Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 25, No. 2, March 2026, pp. 275∼286
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI: 10.30812/matrik.v25i2.5988

Optimizing Content Recommendations Using a Hybrid Filtering Algorithm to Enhance User Relevance and Engagement

Lusiana Efrizoni, Junadhi, Agustin
Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia

Article Info

Article history:

Received December 06, 2025
Revised January 22, 2026
Accepted February 13, 2026

Keywords:

Content Recommendation;
Hybrid Filtering;
Performance Evaluation;
Recommender System;
User Engagement.

ABSTRACT

Recommender systems play an important role in helping users discover relevant content in environments characterized by information overload. However, existing approaches often struggle to balance recommendation relevance and user engagement. Collaborative filtering is constrained by data sparsity and the cold-start problem, whereas content-based methods that rely on textual features may not fully capture dynamic user preferences. This study aims to develop a hybrid deep learning-based recommendation model that improves both recommendation relevance and user engagement. The proposed method integrates collaborative filtering via Neural Matrix Factorization (NeuMF) with content-based filtering via a Long Short-Term Memory (LSTM) text encoder, employing an early-fusion strategy. An experimental research method was applied using synthetic user–item interaction data. Model performance was evaluated using ranking metrics (Precision@10, Recall@10, and NDCG@10) and engagement metrics (Click-Through Rate and Average Completion Ratio). The results show that the hybrid model outperforms the baseline models. It achieves Precision@10 of 0.143, Recall@10 of 0.112, and NDCG@10 of 0.139, which exceed those of both the NeuMF-only and LSTM-only models. In terms of engagement, the hybrid model also records the best performance with a CTR of 0.0017 and an ACR of 0.0090. These findings indicate that integrating user–item interaction patterns with semantic content representations can significantly enhance recommendation quality and user engagement, providing a more effective solution for content-rich digital platforms.

Corresponding Author:

Lusiana Efrizoni, +62857-6743-4270,
Department of Informatics Engineering, Faculty of Engineering and Informatics,
Universitas Sains dan Teknologi Indonesia, Pekanbaru, Indonesia,
Email: lusiana@usti.ac.id

How to Cite:
L. Efrizoni, Junadhi, and Agustin, ”Optimizing Content Recommendations Using a Hybrid Filtering Algorithm to Enhance User Relevance and Engagementt”, *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 25, No. 2, pp. 275-286, March, 2026.

1. INTRODUCTION

In today's digital era, users are confronted with an enormous volume of content across online platforms-ranging from articles, videos, and social media to learning materials. The phenomenon of information overload necessitates recommendation systems capable of filtering and presenting relevant content to users, thereby enabling a more personalized and meaningful user experience [1, 2]. Traditional recommendation systems primarily focus on predicting interaction metrics, such as clicks or ratings, but recent studies show that optimizing solely for clicks or immediate engagement is insufficient to sustain users' long-term engagement with a platform. A study by [3] shows that although many recommender systems successfully increase direct engagement metrics, they often fail to maintain long-term user involvement. In addition, research in [4] highlights that users' affective and emotional responses toward content significantly influence perceived relevance, indicating that click-based prediction alone no longer adequately represents user satisfaction.

Within digital platforms such as OTT services, recommendation algorithms play a critical role not only in content discovery but also in strengthening user loyalty and retention [5]. Nevertheless, existing studies reveal that most recommendation systems continue to evaluate performance primarily in terms of accuracy and computational efficiency, with limited attention to holistic user engagement [6]. Similar limitations are observed in educational platforms, e-learning environments, and other content-based systems, where deeper engagement indicators such as content completion, repeated interactions, and sustained usage behavior are often overlooked. As a result, many systems are able to recommend relevant content but remain ineffective in fostering meaningful and long-term user engagement [7–9].

Hybrid recommendation approaches that combine Collaborative Filtering and Content-Based Filtering have been widely explored to overcome the limitations of single-method models. Prior studies consistently report improvements in recommendation accuracy, diversity, and user satisfaction when hybrid strategies are applied [10, 11]. However, most existing hybrid models still rely on conventional or shallow feature representations, thereby limiting their ability to capture semantic content, sequential user behavior, and engagement-oriented interaction patterns. Consequently, these approaches remain constrained in predicting not only what users are likely to click, but also how deeply users will engage with the recommended content.

