Acta Informatica Pragensia
2026, Volume 15, Issue 1, 1–21
https://doi.org/10.18267/j.aip.280

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Article Open Access

ResNetMF: Improving Recommendation Accuracy and Speed with Matrix Factorization Enhanced by Residual Networks

Mustafa Payandenick [1], **YinChai Wang** [1], **Mohd Kamal Othman** [2], **Muhammad Payandenick** [1]

[1] Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, Sarawak, Malaysia
[2] Faculty of Cognitive Sciences and Human Development, Universiti Malaysia Sarawak, Sarawak, Malaysia

Corresponding author: Mustafa Payandenick (19020070@siswa.unimas.my)

Editorial Record

First submission received:
April 28, 2025

Revision received:
June 8, 2025

Accepted for publication:
July 6, 2025

Academic Editor:
Zdenek Smutny
Prague University of Economics and Business, Czech Republic

This article was accepted for publication by the Academic Editor upon evaluation of the reviewers' comments.

How to cite this article:
Payandenick, M., Wang, Y., Othman, M. K., & Payandenick, M. (2026). ResNetMF: Improving Recommendation Accuracy and Speed with Matrix Factorization Enhanced by Residual Networks. *Acta Informatica Pragensia*, 15(1), 1–21. https://doi.org/10.18267/j.aip.280

Abstract

Background: Recommendation systems are essential for personalized user experiences but struggle to balance accuracy and efficiency.

Objective: This paper presents ResNetMF, an innovative hybrid framework designed to address these limitations by combining the strengths of matrix factorization (MF) and deep residual networks (ResNet). Matrix factorization excels at capturing explicit linear relationships between users and items, while ResNet is employed to model non-linear residuals.

Methods: By focusing on refining the baseline MF output through incremental improvements, ResNetMF minimizes redundant computations and significantly enhances recommendation accuracy. The unique architecture of the framework allows it to capture and represent both linear and non-linear relationships between users and items, ensuring robust and scalable performance. Extensive experiments conducted on the widely used MovieLens dataset demonstrate the superiority of ResNetMF over existing methods.

Results: Specifically, it achieves a minimum improvement of 7.95% in root mean square error compared to neural collaborative filtering and outperforms other state-of-the-art techniques in key metrics such as precision, recall and training efficiency. These results highlight the ability of ResNetMF to deliver highly accurate recommendations while maintaining computational efficiency, making it an efficient approach to real-world application of recommendation systems.

Conclusion: By addressing the dual challenges of accuracy and efficiency, ResNetMF offers a balanced and scalable approach to personalized recommendation systems.

Index Terms

Recommendation systems; Matrix factorization; Residual networks; ResNet; Hybrid framework; Training efficiency; Personalized recommendations.

1 INTRODUCTION

Recommendation systems play an essential role in enabling personalized information filtering by suggesting relevant choices to users from vast collections (Urdaneta-Ponte et al., 2021). These systems are designed to alleviate the overwhelming nature of extensive selection options and provide tailored recommendations based on user preferences (Zhu, Jiang, et al., 2019). Recommendation systems are generally categorized into three main types: content-based filtering, collaborative filtering and hybrid approaches; each approach employs distinct methodologies to deliver personalized recommendations (Manikantan, 2021).

Content-based filtering focuses on user-centric approaches that use user profiles and item descriptions to recommend choices based on individual preferences (Eliyas & Ranjana, 2022; Yadalam et al., 2020). By analysing explicit ratings or implicit indicators such as user click rates, content-based filtering creates comprehensive user profiles and tailors recommendations accordingly (Eliyas & Ranjana, 2022). Collaborative filtering, recognized as one of the most commonly used techniques in developing recommendation systems (Guo & Yan, 2020; Z. Zhang et al., 2020), relies on historical data of items or users to generate recommendations (Nallamala et al., 2020). To overcome the limitations of individual methods, hybrid filtering techniques have emerged as a practical solution (H. Li & Han, 2020). Hybrid models combine two or more recommendation algorithms to enhance recommendation accuracy and reduce information loss (Duong et al., 2021).

In recent years, deep learning-based recommendation systems are gaining popularity in recommendation tasks (Nan et al., 2022). These systems use neural networks to identify complex patterns in user interactions and item attributes, enabling them to provide tailored recommendations (Song, 2020). Deep learning-driven recommendation systems present exciting prospects for enhancing the accuracy, customization and flexibility of recommendations (L. Wu et al., 2023).

However, recommendation systems face significant challenges in balancing accuracy and computational efficiency (L. Wu et al., 2023). Traditional collaborative filtering methods, such as matrix factorization (MF), effectively capture linear user-item interactions using low-dimensional latent factors (Chen et al., 2024) but fail to model complex non-linear relationships (L. Wu et al., 2023. In contrast, deep learning models can learn intricate non-linear patterns (Zeng et al., 2019) but suffer from vanishing gradients, training instability and high computational costs, particularly when scaling to large datasets (Tan et al., 2023; L. Wu et al., 2023). Key limitations of current approaches are summarized as follows.

- Inadequate modelling of complex interactions – shallow models such as MF cannot capture multi-layered user-item dependencies (L. Wu et al., 2023), while deep networks (e.g., CNNs, RNNs) degrade in performance with increasing depth due to vanishing gradients (Luo, 2024).
- Computational inefficiency – large user-item matrices require extensive memory and training deep networks on sparse data further intensifies resource demands (Xia et al., 2023).
- Suboptimal hybrid architectures – existing hybrid approaches often concatenate MF and neural components without effectively integrating them (Zhou et al., 2019). The absence of residual mechanisms results in incomplete fusion of linear and non-linear features, leading to suboptimal performance and excessive memory consumption (Du et al., 2023).

This study addresses these gaps by proposing ResNetMF, a hybrid framework that integrates matrix factorization with deep residual networks (ResNet) to jointly capture linear and non-linear interactions while ensuring memory efficiency and robust training dynamics. ResNetMF overcomes these limitations by unifying MF with a residual network. The MF component captures explicit linear patterns, while the ResNet learns residuals—non-linear discrepancies between MF predictions and actual ratings. The residual learning mechanism of ResNet explicitly learns residual errors, allowing it to focus on incremental improvements to the baseline MF output. This targeted error correction reduces redundant computations and improves recommendation accuracy.

2 RELATED WORKS

In this section, we begin by discussing the recommendation system based on matrix factorization. Furthermore, we discuss deep learning-based recommendation systems.

2.1 Matrix factorization-based recommendation systems

Matrix factorization is the most commonly employed algorithm for model-based recommendation systems, valued by researchers for its efficiency and speed (H. Liu et al., 2022). The fundamental matrix factorization model estimates user preferences by breaking down the user-item rating matrix into two matrices with lower dimensions; this method has been expanded to include implicit user-item feedback (Wan et al., 2021). Two widely adopted matrix factorization techniques in the recommendation field are singular value decomposition (SVD) and non-negative matrix factorization (NMF) (Muhammet & Arıcan, 2024).

