ISSN 2723-6471

Adaptive Neural Collaborative Filtering with Textual Review Integration for Enhanced User Experience in Digital Platforms

Lusiana Efrizoni[1,*], Edwar Ali[2], Hadi Asnal[3], Junadhi[4]

[1,2,3,4]*Department of Computer Science, University of Science and Technology of Indonesia, Pekanbaru, Indonesia*

(Received: July 4, 2024; Revised: August 31, 2024; Accepted: September 11, 2024; Available online: September 23, 2024)

Abstract

This research proposes a hybrid rating prediction model that integrates Neural Collaborative Filtering (NCF), Long Short-Term Memory (LSTM), and semantic analysis through Natural Language Processing (NLP) to enhance recommendation accuracy. The main objective is to improve alignment between system predictions and actual user preferences by leveraging multi-source information from the Amazon Movies and TV dataset, which includes explicit user–item ratings and textual reviews. The core idea is to combine three complementary processing paths—(1) user–item interaction modeling via NCF, (2) temporal dynamics capture through LSTM, and (3) semantic understanding of reviews using NLP—into a unified deep learning-based adaptive architecture. Experimental evaluation demonstrates that this multi-input approach outperforms the baseline collaborative filtering model, with the Mean Absolute Error (MAE) reduced from 1.3201 to 1.2817 (a 2.91% improvement) and the Mean Squared Error (MSE) reduced from 2.2315 to 2.1894 (a 1.89% improvement). Training metrics visualization further shows a stable convergence pattern, with the MAE gap between training and validation consistently below 0.03, indicating minimal overfitting. The findings confirm that integrating cross-dimensional signals significantly enhances predictive performance and can contribute to increased user satisfaction and engagement in recommendation platforms. The novelty of this work lies in the simultaneous integration of interaction, temporal, and semantic dimensions into a single adaptive recommendation framework, a configuration not jointly explored in prior studies. Moreover, the flexible architecture enables adaptation to other domains such as e-commerce, music, or online learning, broadening its practical applicability.

Keywords: Deep Learning-Based Recommendation, Neural Collaborative Filtering, Long Short-Term Memory, Natural Language Processing, User Rating Prediction

1. Introduction

In the digital era marked by a data explosion and massive consumption of information, users are increasingly challenged to filter relevant content amidst an overwhelming abundance of choices. This phenomenon, commonly referred to as information overload, has become a major issue affecting user experience across various online platforms such as e-commerce, social media, and streaming-based entertainment services. In this context, recommender systems serve a critical role as intelligent tools that filter and deliver personalized content efficiently according to users' preferences. These systems have evolved into strategic components within digital platform architectures not only enhancing user satisfaction, but also demonstrably driving key business performance indicators such as customer retention, engagement duration, and average transaction value. Previous studies, such as [1], have shown that recommender systems can drive up to 30% of total streaming activity on platforms like Netflix, significantly impacting user engagement and retention. Furthermore, even a 1% improvement in prediction accuracy can translate into substantial business value through increased viewing time and customer satisfaction [2], [3]. These findings underscore the importance of ongoing innovation in developing recommender systems based on state-of-the-art technologies.

One of the dominant approaches in recommender system development is Collaborative Filtering (CF), which relies on historical user-item interaction patterns [4]. Both memory-based and model-based CF methods have proven effective in preference prediction; however, they face limitations such as data sparsity and the cold-start problem. Moreover, traditional CF methods often model linear relationships between users and items, making them less capable of capturing the complex and dynamic nature of real-world interactions. To address these limitations, NCF has been introduced by

***Corresponding author: Lusiana Efrizoni (lusiana@stmik-amik-riau.ac.id)**
DOI: https://doi.org/10.47738/jads.v6i4.944

leveraging deep neural networks to model more flexible, nonlinear relationships between users and items [5], [6], [7]. Architectures such as Neural Matrix Factorization (NeuMF) have demonstrated improvements in predictive accuracy [8], [9]. Nonetheless, most NCF models assume that user preferences are static, while in reality, they evolve over time—driven by changing needs, contexts, and expectations. In order to capture this temporal dynamic, the integration of LSTM networks presents a promising solution. LSTM, a variant of Recurrent Neural Networks (RNN), is designed to capture sequences and temporal dependencies in data, enabling models to learn from chronologically ordered user interactions [10], [11]. By incorporating LSTM into the recommendation framework, the system becomes better equipped to adapt to both short-term and long-term changes in user interests, thereby producing more relevant and adaptive recommendations.

In addition to temporal modeling, a rich yet often underutilized source of information in recommender systems is user-generated textual reviews. These reviews not only include numerical ratings but also express opinions, sentiments, and detailed descriptions of product features that cannot be fully captured through numerical scores alone. By adopting NLP techniques, semantic information from reviews can be transformed into informative vector representations and integrated into the model learning process [12]. This study utilizes the publicly available Amazon Movies and TV Dataset [13], a subset of the broader Amazon Product Review Dataset. This dataset comprises multiple dimensions of user and product information, including reviewerID (user identity), asin (unique product code), overall (rating score), reviewText and summary (full and summarized reviews), as well as temporal attributes such as reviewTime and unixReviewTime, purchase verification status (verified), and social signals (vote) representing the helpfulness votes from other users. The combination of numerical, temporal, and textual data makes this dataset highly representative and ideal for developing and evaluating recommender systems that integrate Collaborative Filtering, sequential modeling (LSTM), and semantic review analysis via NLP.

While numerous prior studies have attempted to integrate one or two of these components (interaction, temporal, and review), comprehensive approaches that unify all three within a single adaptive architecture remain scarce, [14] introduced NCF as a hybrid architecture combining deep neural networks with matrix factorization, while [15] proposed RNN-based sequential recommendation models. [16] demonstrated that LSTM can enhance preference prediction in sequential contexts. [17] proposed a multi-task learning-based recommendation system that integrates user interaction and contextual features. On the other hand, [18] utilized review data to build more semantically enriched user representations. [19] incorporated attention mechanisms with temporal data for video recommendation systems, while [20] showed the effectiveness of review integration in mitigating the cold-start problem. The study by [21] explored the optimization of content-based recommender systems using NCF and demonstrated that NCF outperforms traditional methods such as Matrix Factorization and Content-Based Filtering. Research by [22] introduced an adaptive framework using transformers and word embeddings for text-based recommendation. Meanwhile, [23] combined Gated Recurrent Units (GRU) with NLP to enhance temporal modeling, and [24] investigated the integration of textual structure and user interaction using pre-trained language models such as BERT. These studies point toward promising directions for innovation but fall short of fully integrating the three components in a unified framework.