Recent advances in deep learning provide new opportunities to address these limitations through representation learning, sequential modeling, and semantic text understanding [12]. Several studies have explored the integration of Neural Collaborative Filtering with text-based content representations and demonstrated improvements in recommendation relevance [13]. Nevertheless, most existing work focuses on relevance-based optimization, whereas user engagement is often treated as an indirect outcome rather than as an explicit modeling objective. In addition, prior hybrid deep learning approaches rarely employ early-fusion strategies that integrate latent user–item interaction representations and sequential textual embeddings at the representation level, which limits their ability to capture fine-grained engagement signals. Moreover, empirical evaluations in existing studies predominantly rely on ranking-based accuracy metrics, with limited validation using engagement-oriented indicators such as content completion and repeated interactions [14, 15].

Based on these observations, a clear research gap remains in developing recommendation systems that jointly model user–item interactions, semantic content representations, and engagement-aware signals within a unified deep learning framework. This study addresses this gap by proposing a Hybrid Deep Neural Recommender that integrates Neural Matrix Factorization and an LSTM-based text encoder through an early-fusion strategy. Unlike prior hybrid approaches, the proposed model is designed to simultaneously enhance recommendation relevance and explicitly capture users' engagement potential, including content completion behavior and repeated interactions. By shifting the focus from accuracy-only optimization toward relevance and engagement-aware recommendation, this research contributes to the development of more adaptive, contextual, and user-centric recommender systems capable of delivering a holistic user experience.

2. RESEARCH METHOD

This study employs an experimental approach with a structured workflow consisting of four main stages. The first stage is data acquisition and preprocessing, which includes collecting user–item interaction data, followed by data cleaning, variable transformation, encoding, and preparing textual representations to be processed by the LSTM Text Encoder [16]. The second stage is hybrid model construction, where the recommendation architecture is designed by integrating NeuMF, which models user–item interactions through embeddings, with LSTM, which processes content descriptions to produce semantic representations [17]. The third stage involves training and optimization, covering model training using modern optimization algorithms, early stopping mechanisms, learning-rate adjustments, and regularization techniques to prevent overfitting [18]. The fourth stage is evaluation, which assesses model performance across two key aspects: recommendation relevance, measured by metrics such as Precision@K, Recall@K, and NDCG@K, and user engagement, measured by CTR and ACR. This sequence of stages is designed to provide a comprehensive evaluation of the effectiveness of the Hybrid Deep Neural Recommender approach in improving recommendation quality while simultaneously enhancing user engagement. The system architecture of the proposed hybrid recommender model is illustrated in Figure 1.

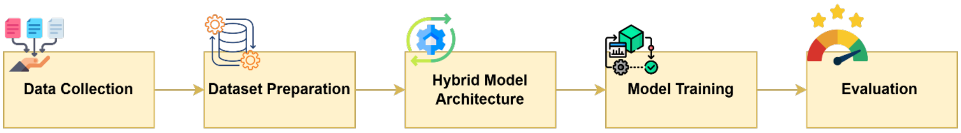

Figure 1. Research flow diagram

2.1. Data Collection

The data collection process was carried out by gathering 1,000 user interaction records, which in this study are synthetic datasets created in a controlled manner to simulate user activity toward content on a digital platform [19]. The use of synthetic data was chosen due to the unavailability of direct access to real interaction data, allowing researchers to flexibly control label distribution, engagement patterns, and interaction characteristics while still representing conditions that closely resemble real-world environments. Each data entry represents a single user activity—such as viewing content, reading, responding, or performing a specific action—with four main components: user_id, item_id, engagement_label, and relevance_label. An example of a single data row is: (user_id: 15, item_id: 203, engagement_label: 1, relevance_label: 1). The dataset contains two main target variables: engagement label (0 = low, 1 = high), relevance label (0 = not relevant, 1 = relevant).

The distribution of these labels is visualized in separate charts, showing that high-engagement data is slightly more common than low-engagement data, with approximately 470 entries labeled 0 and 530 labeled 1. A similar pattern is observed for the relevance label, with around 380 entries marked as not relevant and 620 marked as relevant, indicating that users tend to engage more with content perceived as relevant. Additionally, the interaction distribution across users and content items is shown in Figure 2.