Singular value decomposition (SVD) simplifies a user-item interaction matrix by breaking it down into lower-dimensional forms, retaining critical features and eliminating noise (Ferreira et al., 2020). It is especially useful in collaborative filtering, where the matrix structure consists of users in rows and items in columns, with elements representing user ratings for items (Przystupa et al., 2021). SVD enables the identification of latent factors that can predict user preferences effectively (Ferreira et al., 2020).

Non-negative matrix factorization (NMF) ensures that the values obtained from factorization are constrained to be non-negative, which enhances the interpretability of the resulting components (Alphonse & Verma, 2023). This approach is particularly suitable for applications such as music and film recommendations, where the absence of negative values makes it easier to analyse and understand user preferences and item features (Alphonse & Verma, 2023).

Despite its advantages, the application of matrix factorization is not without challenges, particularly concerning the trade-offs between speed and accuracy (Hu et al., 2020). While high accuracy is crucial for delivering relevant recommendations, computational speed is equally important for providing real-time feedback, essential for maintaining user satisfaction and engagement (Hu et al., 2020).

Moreover, the inherent complexity of human preferences presents a formidable challenge for matrix factorization techniques. Users may have nuanced opinions about items that are difficult to capture with simplistic models (Bachiri et al., 2023). Therefore, there is a need for advanced approaches that can effectively model the multifaceted nature of user preferences, incorporating aspects such as context and emotional responses, to provide more accurate and personalized recommendations (Bachiri et al., 2023).

Matrix factorization techniques are often enhanced through integration with various advanced methods to improve speed and accuracy in recommendation systems (Muhammet & Arıcan, 2024). Deep learning methods have become increasingly popular in the field of collaborative filtering (Oggretir & Cemgil, 2017). These methods are particularly beneficial for overcoming the limitations of basic matrix factorization, as neural networks can approximate complex relationships within data more effectively (Suzuki & Ozaki, 2017). Combining matrix factorization with deep learning not only exploits the strengths of both methodologies but also enhances predictive accuracy (Wan et al., 2021). Therefore, advancements in deep learning and hybrid approaches are explored to mitigate these challenges and further enhance the accuracy and efficiency of matrix factorization in recommendation systems (Hu et al., 2020).

2.2 Deep learning in recommendation systems

Deep learning has significantly enhanced the capabilities of recommendation systems, enabling them to gain deeper insights into user preferences and item characteristics by means of sophisticated modelling techniques (W. Huang et al., 2023). The strength of deep learning in recommendation tasks stems from its capability to autonomously derive features from raw data (Arunkumar et al., 2024; Eswaraiah & Syed, 2024), handle large-scale datasets (Sakboonyarat & Tantatsanawong, 2022) and model intricate, non-linear relationships among users and items (Khan et al., 2023). Deep learning methods, including neural collaborative filtering and transformer models, have gained prominence as effective alternatives, utilizing intricate user-item relationships to enhance prediction accuracy substantially (Tilahun et al., 2017).

One of the notable deep learning approaches is neural collaborative filtering (NCF), which integrates user and item embeddings with neural networks (Chang et al., 2023). This enables the model to capture non-linear connections between users and items, leading to more accurate user preference predictions and improving overall recommendation quality (G. Wu et al., 2019).

Recurrent neural networks (RNN) are particularly effective for tasks involving sequences and context (Mienye et al., 2024), such as natural language processing (Yu et al., 2019) and contextual recommendations (Mienye et al., 2024). By employing memory mechanisms such as long short-term memory (LSTM) or gated recurrent units (GRU), RNNs can learn temporal dependencies in user interactions, which is crucial for applications such as next-item recommendations (Zangerle & Bauer, 2023).

Transformer models, such as BERT (bidirectional encoder representations from transformers), offer an alternative to RNNs by utilizing attention mechanisms to concentrate on the most significant aspects of the input data (Zangerle & Bauer, 2023). This method has proven effective in parsing and understanding complex sequences, thus enhancing the ability of the model to generate precise recommendations based on user preferences and historical data (Sujaykumar Reddy et al., 2024).

Despite the advancements in accuracy, deep learning-based recommendation systems face substantial challenges, particularly concerning training speed and computational efficiency (Y. Liu, 2024). The complexity of deep learning models often results in longer training times compared to traditional methods, which can hinder their application in real-time environments (Nagraj & Palayyan, 2024).

As highlighted in various studies, the pursuit of higher accuracy often leads to increased computational demands, resulting in longer training times (Zangerle & Bauer, 2023). Chen and Liu (2017) emphasized the need for performance evaluations that balance accuracy with computational efficiency, suggesting that the implementation of faster algorithms can help mitigate these issues without significantly compromising the quality of recommendations.

Achieving a balance between training speed and accuracy remains a persistent dilemma in the development of recommendation systems. For instance, setting a low accuracy threshold may enhance speed but increase error rates, while a higher threshold may slow down responses. Research into deep learning recommendation systems often focuses on enhancing model accuracy, but an equally critical area is improving training speed without sacrificing performance.

2.3 Residual learning in recommendation systems

Residual learning-based recommender systems are advanced algorithms designed to enhance personalized recommendations by utilizing deep learning techniques, particularly deep residual networks (Tai et al., 2025). These systems address common challenges faced by traditional recommendation methods, such as sparsity and scalability, through their ability to model intricate user-item interactions effectively (Shen et al., 2023; Tai et al., 2025). As the demand for personalized user experiences grows across various sectors, residual learning-based systems have emerged as a notable solution, significantly improving recommendation accuracy and user satisfaction (Tai et al., 2025).

Residual learning was introduced to overcome the difficulties of training very deep neural networks by allowing gradients to flow through the network more easily (Xie et al., 2025). This is achieved by employing shortcut connections that skip one or more layers, enabling the network to learn residual mappings instead of trying to learn the desired underlying mappings directly (Z. Xu & Geng, 2024). The use of shortcut connections allows gradients to flow more easily through deeper neural networks, which is crucial for capturing complex relationships in data (Roy & Dutta, 2022; Z. Xu & Geng, 2024).

This architectural innovation facilitates training of deeper networks, which is critical for capturing intricate patterns in user preferences and item characteristics in recommender systems (Z. Xu & Geng, 2024). This architectural innovation facilitates faster training and enhances the capacity of models to learn generalizable representations, thus providing accurate recommendations even in dynamic environments (Carole et al., 2024). The strength of these systems lies in their improved accuracy, scalability and adaptability, making them suitable for diverse applications, from recommending products on e-commerce platforms to suggesting media content on streaming services (Priya Thota & Devi G, 2024; H. Zhang et al., 2024).

One of the primary advantages of residual learning-based recommender systems is their ability to achieve higher accuracy in recommendations compared to traditional models (Pireci Sejdiu et al., 2022). Residual networks utilize skip connections that facilitate the flow of gradients during training, thereby achieving faster convergence (Yun, 2024). This results in shorter training times and reduces the computational costs associated with model training (Yun, 2024). Due to their unique structure, residual learning models tend to generalize better across unseen data (Tai et al., 2025). By utilizing deep learning techniques, residual learning-based recommender systems can analyse more complex user preferences and behaviour (Sharma et al., 2023). These systems can capture intricate relationships and interactions between users and items, thereby providing more nuanced recommendations that align with users' needs and expectations (Chronis et al., 2024). Despite the challenges associated with large-scale data, residual learning-based systems can effectively manage increasing data volumes (Gibril et al., 2022).