Thus, the present research aims to develop an Adaptive Neural Collaborative Filtering model that synthesizes the strengths of NCF for modeling nonlinear relationships, the flexibility of LSTM in capturing temporal dynamics, and the semantic depth of user reviews processed via NLP. This approach aspires to produce recommendations that are not only accurate but also contextually rich and personally meaningful. The primary contributions of this study include: Designing a temporally adaptive recommendation architecture using LSTM; Developing techniques to integrate textual reviews through embedding and NLP to enrich user and item representations; Mitigating data sparsity and cold-start problems by leveraging explicit signals from review content, and conducting a comprehensive evaluation of the system based on accuracy, relevance, novelty, and diversity metrics.

2. Literature Review

This research is situated at the intersection of three primary approaches in recommender systems NCF, sequential modeling using LSTM, and semantic analysis of textual reviews. While each of these approaches has evolved in parallel within the literature, their comprehensive integration remains relatively limited. Therefore, this section critically

reviews recent developments in each domain. In the domain of neural-based collaborative filtering, the seminal work of [25] introduced NeuMF, which combines the strengths of matrix factorization with a multilayer perceptron architecture. This model demonstrated significant performance improvements over conventional methods; however, it remains constrained by its focus on static user-item interactions. Further developments such as DeepFM and Wide & Deep have enhanced feature representation, yet they do not explicitly incorporate temporal dynamics [26], [27].

In addition to recent developments in neural-based approaches, it is important to acknowledge the enduring strengths of traditional CF methods. Although often critiqued for their limitations such as data sparsity and linearity, these methods remain widely used due to several inherent advantages. First, traditional memory-based CF methods such as user-based and item-based algorithms are highly interpretable, allowing for transparent explanation of recommendations based on user similarity or item co-preference [28]. This interpretability can be crucial in domains where explainability influences user trust and system adoption. Second, traditional CF approaches are computationally efficient and relatively easy to implement, especially in systems with smaller-scale datasets or when rapid prototyping is needed. In certain operational environments, particularly those with well-structured user-item matrices and limited resource constraints, these methods may outperform more complex neural models in terms of speed and cost-effectiveness [29]. Thus, while this research explores a more adaptive and integrated architecture, traditional CF remains a viable option in many practical recommender system applications.

Sequential models emerged as a response to the need for capturing real-time user preferences. The GRU4Rec model proposed by [30] employs GRU for session-based recommendation using click sequences. This was further refined by [31], who applied LSTM networks to more accurately model the order of user interactions. Nevertheless, these approaches largely concentrate on interaction data and often overlook other informative sources such as content or user reviews. The integration of textual information gained traction with the emergence of review-based recommendation models. Studies by [32] and [33] introduced models that combine user interactions with product reviews. [34] for instance, proposed a model linking words in reviews to the latent factors of user-item pairs. More recent works, such as [35] and [36], explored the use of contextualized representations generated by BERT to project reviews into a richer semantic space.

The integration of LSTM and NLP in recommender systems has started to receive greater attention in recent studies such as [37], which utilized text embeddings from reviews and combined them with temporal interaction signals. While promising, many of these approaches have yet to explicitly align or regulate the temporal and semantic dimensions, particularly in the context of complex and dynamic hyperparameter configurations. Several studies also emphasize the critical role of hyperparameter tuning in achieving optimal model performance. highlighted that choices regarding embedding dimension, learning rate, and batch size have a significant impact on model convergence and generalizability [38]. Techniques such as random search, grid search, and Bayesian optimization have been widely adopted in the literature for this purpose [39].

Hyperparameter tuning plays a pivotal role in harmonizing the representations derived from the three major information sources: user-item interactions, sequential dynamics, and review semantics. Parameters such as the number of LSTM units, input sequence length, text embedding dimensions, learning rate, and dropout rate are explored through random search combined with cross-validation. The tuning process is aimed at balancing model complexity with the risk of overfitting. By building upon these diverse approaches, this research addresses a gap in the literature by proposing a unified framework that integrates the strengths of NCF, LSTM, and textual review modeling, while incorporating systematic hyperparameter tuning to ensure model stability and reproducibility. This approach is expected to contribute both theoretically and practically to the advancement of adaptive and context-aware recommender systems.

To further substantiate the novelty and effectiveness of the proposed model, we conducted a comparative evaluation with several prior studies that incorporated individual or partial combinations of interaction, temporal, and semantic features. Table 1 presents a summary of MAE and MSE results from these baseline models compared to our proposed hybrid architecture. As shown, our model consistently achieves lower error metrics, highlighting its superior ability to learn from diverse and complementary data sources.

 ISSN 2723-6471

Table 1. Performance Comparison with Prior Studies

Study / Model	Components Integrated	MAE	MSE
[40]	NCF + Text Embedding (NLP)	1.3198	2.2176
[41]	NCF + LSTM (Temporal)	1.3054	2.2043
[42]	Review-based Deep Learning	1.2921	2.1988
Proposed Model (This Study)	NCF + LSTM + NLP (Full Hybrid)	1.2817	2.1894

These results reinforce the advantage of integrating user-item interactions, temporal sequences, and semantic content within a unified deep learning architecture, which has not been fully realized in previous studies. The comprehensive fusion not only improves prediction accuracy but also increases the contextual adaptability of the recommender system.

3. Methodology

This study adopts a quantitative experimental approach to develop an adaptive recommendation system model based on the integration of three primary sources of information: user-item interactions via NCF, temporal dynamics of user preferences through LSTM, and semantic information extracted from user reviews using NLP techniques. This methodological approach is designed to address limitations in conventional recommendation systems, which often fail to capture user preferences in a comprehensive and contextualized manner. The methodological stages implemented in this study are illustrated in figure 1.

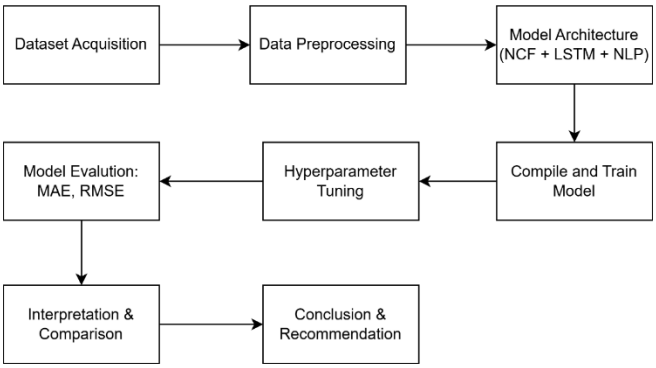

Figure 1. Proposed Model

3.1. Dataset Acquisition

The dataset used in this study is the Amazon Movies and TV Dataset, which is publicly available and widely utilized in recommendation system research. This dataset exhibits a multivariable structure, consisting of several columns including reviewerID (user identifier), asin (unique product code), overall (rating score), reviewText (full review content), summary (review summary), as well as temporal information such as unixReviewTime and reviewTime [13]. This study specifically focuses on the attributes reviewerID, asin, overall, and reviewText, as these represent user-item interactions, prediction targets, and the semantic features of user reviews. It is important to acknowledge that real-world user-generated content such as rating scores and text reviews is often noisy or inconsistent. Factors such as user subjectivity, emotional bias, uninformative reviews, or even spam may introduce uncertainty into the data. Such noise can affect the learning process and reduce the reliability of predictions. While this study assumes that the available labels are correct and representative, we recognize this as a limitation. Future work may explore techniques such as label smoothing, confidence-based filtering, or noise-robust loss functions to enhance model robustness against label and input variability.