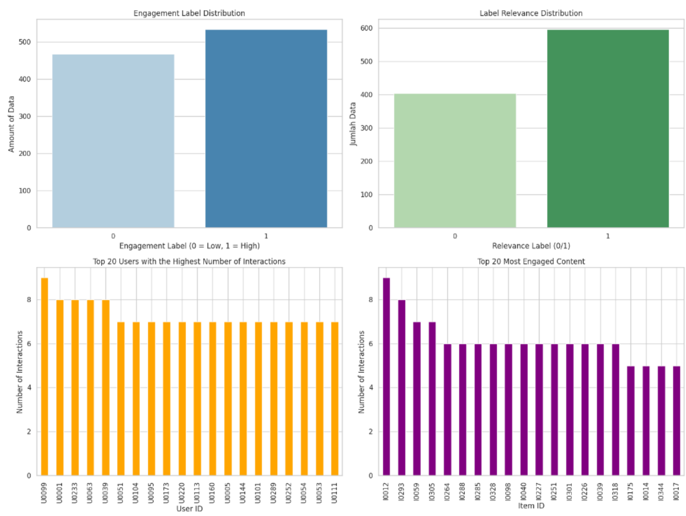

Figure 2. Top 20 users with the highest number of interactions

The Top 20 Users with the Highest Number of Interactions chart indicates that a small subset of users account for most interactions, with counts ranging from 7 to 10. This indicates the presence of power users who contribute significantly to overall system activity. Meanwhile, the Top 20 Most Engaged Content chart reveals that certain content items receive far more interactions than others, exhibiting a similar pattern in the range of 6–10 interactions. This information is crucial for understanding user preferences and for developing a hybrid recommendation model. The complete dataset visualizations are presented in the following charts to provide a comprehensive overview of the data characteristics used in this study.

2.2. Dataset Preparation

The raw data obtained from user interactions was processed to prepare it for the modeling phase. Of the 1,000 interaction records, a completeness check (data cleaning) was conducted to ensure that there were no missing values, duplicates, or inconsistencies in columns such as user_id, item_id, engagement_label, and relevance_label. Next, all categorical variables-such as user IDs and content IDs-were encoded using label encoding so they could be processed by numerical models. At this stage, the data was also normalized using Min-Max Scaling to align feature scales, enabling the model to learn more stably without being affected by differences in value ranges across features. Subsequently, the dataset was divided into three subsets: 70% for training, 15% for validation, and 15% for testing. This split ensures that model performance can be evaluated objectively without bias from the training data.

Additionally, to support the hybrid model, extra feature representations were generated, including interaction frequency, user historical engagement, and content popularity score, extracted from the interaction patterns observed in the previous graphs. These steps produced a clean, structured, and optimized dataset suitable for training the hybrid recommender model [20]. The results of this dataset preparation process are presented in Figure 3.

```
# ====== CELL 5: Prepare arrays and split ======
X_user = df['user_idx'].values
X_item = df['item_idx'].values
X_text = np.vstack(df['seq'].values)
y = df['engagement_label'].astype(int).values  # main target (binary)

# Stratified split (train/val/test)
X_user_train, X_user_temp, X_item_train, X_item_temp, X_text_train, X_text_temp, y_train, y_temp = \
    train_test_split(X_user, X_item, X_text, y, test_size=0.3, random_state=42, stratify=y)

X_user_val, X_user_test, X_item_val, X_item_test, X_text_val, X_text_test, y_val, y_test = \
    train_test_split(X_user_temp, X_item_temp, X_text_temp, y_temp, test_size=0.5, random_state=42, stratify=y_temp)

print("Dataset sizes -> Train:", len(y_train), "Val:", len(y_val), "Test:", len(y_test))

Dataset sizes -> Train: 700 Val: 150 Test: 150
```

Figure 3. Dataset preparation workflow

2.3. Hybrid Model Architecture

The hybrid model architecture used in this study combines the strengths of collaborative and content-based filtering within a unified prediction framework. Overall, the model consists of two main components: NeuMF (Neural Matrix Factorization), which models historical user-item interaction dependencies, and the LSTM Text Encoder, which extracts semantic features from item descriptions [21, 22]. In the NeuMF component, user and item embeddings are generated through two pathways: Generalized Matrix Factorization (GMF) and a Multi-Layer Perceptron (MLP). GMF learns linear relationships between entities, whereas the MLP captures more complex nonlinear patterns through multiple hidden layers. These two pathways are then concatenated to produce a richer interaction representation, consistent with the standard NeuMF architecture. Meanwhile, the content-based component is implemented using a Long Short-Term Memory (LSTM) network to process item descriptions. Each description is converted into a sequence of tokens, embedded, and passed through one or two LSTM layers to extract the semantic context relevant to user preferences. The final LSTM output produces a content vector that reflects linguistic characteristics of the content, such as theme, topic, and writing style.

A full visualization of the integration flow between these two components is shown in Figure 4, which illustrates the process of combining user-item embeddings from the NeuMF module with the content vector produced by the LSTM via an early-fusion mechanism before entering the fully connected layer. This fusion is performed prior to the prediction layer, allowing the final layer to learn both interaction and content information simultaneously. It uses a sigmoid activation function to predict the probability of a match between a user and an item. This hybrid architecture offers advantages in learning historical interaction patterns while simultaneously understanding content characteristics at the semantic level, thereby generating more relevant recommendations and increasing user engagement.

