhance the models' ability to account for user emotions, adding a layer of personalization that prior models lacked. By evaluating each model on standardized error metrics—such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE)— this study aims to identify the most effective model for delivering accurate and sentiment-aware recommendations. Through this rigorous approach, our work sets a benchmark for future advancements in movie recommendation systems by emphasizing the impact of sentiment-driven personalization on recommendation quality.

## 2. Preliminaries

This paper presents a collaborative filtering recommendation algorithm [1] which integrates attention mechanisms within a Gated Recurrent Unit (GRU) framework and employs adversarial learning techniques. Here, the proposed model's aim is to enhance user-item interactions that focus on important features and reduce the noise from irrelevant data. In this, the results prove the enhanced performance in the context of recommendation accuracy compared to traditional methods, showcasing the effectiveness of the attention mechanism and adversarial learning in capturing user preferences.

This study proposes a multi-modal transformer framework [2] which leverages both textual and visual features from movie posters which are used to enhance the recommendation performance. In this, by employing an attention mechanism, the model's concentration is on prominent features from the posters while integrating them with textual data that is obtained from movie descriptions. The experimental results here illustrate that the proposed approach implicitly outperforms existing methods, mostly in scenarios where visual data plays a vital role in user preference prediction.

It explores the application of Variational Autoencoders (VAEs) [3] in the context of top-K recommendation systems using implicit feedback. In this the authors introduce an innovative VAE architecture that successfully models user preferences and item characteristics while also addressing challenges associated with implicit feedback, such as data sparsity. The experiments disclose that the proposed VAE-based method attains competitive results in top-K recommendation tasks, demonstrating its capability to generalize well to the hidden data.

This survey [4] provides an all-inclusive overview of various deep learning models, which includes Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Gated Recurrent Units (GRUs). In this the authors discuss the strengths and weaknesses of individual models, their applications in different domains, and comparative performance metrics. This paper contributes as a valuable resource for researchers and practitioners, for those who seek to understand the landscape of deep learning architectures.

In this work, the authors propose a personalized movie recommendation system [5] that combines LSTM and CNN architectures to capture both sequential and contextual features from user interactions. The model is designed here to extract temporal patterns in user behavior while considering the content of movies, as well. Experimental results indicate that the proposed system outperforms traditional collaborative filtering methods, particularly in capturing user preferences over time.

This paper investigates the use of Variational Autoencoders (VAEs) [6] for collaborative filtering tasks. The authors proposed a model here that collectively learns user and item representations while incorporating uncertainty into the recommendations. The VAE framework here addresses challenges as well, such as sparsity and cold-start problems in collaborative filtering, resulting in the enhanced recommendation accuracy. The discoveries suggest that VAEs can efficiently model user-item interactions in collaborative filtering scenarios.

This study focuses on improving movie recommendation systems [7] by integrating deep learning techniques and KMeans clustering. Here, the authors develop a sequence-based recommendation model that captures user preferences over time and apply KMeans to group similar users. The results demonstrate that the hybrid approach which yields superior recommendation performance compared to traditional methods, mostly in dealing with sequential data.

This paper explores the integration of sentiment analysis into recommender systems [8] which is used to enhance user experience. The authors propose a deep learning framework that incorporates sentiment scores from user reviews to refine recommendations, particularly for items with varying emotional tones. Here, the experiments show that incorporating sentiment analysis leads to more personalized and appropriate recommendations, highlighting the importance of emotional context in user preferences.

This work presents an approach [9] to improve movie recommendation systems by leveraging deep learning models alongside sentiment analysis. The authors demonstrate that incorporating sentiment data from user reviews significantly enhances the accuracy and relevance of recommendations. The findings underscore the potential of combining different data sources to create more effective recommendation algorithms. The SVM and CNN algorithms were implemented for the Movie Recommendation System to recommend the most relevant films for a given movie. Even after extensive testing, CNN classifier has produced decent findings in terms of suggesting the films.

This paper proposes the GANCF model [10], which combines user and item latent vectors with auxiliary information to enhance recommendation performance through deep non-linear learning. Here, experimental results show better outcomes on two datasets, validating the benefit of auxiliary data. Future work will discover time-based mechanisms and integrate multi-source heterogeneous data for better capturing dynamic user interests.

This study addresses the cold start [11] problem in recommendation systems and proposes a deep learning approach that builds user profiles from demographic attributes. Here, a modified ANN model clusters users by demographics, which is used to provide personalized movie recommendations. It also demonstrates strong performance across multiple evaluation metrics.

By human brain function, Convolutional Neural Networks (CNNs) are inspired [12,13] and They effectively handle grid-like structured data, too [14]. The CNN architecture has 3 types of dimensions: 1D is for processing text and signals, 2D is for images and audio, whereas 3D is for videos. While CNNs are chiefly used in computer vision tasks, e.g., image classification [15], they also perform well in text classification using word vectors formed through concatenation [16].

Google's TensorFlow framework [17], developed for machine learning, underlines tensors, which generalize vectors and matrices for managing flexible dimensions. This study leverages the Keras TensorFlow library which is mostly used to build CNN models with layers via including input, convolution, max-pooling, flatten, dense, dropout, and output [18], here each layer processes input data through the network [19]. Given the one-dimensional nature of text data, a 1D convolutional layer is used for the same, which is letting the model to extract composite patterns in the data [20].

To prevent overfitting, a dropout rate of 0.5 is applied after each Conv1D layer, deactivating a portion of neurons [21]. To reduce dimensionality, MaxPooling1D takes the maximum value in each pooling window. To further minimize the risk of overfitting, another dropout layer is added afterward MaxPooling. Then, a flattening layer converts the feature matrix into a vector, followed by another dropout at 0.5. Here, a dense layer with 64 units and ReLU activation processes this vector. The final dense layer with 3 units and sigmoid activation produces class probabilities. CNN was chosen for its aptitude to achieve greater accuracy and it effectively recognizes patterns, even in rating data.

The content-based recommendation system [22] is developed using CNN, which is combining TF-IDF and RoBERTa for pattern recognition in movie review data obtained from Twitter. This augmented CNN

model with SMOTE and an SGD optimizer achieved an accuracy of 86.41 %, efficaciously providing accurate movie recommendations.

DistilBERT is a reorganized version of BERT, achieving a 40 % reduction in size and a 60 % increase in speed while conserving 97 % of BERT's language comprehension abilities [23]. It is trained through distillation, it's highly efficient and well-suited for edge deployments. Summarization can be extractive (selecting key sentences) or abstractive (rephrasing content). This study assesses BERT-based models and it introduces "SqueezeBERTSum," [24] which is a streamlined summarization model that retains 98 % of BERTSum's performance with 49 % fewer parameters. ArDBertSum, an Arabic text summarization model based on fine-tuned DistilBERT [25], enhanced with the SCSAR technique for sentence segmentation. It is evaluated on the EASC corpus, it beats other Arabic summarizers, and the future work will focus on expanding datasets, filtering evaluation methods, and discovering other pre-trained models.

This paper offers a widespread comparison of cutting-edge deep learning models for movie recommendation, encompassing BiLSTM, CNN + LSTM, CNN + GRU, CNN + Attention, CNN, VAE, Simple RNN, GRU + Attention, Transformer Encoder, FNN, and ResNet.

### 2.1. BiLSTM (Bidirectional long short-term memory)

BiLSTM networks are a variant of LSTMs [26] which process data in both directions - forward and backward, making them particularly effective in capturing long-range dependencies within sequences. In recommendation systems, BiLSTM is used to understand the consecutive patterns in user interactions (e.g., movie-watching behavior) over time. This bidirectional approach is helpful in capturing the complete context of user preferences, mainly for the tasks such as sequential recommendation.

### 2.2. CNN + LSTM

The CNN + LSTM model attaches Convolutional Neural Networks (CNNs) with LSTMs [5] which is allowing the system to handle both spatial and sequential data efficiently. CNNs are typically applied first to capture features from movie details (e.g., visual or textual features), followed by LSTMs to understand the chronological dependencies in user interaction data. This combination is powerful for multimedia recommendation systems where both temporal dynamics and content features (such as movie genres or user reviews) impact the movie recommendations.

### 2.3. CNN + GRU

This architecture pairs CNNs with Gated Recurrent Units (GRUs), where CNNs capture spatial features from input data, whereas GRUs manage sequential dependencies. The GRU, a simplified version of LSTM [10], combines the forget and input gates into an update gate and merges cell and hidden states. This kind of design allows it to capture long-term dependencies effectively while reducing issues like gradient vanishing and explosion. Here, GRUs are computationally more efficient than LSTMs, as they contain fewer parameters. The CNN + GRU model is thus appreciated for the cases where movie recommendation systems need to balance temporal insights with efficient processing of rich content data, like user comments / reviews.

### 2.4. CNN + attention

In this model, CNNs are coupled with an attention mechanism, which is used to prioritize important features in the data. First, CNNs extract core features, which are then weighted by the attention mechanism, allowing the model to focus on the utmost pertinent information. For movie recommendations, CNN + Attention mechanism can highlight detailed aspects of user preferences, such as genre or specific movie features, which is used here to provide more relevant suggestions based on past communications.