However, residual learning-based recommender systems are not without their limitations. They often face challenges related to computational complexity, as deeper architectures require significant resources for training and inference, raising concerns about accessibility for smaller organizations (Naik et al., 2025). Additionally, the risk of overfitting when dealing with sparse data and the enduring cold-start problem for new users or items can hinder their performance in certain contexts (Frausto-Solís et al., 2024).

Table 1. *Summary of key aspects of discussed methods.*

Method	Key features	Strengths	Limitations	Comparison with ResNetMF
Matrix factorization (SVD, NMF)	Decomposes user-item matrix into lower-dimensional latent factors.	- Efficient and fast. - Works well with explicit feedback. - Interpretable (NMF).	- Struggles with non-linear relationships. - Limited by data sparsity. - Cold-start problem.	ResNetMF **extends** MF by integrating deep learning to handle non-linear patterns.
Neural collaborative filtering (NCF)	Uses neural networks to learn user-item interactions.	- Captures non-linear patterns. - Better accuracy than traditional MF.	- Computationally expensive. - Slower training. - May overfit.	ResNetMF **improves efficiency** via residual learning, enabling faster convergence.
RNN/LSTM for recommendations	Models sequential user behaviour (e.g., session-based recommendations).	- Effective for temporal data. - Captures long-term dependencies.	- High training cost. - Struggles with long sequences (vanishing gradients).	ResNetMF **avoids RNN limitations** by using residual connections for stable deep learning.
Transformer-based (e.g., BERT)	Uses self-attention for sequential recommendations.	- Handles complex patterns. - Strong performance on sequential data.	- Heavy computational demand. - Requires large datasets.	ResNetMF is **lighter** and **faster** while still capturing deep interactions.
Residual learning (deep ResNet)	Uses skip connections to enable deeper networks.	- Faster convergence. - Better generalization. - Handles sparse data.	- Computationally complex. - Risk of overfitting.	ResNetMF **builds on this** but combines it with MF for better linear and non-linear modelling.
Proposed ResNetMF	Hybrid of **matrix factorization + deep residual networks** (dual-branch architecture).	**- Balances speed and accuracy.** - Handles both linear and non-linear patterns. - Faster training (residual learning).	- Still faces cold-start issues. - Requires tuning for optimal fusion.	**Key Contribution:** Combines MF efficiency with deep learning power in a unified framework.

Integration of residual networks with matrix factorization presents a promising advancement in recommender systems, addressing critical challenges in accuracy and training efficiency (Bobadilla et al., 2024; Karimian & Hosseini Kordkheili, 2025). Existing research demonstrates that residual learning enhances deep recommendation models by enabling stable training of deeper architecture through skip connections, leading to better feature representation and faster convergence (G. Xu et al., 2025). Meanwhile, matrix factorization remains effective in capturing latent user-item interactions, particularly in sparse data scenarios. This study explores adaptive fusion techniques and lightweight architectures to maximize both accuracy and training speed, ensuring broader applicability across diverse recommendation domains. Table 1 below provides a comparative summary of the key aspects of the discussed methods, highlighting the distinctions and contributions of the proposed ResNetMF approach.

3 RESIDUAL NETWORK MATRIX FACTORIZATION (RESNETMF)

This study introduces ResNetMF, a multi-criteria recommendation system that combines matrix factorization (MF) with deep residual networks (ResNet) to enhance recommendation accuracy and training efficiency. By integrating the ability of ResNet to simplify deep network training and capture complex patterns, ResNetMF effectively models both linear and non-linear relationships in data.

Traditional matrix factorization excels at learning linear interactions but struggles with non-linear dependencies. In contrast, deep residual networks specialize in modelling intricate non-linear patterns. ResNetMF bridges this gap through a dual-branch architecture: one branch employs MF for linear relationships, while the other utilizes ResNet for non-linear feature extraction. This fusion enables more accurate and robust recommendations by jointly optimizing both types of patterns.

3.1 Matrix factorization component

Matrix factorization breaks down the user-item interaction matrix R into two lower-dimensional latent factor matrices:

- U: A user latent factor matrix with dimensions $Y \times K$, where Y indicates the total count of users and K denotes the latent factor count.
- I: Item latent factor matrix of size $X \times K$, with X representing the total count of items, with K signifying the latent factor count.

The estimated rating $\hat{r}_{u,i}$ for the user u and the item i is determined by computing the dot product of their respective latent factor vectors:

$$\hat{r}_{u,i} = U_u \,.\, I_i^T \qquad (1)$$

where U_u represents the latent factor vector corresponding with the user u and I_i represents the latent factor vector corresponding with the item.

The T in I_i^T represents the **transpose** of the item latent factor vector I_i. The transpose operation flips the vector over its diagonal, converting a column vector into a row vector or vice versa. In this equation,

- U_u is typically a **row vector** (dimensions $1 \times K$),
- I_i is typically a **column vector** (dimensions $K \times 1$),
- I_i^T converts I_i into a row vector (dimensions $1 \times K$).

The dot product $U_u \,.\, I_i^T$ is computed as

$$U_u \,.\, I_i^T = \sum_{k=1}^{k} U_{u,k} \,.\, I_{i,k} \qquad (2)$$

where $U_{u,k}$ and $I_{i,k}$ correspond to the k-th elements of the latent factor vectors for both users and items, respectively. This dot product quantifies the interaction strength between the user u and the item i within the latent factor space.

Matrix factorization excels at capturing latent factors within user-item interactions. It identifies linear data patterns and creates feature vectors representing users and items, which are then fed into the residual neural network. This factorization greatly lowers the dimensionality of data, enhancing computational efficiency.

3.2 Residual network component

The residual network (ResNet) is a deep learning architecture designed to identify intricate, non-linear patterns within data. It takes the latent factors U_u and I_i as input and learns non-linear relationships through a series of layers. Let $f_{\text{ResNet}}(\cdot)$ represent the ResNet function. The output of the ResNet is:

$$f\text{ResNet}\ (U_u, I_i) = \text{ResNet}(U_u \oplus \ I_i)$$

where:

- $\oplus$ denotes the concatenation of user and item latent factors.
- ResNet(·) consists of:
    - **Input layer:** Accepts the concatenated latent factors as input.
    - **Feature extraction layers:** Multiple layers of convolutional, pooling and activation operations to learn high-level features from the input data.
    - **Residual blocks:** Stacked layers with skip connections to capture residual information and enable deeper learning. These blocks enable the network to capture intricate patterns while preventing the vanishing gradient issue.
    - **Global pooling layer:** Aggregates the features across spatial dimensions and generates a fixed-length feature vector.
    - **Fully connected layers:** Transform the extracted features into a format suitable for the final output.
    - **Output layer:** Produces the final recommendation scores or ratings.