Figure 4. Hybrid recommendation system architecture

To clarify the processing flow of the hybrid model used in this study, the overall procedure is formally described in Algorithm 1. Algorithm 1 presents a summary of the main stages in constructing the Hybrid NeuMF + LSTM architecture, illustrating how user and item embeddings are processed through the generalized matrix factorization and multilayer perceptron branches in the neural matrix factorization component, then combined with the semantic content representation produced by the long short-term memory text encoder via an early-fusion mechanism before entering the final prediction layer.

Table 1. Pembagian data untuk Training dan Testing

Algorithm 1. Hybrid NeuMF + LSTM Text Encoder
Initialize all model parameters θ randomly.
♯ Building Embeddings
Create embedding vectors for users and items.
Create text embedding layer for content sequences.
♯ Pretraining LSTM Text Encoder
for epoch = 1 to E_pre do
Sample mini-batches (text_seq, y_text) from D.
Forward pass through LSTM to obtain content vector.
Compute loss using binary cross-entropy.
Update LSTM parameters θ_LSTM using optimizer.
end for
♯ Training Hybrid NeuMF + LSTM
for epoch = 1 to E do
Sample mini-batches (u, i, text_seq, y) from D.

(dilanjutkan di halaman berikutnya)

Tabel 1 (lanjutan)

Algorithm 1. Hybrid NeuMF + LSTM Text Encoder

```
♯ NeuMF Branch (GMF + MLP)
u_vec ⟵ user_embedding(u)
i_vec ⟵ item_embedding(i)

gmf_vec ⟵ elementwise_multiply(u_vec, i_vec)
mlp_vec ⟵ MLP(concat(u_vec, i_vec))

cf_vec ⟵ concat(gmf_vec, mlp_vec)

♯ LSTM Text Encoder
text_emb ⟵ embed(text_seq)
text_vec ⟵ LSTM(text_emb)

♯ Early Fusion
fusion_vec ⟵ concat(cf_vec, text_vec)
h ⟵ DenseLayers(fusion_vec)

♯ Prediction
y_pred ⟵ sigmoid(Dense(h))

♯ Compute Training Loss
loss ⟵ BCE(y, y_pred)

♯ Backpropagation
Update all parameters θ with optimizer using loss.

end for

return θ
```

2.4. Model Training

The model was trained using a supervised approach, with the data split into training and validation sets. As shown in Figure 5, the model was trained for up to 20 epochs with a batch size of 64. To prevent overfitting and preserve the best weights, two callback mechanisms were used: Early Stopping (patience = 4) and ReduceLROnPlateau, which adaptively reduces the learning rate when the validation loss does not improve. During training, the three inputs (user, item, and text sequence) were processed in parallel and optimized using the Adam optimizer, with binary cross-entropy as the loss function.

```python
# ====== CELL 7: Train model ======
batch_size = 64
epochs = 20
callbacks = [
    EarlyStopping(monitor='val_loss', patience=4, restore_best_weights=True, verbose=1),
    ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2, verbose=1)
]

history = model.fit(
    x=[X_user_train, X_item_train, X_text_train],
    y=y_train,
    validation_data=([X_user_val, X_item_val, X_text_val], y_val),
    batch_size=batch_size,
    epochs=epochs,
    callbacks=callbacks,
    verbose=2
)
```

Figure 5. Model training

2.5. Evaluation

The model was evaluated using Top-K recommendation metrics, specifically Click-Through Rate (CTR) and Average Completion Ratio (ACR). A value of K = 10 was chosen because it aligns with common practices in recommendation system research and reflects a realistic number of recommendations typically presented to users [23, 24]. During the evaluation process, the model predicts relevance scores for all items for each user, then selects the Top-10 items with the highest scores as recommendations. Mathematically, Precision@10 measures the proportion of relevant items in the top 10 and is defined in Equation 1.

$$Precision@10 = \frac{|\{\text{relevant items in Top - 10}\}|}{10} \tag{1}$$

Meanwhile, Recall@10 measures the extent to which the system successfully retrieves relevant items that should have been recommended and is formulated as defined in Equation 2. This metric assesses the model's ability to include the set of relevant items for a given user among the top ten recommendations, thereby reflecting the extent to which the recommendation system captures user preferences. A higher Recall@10 value indicates that a larger proportion of relevant items are included in the recommendation list, which is particularly important in scenarios where missing relevant content may negatively affect user satisfaction and perceived recommendation quality.