### 2.5. CNN (Convolutional neural network)

Originally developed for image processing, CNNs are proficient at recognizing spatial hierarchies in data. In context of movie recommendation systems, CNNs [27] are used for feature extraction from text-based reviews/ user comments, or visual features related to movie posters. Though CNNs do not inherently capture sequential information, they provide valuable acumens into content-related features that impact recommendations.

### 2.6. VAE (Variational autoencoder)

VAEs are probabilistic models, which are designed for dimensionality reduction and data generation. They use latent variable representations, which are particularly useful for collaborative filtering because they can capture the hidden factors which are driving user preferences. In movie recommendation systems, VAEs allow for the modeling of complex, implicit feedback, creating robust representations of user preferences that adapt well to various types of recommendation tasks.

### 2.7. Simple RNN (Recurrent neural network)

RNNs [4] are a foundational architecture which is used for sequential data processing, where each node's output is fed as input to the next node. While they are effective for short sequences, RNNs are prone to matters like vanishing gradients, which is limiting their ability to capture long-term dependencies. In context of recommendation task, Simple RNNs can offer elementary insights into sequential patterns. still, they are typically lacking in efficiency than more advanced recurrent models like GRUs or LSTMs.

### 2.8. GRU + attention

This model combines GRUs [4] with an attention layer because they want to prioritize significant sequential data. GRUs efficiently handle sequential dependencies as well, while the attention layer boosts interpretability by highlighting the most influential user interactions or content features. This kind of combination is ideal for recommendation systems that require efficient temporal modeling and the ability to focus on key preferences in the user's viewing history as well.

### 2.9. Transformer encoder

This is an element of the Transformer architecture that relies solely on self-attention mechanisms by discarding recurrence entirely. This kind of architecture allows parallel processing of input data, making it efficient and extremely scalable. Transformer Encoders are particularly effective in capturing complex dependencies in user interactions over time, making them suitable for large-scale recommendation systems and these systems need to analyze diverse content features simultaneously.

### 2.10. FNN (Feedforward neural network)

Feedforward Neural Networks are simple neural networks containing fully connected layers, mainly used for classification or regression tasks. In recommendation systems, FNNs can be beneficial for basic collaborative filtering tasks or as supplementary layers in hybrid models, though they lack the sequential or hierarchical structure required for complex, multi-faceted recommendation tasks.

### 2.11. ResNet (Residual neural network)

ResNet is DNN model which is known for its residual connections. it

helps to mitigate issues like vanishing gradients in very deep networks. In recommendation contexts, ResNet can be applied to extract robust, hierarchical features from high-dimensional data, e.g., movie posters or other multimedia content. Its depth makes it especially effective for learning complex feature, contributing to high-quality recommendations.

## 3. DATASET preparation

Here, we present a movie recommendation system that integrates movie review datasets from numerous sources. This includes a dataset of over 5000 movies from data source Kaggle [29] up to year 2017, alongside movie metadata [30] and additional data from Wikipedia for movies released between the years 2018 [31], 2019 [32] and 2020 [33]. Additionally, we collect reviews for sentiment analysis from the TMDB website using the TMDB API [28].

## 4. Feature engineering

In this project as shown in Fig. 1, the focus is on building an inclusive and sophisticated framework, for movie rating prediction by integrating various machine learning as well as deep learning models with sentiment analysis and feature extraction techniques. Initially, we preprocess the data using DistilBERT, which is a Transformer-based model used to extract sentiment scores and labels, enhancing the dataset with valuable contextual insights from movie reviews. Additionally, we apply TF-IDF vectorization, which is combined with SVD for the reduction of dimensionality, resulting in a streamlined feature set.

The experiments conducted in this study reveal the tangible impact of sentiment analysis on the performance of the classification model. Without sentiment analysis, the model struggled to generalize effectively, especially in handling imbalanced classes. However, integrating sentiment embeddings generated by DistilBERT led to a noticeable improvement in performance metrics, as detailed below.

The supplementary experiments on sentiment analysis, as shown in Table 1, demonstrate a clear improvement in both classification and regression tasks. The observed improvements in both classification and regression tasks suggest that sentiment embeddings contribute beyond sentiment polarity detection, directly enhancing rating prediction accuracy. While the classification accuracy increases from 85.75 % to 91.75 % with DistilBERT, demonstrating better sentiment differentiation, the key takeaway is its impact on regression performance. The reduction in MSE (from 0.1224 to 0.0743), MAE (from 0.2552 to 0.1595), and RMSE (from 0.3498 to 0.2726) underscore how refined sentiment representations lead to more precise numerical predictions.

Rather than treating classification accuracy as a standalone metric, it should be interpreted as a validation of sentiment embedding quality. Higher classification accuracy indicates that the embeddings capture

**Algorithm for Sentiment Analysis and Rating Prediction**

1. **Data Preparation:**
   - Load movie review data with user ratings.
   - Preprocess data:
     - Clean and tokenize reviews.
     - Handle missing values.
     - Create a combined feature vector including review text, sentiment, and other movie attributes.
2. **Sentiment Analysis:**
   - Use DistilBERT to extract sentiment scores and labels from reviews.
   - Assign positive or negative labels based on the sentiment score.
3. **Feature Engineering:**
   - Combine sentiment scores, labels, and other features into a single vector.
   - Apply TF-IDF and SVD for feature extraction and dimensionality reduction.
4. **Model Selection and Training:**
   - Choose from various deep learning models:
     - BiLSTM, CNN-LSTM, CNN-GRU, CNN-Attention, CNN, VAE, SimpleRNN, GRU-Attention, Transformer Encoder, FNN, ResNet.
   - Train each model on the prepared dataset with early stopping to prevent overfitting.
5. **Model Evaluation:**
   - Evaluate models on a test set using metrics like MAE, MSE, RMSE, R2, and Explained Variance.
   - Compare performance across different models and sample sizes.
6. **Result Analysis:**
   - Analyze the results to identify the best-performing model.
   - Visualize the performance metrics to gain insights.
   - Consider factors like model complexity, training time, and dataset size.
7. **Deployment:**
   - Deploy the chosen model to a production environment.
   - Use the model to predict user ratings for new movie reviews.

**Fig. 1.** Algorithm of Sentiment Analysis & Movie rating Prediction.

**Table 1**
Performance comparison with and without DistilBERT sentiment classifier.

| Model Variant | Accuracy | MSE | MAE | RMSE | Precision (Class 1) | Recall (Class 1) | F1-score (Class 1) |
|---|---|---|---|---|---|---|---|
| Without DistilBERT | 0.8575 | 0.1224 | 0.2552 | 0.3498 | 0.86 | 1.00 | 0.92 |
| With DistilBERT | 0.9175 | 0.0743 | 0.1595 | 0.2726 | 0.91 | 1.00 | 0.95 |

nuanced sentiment variations more effectively, which, in turn, enrich the feature representations used in regression. This improved representation reduces prediction errors by aligning extracted sentiment information more closely with actual user ratings.

The CNN + LSTM model, trained with TF-IDF embeddings serves as a baseline to illustrate this relationship. While TF-IDF captures word frequency-based sentiment cues, DistilBERT embeddings offer a more contextualized understanding, leading to improvements across both classification and regression tasks Therefore, sentiment analysis should be framed primarily in terms of its role in refining feature extraction, ensuring consistency with the study's core regression evaluation metrics.

To enhance clarity, the discussion will emphasize how improvements in sentiment classification contribute to better rating predictions. This reinforces the alignment between sentiment analysis and the study's primary regression objectives.

## 5. MODEL development & results discussion

This dataset is then used to train multiple models encompassing BiLSTM, CNN + LSTM, CNN + GRU, CNN + Attention, CNN, VAE, Simple RNN, GRU + Attention, Transformer Encoder, FNN, and ResNet. Each model here explores different mechanisms for capturing dependencies within the data. For example, the CNN + Attention model utilizes self-attention which is used to identify relationships within the data, while the BiLSTM model captures dependencies which are in both forward & backward directions. Moreover, using a VAE-based model allows us to integrate generative elements, creating a more robust feature representation that can potentially improve prediction accuracy.

Here the evaluation criteria include MSE, MAE, RMSE, R-squared, and explained variance score, which help us to analyze model performance and provide insight into each model's suitability for the task. With this approach, our goal is to establish a strong baseline and identify the best-performing model, contributing to advanced movie recommendation systems that align closely with user preferences and actual ratings.

Here this pipeline allows for an inclusive assessment of innumerable deep learning models on the movie rating prediction task. It integrates both of the traditional architectures (like BiLSTM and CNN) and the advanced approaches (like Attention mechanisms, VAE, and GAN), with an emphasis on balancing performance and interpretability. Each model shown here is designed to address specific data characteristics, such as sequence information given in movie reviews, making the framework much flexible for various text-heavy recommendation systems.