The ResNet is designed to discover previously inconceivable relationships in underlying data by generating additional features that describe observations better than the original data. These additional features aid in determining which items should be added or removed from the top recommendation list based on user preferences.

Instead of directly predicting the rating, the ResNet learns the residual error $\epsilon_{u,i}$ between the actual rating $r_{u,i}$ and the matrix factorization prediction $\hat{r}_{u,i}$

$$\epsilon_{u,i} = r_{u,i} - \hat{r}_{u,i} \tag{3}$$

The ResNet is trained to predict this residual:

$$f\text{ResNet}(U_u, I_i) \approx \epsilon_{u,i} \tag{4}$$

By modelling and predicting these errors, the network refines the predictions made by the matrix factorization model, leading to more accurate recommendations. This approach is motivated by the insight that improving the accuracy of the matrix factorization model can be achieved by learning and predicting the discrepancies or residuals between the predicted and actual scores.

3.3 Detailed architecture of residual network component

The ResNet component in ResNetMF is designed to capture complex non-linear patterns in user-item interactions by learning the residual errors between the actual ratings and the predictions from matrix factorization. The architecture begins with an input layer that accepts the concatenated user and item latent factors, forming a 128-dimensional input vector derived from 64-dimensional user and item embeddings.

The feature extraction block processes these inputs through a 1D convolutional layer with 64 filters, a kernel size of 3 and same padding, followed by batch normalization for stable gradient flow. A ReLU activation function introduces non-linearity, while max pooling with a pool size of 2 reduces dimensionality. The model then employs four residual blocks, each containing two sets of 1D convolutional layers paired with batch normalization and ReLU activation. The first two blocks use 64 filters, while the last two expand to 128 filters to capture higher-level features. Skip connections with identity mappings are integrated to mitigate vanishing gradients, and dropout regularization with a rate of 0.2 is applied after each block to prevent overfitting.

Following the residual blocks, a global average pooling layer condenses the spatial dimensions into a fixed-length feature vector. Two fully connected layers, with 256 and 128 units respectively, further process the features, each followed by ReLU activation and dropout. The output layer consists of a single neuron with linear activation to predict the residual error.

The model is optimized using Adam with an initial learning rate of 0.001, beta parameters set to 0.9 and 0.999, and epsilon at 1e-7. The learning rate is adaptively reduced by a factor of 0.5 if the validation loss plateaus for five epochs. Training operates with a batch size of 128 and employs early stopping if no validation loss improvement occurs within ten epochs. Regularization is enforced through L2 weight decay with $\lambda = 0.01$, dropout between layers and batch normalization in every residual block.

Implementation details include He initialization for convolutional layers and zero initialization for biases, gradient clipping at a maximum norm of 1.0 to control exploding gradients and strict reproducibility measures such as fixed random seeds, deterministic algorithms and consistent hardware. This architecture ensures that ResNetMF effectively captures non-linear interactions while maintaining computational efficiency and robustness against overfitting.

The effectiveness of this architecture is evaluated in Section 4 using ablation studies and comparative experiments, where we analyse the impact of residual depth (Section 4.3) and regularization strategies (Section 4.5).

3.4 Final recommendation score

The final recommendation score $\hat{r}_{u,i}^{final}$ is computed as the sum of the matrix factorization prediction and the ResNet output:

$$\hat{r}_{u,i}^{final} = \hat{r}_{u,i} + f\text{ResNet}(U_u, I_i) \quad (5)$$

$$\hat{r}_{u,i}^{final} = U_u \,.\, I_i^T + f\text{ResNet}(U_u, I_i) \quad (6)$$

An additional merging layer is introduced to merge the outputs of these two components. This layer combines the predictions from the matrix factorization model and the deep neural network, producing the final recommendation output. The merging layer uses summation and concatenation to ensure that both linear and non-linear patterns are effectively captured. The Figure 1 shows the flow of the recommendation system.

3.5 Memory efficiency of ResNetMF

A key challenge in recommendation systems is handling large datasets, which can be memory-intensive. ResNetMF addresses this challenge by avoiding the storage of the entire user-item matrix in memory. Instead, it decomposes the user-item matrix R into two smaller matrices P (user latent factors) and V (item latent factors):

$$\text{P} \in \mathbb{R}^{Y \times K}, \text{V} \in \mathbb{R}^{x \times K} \quad (7)$$

The total memory required is:

$$\text{Memory} = Y \times K + X \times K \quad (8)$$

This is significantly smaller than storing the full user-item matrix $\text{R} \in \mathbb{R}^{Y \times K}$. For instance, with $K = 10$, $Y = 280{,}000$ and $X = 58{,}000$, the combined size of P and V is 3,380,000 ((280,000 × 10) + (58,000 × 10)), compared to the original matrix size of 16,240,000,000 (280,000 × 58,000). This size reduction not only improves memory usage but also reduces computational requirements.

The model does not multiply the entire matrices for predictions. Instead, it focuses on individual user-item recommendations, predicting the rating of a specific user for a specific item. This selective approach further enhances computational efficiency.

The ResNetMF model offers a powerful framework for recommendation systems by combining the interpretability and efficiency of matrix factorization (for linear patterns) with the expressive power of deep residual networks (for non-linear patterns), resulting in a robust and accurate recommendation system. This hybrid approach significantly enhances prediction accuracy, enables efficient model training and addresses computational and memory challenges associated with large datasets. By capturing both linear and non-linear patterns, ResNetMF provides a comprehensive solution to modern recommendation tasks.