$$Precision@10 = \frac{|\{\text{relevant items in Top - 10}\}|}{|\{\text{all relevant items for user}\}|} \tag{2}$$

To evaluate ranking quality, NDCG@10 was used, as it considers not only the presence of relevant items but also their positions within the recommendation list. This metric assigns higher importance to relevant items that appear at the top of the ranked list, reflecting the practical scenario in which users are more likely to interact with higher-ranked recommendations. NDCG@10 is defined in Equation 3 and provides a normalized measure of ranking performance, enabling a fair comparison across users with different numbers of relevant items.

$$NDCG@10 = \frac{DCG@10}{IDCG@10} \tag{3}$$

with

$$DCG@10 = \sum_{i=1}^{10} \frac{rel_i}{log_2(i+1)} \tag{4}$$

which represents the discounted gain for each item's position, while IDCG denotes the ideal DCG value. In addition to relevance, the model was evaluated for user engagement. CTR is calculated as the ratio of Top-10 items that were actually clicked by users in the test data, as defined in Equation 5.

$$CTR = \frac{\text{Jumlah klik pada Top-K}}{\text{Jumlah item Top-K yang direkomendasikan}} \tag{5}$$

Meanwhile, the Average Completion Ratio (ACR) is calculated as the average proportion of content completed by items with which users interact, as defined in Equation 6. This metric reflects the extent to which users consume recommended content in its entirety, providing insight into the depth of user engagement beyond initial interaction. A higher ACR value indicates that users are more likely to complete the recommended content, suggesting that the recommendations are perceived as relevant, engaging, and valuable.

$$ACR = \frac{1}{N}\sum_{i=1}^{N} C_i \tag{6}$$

Where C_i is the completion ratio of content *i* i. These two metrics provide a realistic picture of the model's ability not only to attract users' attention (via CTR) but also to encourage deeper engagement with content (via ACR). Thus, the Top-K evaluation reflects the model's comprehensive performance in real-world recommendation scenarios. The Python code is presented in Figure 6.

```python
# ====== CELL 10: Engagement evaluation on Top-K recommendations (CTR, ACR) ======
def compute_ctr_acr(model, users, K=10):
    ctrs, acrs = [], []
    for u in users:
        user_arr = np.full_like(all_items, fill_value=u)
        preds = model.predict([user_arr, all_items, item_texts], verbose=0).reshape(-1)
        topk = np.argsort(-preds)[:K]
        clicks = 0
        completion_vals = []
        for idx in topk:
            # check if this (user,idx) exists in test_df
            mask = (test_df['user_idx']==u) & (test_df['item_idx']==idx)
            if mask.any():
                row = test_df[mask].iloc[0]
                if row['engagement_label'] == 1:
                    clicks += 1
                    # find completion_ratio in original df
                    mask_full = (df['user_idx']==u) & (df['item_idx']==idx)
                    if mask_full.any():
                        completion_vals.append(df[mask_full].iloc[0]['completion_ratio'])
        ctrs.append(clicks / K)
        acrs.append(np.mean(completion_vals) if len(completion_vals)>0 else 0.0)
    return np.mean(ctrs) if len(ctrs)>0 else 0.0, np.mean(acrs) if len(acrs)>0 else 0.0

ctr_top10, acr_top10 = compute_ctr_acr(model, users_unique, K=10)
print(f"Engagement Top-10 -> CTR: {ctr_top10:.4f}, ACR: {acr_top10:.4f}")
```

Figure 6. Model evaluation

3. RESULT AND ANALYSIS

This section presents the experimental results for the developed model, along with an analysis of its performance using the defined evaluation metrics. All findings are organized into several subsections to provide a clear overview of the model's performance, learning patterns, and the effectiveness of the proposed approach.

3.1. Hybrid Model Architecture

The hybrid model architecture in this study integrates collaborative filtering and content-based filtering to produce more comprehensive predictions. The main components of the model consist of NeuMF (GMF–MLP), which learns historical user–item interaction patterns, and an LSTM Text Encoder, which extracts semantic features from content descriptions. In the NeuMF component, user and item embeddings are processed through two pathways: GMF, which models linear relationships, and MLP, which captures more complex non-linear patterns. These two pathways are then combined to produce a more informative interaction representation. In parallel, item text descriptions are processed by an LSTM, which transforms tokens into semantic representations that capture the content's topic and characteristics. The resulting text representation is then merged with the NeuMF interaction representation through an early-fusion mechanism before entering the prediction layer. This integration process allows the model to learn historical and semantic information simultaneously. As illustrated in Figure 3, the hybrid architecture produces a richer, unified latent vector, thereby improving the accuracy of relevance prediction and the prediction of engagement potential. The performance results of this hybrid architecture are presented in Table 2 to assess the model's effectiveness.