Here's a summary of the models and their structures:

- BiLSTM: A Bidirectional LSTM network with 64 units, followed by a Dense layer (32 units) for regression. It uses MSE- Mean Squared Error as the loss function and Adam optimizer.
- CNN + LSTM: Combines Conv1D (64 filters) for feature extraction, followed by LSTM (64 units) for sequence modeling. It uses MSE and Adam optimizer.
- CNN + GRU: Similar to the CNN + LSTM, but replaces LSTM with GRU for sequence modeling. It also uses MSE and Adam.
- CNN + Attention: Uses Conv1D layers for feature extraction followed by a self-attention mechanism, then a Dense layer for regression. It uses MSE and Adam.
- VAE: A Variational Autoencoder with a 32-dimensional latent space. The model uses both reconstruction loss (binary cross entropy) and KL divergence loss, and is trained with the RMSprop optimizer.
- Simple RNN: A Simple RNN layer (64 units) is mainly used for sequence modeling, followed by a Dense layer for the regression. It uses MSE and Adam.
- GRU + Attention: This combines GRU (64 units) with self-attention for sequence modeling, followed by Dense layers for regression. It also uses MSE and Adam.
- FNN (Feedforward Neural Network): A fully connected network with three Dense layers of sizes 128, 64, and 32, trained for regression using MSE and Adam.
- ResNet: A CNN with residual connections and two Conv1D layers followed by MaxPooling, Flatten, and Dense layers for regression. It uses MSE and Adam.
- Transformer Encoder: A simplified transformer with Conv1D layers and Dense layers for regression. It uses MSE and Adam.
- GAN (Generator + Discriminator): The generator creates synthetic data from a latent vector, and the discriminator classifies the data. Both parts are trained using binary cross-entropy loss.

Based on the data provided from Table 2, here a summarized analysis of the models' observations and insights based on the sample size and key metrics (MAE, MSE, RMSE, $R^2$, and Explained Variance) is provided in detail:

### 5.1. Effect of attention mechanism

Models incorporating Attention (e.g., CNN + Attention, GRU + Attention) demonstrate varying degrees of improvement over their non-attention counterparts, particularly in reducing RMSE and improving $R^2$ for larger sample sizes.

However, CNN + Attention and GRU + Attention do not constantly outperform simpler models on smaller datasets, which may point to a need for larger data volumes to fully leverage the benefits of attention.

### 5.2. Performance of simple and advanced models

From analyzing Figs. 2, 3 and 4, As the sample size rises, The BiLSTM model shows consistent performance improvement, with comparatively lower MAE, MSE, and RMSE to other models. For the larger datasets (e.g., 5000 samples), it performs fairly well with $R^2$ values around 0.26.

As Per Fig. 5, Traditional models such as FNN (Feedforward Neural Networks) and ResNet have high error rates and poor $R^2$ values, mainly on smaller datasets, indicating they are less suited for this regression task without additional optimization.

More advanced models, such as the Transformer Encoder, demonstrate potential but generally fall behind BiLSTM and VAE in terms of MAE and RMSE across most sample sizes.

The performance analysis of various models reveals significant variations, highlighting the strengths and limitations of each approach. The VAE model consistently achieves the lowest error values (MAE, MSE, and RMSE), indicating its ability to capture complex patterns effectively. However, its highly negative $R^2$ and Explained Variance scores suggest potential overfitting or difficulties in generalizing to unseen data. This suggests that while VAE is powerful in latent representation learning, it may require additional regularization techniques or fine-tuning for better generalization. In contrast, BiLSTM demonstrates relatively strong performance with lower errors and improved $R^2$ scores, making it

**Table 2**
The evaluation metrics for given Different Models on customized dataset.

| Sample Size | Model | MAE | MSE | RMSE | R2 | Explained Variance |
|---|---|---|---|---|---|---|
| 1000 | BiLSTM | 0.671496 | 0.979911 | 0.989904 | −0.0009 | 0.00753 |
| 1000 | CNN + LSTM | 6.080096 | 37.88085 | 6.154742 | −37.6921 | −0.00128 |
| 1000 | CNN + GRU | 6.024232 | 37.1923 | 6.098549 | −36.9888 | −0.00059 |
| 1000 | CNN + Attention | 3.740283 | 14.59443 | 3.820265 | −13.907 | −0.02247 |
| 1000 | CNN | 1.80834 | 3.964826 | 1.991187 | −3.04974 | −0.05592 |
| 1000 | VAE | 0.406633 | 0.17754 | 0.421355 | −2642.08 | −101.288 |
| 1000 | Simple RNN | 1.64833 | 4.494392 | 2.119998 | −3.59065 | −3.44333 |
| 1000 | GRU + Attention | 1.46563 | 2.814831 | 1.677746 | −1.87512 | 0.000344 |
| 1000 | Transformer Encoder | 0.717499 | 1.079262 | 1.038875 | −0.10238 | −0.09497 |
| 1000 | FNN | 5.951155 | 36.31172 | 6.02592 | −36.0894 | −0.00824 |
| 1000 | ResNet | 2.510871 | 8.314544 | 2.883495 | −7.49262 | −1.15139 |
| 2000 | BiLSTM | 0.59105 | 0.584029 | 0.764218 | 0.173399 | 0.198478 |
| 2000 | CNN + LSTM | 0.746794 | 0.899824 | 0.94859 | −0.27356 | 0.000646 |
| 2000 | CNN + GRU | 5.175412 | 27.42779 | 5.237155 | −37.8197 | 0.000728 |
| 2000 | CNN + Attention | 2.48067 | 6.941842 | 2.634738 | −8.82508 | −0.11759 |
| 2000 | CNN | 2.046236 | 4.959599 | 2.227016 | −6.01953 | −0.12288 |
| 2000 | VAE | 0.343646 | 0.129785 | 0.360257 | −1833.7 | −112.103 |
| 2000 | Simple RNN | 0.975142 | 1.555495 | 1.247195 | −1.20156 | −0.98509 |
| 2000 | GRU + Attention | 1.402577 | 2.512179 | 1.584985 | −2.55559 | −0.00118 |
| 2000 | Transformer Encoder | 0.719324 | 0.901978 | 0.949725 | −0.27661 | −0.16107 |
| 2000 | FNN | 5.946396 | 36.03272 | 6.002726 | −49.9986 | 0.000607 |
| 2000 | ResNet | 0.918815 | 1.375626 | 1.172871 | −0.94698 | −0.81381 |
| 3000 | BiLSTM | 0.557207 | 0.6463 | 0.803928 | 0.235977 | 0.236167 |
| 3000 | CNN + LSTM | 0.998374 | 1.66297 | 1.289562 | −0.96588 | −0.00035 |
| 3000 | CNN + GRU | 2.25313 | 5.928203 | 2.43479 | −6.00802 | −0.00673 |
| 3000 | CNN + Attention | 1.482648 | 2.819197 | 1.679046 | −2.33271 | −0.07432 |
| 3000 | CNN | 1.565222 | 3.075744 | 1.75378 | −2.63599 | −0.06903 |
| 3000 | VAE | 0.263929 | 0.081713 | 0.285856 | −592.812 | −68.7838 |
| 3000 | Simple RNN | 0.809486 | 1.138417 | 1.066966 | −0.34578 | −0.2557 |
| 3000 | GRU + Attention | 0.749757 | 1.08692 | 1.042555 | −0.2849 | −0.00787 |
| 3000 | Transformer Encoder | 1.308901 | 2.337863 | 1.529007 | −1.7637 | −0.10741 |
| 3000 | FNN | 5.481222 | 30.79281 | 5.549127 | −35.4017 | −0.01322 |
| 3000 | ResNet | 1.065996 | 1.987914 | 1.409934 | −1.35001 | −1.25002 |
| 4000 | BiLSTM | 0.566931 | 0.584771 | 0.764703 | 0.158819 | 0.158916 |
| 4000 | CNN + LSTM | 0.779662 | 0.980019 | 0.989959 | −0.40974 | −0.00056 |
| 4000 | CNN + GRU | 1.251231 | 2.081645 | 1.442791 | −1.9944 | −0.01432 |
| 4000 | CNN + Attention | 0.851361 | 1.14326 | 1.069234 | −0.64456 | −0.0598 |
| 4000 | CNN | 0.799943 | 1.067414 | 1.033157 | −0.53545 | −0.07575 |
| 4000 | VAE | 0.187607 | 0.04504 | 0.212225 | −733.128 | −98.2667 |
| 4000 | Simple RNN | 0.805155 | 1.092853 | 1.045396 | −0.57205 | 0.017145 |
| 4000 | GRU + Attention | 0.858142 | 1.138319 | 1.06692 | −0.63745 | 0.001379 |
| 4000 | Transformer Encoder | 0.911492 | 1.340027 | 1.157595 | −0.9276 | −0.18881 |
| 4000 | FNN | 4.218891 | 18.40042 | 4.289572 | −25.4686 | −0.00789 |
| 4000 | ResNet | 1.04947 | 1.803121 | 1.342804 | −1.59375 | −1.08891 |
| 5000 | BiLSTM | 0.537969 | 0.503345 | 0.709468 | 0.26168 | 0.261796 |
| 5000 | CNN + LSTM | 0.681385 | 0.754899 | 0.868849 | −0.10731 | −0.00046 |
| 5000 | CNN + GRU | 0.642445 | 0.693075 | 0.832511 | −0.01662 | −0.00473 |
| 5000 | CNN + Attention | 0.747997 | 0.86742 | 0.931354 | −0.27236 | −0.03289 |
| 5000 | CNN | 0.666162 | 0.722275 | 0.849868 | −0.05945 | −0.02662 |
| 5000 | VAE | 0.12305 | 0.022333 | 0.149444 | −509.388 | −102.912 |
| 5000 | Simple RNN | 0.62642 | 0.663366 | 0.814473 | 0.026956 | 0.028359 |
| 5000 | GRU + Attention | 0.646964 | 0.705254 | 0.839794 | −0.03449 | −0.00028 |
| 5000 | Transformer Encoder | 0.770956 | 0.911904 | 0.954937 | −0.33761 | −0.07195 |
| 5000 | FNN | 1.085351 | 1.72848 | 1.314717 | −1.53538 | −0.08889 |
| 5000 | ResNet | 0.911236 | 1.315809 | 1.147087 | −0.93007 | −0.90149 |