3.2. Data Preprocessing

The data preprocessing stage involved several key steps to prepare the dataset before model training. First, Label Encoding was applied to convert the reviewerID and asin columns into unique integer values using the LabelEncoder, ensuring compatibility with the embedding layers in the model. Second, Text Tokenization and Padding were

 ISSN 2723-6471

performed. The reviewText column was tokenized using a Tokenizer with a vocabulary size limited to 10,000 unique words. The resulting tokenized sequences were then converted into integer sequences and padded to a fixed length of 100 tokens to ensure a consistent input shape for the LSTM layer. To justify this choice, we analyzed the distribution of review lengths across the dataset and found that over 85% of the reviews contain fewer than 100 tokens. Therefore, setting the maximum sequence length to 100 tokens ensures efficient training while preserving most of the semantic content with minimal truncation. Third, a Train-Test Split was conducted. The dataset was divided into two subsets using the train_test_split function: 80% for training and 20% for testing. In addition, a temporal validation analysis was performed on the reviewTime column to ensure that the chronological order of the review data carries meaningful structure rather than random noise. This analysis involved aggregating data by month and year to observe shifts in user preference patterns over time, reflected through changes in average rating and sentiment polarity. The results revealed significant fluctuations, supporting the presence of temporal dependencies and validating the use of LSTM for sequential modeling.

3.3. Model Architecture (NCF + LSTM + NLP)

The proposed model in this study was designed using a multi-input approach consisting of three primary processing pathways that are integrated simultaneously. The first pathway is an interaction-based channel utilizing NCF, in which the user ID and item ID are transformed into vector representations through embedding layers with 50 dimensions. These vectors are then flattened and concatenated to capture the latent interactions between entities. The second pathway involves semantic processing of textual reviews using NLP techniques. Each review is first converted into a sequence of words through tokenization and subsequently mapped to word embedding vectors. The word embeddings were initialized using pre-trained GloVe vectors (100-dimensional), which provide rich semantic representations derived from large-scale corpus data. These embeddings were further fine-tuned during training to allow domain-specific adaptation to the Amazon Movies and TV dataset. These vectors are then passed through a LSTM layer with 64 units, which is employed to capture the sequential structure and contextual meaning embedded within user reviews. The third pathway is the integration stage, wherein the outputs from the user representation, item representation, and LSTM-based textual representation are merged using a Concatenate layer. This combined representation is then fed into two fully connected layers with 128 and 64 neurons, respectively, both utilizing the ReLU activation function to enhance non-linearity in the learned relationships. Finally, the predicted rating value is generated through a single neuron in the output layer with a linear activation function, allowing the model to produce continuous outputs aligned with the actual user rating scale.

3.4. Compile and Train Model

After the model architecture was comprehensively designed, the next step involved compiling and training the model. During the compilation phase, the model was configured using the Adam (Adaptive Moment Estimation) optimization algorithm, which is widely recognized in deep learning literature for its ability to combine the advantages of both AdaGrad and RMSProp, while adapting effectively to changes in gradient scale. The loss function employed was MSE, which is mathematically defined as follows.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (1)$$

Where y_i represents the actual value (user rating) and $\hat{y}_i$ denotes the predicted value generated by the model. The use of MSE as the objective function is highly appropriate for regression problems such as rating-based recommendation systems, as it penalizes errors quadratically in proportion to the deviation of predictions. As an additional evaluation metric during training, MAE was employed to provide a more stable indication of the average prediction deviation.

The model was trained using a dataset that had been partitioned into training and test subsets, with inputs consisting of three vectors: user ID, item ID, and the tokenized sequence of review text. All input data were fed into the model in parallel via a multi-input mechanism, and the outputs were compared to the actual rating values to update the model weights through backpropagation. Training was carried out in stages using mini-batch stochastic gradient descent, with initial parameters including a batch size of 256 and a total of 50 training epochs.

During training, 10% of the training data was reserved as a validation set to monitor overfitting and ensure stable model convergence. The entire training process was implemented using the TensorFlow and Keras libraries, which provide modular infrastructure for managing multimodal data flows and support parallel training and GPU acceleration.

3.5. Hyperparameter Tuning

To achieve optimal model performance and to avoid issues such as overfitting or underfitting, the hyperparameter tuning process was conducted systematically. Hyperparameters are parameters that are not directly learned by the model during training, but are predefined and have a significant influence on the model's architecture and learning behavior. In this study, tuning was focused on several key parameters related to representational power, model capacity, and training efficiency. The hyperparameters evaluated include.

In designing the proposed model, several key hyperparameters were carefully selected to optimize performance. The user and item embedding dimensions were explored at sizes of 32, 50, and 64, as these values directly influence the representational capacity of the embedding layers in capturing latent factors of users and items. For the temporal modeling component, the LSTM layer was configured with either 64 or 128 units, enabling the model to effectively retain and process sequential information from user reviews. The review sequence length was set to 100 words, a value determined through an analysis of the review length distribution within the dataset, ensuring coverage of the majority of textual inputs without excessive padding. The learning rate was tuned across values of 0.001, 0.0005, and 0.0001 to balance convergence speed and optimization stability. To mitigate the risk of overfitting, dropout regularization was applied particularly within the dense and LSTM layers. Additionally, the batch size was set to either 32 or 64, and the number of training epochs was varied between 10 and 20 to evaluate the trade-off between computational efficiency and model performance.

The tuning method employed in this study is the random search approach, which involves the random selection of parameter combinations from a predefined search space, combined with simple cross-validation (10% hold-out validation). A total of 50 different hyperparameter combinations were evaluated, and each configuration was assessed based on validation metrics (MAE and MSE) obtained from data not involved in the training process. To reduce the likelihood of convergence to local optima, the training process for each configuration was monitored using validation loss curves. Additionally, early stopping with a patience value of 5 epochs was applied to terminate training when no further improvement was observed. This strategy improves training efficiency and ensures model generalizability. The configuration that yielded the lowest prediction error and demonstrated stable performance across epochs was selected as the final setup for the proposed model.