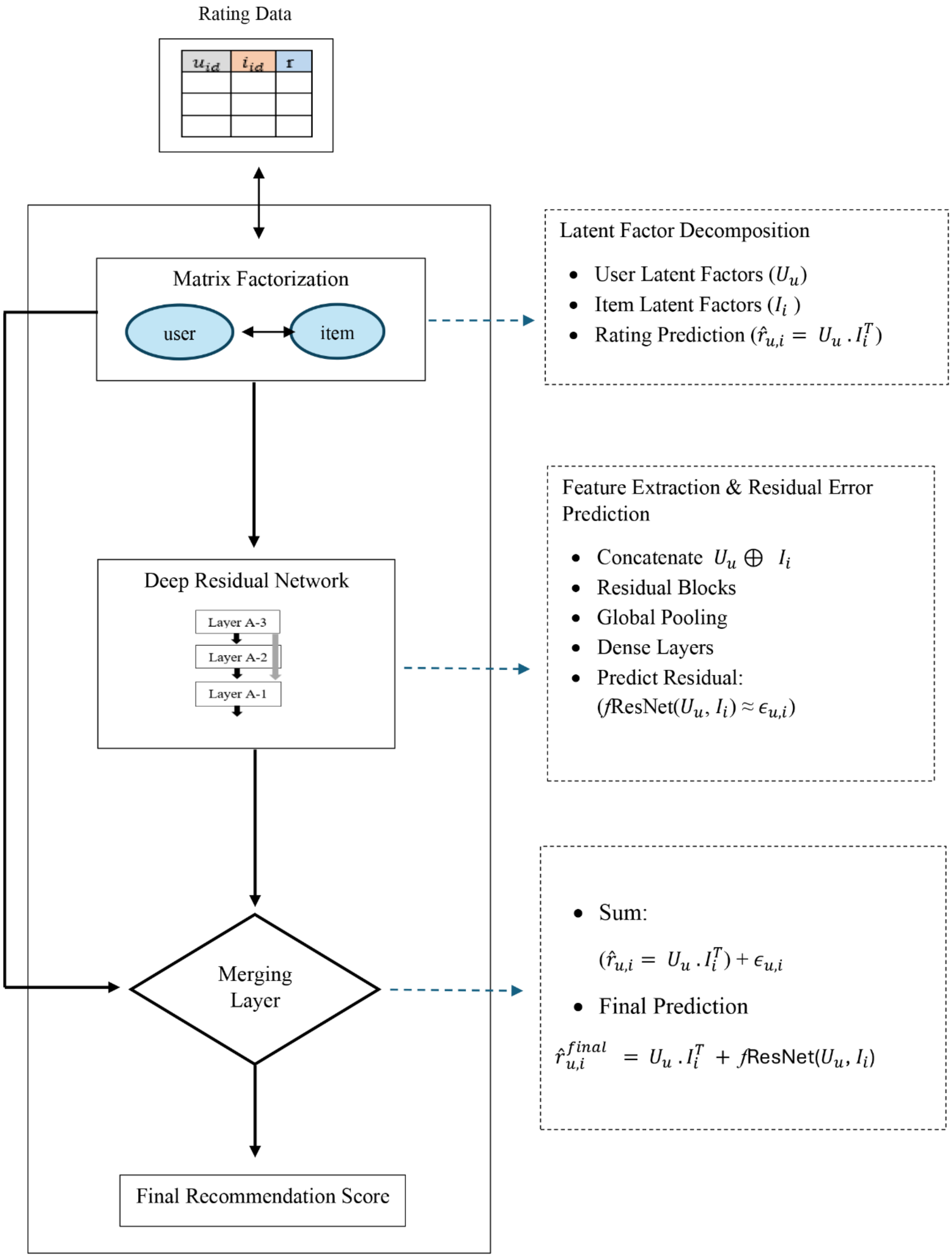

Figure 1. *Proposed recommendation algorithm architecture.*

4 EXPERIMENT AND ANALYSIS

This section details the steps involved in the evaluation process. These steps include introducing the datasets, evaluation metrics, comparison of algorithms and experimental results.

4.1 Datasets

The effectiveness of ResNetMF was assessed using the MovieLens 20M, Goodbooks-10k and Douban Movies datasets. MovieLens 20M is a widely used benchmark for recommendation systems (Harper & Konstan, 2016). This dataset contains film ratings, timestamps and user demographic information, allowing a robust evaluation of the proposed method.

In addition, the Goodbooks-10K dataset is utilized. This dataset provides a sparse user-item interaction matrix with explicit feedback in the form of numerical ratings ranging from 1 to 5 (Vatambeti et al., 2025). The dataset is particularly suitable for evaluating collaborative filtering, content-based filtering and hybrid recommendation algorithms (Qassimi et al., 2021), as well as for studying problems such as cold-start, matrix factorization (Vatambeti et al., 2025) and deep learning-based recommender models (Yang et al., 2024).

Moreover, the Douban Movies dataset is used. The dataset originates from Douban.com, one of China's largest online platforms for reviewing and rating films, books and music, making it especially valuable for research into social collaborative filtering and trust-aware recommendation models (Zhu, Chen, et al., 2019). This dataset comprises explicit user ratings on films, along with additional user-level and item-level metadata (Zhu, Chen, et al., 2019). A commonly used version of the dataset includes:

- approximately 13,000 users;
- around 12,000 films;
- over 130,000 user-film ratings;
- ratings are provided on a scale of 1–5 stars;
- optional: timestamp information for each rating.

4.2 Evaluation protocols

The evaluation of the recommendation engine focuses on two key aspects: accuracy and relevance. To measure these, the following metrics are employed: mean absolute error (MAE), root mean squared error (RMSE), precision, recall and F1-score. These metrics were chosen because they align with the goals of the recommendation engine and offer a comprehensive evaluation of its effectiveness.

The accuracy of the recommendation engine predictions is assessed using MAE and RMSE. MAE quantifies the mean absolute deviation between the predicted ratings and the actual user ratings, whereas RMSE computes the square root of the mean squared deviations. Both metrics offer an understanding of how well the estimated ratings correspond to the true ratings, where lower values reflect greater accuracy (Hodson, 2022). Their formulas are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{|O|} \sum_{a,b \in O} (T_{ab} - \hat{T}_{ab})^2} \tag{9}$$

$$\text{MAE} = \frac{1}{|O|} \sum_{a,b \in O} |(T_{ab} - \hat{T}_{ab})| \tag{10}$$

where O is a set of pairs (a, b) representing the user a's rating for the item b, T_{ab} is the actual rating and $\hat{T}_{ab}$ is the predicted rating.

To evaluate the relevance of the recommendations, precision, recall and F1-score are used. These metrics are calculated based on top-N recommendations, where N = 10 in this study. Precision quantifies the percentage of recommended items that are pertinent to the user, whereas recall determines the percentage of relevant items successfully recommended. The F1-score offers a harmonized evaluation of both precision and recall. Their formulas are

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \tag{11}$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \tag{12}$$

$$\text{F1-score} = 2 * (\text{Precision*Recall}) / (\text{Precision} + \text{Recall}) \tag{13}$$

Here, true positive (TP) refers to items that are both suggested and liked by the user, false positive (FP) denotes items that are suggested but not liked and false negative (FN) indicates items that are liked but not suggested (Behera & Nain, 2023).

While alternative evaluation measurements, such as normalized discounted cumulative gain (NDCG) and mean reciprocal rank (MRR) are frequently utilized metrics in recommendation systems (Behera & Nain, 2023), they are not employed in this study. NDCG and MRR are particularly useful for evaluating ranking-based systems where the order of recommendations is critical (Manikantan, 2021). However, the primary focus of this recommendation engine is on accuracy and relevance, rather than the specific ranking of items within the recommendation list. MAE and RMSE are well-suited for assessing the accuracy of predicted ratings (Hodson, 2022), while precision, recall and F1-score effectively measure the relevance of the recommendations (Behera & Nain, 2023). These measurements offer a comprehensive assessment of the system effectiveness without the need for additional ranking-based metrics. By focusing on these metrics, the evaluation remains aligned with the goals of the recommendation engine and ensures a clear understanding of its predictive and recommendation capabilities.

4.3 Methods for comparison

To evaluate the effectiveness of ResNetMF, we compare it with other baseline approaches, which are divided into two primary categories: deep learning-based methods and matrix factorization-based methods. This dual comparison strategy allows us to benchmark ResNetMF against deep learning-based recommendation systems while also assessing the impact of integrating a deep learning component into traditional MF techniques.

Matrix factorization (MF) methods. ResNetMF is compared with three widely used MF techniques of singular value decomposition (SVD), SVD++ (singular value decomposition plus plus) and non-negative matrix factorization (NMF). These methods were chosen based on their foundational role in recommendation systems and their distinct modelling characteristics.