a reliable choice for sequential data analysis. The GRU + Attention model also performs well by maintaining a balance between accuracy and computational efficiency, selectively focusing on important sequences to enhance predictions. On the other hand, CNN-based models, such as CNN + LSTM and CNN + GRU, exhibit significantly higher errors, particularly for smaller sample sizes, indicating their struggle in capturing long-range dependencies within the dataset. While CNN architectures are effective in feature extraction, their ability to model sequential relationships may be limited, leading to suboptimal performance. Similarly, ResNet, despite its deep learning capabilities, shows inconsistent results, often producing higher errors and poor $R^2$ scores, suggesting that residual learning techniques effective in image processing may not translate well to movie rating predictions. Meanwhile, Transformer Encoder and Simple RNN models perform moderately, though their higher variance in predictions suggests sensitivity to dataset size. Transformers generally require large amounts of data to perform optimally, while Simple RNNs are prone to vanishing gradient issues, making them less effective for long-term dependencies compared to GRU and LSTM-based models.

From a practical perspective, the trade-off between accuracy and generalization is crucial. While VAE provides the best accuracy in terms of error reduction, its poor $R^2$ and explained variance scores indicate that a model with slightly higher errors but better generalization, such as BiLSTM, may be preferable for real-world applications. Additionally, computational efficiency plays a vital role in model selection. Transformer-based architectures and deep models like ResNet, while powerful, are computationally expensive and may not be feasible in resource-constrained environments. In contrast, GRU + Attention offers a reasonable trade-off between accuracy and efficiency, making it a more practical choice. Furthermore, dataset size sensitivity is another

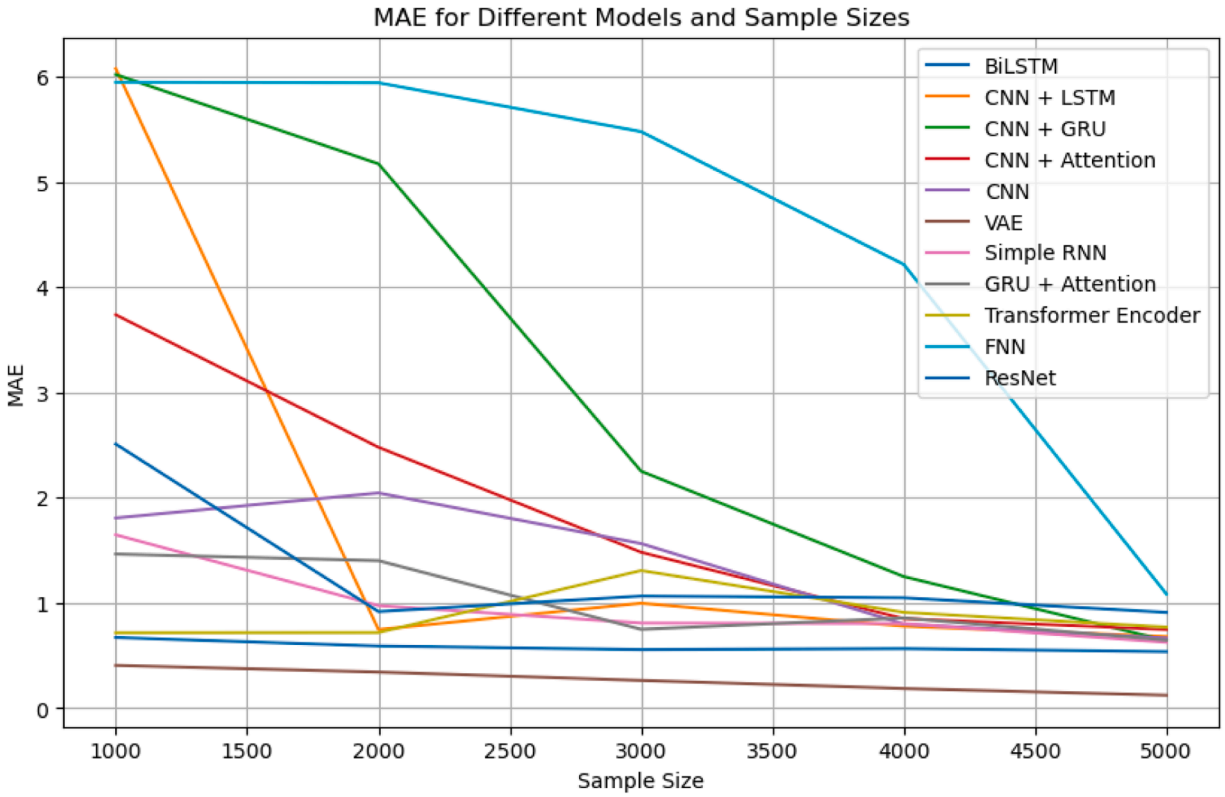


**Fig. 2.** MAE for Different Models and Different Sample Sizes.

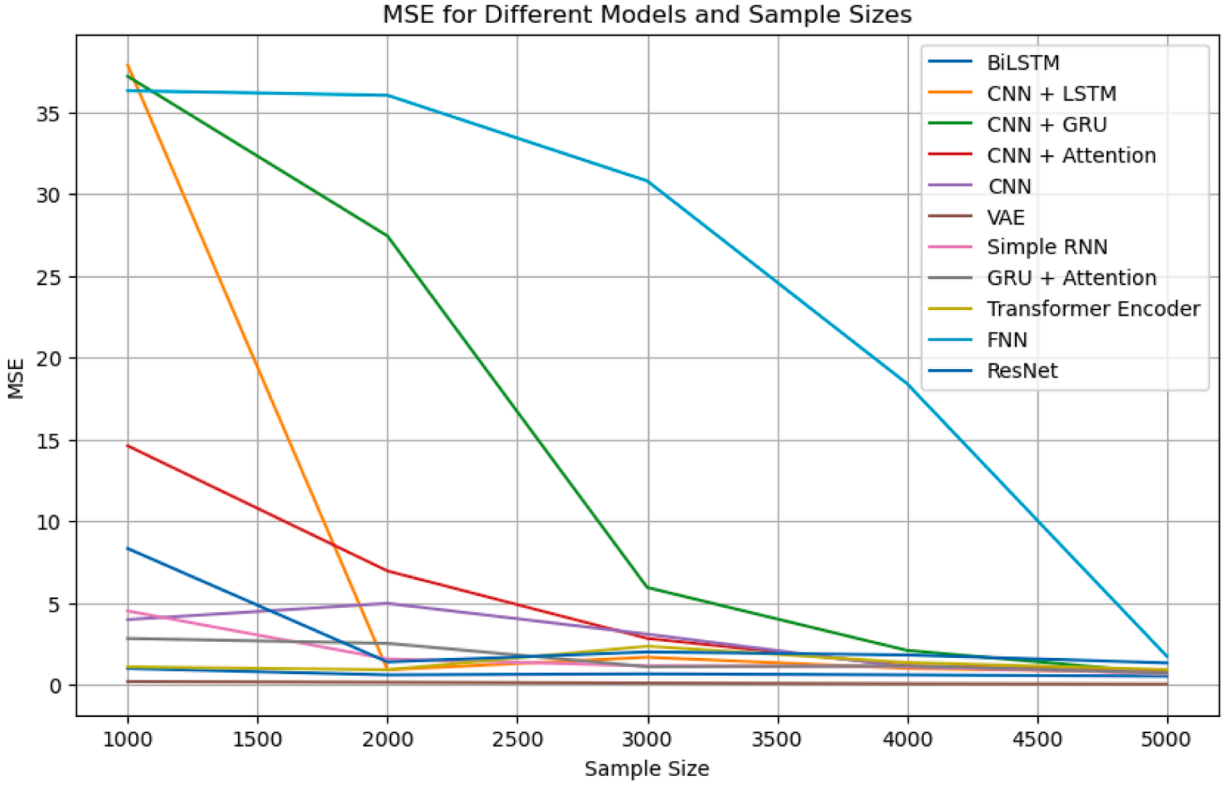


**Fig. 3.** MSE for Different Models and Different Sample Sizes.

critical factor, as models like CNN + LSTM and ResNet show performance degradation with smaller datasets, suggesting that they may require larger data volumes to fully leverage their architectural advantages. These findings underscore the importance of careful model selection based on practical constraints such as computational cost, dataset availability, and generalization ability, rather than relying solely