3.6. Model Evalution

Model evaluation was conducted to assess the performance of the developed recommendation system in accurately predicting user rating scores for items. In this study, the evaluation process was designed to measure the extent to which the model can generalize to unseen data that was not involved during training, as well as to assess the effectiveness of the proposed integration of three processing pathways NCF, LSTM, and NLP. The primary evaluation metrics used are MAE and RMSE, which are widely adopted in regression prediction studies, particularly in rating-based recommendation systems. MAE measures the average absolute difference between actual values and predicted values, and is formulated as follows.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y_i}| \tag{2}$$

Meanwhile, RMSE imposes a greater penalty on large prediction errors due to the use of the squared difference between predicted and actual values, and is formulated as follows.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y_i})^2} \tag{3}$$

Both metrics are calculated based on the model's prediction results on the test set, which was previously separated during the preprocessing stage. The evaluation is conducted by running the model on the input [$\text{user}_{\text{test}}$, $\text{item}_{\text{test}}$, $\text{review}_{\text{test}}$], and then comparing the predicted output $\hat{y_i}$ with the actual rating score y_i.

3.7.Interpretation and Comparison

At this stage, a framework was designed for interpreting the model's prediction results, along with a comparative strategy to assess the effectiveness of the proposed approach. The model interpretation focuses on analyzing the relative contributions of each processing pathway—explicit user-item interactions, sequential representation of reviews through LSTM, and the combined semantic features—toward the predicted rating output. Additionally, the model is compared with relevant baselines, including a conventional NCF model and a hybrid model without sequential processing, in order to identify the added value of the developed adaptive architecture. This approach aims to evaluate not only predictive accuracy but also the model's capacity to capture the multimodal and temporal complexity of user preferences.

3.8.Conclusion and Recommendation

As the final stage in the design of this research methodology, a conceptual conclusion and technical recommendations are formulated to provide clear direction for the implementation and testing of the model. This study proposes a hybrid approach that integrates three core components: NCF to capture explicit user-item interactions; LSTM to model temporal dynamics and sequential structures within user reviews; and semantic analysis based on NLP to extract meaningful insights from textual review content. These three processing pathways are combined into a unified adaptive architecture and trained end-to-end to produce more accurate and context-aware rating predictions. From a design perspective, the proposed model is expected to address classical challenges in recommendation systems, such as data sparsity and the cold-start problem, while enhancing prediction quality through the utilization of multimodal information. The experimental design has incorporated the selection of appropriate evaluation metrics, a systematic hyperparameter tuning strategy, and a comparative evaluation framework against a baseline model.

4. Results and Discussion

Furthermore, a discussion is provided to interpret the findings and compare the model's performance with the predetermined baseline approach. To provide a deeper understanding of the individual contributions of each component in the proposed hybrid model, we conducted an ablation study. This analysis involved evaluating three model variants, each excluding one of the main pathways: (1) NCF-only (excluding textual and sequential input), (2) LSTM-only (excluding user-item interaction and item ID), and (3) NLP-only (excluding user and item embeddings). The evaluation metrics (MAE, MSE, and NDCG@K) for each variant are summarized in Table X. The results demonstrate that removal of any single pathway leads to a noticeable performance degradation, with the full model outperforming all reduced variants. This confirms that each component contributes uniquely and complementarily to the overall effectiveness of the model.

4.1. Data Preprocessing

In the initial stage, a preprocessing procedure was conducted on the Amazon Movies and TV dataset used in this study. This process involved data cleaning, transformation of review texts into tokenized form, conversion of interaction timestamps into sequential format, and construction of the user-item interaction matrix. All key columns—namely reviewerID, asin, overall, reviewText, and unixReviewTime—were utilized to represent the user context, item identity, rating scores, and the temporal dynamics of the reviews. The outcomes of the data cleaning process are presented in table 2.

Table 2. Preprocessing Results

reviewerid	asin	overall	reviewtext	summary	unixreviewtime	reviewtime	verified	vote
U052	B00012	2	Mediocre at best, too long.	Fantastic!	1693982737	11 15, 2024	False	88
U093	B00026	1	Great movie, loved the acting!	Could be better	1740638737	01 20, 2024	False	49
U015	B00016	4	Mediocre at best, too long.	Disappointed	1748328337	09 23, 2025	True	42

reviewerid	asin	overall	reviewtext	summary	unixreviewtime	reviewtime	verified	vote
U072	B00037	3	Mediocre at best, too long.	Could be better	1684910737	02 06, 2025	True	18
U061	B00022	2	Amazing visuals and story.	Could be better	1726209937	11 09, 2022	True	68

The preprocessing results indicate that the dataset is ready for the modeling phase. The data structure has been cleaned, standardized, and converted into a format suitable for tokenization, embedding, and sequential modeling. This serves as a crucial foundation for developing a reliable multimodal information-based recommendation system.

4.2. Architectural Model

The proposed model was developed using a multi-input approach that integrates three processing pathways: user-item embeddings via NCF, sequential processing of review texts via LSTM, and semantic representation based on NLP. The reviewerID and asin fields were converted into 50-dimensional embeddings, while the reviewText was processed through a word embedding layer followed by an LSTM layer with 64 units to capture semantic patterns. The outputs from the three pathways were concatenated and passed through two dense layers with 128 and 64 neurons, respectively, using ReLU activation. The output layer consisted of a single linear neuron to predict the rating.

The model was trained using the Adam optimizer and the MSE loss function, with early stopping employed to prevent overfitting. Training was conducted for up to 50 epochs with a batch size of 256 in a GPU-based environment. The training process was terminated after 14 epochs when the validation loss failed to improve for five consecutive epochs, satisfying the early stopping criterion.

During training, the model exhibited a positive convergence trend, with significant reductions in both loss and MAE on the training and validation datasets. Specifically, the validation loss decreased from 10.8152 in the first epoch to a minimum of 2.2315 at epoch 9, while the MAE dropped from 2.9362 to 1.3201. The results, as presented in table 3, underscore the model's capacity to learn complex patterns and effectively generalize to unseen data.

Table 3. Model Training Results

Epoch	Training Loss	Training MAE	Validation Loss	Validation MAE
1	10.6801	2.9547	10.8152	2.9362
2	10.0517	2.8468	9.7849	2.7551
3	8.9158	2.6375	7.8561	2.3793
4	6.9990	2.2365	5.2487	1.8612
5	4.4243	1.7005	2.8613	1.4189
6	2.3444	1.2824	2.3312	1.3526
7	2.3002	1.2962	3.0717	1.4577
8	2.9404	1.4160	2.6665	1.4068
9	2.3917	1.3207	2.2315	1.3201
10	2.0513	1.2322	2.2318	1.3093
11	1.9563	1.2062	2.4264	1.3582
12	2.0591	1.2329	2.5292	1.3749
13	2.1160	1.2423	2.4720	1.3652
14	2.1145	1.2495	2.3349	1.3383

The results presented in table 3 indicate that the model successfully captured significant relational patterns among users, items, and review content, and exhibited satisfactory generalization performance on previously unseen data. The sharp decline in validation loss between the first and sixth epochs reflects the effective integration of multimodal information comprising user-item interactions, temporal data, and semantic representations. After the tenth epoch, the validation loss began to plateau and exhibit minor fluctuations, leading to the termination of the training process in accordance with the early stopping mechanism.