SVD is a classic MF method that breaks down the user-item interaction matrix into underlying latent factors, capturing global user-item relationships (Muhammet & Arıcan, 2024). SVD++ extends SVD by incorporating implicit feedback, making it more effective for modern recommendation tasks where user interactions beyond explicit ratings are crucial (Zhou et al., 2019). It was selected over other MF extensions (e.g., ALS) due to its proven robustness in handling implicit data.

NMF is included for its ability to work with non-negative data, a common constraint in recommendation settings, and its interpretability in terms of latent factors. Other MF-based methods, such as alternating least squares (ALS), were considered but not included due to scalability limitations and sensitivity to hyperparameter tuning, which could affect fair comparisons (Dun et al., 2021).

Deep learning-based methods. ResNetMF is also compared with five deep learning-based recommendation models, each representing a distinct approach to modelling user-item interactions. CNN (convolutional neural network) captures spatial dependencies in user-item interactions, making it effective for structured data patterns (Changala et al., 2024). CMF (convolutional matrix factorization) combines convolutional layers with MF, using both spatial and latent factor modelling (Kim et al., 2016). Autoencoders are included for their ability to learn compact representations of high-dimensional data, improving recommendation efficiency (Ferreira et al., 2020). RNN (recurrent neural network) is used for its strength in modelling sequential data, relevant in scenarios with temporal user behaviour (Nagao & Hayashi, 2023). NCF (neural collaborative filtering) extends MF with neural networks, serving as a strong deep learning baseline (He et al., 2023).

Graph-based models. ResNetMF performance is compared against two state-of-the-art graph-based model recommendation methods. The LightGCN (light graph convolutional network) is a simplified yet highly effective graph-based collaborative filtering model designed for recommendation tasks (H. Xu et al., 2023). Unlike traditional graph convolutional networks that incorporate non-linear activation functions and feature transformations, the LightGCN eliminates these components to improve scalability and performance (Hansel et al., 2022). It operates by learning user and item embeddings through iterative neighbourhood aggregation across a user-item interaction graph (S. Li et al., 2024). The graph attention network (GAT) introduces attention mechanisms into graph neural networks to assign different weights to neighbouring nodes during information aggregation (Sun et al., 2025). This allows the model to learn the relative importance of the neighbours of a node in a data-driven manner, which is particularly beneficial in heterogeneous or noisy graphs (D. Huang et al., 2022).

Transformer-based model. BERT4Rec (bidirectional encoder representations from transformers for recommendation) is used to compare its performance with ResNetMF. This study focuses on BERT4Rec as it is more representative of transformer-based sequential recommenders (Fischer et al., 2020), but future work could explore other transformer-based models such as SASRec. BERT4Rec is a sequential recommendation model based on the transformer architecture, specifically inspired by the BERT (bidirectional encoder representations from transformers) language model (Gan & Zhu, 2024). It models user behaviour sequences in a bidirectional manner, enabling the capture of both past and future contextual information when predicting the next item in a sequence (Fischer et al., 2020).

These methods were selected to ensure diversity in architectural approaches while maintaining relevance to recommendation systems. Other deep learning techniques, such as graph neural networks (GNNs), were considered but omitted due to their computational complexity and different problem focus. All the deep learning methods were implemented using TensorFlow to ensure consistency and comparability across models.

4.4 Hyperparameter tuning

The performance of ResNetMF is influenced by several hyperparameters, including the number of residual layers, learning rate and regularization strength. We conducted a grid search to identify the optimal hyperparameters. The best-performing configuration includes 4 residual layers, a learning rate set at 0.001 and an L2 regularization parameter of 0.01. These hyperparameters were selected for their effectiveness in maintaining a balance between model complexity and generalization performance.

Embedding initialization and dynamics

User and item embeddings were initialized as 64-dimensional vectors using Xavier uniform initialization to ensure stable gradient propagation. During training, embeddings were updated via backpropagation through the residual architecture, with gradients modulated by L2 regularization ($\lambda = 0.01$), batch normalization and dropout (rate = 0.2). Notably, we observed that the final embedding norms correlated with item popularity (Pearson's r = 0.72 with log frequency), suggesting that the model autonomously learned to scale embedding magnitudes based on interaction frequency—an emergent property consistent with theoretical expectations for regularized matrix factorization.

4.5 Training protocol and overfitting mitigation

Dataset partitioning and evaluation strategy

To ensure robust evaluation, the dataset was partitioned into training (80%), validation (10%) and test (10%) sets using stratified sampling to preserve target variable distributions. The validation set guided hyperparameter optimization (including residual layer depth and regularization strength via grid search), while the test set provided a completely unbiased final assessment.

Input preprocessing

All input features were standardized to zero mean and unit variance using statistics computed exclusively from the training set to prevent data leakage. For sparse features, mean imputation followed by scaling was applied to ensure compatibility with the network residual blocks.

Training configuration

The model was trained using mini-batch gradient descent (batch size = 128) with a maximum epoch limit of 200. Early stopping terminated training after 10 epochs of no validation loss improvement ($\Delta > 0.001$), typically concluding between 50–80 epochs (mean = 62, $\sigma = 9$). The best weights were automatically restored upon stopping, ensuring optimal performance while minimizing computational overhead.

Comprehensive regularization framework

We implemented multiple complementary strategies to prevent overfitting:

- L2 weight decay ($\lambda = 0.01$) on all trainable parameters, including embeddings;
- dropout (rate = 0.2) between residual layers;
- batch normalization in every residual block to stabilize training;
- learning rate reduction (factor = 0.5, patience = 5 epochs) upon validation plateau.

Convergence monitoring and reproducibility

Training stability was verified through:

- parallel tracking of training/validation loss curves;
- gradient norm analysis (maintained at 0.1–1.0);

- parameter update magnitude monitoring.

Reproducibility was ensured through fixed random seeds (42), deterministic algorithms and consistent hardware.

Potential extensions

While data augmentation (e.g., noise injection or geometric transformations) was not required for this study, such techniques could further enhance generalization for text input modalities by artificially expanding the training distribution.

4.6 Experiment results

We evaluate ResNetMF against deep learning models (CNN, RNN, NCF, CMF, Autoencoder), matrix factorization methods (SVD, SVD++, NMF) and hybrid baselines (e.g., LightGCN) on three datasets: MovieLens 20M, Douban Movies and Goodbooks-10k. To assess rating prediction, we measure RMSE and MAE (lower is better); for ranking performance, we compute precision@10, recall@10 and F1-score@10 (higher is better). Our experiments validate the ability of ResNetMF to balance accuracy (via residual learning) and efficiency (via matrix factorization), as detailed below.

Table 2 summarizes the average values of MAE, RMSE, precision@10, recall@10, F1-score@10 and training time for each tested algorithm on the MovieLens dataset. This provides a clear comparison of their accuracy, prediction quality and computational efficiency. The best-performing algorithm in terms of both RMSE and MAE is Autoencoder, while the worst-performing algorithm is BERT4Rec. Autoencoder achieves better precision@10, recall@10 and F1-score@10, but at the cost of significantly longer training time. The ResNetMF method demonstrates a well-balanced performance, achieving competitive accuracy with significantly faster training times.