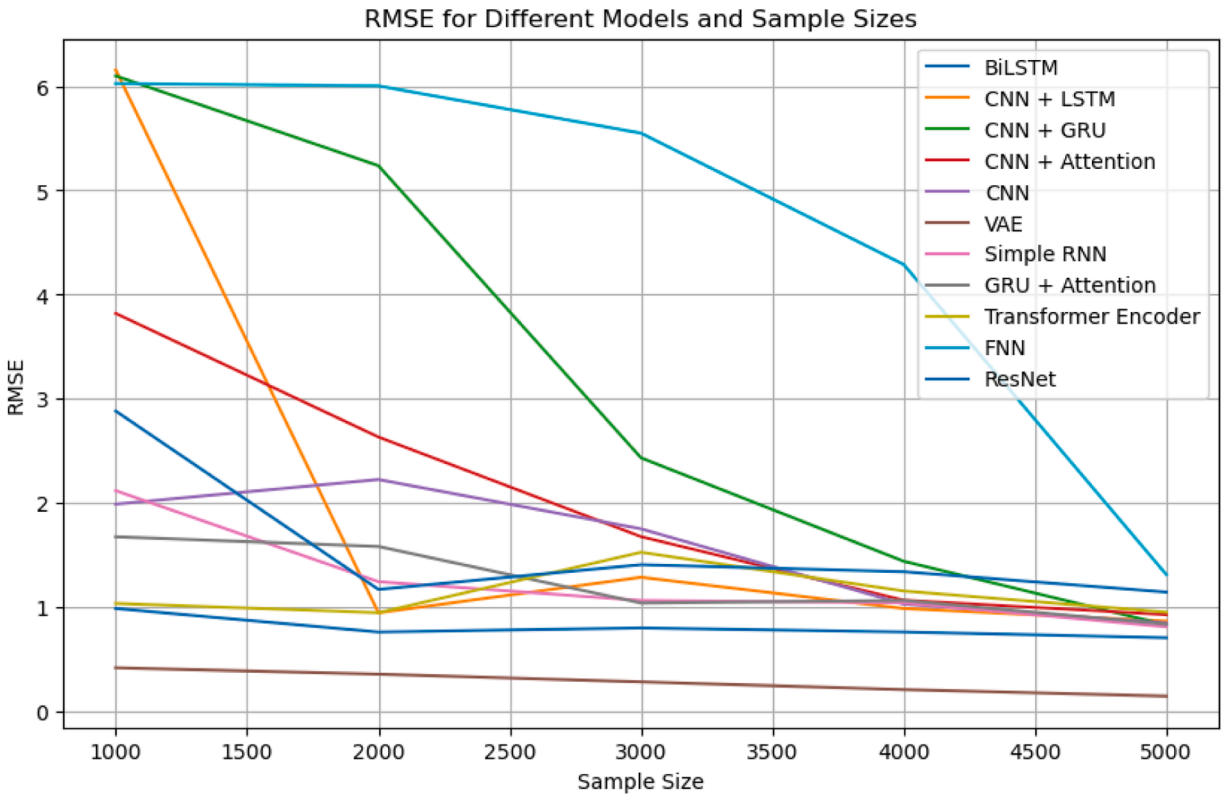


**Fig. 4.** RMSE for Different Models and Different Sample Sizes.

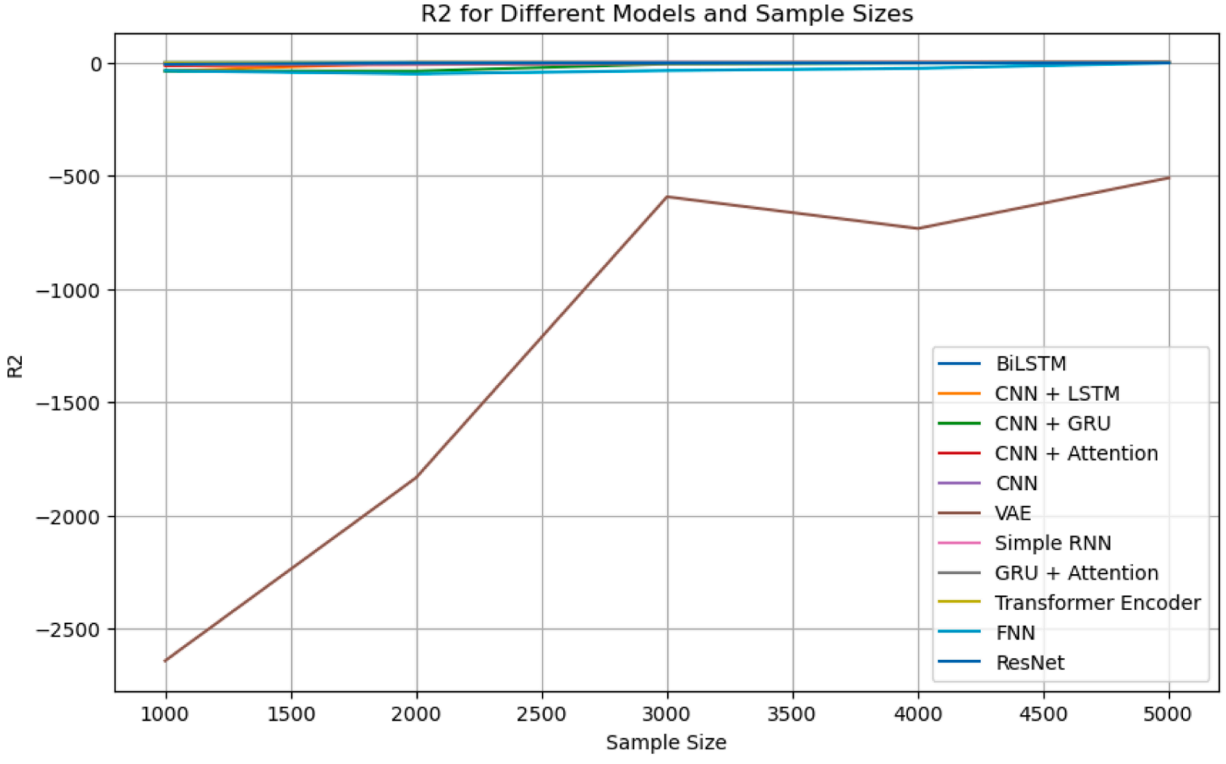


**Fig. 5.** R2 for Different Models and Different Sample Sizes.

on error metrics.

### 5.3. Model consistency

VAE demonstrates remarkably low MAE, MSE, and RMSE values across different sample sizes, indicating strong predictive performance

in minimizing absolute errors. This suggests that VAE effectively captures latent patterns in the data, leading to precise individual predictions.

However, the model exhibits extremely negative $R^2$ values, indicating a significant discrepancy between the variance of the predicted and actual ratings. While this might initially suggest overfitting, it is more likely due to structural characteristics of the VAE rather than traditional overfitting in a deterministic model. Unlike standard regression-based models, VAE prioritizes reconstructing input features rather than directly optimizing for rating prediction, which may result in uninformative or misaligned feature representations.

Several factors could contribute to this behaviour:

- **Poor Latent Space Representation** – The learned latent space may not effectively capture the global variance of the target ratings, leading to inconsistent predictions.
- **Overly Strong KL Divergence Regularization** – Excessive regularization can force the latent space distribution too close to a prior (e.g., isotropic Gaussian), potentially limiting the expressiveness of the learned representations.
- **Mismatch Between Generative and Predictive Objectives** – VAE's primary goal is to generate meaningful representations of input data rather than directly minimize rating prediction error, which may cause it to underperform in tasks requiring strict numerical alignment.
- **Improper Feature Scaling or Suboptimal Hyperparameters** – Poorly scaled features, an inappropriate latent dimension size, or insufficient tuning of key hyperparameters may further degrade predictive performance.

To address these issues, future work could explore fine-tuning techniques such as adjusting the KL divergence weight, optimizing the latent space dimensionality, refining hyperparameters, and incorporating hybrid models that balance generative representation learning with explicit predictive objectives. These improvements could enhance VAE's interpretability while preserving its ability to capture complex feature interactions.

### 5.4. Practical implications and trade-offs

- Accuracy vs. Generalization: While VAE provides the best accuracy, its poor $R^2$ and explained variance scores highlight the importance of evaluating generalization. A model with slightly higher errors but better $R^2$, such as BiLSTM, may be preferable in real-world applications.
- Computational Cost vs. Performance: Transformer-based models and deep networks like ResNet are computationally expensive, making them impractical for resource-constrained environments. In contrast, GRU + Attention offers a reasonable trade-off between accuracy and efficiency.
- Dataset Size Sensitivity: Some models, such as CNN + LSTM and ResNet, perform worse on smaller datasets, indicating that they may require larger data volumes to leverage their architectural strengths effectively.

### 5.5. Impact of sample size

Increasing the sample size generally improves the performance of all the models, especially in context of reducing MAE and RMSE. However, from the Analysis of the Fig. 6, the improvement in $R^2$ and Explained Variance is not uniform, as some models still show adverse $R^2$, indicating poor fit despite the larger dataset.

### 5.6. Noteworthy performance

VAE consistently has low error metrics (MAE, MSE, RMSE), making it a candidate for further tuning, though the highly negative $R^2$ suggests it might require regularization or additional feature engineering to generalize better.