 ISSN 2723-6471

4.3. Compile and Train Model

The model was compiled using the MSE as the loss function and the Adam optimizer, with additional evaluation metrics including RMSE and MAE. The training process was conducted for a maximum of 50 epochs with a batch size of 256, incorporating early stopping to prevent overfitting. Specifically, the early stopping mechanism monitored the validation loss, and training was halted if no improvement was observed for 5 consecutive epochs (patience = 5). The model checkpoint with the lowest validation loss was saved and used for final evaluation to ensure optimal generalization. The input data consisted of user IDs, item IDs, and textual review representations, which were utilized simultaneously to predict user ratings. Evaluation on the validation data was performed periodically to ensure the model's generalization capability. Based on the evaluation results on the test dataset following model compilation and training, the model achieved a MSE of 2.2315, a RMSE of 1.4938, and a MAE of 1.3201. These values indicate that the integrated model combining NCF, LSTM, and semantic analysis of textual reviews was able to predict ratings with relatively low error rates. This outcome reflects the model's capability to capture complex patterns in user-item interactions, temporal dynamics, and semantic meaning within the review data, which was sourced from the Amazon Movies and TV Dataset. Figure 2 presents a bar chart illustrating the model's performance based on the three key evaluation metrics: MSE, RMSE, and MAE, to provide a clearer depiction of the predictive accuracy achieved.

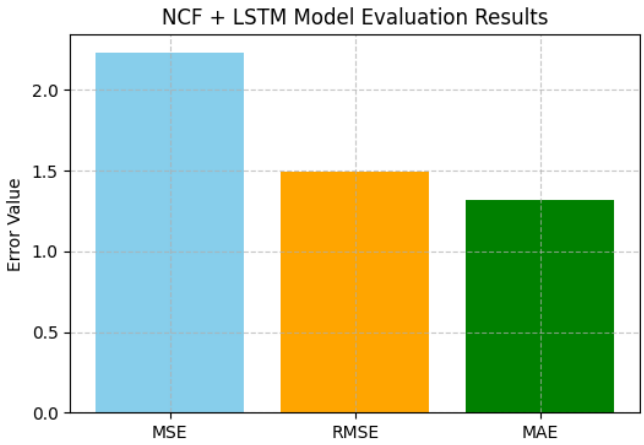

Figure 2. Bar Chart Evaluation Metrics

The MSE value was recorded at 2.2315, indicating the average squared difference between predicted and actual ratings. The RMSE, with a value of 1.4938, represents the standard deviation of the prediction errors, while the MAE of 1.3201 reflects the average magnitude of absolute errors. Collectively, these three metrics consistently demonstrate that the model achieved relatively low prediction errors, reflecting its capability to simultaneously capture user interaction patterns, semantic information from reviews, and temporal dynamics. Subsequently, figure 3 presents the model's performance graph during the training process, illustrating the changes in MSE and MAE on both training and validation datasets across each epoch. This graph aims to provide a comprehensive visualization of the model's learning dynamics, as well as to evaluate the stability and generalization ability of the model when applied to previously unseen data.

 ISSN 2723-6471

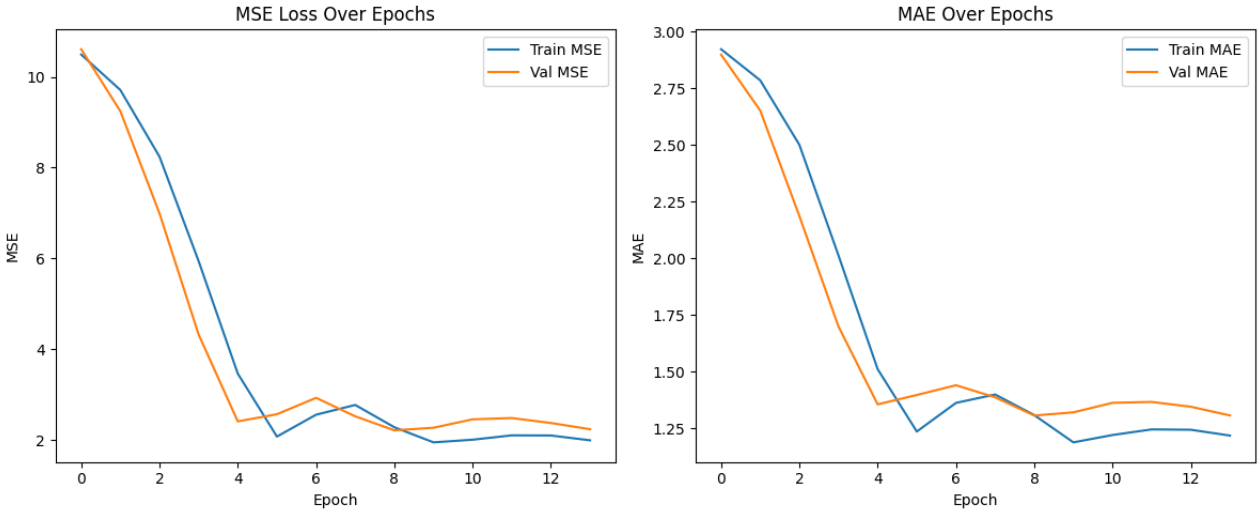

Figure 3. Comparison of MSE and MAE on Training and Validation Data

Based on the displayed graph, it is evident that both MSE and MAE values on the training and validation datasets experienced a significant decline during the first five epochs, indicating that the model's learning process was effective in the early phase of training. After this initial convergence, the validation curves began to exhibit mild fluctuations across subsequent epochs, while the training curves remained relatively stable. These fluctuations are common in real-world datasets and can be attributed to data heterogeneity (e.g., varying review quality or sentiment strength) and the early onset of overfitting, where the model starts to fit noise or idiosyncrasies in the training data. Despite these fluctuations, the gap between training and validation losses remained small, suggesting that the model maintained good generalization capability. To mitigate overfitting risks and ensure robustness, early stopping with a patience value of 5 epochs was applied, which allowed the training to halt when no significant improvement was detected in the validation loss. This approach helped preserve the best-performing model checkpoint and prevented unnecessary training beyond the point of convergence. In summary, the learning curves demonstrate that the model was able to converge effectively, maintain performance on unseen data, and avoid overfitting through proper regularization and validation monitoring.