Table 2. Comparison of Hybrid Model Performance vs. Baseline

Model	Precision@10	Recall@10	NDCG@10	CTR	ACR
Baseline CF (NeuMF)	0.118	0.091	0.104	0.0011	0.0064
Baseline LSTM Content	0.102	0.083	0.097	0.0010	0.0059
Hybrid (NeuMF + LSTM)	0.143	0.112	0.139	0.0017	0.0090

The study results show that the hybrid model combining NeuMF and LSTM achieves the best performance compared with the baseline models. This is evident from higher Precision@10, Recall@10, and NDCG@10, indicating that the model is more effective at identifying and ranking relevant items. In addition, improvements in CTR and ACR indicate that the hybrid model generates recommendations that are more engaging and relevant to users, thereby encouraging higher levels of user interaction. Overall, the combination of collaborative filtering and content-based approaches is more effective than using a single model.

3.2. Evaluation

Evaluation of the model's performance using the training history plot indicates that the training process ran stably and effectively. This is evident from the consistent decrease in training loss from the early to the final epochs, indicating that the model learned data patterns effectively. Training accuracy also increased significantly—especially after learning rate adjustments-showing that the model responded well to the optimization strategy. On the other hand, validation accuracy increased before slightly declining in the final epoch, whereas validation loss increased steadily, indicating mild overfitting toward the end of training. However, the early stopping mechanism successfully halted training at the optimal point, ensuring that the model maintained its best performance without excessive overfitting. Overall, the graph pattern confirms that the model learned effectively and achieved a good balance between training and validation. The training history graph is presented in Figure 7.

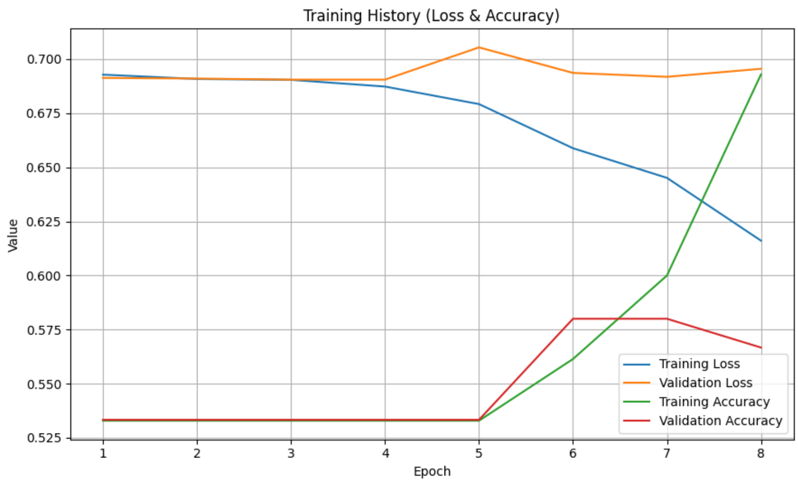

Figure 7. Training history (loss & accuracy)

The training history visualization indicates that the model underwent stable, effective training. This is evident from the consistent decrease in training loss and the increase in accuracy from the beginning to the end of training. Validation accuracy also improved during the initial epochs, although it declined slightly toward the end, which remains within the normal range. Meanwhile, validation loss increased, indicating the emergence of mild overfitting after certain epochs. However, the early stopping mechanism and learning rate adjustments effectively identified the optimal epoch before overfitting became more pronounced. Overall, this pattern demonstrates that the model learned effectively and achieved optimal performance at the epoch selected by early stopping. Unlike previous studies, which focused solely on optimizing recommendation accuracy, this research provides stronger empirical evidence that integrating interaction-based and semantic content representations yields more stable learning performance while also improving relevance and user engagement.

4. CONCLUSION

This study successfully developed a hybrid recommendation model that integrates Collaborative Filtering based on NeuMF with a Content-Based LSTM approach. The evaluation results indicate that the hybrid model achieves the best performance compared with the two baseline models. In ranking evaluation metrics, the hybrid model achieved a Precision@10 of 0.143, Recall@10 of 0.112, and NDCG@10 of 0.139, which are higher than those of the NeuMF baseline (0.118; 0.091; 0.104) and the LSTM baseline (0.102; 0.083; 0.097). In the user engagement evaluation, the hybrid model also achieved the highest performance, with a CTR of 0.0017 and an ACR of 0.0090, indicating that the recommendations produced are not only accurate but also more effective at driving user interaction. The training history analysis confirms that the model learned in a stable manner, as evidenced by a reduction in training loss from 0.6928 to 0.6161 and an increase in training accuracy from 0.5329 to 0.6929. Although the validation loss increased during several of the final epochs—indicating mild overfitting-the early stopping mechanism halted training at the optimal epoch (epoch 4), thereby preserving the model's performance. Overall, this study demonstrates that combining user interaction features with content information significantly improves recommendation quality and provides a strong foundation for developing more adaptive and effective recommendation systems.