BiLSTM and GRU + Attention appears to be more balanced choices,

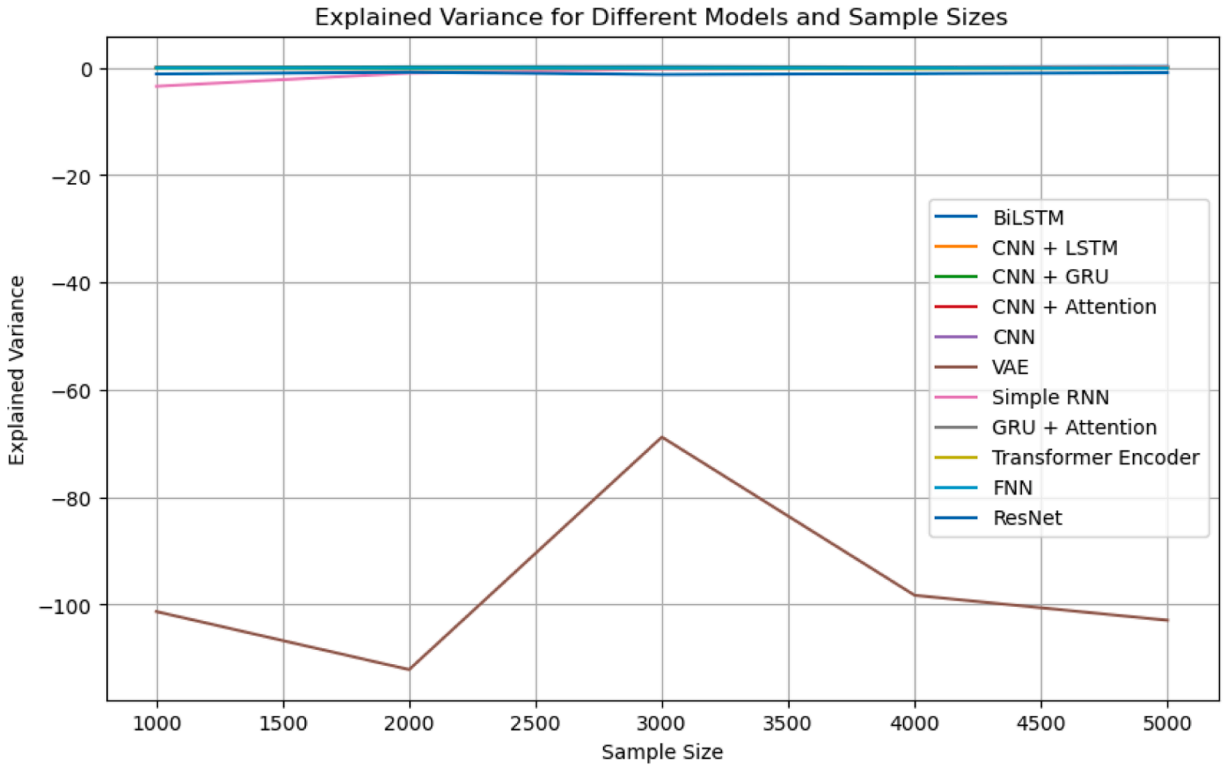


**Fig. 6.** Explained Variance for Different Models and Different Sample Sizes.

with moderate error metrics and reasonable $R^2$ values, representing both accuracy and generalizability.

Here are the context-specific definitions of the evaluation metrics, tailored for actual movie ratings (yi) predicted movie ratings (ŷi) :

Mean Absolute Error (MAE) measures the average magnitude of the errors which are in a set of predictions.

$$\mathbf{MAE} = (1/\mathbf{n}) * \Sigma|\mathbf{yi} - \hat{\mathbf{y}}\mathbf{i}| \quad (1)$$

Where, n: number of data points, yi: actual rating, ŷi: predicted rating.

Mean Squared Error (MSE) - measures the average squared difference, which is between the actual and predicted values.

$$\mathbf{MSE} = (1/\mathbf{n}) * \Sigma(\mathbf{yi} - \hat{\mathbf{y}}\mathbf{i})^2 \quad (2)$$

Root Mean Squared Error (RMSE) is the square root value, which is of the MSE, which is providing an error measure in the same units like as the original data.

$$\mathbf{RMSE} = \sqrt{\mathbf{MSE}} \quad (3)$$

R-squared ($R^2$) measures the proportion of the variance in the dependent variable (actual ratings) that is explained by the independent variable (predicted ratings).

$$\mathbf{R}^2 = 1 - \left(\frac{\mathbf{SSR}}{\mathbf{SST}}\right) \quad (4)$$

Where, SSR: Sum of Squared Residuals = Σ (yi - ŷi) ², SST: Total Sum of Squares = Σ(yi - ȳ)² , ȳ: mean of actual ratings

Explained Variance Score measures the proportion of variance in the dependent variable, explained the model predictions.

$$\textbf{Explained Variance Score} = 1 - \left(\frac{\mathbf{Var}(\mathbf{y} - \hat{\mathbf{y}})}{\mathbf{Var}(\mathbf{y})}\right) \quad (5)$$

Where, Var (y - ŷ): variance of the residuals, Var(y): variance of the actual ratings

By calculating these kinds of metrics, we can compute the accuracy and reliability of our movie rating prediction model and make knowledgeable decisions about its performance and probable improvements.

In Fig. 7, The number of attention heads' sensitivity analysis in the CNN + Attention model shows that use of 4 attention heads results in the lowest loss (0.667), offering the best performance. Increasing the amount of attention heads beyond 4 leads to diminishing returns, with performance slightly degrading. Therefore, 4 heads strike the best balance between model complexity and performance.

The Summary of Differences between the given Models are:

VAE excels at handling uncertainty and missing data, and it is generative, meaning it is able to create new samples. This helps it overcome cold start problems more effectively than models like CNN + LSTM, CNN + GRU, or BiLSTM, which are not designed for generative tasks.

CNN-based models (LSTM, GRU, Attention) are intended for sequential data processing (such as text or time-series) and capture temporal dependencies. However, they still struggle with cold-start problems as they do not generate new data which relies heavily on the availability of historical data.

BiLSTM and CNN + Attention are mainly decent at capturing complex sequential dependencies, but they are not inherently generative as VAE. Their attention mechanisms help the model focus on important sequences or features, but they still require explicit handling of missing data and cold-start issues.

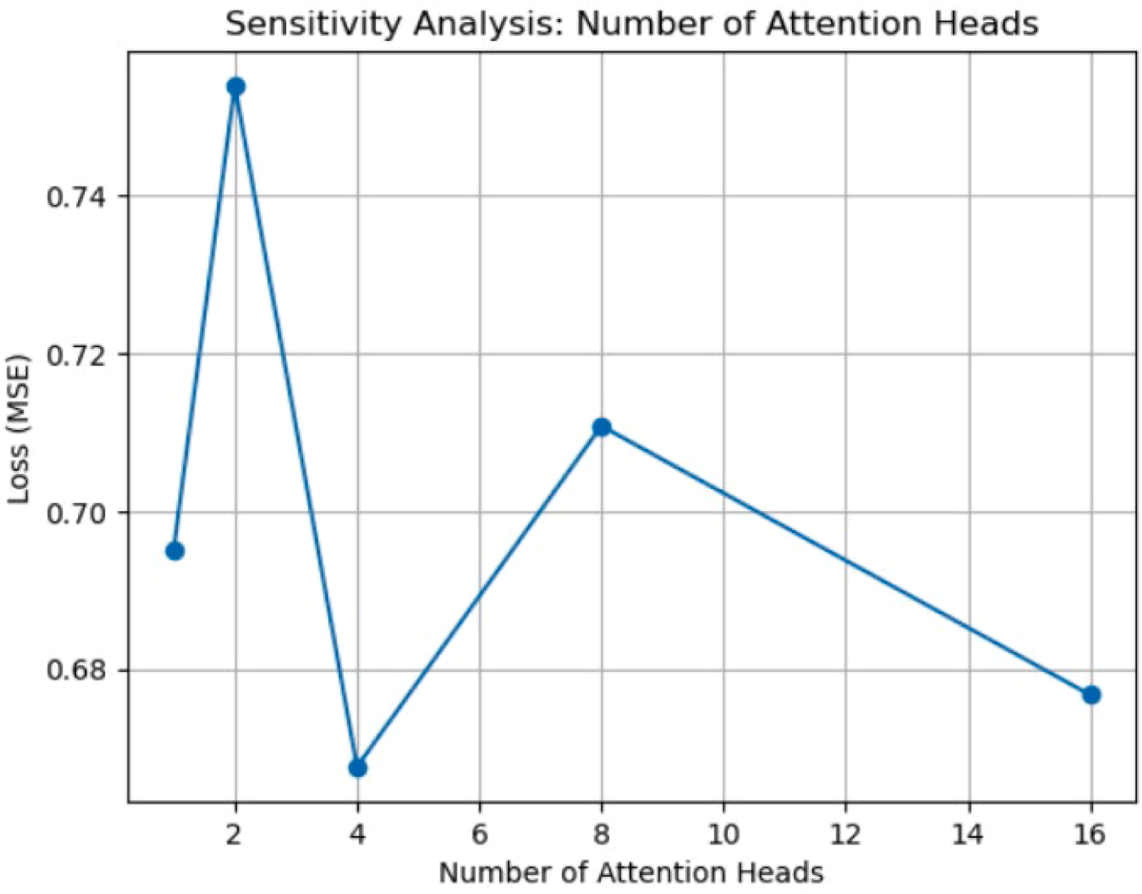


**Fig. 7.** Sensitivity Analysis:Number of Attention Heads.

Training Time fluctuates significantly. VAE tends to be the slowest due to its complex training process which involves variational inference. The CNN-based models and BiLSTM can have moderate training times.