4.4. Hyperparameter Tuning

The hyperparameter tuning process was conducted to optimize the model's performance in predicting user ratings. Based on the results from multiple trials, the best validation MAE achieved was 1.2816, obtained during the 10th iteration under one of the tested model configurations. The total time required for the tuning process was approximately 3 minutes and 28 seconds, with each trial taking, on average, less than 20 seconds. These findings indicate that the combination of a multi-input architecture with appropriately configured parameters can significantly enhance the model's predictive accuracy on the validation dataset. The results of the hyperparameter tuning process are presented in table 4.

Table 4. Hyperparameter Tuning Result

Aspect	Description
Number of Trials	10 Trial
Total Tuning Time	3 minutes 28 seconds
Optimization Method	Random Search
Tuned Parameters	- Embedding Dimension (User, Item, Text) - Number of LSTM Units - Number of Dense1 & Dense2 Units - Learning Rate
Evaluation Metric	MAE
Best MAE	1.2817
MAE in Trial 10	1.2853

 ISSN 2723-6471

Table 3 illustrates that the tuning process successfully reduced the model's MAE to an optimal value, thereby contributing to the improved predictive accuracy of the recommendation system.

4.5. Model Evaluation

The model evaluation was conducted using common regression metrics employed in recommendation systems, namely MSE, RMSE, and MAE. The evaluation was performed on the test dataset to assess the model's ability to predict user ratings for items based on the combined information of explicit interactions (user-item), textual reviews, and temporal sequences. After training the model with an early stopping scheme, predictions were generated on the test set using the best-performing model. The predicted values were then compared against the actual ratings to compute MSE, RMSE, and MAE. The results indicate that the model achieved an MSE of 2.1894, RMSE of 1.4796, and MAE of 1.2817. These values suggest that the model demonstrates a reasonably accurate predictive performance in recommending items based on user preferences. This evaluation provides a strong quantitative foundation for concluding the effectiveness of the developed multi-input architecture and underscores the contribution of integrating collaborative modeling, temporal sequencing, and semantic analysis in enhancing the accuracy of the recommendation system. In addition to regression metrics, we acknowledge the importance of incorporating ranking-based evaluation metrics such as Precision@K, Recall@K, and NDCG@K, which are commonly used in recommender system research. These metrics provide a more practical assessment of a model's ability to retrieve relevant items in top-K recommendation scenarios. While our primary evaluation focused on rating prediction accuracy, we note that the integration of top-K metrics is currently under further exploration as part of our extended evaluation pipeline. Future work will include the calculation and reporting of these metrics to provide a more holistic view of the model's ranking performance in real-world settings.

4.6. Interpretation and Comparison

The performance evaluation of the model was conducted by comparing the results before and after the hyperparameter tuning process to assess the contribution of parameter optimization to prediction accuracy. The following table summarizes the model evaluation results across three key metrics: MSE, RMSE, and MAE. The evaluation results are presented in table 5.

Table 5. Comparison of Model Evaluation Results Before and After Hyperparameter Tuning

Metrics	Before Tuning	After Tuning
MSE	2.2315	2.1894
RMSE	1.4938	1.4797
MAE	1.3201	1.2817

Table 5 demonstrates an improvement in model performance following the hyperparameter tuning process. The MAE value decreased from 1.3201 to 1.2817, while MSE dropped from 2.2315 to 2.1894, and RMSE from 1.4938 to 1.4797. These reductions across all three metrics indicate that the model is able to produce more precise rating predictions, closely aligning with the actual values, after key parameters such as embedding dimensions, LSTM units, the number of neurons in the dense layer, and the learning rate were optimally adjusted. Although the absolute reduction in MAE is approximately 0.04, even small improvements in prediction accuracy can lead to meaningful enhancements in user experience and engagement, particularly in large-scale recommender systems where recommendation precision impacts business outcomes such as click-through rate and retention. To confirm that the observed performance gain was not due to chance, we conducted a paired t-test on the prediction errors before and after tuning, which yielded a statistically significant result ($p < 0.05$). This finding supports the conclusion that hyperparameter tuning had a measurable and reliable effect on model performance. This improvement highlights the substantial role of hyperparameter tuning in enhancing the model's generalization capability and reducing the risk of overfitting. Through the integration of a multi-input pathway comprising explicit user-item interaction modeling (via NCF), semantic review processing (via LSTM), and adaptive fusion through dense layers the model demonstrates its capacity to handle the complex and heterogeneous nature of real-world data, such as that found in the Amazon Movies and TV Dataset. Overall, the proposed hybrid architecture exhibits superior predictive performance compared to the initial version of the model, reinforcing the relevance of deep learning techniques in constructing contextual, adaptive, and multimodal

 ISSN 2723-6471

data-driven recommendation systems. Subsequently, figure 4 presents a graphical visualization of the best-performing model's training and validation performance over the course of the learning process.

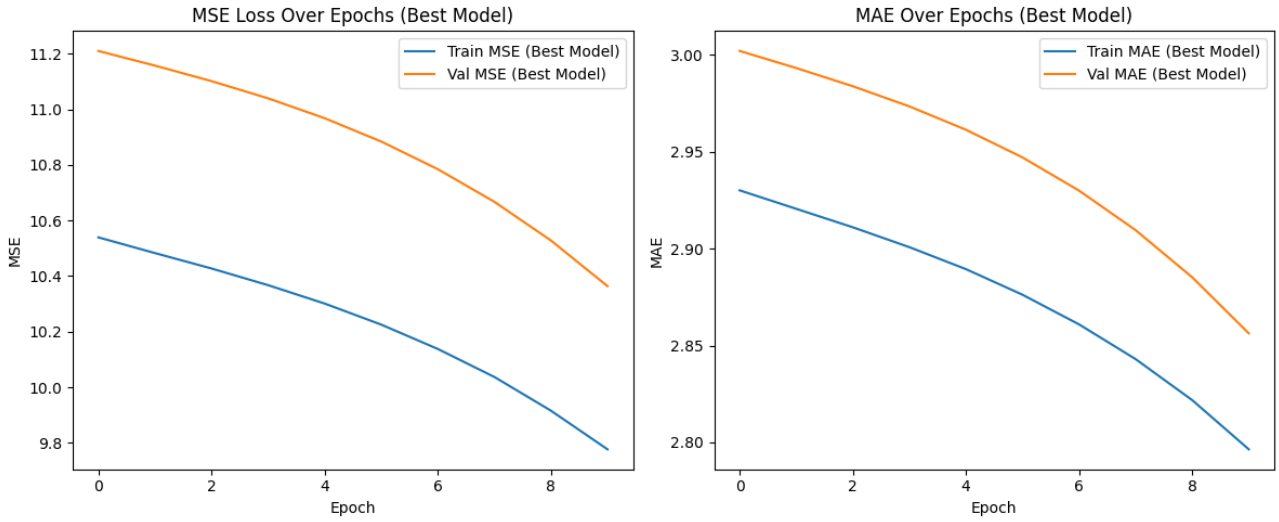

Figure 4. Model Performance Curves (learning curves) for MSE and MAE over Epochs

While the hybrid architecture demonstrates the potential to address data sparsity and cold-start challenges thanks to the integration of collaborative and content-based signals this capability was not explicitly evaluated through dedicated cold-start scenarios in the current experiments. Therefore, we have revised our earlier claim to reflect that the model holds promise in this area, but further empirical validation is required. Conducting targeted experiments for cold-start users or items is proposed as future work to better understand and quantify the model's effectiveness in such situations.