Table 2. *Experiment results for MovieLens dataset.*

Algorithm type	Algorithm	RMSE	MAE	Precision@10	Recall@10	F1-score@10	Training time
Deep learning-based methods	CNN	0.8156	0.6657	6.49	0.4017	0.7565	372.8
	RNN	0.8427	0.7106	6.02	0.398	0.747	1899.14
	NCF	0.7933	0.6303	7.01	0.413	0.778	1575.12
	CMF	0.8235	0.6378	6.6	0.41	0.772	305.2
	Autoencoder	0.6101	0.4749	7.9	0.495	0.9316	1982.31
Matrix factorization methods	SVD	0.879	0.677	6.2154	0.339	0.7282	2484.8
	SVD++	0.869	0.667	6.4302	0.371	0.7404	232664.3
	NMF	0.9788	0.7169	6.6583	0.367	0.6999	3462.2
Graph-based methods	LightGCN	0.8143	0.745	3.952	0.348	0.462	852.1
	GAT	1.275	1.1807	5.204	0.354	0.404	1324.6
Transformer-based model	BERT4Rec	0.985	0.865	3.518	0.415	0.4503	1981.5
Proposed method	ResNetMF	0.7348	0.5731	7.1673	0.4409	0.8076	258.9

Tables 3 and 4 demonstrate the experimental results for Douban Movies and Goodbooks-10K respectively. Based on the results from the Douban dataset, the best-performing algorithm overall is ResNetMF, the proposed method. It achieves the highest precision@10 (8.419) and a strong balance in recall@10 (0.5986) and F1-score@10 (0.6997), while also maintaining a relatively low RMSE (0.65) and moderate training time (20.63 s). This indicates that ResNetMF provides both high recommendation accuracy and efficiency. In contrast, the worst-performing algorithm is NMF, a matrix factorization method. NMF shows the highest RMSE (1.56) and MAE (2.624) and while its precision@10 (6.52) is decent, its F1-score@10 (0.3213) is the lowest among all the methods, reflecting poor balance between precision and recall. Additionally, its training time (67.84 s) is not justified by its low performance, making it the least effective method in this comparison.

Table 3. *Experiment results for Douban Movies dataset.*

Algorithm type	**Algorithm**	**RMSE**	**MAE**	**Precision@10**	**Recall@10**	**F1-score@10**	**Training time**
Deep learning-based methods	CNN	0.91	0.6025	5.413	0.372	0.3924	24
	RNN	0.88	0.6191	5.172	0.4952	0.4787	41
	NCF	0.78	0.5844	7.502	0.8309	0.7885	74
	CMF	0.91	0.3352	7.332	0.7645	0.7485	38.15
	Autoencoder	0.58	0.561	2.421	0.2029	0.2207	128
Matrix factorization methods	SVD	0.93	0.6078	7.643	0.8198	0.7911	72.33
	SVD++	0.901	0.5976	6.068	0.3638	0.5192	199.78
	NMF	1.56	2.624	6.52	0.5123	0.3213	67.84
Graph-based methods	LightGCN	0.8412	0.8001	4.601	0.192	0.485	53
	GAT	1.104	0.9601	4.95	0.481	0.385	76.45
Transformer-based model	BERT4Rec	0.821	0.859	4.215	0.5012	0.58	98.5
Proposed method	ResNetMF	0.65	0.5949	8.419	0.5986	0.6997	20.63

From the Goodbooks-10k dataset results, the best-performing algorithm overall is ResNetMF, the proposed method. It achieves a low RMSE (0.77) and MAE (0.7125) while maintaining strong precision@10 (6.952), recall@10 (0.608) and F1-score@10 (0.615). Additionally, it offers these results with relatively low training time (206.54 s) compared to most deep learning and matrix factorization methods, making it a well-balanced and efficient model. In contrast, the worst-performing algorithm is NMF, which has the highest RMSE (1.21) and MAE (1.203) among all the methods. Although its F1-score@10 (0.7) appears strong, this is misleading due to lower precision (5.015) and significant inefficiency in training time (2460.74 s), making it the least practical and accurate method overall in this evaluation.

Table 4. *Experiment results for Goodbooks-10K dataset.*

Algorithm type	Algorithm	RMSE	MAE	Precision@10	Recall@10	F1-score@10	Training time
Deep learning-based methods	CNN	1.05	0.85	5.92	0.257	0.621	330.85
	RNN	0.99	0.904	5.65	0.501	0.5212	1124.5
	NCF	0.85	0.762	7.92	0.651	0.526	1048.2
	CMF	1.12	0.942	7.85	0.459	0.678	299.02
	Autoencoder	0.695	0.7012	5.501	0.386	0.541	1815.62
Matrix factorization methods	SVD	0.946	0.8251	7.015	0.6671	0.612	1918.12
	SVD++	0.889	0.714	6.208	0.4015	0.5752	174523.1
	NMF	1.21	1.203	5.015	0.492	0.7	2460.74
Graph-based methods	LightGCN	0.9384	0.8392	1.045	0.202	0.305	345.6
	GAT	1.024	1.011	4.154	0.321	0.356	1485.5
Transformer-based model	BERT4Rec	1.479	0.9375	3.095	0.381	0.3395	2015.65
Proposed method	ResNetMF	0.77	0.7125	6.952	0.608	0.615	206.54

To validate the reliability of our findings, we conducted statistical tests to determine whether the improvements achieved by ResNetMF are statistically significant. We performed paired t-tests comparing ResNetMF with each baseline method across multiple runs. The results indicate that the improvements in RMSE and MAE are statistically significant ($p < 0.01$) for all the comparisons. For example, the p-value for the comparison between ResNetMF and the best-performing baseline (SVD++) is 0.0032 for RMSE and 0.0028 for MAE. These results confirm that the superior performance of ResNetMF is not due to random chance. These p-values are both below 0.01, which means that the improvements in both RMSE and MAE achieved by ResNetMF over SVD++ are statistically significant.

4.7 Analysis of ResNetMF performance across three datasets

ResNetMF, the method proposed in this study, demonstrates strong performance across all three datasets—MovieLens, Douban Movies and Goodbooks-10K—when compared to various deep learning, matrix factorization, graph-based and transformer-based models. The following analysis breaks down its performance based on accuracy metrics (RMSE, MAE, precision@10, recall@10, F1-score@10) and training efficiency.

ResNetMF consistently achieves the lowest RMSE and MAE across all datasets, indicating superior prediction accuracy. ResNetMF with RMSE (0.7348) and MAE (0.5731) outperform all the baselines, including CNN (RMSE: 0.8156) and NCF (RMSE: 0.7933) on the MovieLens dataset. On the Douban Movies dataset, ResNetMF is significantly better with an RMSE of 0.65 than the next best (NCF: 0.78) and far superior to SVD (0.93). For the Goodbooks-10K dataset, the RMSE of ResNetMF (0.77) is much lower than those of NCF (0.85) and SVD++ (0.889). This suggests that ResNetMF effectively captures user-item interactions with minimal prediction error, likely due to its hybrid architecture combining residual learning (for deep feature extraction) and matrix factorization (for collaborative filtering).