CNN + LSTM efficiently extracts features and captures temporal patterns, but struggles with long sequences. CNN + GRU offers better efficiency but may miss long-range dependencies. CNN + Attention improves performance with a focus mechanism, though it adds complexity. VAE excels at learning a regularized latent space but can struggle with noisy data. Simple RNNs are efficient but fail with long-term dependencies. GRU + Attention combines efficiency with attention but still faces long-range challenges. FNN is simple but lacks the ability to model complex relationships, while ResNet helps with gradient flow but can lead to overfitting. Transformer models capture long-range dependencies well but are computationally expensive, and GANs are powerful but often unstable during training.

The VAE model performs significantly better than SVR on the basis of provided MAE and MSE. The VAE's MAE decreases from 0.4066 to 0.1231, and MSE drops from 0.1775 to 0.0223, indicating improved accuracy with each iteration. In contrast, as shown in [34], the SVR model has higher MAE (0.787) and MSE (1.097), demonstrating that VAE minimizes prediction errors more effectively. Table 3.

This extensive evaluation and comparison framework provides the most effective Deep Learning Model for Movie rating Predictions.

### 5.7. Comparison with other benchmark dataset

To ensure a thorough comparison, we evaluate our proposed approach against the widely used MovieLens [35] benchmark. The MovieLens-based method chiefly relies on structured numerical and categorical features such as age, gender, occupation, and movie year, with TF-IDF vectorization applied to movie titles, followed by dimensionality reduction using Truncated SVD and feature scaling. In contrast, proposed customized database-based method integrates both structured and unstructured data, incorporating user reviews, sentiment scores extracted using DistilBERT, and user ratings. By leveraging TF-IDF vectorization with a higher dimensionality (10,000 features) and applying Truncated SVD (200 components), our method captures richer contextual information from textual data. Unlike MovieLens, which focuses on predefined user and item attributes, our approach enhances predictive performance by incorporating sentiment polarity and user-generated content, making it more effective in capturing nuanced user preferences.

**Table 4**
The evaluation metrics for given different models on movielens dataset.

| Model | MSE | MAE | RMSE |
|---|---|---|---|
| BiLSTM | 1.263141 | 0.943774 | 1.123895 |
| CNN + LSTM | 1.263239 | 0.941157 | 1.123939 |
| CNN + GRU | 1.263117 | 0.941631 | 1.123885 |
| CNN + Attention | 1.263012 | 0.942678 | 1.123838 |
| CNN | 1.263223 | 0.941211 | 1.123932 |
| Simple RNN | 1.263106 | 0.941687 | 1.123880 |
| GRU + Attention | 1.263023 | 0.942980 | 1.123843 |
| FNN | 1.265930 | 0.947997 | 1.125135 |
| ResNet | 1.263187 | 0.941339 | 1.123916 |
| VAE | 0.084038 | 0.250673 | 0.289893 |
| Transformer | 0.080232 | 0.242910 | 0.283253 |

**Table 3**
Comparing Models with respect to features.

| Feature | VAE | CNN + LSTM | CNN + GRU | CNN + Attention | BiLSTM |
|---|---|---|---|---|---|
| Model Type | Probabilistic Deep Learning Model | Convolutional + Recurrent Model | Convolutional + Recurrent Model | Convolutional + Attention-based Model | Recurrent Deep Learning Model |
| Latent Space Representation | Latent probabilistic space | Does not have explicit latent space | Does not have explicit latent space | Attention mechanism instead of latent space | Sequential hidden states |
| Handling Uncertainty | Models' uncertainty using latent space | Does not model uncertainty | Does not model uncertainty | Attention weights can focus on key areas, but no explicit uncertainty modeling | Does not model uncertainty |
| Generative Aspect | Generates new data (user-item interaction) | Not generative, focuses on prediction | Not generative, focuses on prediction | Focused on learning attention-based relationships | Not generative, focuses on prediction |
| Regularization | KL divergence regularization for smoothness | Regularization through dropout and L2 | Regularization through dropout and L2 | Regularization via attention and dropout | Regularization via L2 and dropout |
| Handling Missing Data | Handles missing data via latent space representation | Requires imputation or missing data strategy | Requires imputation or missing data strategy | Requires imputation or missing data strategy | Requires imputation or missing data strategy |
| Data Type Handling | Can handle complex data distributions due to probabilistic nature | Focuses on sequential data (e.g., time series, text) | Focuses on sequential data (e.g., time series, text) | Focuses on sequential data with attention mechanism | Focuses on sequential data (text or time series) |
| Feature Interactions | Models complex interactions in latent space | Captures spatial and temporal interactions | Captures spatial and temporal interactions | Focuses on key features using attention weights | Models sequential interactions |
| Scalability | Can scale but may be slow due to sampling in training | Scales well but requires sufficient computational resources | Scales well but requires sufficient computational resources | Scales well with attention mechanism, but needs careful tuning | Scales well for sequential data |
| Overfitting Prevention | KL Divergence term helps to prevent overfitting | Dropout layers for regularization | Dropout layers for regularization | Dropout + attention regularization | Dropout regularization |
| Training Complexity | High computational complexity due to sampling from latent space | Requires tuning of both CNN and LSTM parameters | Requires tuning of both CNN and GRU parameters | Requires tuning of CNN + Attention weights | Requires tuning of LSTM parameters |
| Flexibility | High flexibility due to the generative model | Flexible for sequence-based problems | Flexible for sequence-based problems | Flexible for sequence-based problems with attention focus | Flexible for sequence-based problems |
| Cold Start Problem | Handles cold start better through generative nature | Struggles with cold start if no historical data | Struggles with cold start if no historical data | Struggles with cold start if no historical data | Struggles with cold start if no historical data |
| Hyperparameter Tuning | Needs tuning of latent space size, learning rate, and regularization terms | Needs tuning of CNN layers, LSTM parameters | Needs tuning of CNN layers, GRU parameters | Needs tuning of CNN layers, attention parameters | Needs tuning of LSTM parameters |
| Interpretability | Lower interpretability due to the complex latent space and probabilistic nature | Moderate interpretability in terms of learned filters and sequential patterns | Moderate interpretability in terms of learned filters and sequential patterns | Lower interpretability due to attention mechanisms being black-box | Moderate interpretability in terms of sequential patterns |
| Training Time | Can be slow due to variational inference and sampling steps | Moderate training time due to sequential data processing | Moderate training time due to sequential data processing | Moderate to high depending on the attention complexity | Moderate to high depending on data size |

Here is the formatted Table 4 with all the models and their evaluation metrics:

The VAE and Transformer models have significantly lower errors than the others, indicating superior performance.

Although customized database-based method requires higher computational resources due to transformer-based sentiment analysis, it provides a more comprehensive understanding of user sentiment and engagement, demonstrating its advantage in real-world movie recommendation scenarios where textual opinions significantly influence user decisions.

The evaluation dataset is designed to ensure diversity by incorporating a broad range of movies across multiple genres, different user demographics, and varying sentiment expressions in reviews. Unlike traditional datasets that primarily rely on structured numerical features (e.g., MovieLens), proposed dataset integrates textual data from IMDb user reviews, enriched with sentiment scores extracted using DistilBERT. This approach enables a more nuanced analysis of user preferences beyond explicit ratings. Additionally, the dataset includes movies from different years and a variety of user profiles, ensuring that the proposed method is robust across different audience segments and rating behaviors. By leveraging both structured and unstructured data, our evaluation framework effectively highlights the strengths of different models in handling diverse user interactions and contextual factors in movie recommendation.

The dataset is randomly sampled from a large corpus to ensure diversity across different attributes like genres, directors, actors, and sentiments. Additionally, DistilBERT-based sentiment extraction captures nuanced variations, and TF-IDF with SVD retains key textual diversity. Expanding the sample size or incorporating stratified sampling can further enhance representativeness.

MovieLens 100 K Dataset (structured format with user-item interactions) Fields are: user_id, item_id, rating, timestamp, movie_title, year, age, gender, occupation, zip_code. Models like BiLSTM, CNN + LSTM, CNN + GRU, CNN + Attention, etc., were evaluated on MSE, MAE, and RMSE. VAE and Transformer models performed significantly better.

The Proposed Dataset with Movie Metadata & Sentiment Analysis Fields are director_name, actor_1_name, actor_2_name, actor_3_name, genres, movie_title, comb, User_Score, Review, Review_Sentiment Performance was measured across different sample sizes (1000 to 5000) for multiple models. VAE again showed the best performance, with Transformer Encoder also performing well.