5. ACKNOWLEDGEMENTS

The author expresses sincere gratitude to Universitas Sains dan Teknologi Indonesia (USTI) for its support, facilities, and academic environment, which made this research possible. Appreciation is also extended to all supervising lecturers, fellow students, and everyone who provided guidance, feedback, and assistance throughout the research process. It is hoped that the findings of this study will contribute to the advancement of scientific knowledge and the future application of technology.

6. DECLARATIONS

AI USAGE STATEMENT

Part of the writing and refinement process for this research report used Artificial Intelligence (AI) technologies, particularly language models, to assist with phrasing, text summarization, and grammar improvements. All analyses, experimental results, model design, and data interpretation were conducted by the researcher. The use of AI served solely as a writing aid and did not replace critical thinking or the researcher's scientific contribution. The researcher takes full responsibility for the content, originality, and scientific integrity of all research results presented in this document.

AUTHOR CONTIBUTION

Lusiana Efrizoni contributed to the formulation of the research idea, the design of the methods, the execution of the experiments, and the analysis of the results. Junadhi contributed to validating the methodology, providing technical supervision, and assisting with the writing and review of the research manuscript. Agustin contributed to data collection, dataset processing, and the preparation and refinement of supporting documentation for the final report.

FUNDING STATEMENT

This research did not receive any specific funding from any funding agency, public or private.

COMPETING INTEREST

The author declares that the entire research process, analysis, and manuscript preparation were conducted without any conflicts of interest that could affect the academic and scientific integrity of this article.

REFERENCES

[1] X. He, Q. Liu, and S. Jung, "The impact of recommendation system on user satisfaction : A moderated mediation approach," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 19, no. 1, pp. 448–466, Feb. 2024, https://doi.org/10.3390/jtaer19010024.

[2] M. M. Lesan Sedgh, A. Latif, and S. Emadi, "A novel method for a technology enhanced learning recommender system considering changing user interest based on neural collaborative filtering," *Data Science and Management*, vol. 8, no. 2, pp. 196–206, Jun. 2025, https://doi.org/10.1016/j.dsm.2024.09.004.

[3] W. Li, "Analyzing the Impact of Information Features on User Continuance Intent in Recommendation Systems," *International Journal on Semantic Web and Information Systems*, vol. 20, no. 1, pp. 1–36, Sep. 2024, https://doi.org/10.4018/IJSWIS.353905.

[4] J. Yin, X. Qiu, and Y. Wang, "The impact of ai-personalized recommendations on clicking intentions : Evidence from chinese e-commerce," *Journal of Theoretical and Applied Electronic Commerce Research*, vol. 20, no. 1, p. 21, Feb. 2025, https://doi.org/10.3390/jtaer20010021.

[5] D. Chakraborty, M. Siddiqui, A. Siddiqui, J. Paul, G. Dash, and F. D. Mas, "Watching is valuable: Consumer views content consumption on ott platforms," *Journal of Retailing and Consumer Services*, vol. 70, p. 103148, Jan. 2023, https://doi.org/10.1016/j.jretconser.2022.103148.

[6] L. Efrizoni, "Adaptive neural collaborative filtering with textual review integration for enhanced user experience in digital platforms," *Journal of Applied Data Sciences*, vol. 6, no. 4, pp. 2696–2710, Dec. 2025, https://doi.org/10.47738/jads.v6i4.944.

[7] A.-L. Camerini, S. Morlino, and L. Marciano, "Boredom and digital media use: A systematic review and meta-analysis," *Computers in Human Behavior Reports*, vol. 11, p. 100313, Aug. 2023, https://doi.org/10.1016/j.chbr.2023.100313.

[8] D. P. Sakas, D. P. Reklitis, N. T. Giannakopoulos, and P. Trivellas, "The influence of websites user engagement on the development of digital competitive advantage and digital brand name in logistics startups," *European Research on Management and Business Economics*, vol. 29, no. 2, p. 100221, May 2023, https://doi.org/10.1016/j.iedeen.2023.100221.