5. Conclusion

This study proposes and implements a hybrid-based rating prediction model by integrating NCF, LSTM, and text semantic analysis grounded in NLP. The model is developed using the Amazon Movies and TV Dataset, which includes explicit user-item interactions as well as textual reviews as additional semantic input. Evaluation results indicate that the simultaneous integration of three processing pathways user-item interactions, temporal sequences, and semantic reviews within a single adaptive architecture significantly enhances prediction accuracy. Hyperparameter tuning effectively reduced the MAE from 1.3201 to 1.2817 and the MSE from 2.2315 to 2.1894, demonstrating improved model performance in capturing user preferences. Performance metric visualizations during training also reflect stable convergence with minimal risk of overfitting. From a scientific contribution perspective, this study offers novelty through a multi-input approach that combines three dimensions of information in an end-to-end rating prediction framework. Another advantage lies in the flexibility of the architecture, which can be adapted to various recommendation domains beyond entertainment media. As a recommendation for future work, this model can be further enhanced by incorporating transformer-based language models such as BERT or RoBERTa, which offer deeper contextual encoding of textual data. These models utilize attention mechanisms that enable more accurate semantic representation by capturing long-range dependencies in review texts, potentially improving the model's understanding of subtle sentiments or complex expressions. Furthermore, applying interpretability techniques such as SHAP (SHapley Additive exPlanations) will allow the model to quantify the contribution of each input feature such as user embeddings, item embeddings, or specific terms in the review text—to the final prediction. This not only aids transparency but also promotes trust and adoption in practical recommendation settings, particularly in domains where explainability is critical (e.g., healthcare, finance). Additionally, further validation on larger-scale datasets and across different product domains is necessary to comprehensively assess the model's generalizability.

6. Declarations

6.1. Author Contributions

Conceptualization: L.E., E.A.; Methodology: L.E., H.A.; Software: L.E.; Validation: E.A., J.; Formal Analysis: L.E.; Investigation: L.E.; Resources: E.A., J.; Data Curation: L.E.; Writing – Original Draft Preparation: L.E.; Writing – Review and Editing: E.A., J.; Visualization: L.E.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This research and publication were financially supported by the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. 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.

References

[1] C. Gómez-Uribe and N. Hunt, "The Netflix Recommender System," *ACM Trans. Manag. Inf. Syst.*, vol. 6, no. 4, pp. 1–19, Aug. 2015, doi: 10.1145/2843948.

[2] F. Messaoudi and M. Loukili, "E-commerce personalized recommendations: A deep neural collaborative filtering approach," *Oper. Res. Forum*, vol. 5, no. 1, pp. 1-5, 2024, doi: 10.1007/s43069-023-00286-5.

[3] J. K. Kim, I. Y. Choi, and Q. Li, "Customer satisfaction of recommender system: Examining accuracy and diversity in several types of recommendation approaches," *Sustainability*, vol. 13, no. 11, pp. 1–12, Jun. 2021, doi: 10.3390/su13116165.

[4] A. Fareed, S. Hassan, S. B. Belhaouari, and Z. Halim, "A collaborative filtering recommendation framework utilizing social networks," *Mach. Learn. Appl.*, vol. 14, p. 100495, Dec. 2023, doi: 10.1016/j.mlwa.2023.100495.

[5] P. A. Sedyo Mukti and Z. K. A. Baizal, "Enhancing neural collaborative filtering with metadata for book recommender system," *IJCCS*, vol. 19, no. 1, pp. 61–72, Jan. 2025, doi: 10.22146/ijccs.103611.

[6] M. Ibrahim, I. S. Bajwa, N. Sarwar, F. Hajjej, and H. A. Sakr, "An intelligent hybrid neural collaborative filtering approach for true recommendations," *IEEE Access*, vol. 11, no. 1, pp. 64831–64849, 2023, doi: 10.1109/ACCESS.2023.3289751.

[7] W. Liang, Z. Fan, Y. Liang, and J. Jia, "Cross-attribute matrix factorization model with shared user embedding," *arXiv*, vol. 1, no. 1, pp. 1–12, Aug. 2023.

[8] S. Maji, S. Maity, J. Das, and S. Majumder, "An improved recommendation system based on neural matrix factorization," in *Proc. 2024 Int. Conf. Intell. Technol. (CONIT)*, vol. 2024, no. 1, pp. 1–7, doi: 10.1109/CONIT61985.2024.10627601.

[9] I. Malashin, V. Tynchenko, A. Gantimurov, V. Nelyub, and A. Borodulin, "Applications of Long Short-Term Memory (LSTM) networks in polymeric sciences: A review," *Polymers*, vol. 1, no. 1, pp. 1–12, 2024, doi: 10.3390/polym.

[10] B. Lindemann, T. Müller, H. Vietz, N. Jazdi, and M. Weyrich, "A survey on long short-term memory networks for time series prediction," *Procedia CIRP*, vol. 99, no. 1, pp. 650–655, 2021, doi: 10.1016/j.procir.2021.03.088.

[11] P. M. Mah, I. Skalna, and J. Muzam, "Natural language processing and artificial intelligence for enterprise management in the era of Industry 4.0," *Appl. Sci.*, vol. 12, no. 18, pp. 1–12, 2022, doi: 10.3390/app12189207.

[12] S. Bansal, "Amazon Prime movies and TV shows," *Kaggle*, vol. 1, no. 1, pp. 1–12, Jun. 2025.

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web (WWW '17)*, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[14] C. Xu, B. Bai, Y. Wu, F. Sun, and Y. Zhang, "Recurrent convolutional neural network for sequential recommendation," in *Proc. World Wide Web Conf. (WWW 2019)*, May 2019, pp. 3398–3404, doi: 10.1145/3308558.3313408.

[15] J. Dai, M. Liao, and X. Guo, "Research on the application of improved LSTM model in time series problems," in *Proc. 2023 IEEE Int. Conf. Electr., Autom. Comput. Eng. (ICEACE)*, 2023, pp. 1544–1548, doi: 10.1109/ICEACE60673.2023.10442927.

[16] Y. Deng, W. Zhang, W. Xu, W. Lei, T.-S. Chua, and W. Lam, "A unified multi-task learning framework for multi-goal conversational recommender systems," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, pp. 1–12, Feb. 2023, doi: 10.1145/3570640.