Both MovieLens 100 K and our dataset support movie rating prediction, but MovieLens 100 K focuses on structured user-item interactions, while our dataset integrates metadata and sentiment analysis. Unlike MovieLens, our dataset leverages textual and semantic features, improving model performance, especially for VAE and Transformer-based models. This broader feature set provides a richer benchmark, capturing deeper user preferences beyond explicit ratings.

- **Similarities:** Both MovieLens 100 K and our dataset serve as benchmarks for movie rating prediction and support various deep learning models.
- **Differences:** MovieLens 100 K is structured around user-item interactions, including demographic information, whereas our dataset incorporates additional metadata (director, actors, genres) and sentiment analysis from reviews, providing richer contextual information.
- **Advantages:** The inclusion of sentiment-based and metadata-driven features enhances predictive performance, particularly for complex models like VAE and Transformer Encoder. This broader feature representation enables a more nuanced understanding of user preferences beyond explicit numerical ratings, making our dataset a more comprehensive benchmark.

## 6. Conclusion

This study delivers a thorough performance analysis of numerous deep learning models for movie rating prediction, by examining architectures such as BiLSTM, CNN + GRU, CNN + LSTM, CNN + Attention, VAE, and other advanced frameworks. Through evaluating these kinds of models across manifold metrics, including MAE, MSE, RMSE, $R^2$, and Explained Variance, clear patterns in model performance are identified and effectiveness is found for accurate rating prediction. The results show that while VAE steadily attains the highest accuracy, attention-based models offer valuable improvements in interpretability as well as adaptability to varying input sequences. Models like CNN and BiLSTM also demonstrate reliable performance, and they are also balancing accuracy with computational efficiency. These types of findings underscore the importance of picking the accurate architecture based on the specific requirements of recommendation systems, whether prioritizing prediction accuracy, interpretability, or computational efficiency. This study very well contributes a benchmark for deep learning models in movie rating prediction, which is guiding researchers and practitioners toward optimized model selection in personalized recommendation contexts. Based on the evaluation metrics, the VAE model constantly outperforms others across all sample sizes with the lowermost MAE (0.123 for 5000 samples), MSE (0.022), and RMSE (0.149), which is demonstrating its superior predictive accuracy. However, its negative $R^2$ and Explained Variance still suggest potential limitations in capturing data variability, which is warranting further exploration. The proposed approach integrating sentiment analysis improves movie rating prediction accuracy compared to traditional methods and outperforms benchmark datasets like MovieLens in capturing user preferences. Future research may discover integrating these kinds of various models or incorporating hybrid architectures to further improve the evaluation measure like prediction accuracy and model robustness. The paper could benefit from outlining specific improvements for hybrid models, by integrating reinforcement learning for adaptive recommendations and addressing data sparsity issues. Exploring hybrid models with advanced optimization techniques, such as Bayesian optimization, could enhance accuracy. Additionally, incorporating real-world factors like user behavior patterns and explainable AI techniques can make the system more practical and interpretable.

## Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used ChatGPT in order to improve grammar. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

## CRediT authorship contribution statement

**Manisha Valera:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Dr. Rahul Mehta:** Writing – review & editing, Validation, Supervision, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data Available Upon Request.

## References


[1] H. Xia, J.J. Li, Y. Liu, Collaborative filtering recommendation algorithm based on attention GRU and adversarial learning, IEEE Access. 8 (2020) 208149–208157, https://doi.org/10.1109/ACCESS.2020.3038770.

[2] Xia L., Yang Y., Chen Z., Yang Z., Zhu S. Movie recommendation with poster attention via multi-modal transformer feature fusion. arXiv preprint arXiv:2407.09157. (2024) Jul 12.

[3] Bahare Askari, Jaroslaw Szlichta, Amirali Salehi-Abari, Variational autoencoders for top-k recommendation with implicit feedback, in: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval, 2021.

[4] Shiri, Farhad Mortezapour, Thinagaran Perumal, Norwati Mustapha, and Raihani Mohamed, A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU, arXiv preprint arXiv:2305.17473, (2023).

[5] H. Wang, N. Lou, Z. Chao, A personalized movie recommendation system based on LSTM-CNN, in: 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2020 (2020).

[6] D. Liang, R. Krishnan, T. Jebara, Variational autoencoders for collaborative filtering, in: Proceedings of the 2018 World Wide Web Conference (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, 2018, pp. 689–698. CHE.

[7] Sophort Siet, Sony Peng, Sadriddinov Ilkhomjon, Misun Kang, Doo-Soon Park, Enhancing sequence movie recommendation system using deep learning and kmeans, Appl. Sci. 14 (6) (2024) 2505.

[8] C. Dang, M. Moreno-García, F. De la Prieta, An approach to integrating sentiment analysis into recommender systems, Sensors 21 (16) (2021) 5666.

[9] Chandrakala Arya, Enhancing movie recommendation systems with deep learning and sentiment analysis, Int. J. Mech. Eng. 6 (3) (2021). ISSN: 0974-5823.

[10] H. Xia, Y. Luo, Y. Liu, Attention neural collaboration filtering based on GRU for recommender systems, Complex. Intell. Systems. (2021) 1367–1379.

[11] J. Panchal, S. Vanjale, A deep learning approach towards cold start problem in movie recommendation system, Int. J. Recent Innov. Trends Comput. Commun. 11 (8) (2023). VolumeIssueISSN: 2321-8169.

[12] D. Alsaleh, S. Larabi-Marie-Sainte, Arabic text classification using convolutional neural network and genetic algorithms, IEEE Access. 9 (2021) 91670–91685.

[13] J. Kufel, et al., What is machine learning, artificial neural networks and deep learning?—Examples of practical applications in medicine, Diagnostics 13 (15) (2023) 2582.

[14] J. Heaton, Ian Goodfellow, Yoshua Bengio, Aaron Courville, Deep learning, Genet. Program. Evolvable Mach. 19 (1–2) (2018) 305–307.

[15] S.V. Georgakopoulos, S.K. Tasoulis, A.G. Vrahatis, V.P. Plagianakos, Convolutional neural networks for toxic comment classification, in: Proceedings of the 10th Hellenic conference on artificial intelligence, 2018, pp. 1–6.

[16] N.I. Widiastuti, "Convolution Neural Network for Text Mining and natural language Processing, IOP Conference Series: materials Science and Engineering, (2019), no. 5, p. 052010.

[17] O.A. Montesinos López, A.M. López, and J. Crossa, Convolutional neural networks, multivariate statistical machine learning methods for genomic prediction, (2022) , pp. 533–577.

[18] I. Dhall, S. Vashisth, G. Aggarwal, Automated hand gesture recognition using a deep convolutional neural network model, in: 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2020, pp. 811–816.

[19] E.C. Nisa, Y.Der Kuan, Comparative assessment to predict and forecast water-cooled chiller power consumption using machine learning and deep learning algorithms, Sustain. (Switzerland) 13 (2) (2021) 1–18.

[20] M. Azizjon, A. Jumabek, W. Kim, 1D CNN based network intrusion detection with normalization on imbalanced data, in: International Conference on Artificial Intelligence in Information and Communication, ICAIIC, 2020, pp. 218–224.

[21] W. Ramadhanti, E.B. Setiawan, Topic detection on twitter using deep learning method with feature expansion GloVe, Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI) 9 (3) (2023) 780–792.

[22] Arliyanna Nilla, Erwin Budi Setiawan, Film recommendation system using content-based filtering and the convolutional neural network (CNN) classification methods, Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI) (2024) 17–29.

[23] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv preprint (2020). arXiv:1910.01108v4 [cs.CL].

[24] S. Abdel-Salam and A. Rafea, Performance study on extractive text summarization using BERT models, information, (2022), 13(2), 67.

[25] A. Alshanqiti, A. Namoun, A. Alsughayir, A. Almashreqi, A. Gilal, S. Albouq, Leveraging DistilBERT for summarizing arabic text: an extractive dual-stage approach, IEEE Access. 10 (2021) 312–325.

[26] Zakir Mujeeb Shaikh, Suguna Ramadass, Unveiling deep learning powers: LSTM, BiLSTM, GRU, BiGRU, RNN comparison, Indones. J. Electr. Eng. Comput. Sci. 35 (1) (2024) 263–273. ISSN: 2502-4752.

[27] Haitao He, Zhifu Shang, Mingjie Wu, Yuling Zhang, Movie recommendation system based on traditional recommendation algorithm and CNN model, Highl. Sci. Eng. Technol. 34 (2023). Volume.

[28] TMDB Website, Available: https://www.themoviedb.org/.

[29] imdb-5000-movie-dataset, Available: https://www.kaggle.com/datasets/carolzhangdc/imdb-5000-movie-dataset.

[30] The Movies Dataset, Available: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset.

[31] List of American films of 2018, Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2018.

[32] List of American films of 2019, Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2019.

[33] List of American films of 2020, Available: https://en.wikipedia.org/wiki/List_of_American_films_of_2020.

[34] Manisha Valera, Rahul Mehta, Comprehensive assessment and optimization of sentiment analysis models for movie reviews with enhanced movie recommendation systems, SSRG int. J. Electron. Commun. Eng. 11 (12) (2024) 258–271, https://doi.org/10.14445/23488549/IJECE-V11I12P124. Crossref.

[35] https://grouplens.org/datasets/movielens/100k/.