[9] J. E. Anderson, C. A. Nguyen, and S. Anderson, "Like, comment, and subscribe: Investigating the effectiveness of digital engagement prompts," *Journal of Research in Interactive Marketing*, vol. 20, no. 1, pp. 49–67, february 2026, https://doi.org/10.1108/JRIM-12-2024-0573.

[10] A. Fareed, S. Hassan, S. B. Belhaouari, and Z. Halim, "A collaborative filtering recommendation framework utilizing social networks," *Machine Learning with Applications*, vol. 14, p. 100495, Dec. 2023, https://doi.org/10.1016/j.mlwa.2023.100495.

[11] R. Widayanti, M. H. R. Chakim, C. Lukita, U. Rahardja, and N. Lutfiani, "Improving recommender systems using hybrid techniques of collaborative filtering and content-based filtering," *Journal of Applied Data Sciences*, vol. 4, no. 3, pp. 289–302, Sep. 2023, https://doi.org/10.47738/jads.v4i3.115.

[12] "Sequential recommendation system based on deep learning a survey," *Electronics*, vol. 14, no. 11, p. 2134, May 2025, https://doi.org/10.3390/electronics14112134.

[13] L. Efrizoni, J. Junadhi, and A. Agustin, "Optimization of content recommendation system based on user preferences using neural collaborative filtering," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 24, no. 2, pp. 309–320, Mar. 2025, https://doi.org/10.30812/matrik.v24i2.4775.

[14] Y. Liu, S. Wang, M. S. Khan, and J. He, "A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering," *Big Data Mining and Analytics*, vol. 1, no. 3, pp. 211–221, Sep. 2018, https://doi.org/10.26599/BDMA.2018.9020019.

[15] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network," *Simulation Modelling Practice and Theory*, vol. 113, p. 102375, december 2021, https://doi.org/10.1016/j.simpat.2021.102375.

[16] A. Mumuni and F. Mumuni, "Automated data processing and feature engineering for deep learning and big data applicationsa survey," *Journal of Information and Intelligence*, vol. 3, no. 2, pp. 113–153, Mar. 2025, https://doi.org/10.1016/j.jiixd.2024.01.002.

[17] J. Ni, Z. Huang, Y. Hu, and C. Lin, "A two-stage embedding model for recommendation with multimodal auxiliary information," *Information Sciences*, vol. 582, pp. 22–37, Jan. 2022, https://doi.org/10.1016/j.ins.2021.09.006.

[18] A. A. Adekunle, r. I. Fofana, P. Picher, E. M. Rodriguez-Celis, O. H. Arroyo-Fernandez, and R. Zemouri, "Optimizing deep learning predictive models: A comprehensive review of rnn and its variant architectures," *Applied Soft Computing*, vol. 185, p. 114015, december 2025, https://doi.org/10.1016/j.asoc.2025.114015.

[19] W. Li, "Analyzing the Impact of Information Features on User Continuance Intent in Recommendation Systems," *International Journal on Semantic Web and Information Systems*, vol. 20, no. 1, pp. 1–36, Sep. 2024, https://doi.org/10.4018/IJSWIS.353905.

[20] W. Xue, Q. Cai, Z. Xue, S. Sun, S. Liu, D. Zheng, P. Jiang, K. Gai, and B. An, "Prefrec recommender systems with human preferences for reinforcing long-term user engagement," pp. 2874–2884, Aug. 2023, https://doi.org/10.1145/3580305.3599473.

[21] Y. Wang, S. Ke, C. An, Z. Lu, and J. Xia, "A hybrid framework combining lstm nn and bnn for short-term traffic flow prediction and uncertainty quantification," *KSCE Journal of Civil Engineering*, vol. 28, no. 1, pp. 363–374, Jan. 2024, https://doi.org/10.1007/s12205-023-2457-y.

[22] Y. Xiao and Y. Yin, "Hybrid lstm neural network for short-term traffic flow prediction," *Information*, vol. 10, no. 3, p. 105, Mar. 2019, https://doi.org/10.3390/info10030105.

[23] A. Sami, W. E. Adrousy, S. Sarhan, and S. Elmougy, "A deep learning based hybrid recommendation model for internet users," *Scientific Reports*, vol. 14, no. 1, p. 29390, Nov. 2024, https://doi.org/10.1038/s41598-024-79011-z.

[24] P. N. Vicente and C. D. Burnay, "Recommender systems and over-the-top services a systematic review study (2010–2022)," *Journalism and Media*, vol. 5, no. 3, pp. 1259–1278, Sep. 2024, https://doi.org/10.3390/journalmedia5030080.

[This page intentionally left blank.]