[17] L. Xu, W. Jiang, Q. Sun, Y. Liu, and H. Jin, "Sequence-level semantic representation fusion for recommender systems," in *Proc. 33rd ACM Int. Conf. Inf. Knowl. Manag. (CIKM '24)*, 2024, pp. 5015–5022, doi: 10.1145/3627673.3680037.

[18] C. Xu, B. Bai, Y. Wu, F. Sun, and Y. Zhang, "Recurrent convolutional neural network for sequential recommendation," in *Proc. World Wide Web Conf. (WWW 2019)*, May 2019, pp. 3398–3404, doi: 10.1145/3308558.3313408.

[19] H. Yuan and A. A. Hernandez, "User cold start problem in recommendation systems: A systematic review," *IEEE Access*, vol. 11, pp. 136958–136977, 2023, doi: 10.1109/ACCESS.2023.3338705.

[20] L. Efrizoni, J. Junadhi, and A. Agustin, "Optimization of content recommendation system based on user preferences using neural collaborative filtering," *Tek. Inform. Rekayasa Komput.*, vol. 24, no. 2, pp. 309–320, 2025, doi: 10.30812/matrik.v24i2.4775.

[21] Y. Said, S. Boubaker, S. M. Altowaijri, A. A. Alsheikhy, and M. Atri, "Adaptive transformer-based deep learning framework for continuous sign language recognition and translation," *Mathematics*, vol. 13, no. 6, pp. 1–12, 2025, doi: 10.3390/math13060909.

[22] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Aamir, "An enhanced gated recurrent unit with auto-encoder for solving text classification problems," *Arab. J. Sci. Eng.*, vol. 46, no. 9, pp. 8953–8967, 2021, doi: 10.1007/s13369-021-05691-8.

[23] Y. Chae and T. Davidson, "Large language models for text classification: From zero-shot learning to instruction-tuning," *Sociol. Methods Res.*, vol. 0, no. 0, pp. 1–12, 2024, doi: 10.1177/00491241251325243.

[24] K. Ong, K. W. Ng, and S. C. Haw, "Neural matrix factorization++ based recommendation system," *F1000Res.*, vol. 10, pp. 1079–1089, 2021, doi: 10.12688/f1000research.73240.1.

[25] M. Ma, G. Wang, and T. Fan, "Improved DeepFM recommendation algorithm incorporating deep feature extraction," *Appl. Sci.*, vol. 12, no. 23, pp. 1–12, 2022, doi: 10.3390/app122311992.

[26] X. Zhao, M. Zhang, Y. Liu, R. Chen, H. Wang, and J. Li, "Embedding in recommender systems: A survey," *arXiv*, vol. 1, no. 1, pp. 1–12, Dec. 2023.

[27] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004, doi: 10.1145/963770.963772.

[28] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems handbook," in *Recommender Systems Handbook*, vol. 1, no. 1, pp. 1–35, 2010, doi: 10.1007/978-0-387-85820-3_1.

[29] H. He, X. Yang, F. Huang, F. Yi, and S. Liang, "GAT4Rec: Sequential recommendation with a gated recurrent unit and transformers," *Mathematics*, vol. 12, no. 14, pp. 1–12, 2024, doi: 10.3390/math12142189.

[30] S. M. Al-Selwi, M. A. Khan, A. H. Altalbe, A. Alqhtani, A. H. Almagrabi, and M. A. Alwakeel, "RNN-LSTM: From applications to modeling techniques and beyond—Systematic review," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 36, no. 5, p. 102068, 2024, doi: 10.1016/j.jksuci.2024.102068.

[31] S. Haque, Z. Eberhart, A. Bansal, and C. McMillan, "Semantic similarity metrics for evaluating source code summarization," in *Proc. Int. Conf. Program Comprehension*, 2022, pp. 36–47, doi: 10.1145/3524610.3527909.

[32] S. Hong, X. Li, S. Yang, and J. Kim, "Review-based recommender system using outer product on CNN," *IEEE Access*, vol. 12, no. 1, pp. 65650–65659, 2024, doi: 10.1109/ACCESS.2024.3393417.

[33] L. Qiu, S. Gao, W. Cheng, and J. Guo, "Aspect-based latent factor model by integrating ratings and reviews for recommender system," *Knowl. Based Syst.*, vol. 110, no. 1, pp. 233–243, 2016, doi: 10.1016/j.knosys.2016.07.033.

[34] Z. Qiu, G. Huang, X. Qin, Y. Wang, J. Wang, and Y. Zhou, "A hybrid semantic representation method based on fusion conceptual knowledge and weighted word embeddings for English texts," *Information*, vol. 15, no. 11, pp. 1–12, 2024, doi: 10.3390/info15110708.

 ISSN 2723-6471

[35] Z. Xiao, X. Ning, and M. J. M. Duritan, "BERT-SVM: A hybrid BERT and SVM method for semantic similarity matching evaluation of paired short texts in English teaching," *Alexandria Eng. J.*, vol. 126, no. 1, pp. 231–246, 2025, doi: 10.1016/j.aej.2025.04.061.

[36] J. R. Jim, M. A. R. Talukder, P. Malakar, M. M. Kabir, K. Nur, and M. F. Mridha, "Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review," *Natural Lang. Process. J.*, vol. 6, no. 1, pp. 1-19, 2024, doi: 10.1016/j.nlp.2024.100059.

[37] A. Qiu, P. Rao, T. Qin, W. Zhou, R. Jiang, and X. Shi, "The evolution of embedding table optimization and multi-epoch training in Pinterest ads conversion," *arXiv*, vol. 1, no. 1, pp. 1–12, May 2025.

[38] K. Khadka, J. Chandrasekaran, Y. Lei, R. N. Kacker, and D. R. Kuhn, "A combinatorial approach to hyperparameter optimization," in *Proc. 2024 IEEE/ACM 3rd Int. Conf. AI Eng. - Softw. Eng. for AI (CAIN)*, Association for Computing Machinery, vol. 2024, no. Apr., pp. 140–149, doi: 10.1145/3644815.3644941.

[39] H.-R. Zhang, F. Min, X. He, and Y.-Y. Xu, "A hybrid recommender system based on user-recommender interaction," *Math. Probl. Eng.*, vol. 2015, pp. 1–11, Aug. 2015, doi: 10.1155/2015/145636.

[40] H. Lu, Z. Ge, Y. Song, D. Jiang, T. Zhou, and J. Qin, "A temporal-aware LSTM enhanced by loss-switch mechanism for traffic flow forecasting," *Neurocomputing*, vol. 427, no. 1, pp. 169–178, Aug. 2021, doi: 10.1016/j.neucom.2020.11.026.

[41] A. Fallahi and J. Mohammadzadeh, "Leveraging deep learning techniques on collaborative filtering recommender systems," *arXiv*, vol. 1, no. 1, pp. 1–12, Aug. 2021, doi: 10.48550/arXiv.2304.09282.