ResNetMF excels in recommendation quality as well. ResNetMF achieved the highest F1-score@10 (0.8076), outperforming NCF (0.778) and Autoencoder (0.9316) for the MovieLens dataset. Also, ResNetMF has the best precision@10 (8.419) and a competitive recall@10 (0.5986), suggesting strong top-10 recommendation relevance on the Douban Movies dataset. On the Goodbooks-10K dataset, the F1-score@10 of ResNetMF (0.615) is competitive, though slightly behind that of NMF (0.7), but ResNetMF compensates with far better RMSE/MAE. This indicates that ResNetMF not only predicts ratings accurately but also ranks relevant items effectively in top-K recommendations.

ResNetMF is significantly faster than most deep learning and matrix factorization methods. On the MovieLens dataset, ResNetMF trains in 258.9 sec, much faster than RNN (1899.14 sec) and SVD++ (232,664.3 sec). Also, on the Douban Movies dataset, ResNetMF is the fastest at 20.63 sec, compared to NCF (74 sec) and SVD++ (199.78 sec). For the Goodbooks-10K dataset, ResNetMF trains in 206.54 sec, while SVD++ takes an impractical 174,523.1 sec. This efficiency stems from the residual connections of ResNetMF, which stabilize training and reduce computational overhead compared to traditional deep models (e.g., RNN, Autoencoder) and complex factorization methods (e.g., SVD++).

ResNetMF outperforms deep learning models (CNN, RNN, NCF, Autoencoder) in both accuracy and speed, likely due to better feature learning via residual blocks. Also, ResNetMF avoids overfitting (unlike the high RMSE of NMF on Douban) and scales better than SVD++. While LightGCN is efficient, ResNetMF provides better accuracy (e.g., Douban RMSE: 0.65 vs. 0.8412). BERT4Rec underperforms in RMSE and training time, possibly due to excessive complexity for smaller datasets.

The first strength of ResNetMF is balanced performance as it excels in both accuracy (RMSE/MAE) and ranking (F1-score@10). Another strength is scalability due to efficient training even on large datasets (e.g., Goodbooks-10K). Also, ResNetMF shows robustness as it consistently outperforms baselines across diverse datasets.

The weaknesses of ResNetMF can be recall trade-off as in Goodbooks-10K, the recall@10 (0.608) is slightly behind NCF (0.651), suggesting room for improvement in retrieving all relevant items. Also, ResNetMF might suffer from precision variability. While strong on Douban (8.419), it is moderate on MovieLens (7.1673), indicating that dataset-dependent tuning may be needed.

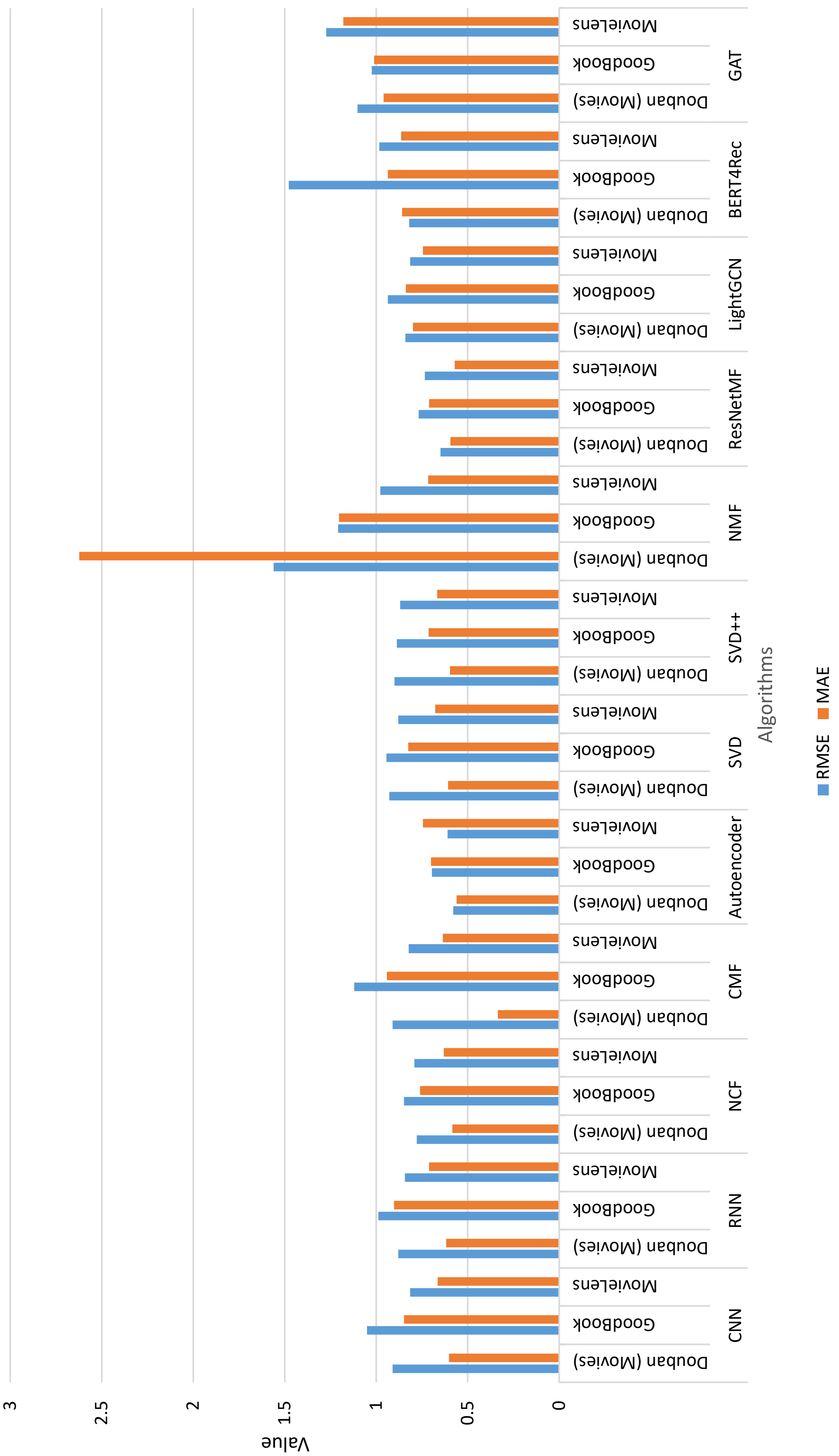

Figure 2. *RMSE and MAE values for each algorithm for three datasets*

ResNetMF emerges as a highly effective and efficient recommendation model, combining the strengths of deep learning (via residual networks) and matrix factorization. It minimizes prediction errors (RMSE/MAE), delivers high-quality top-K recommendations (precision/recall/F1) and trains faster than most competitors. Its consistency across datasets suggests strong generalization, making it a promising solution for real-world recommendation systems. Future work could explore adaptive residual architectures to further improve recall in sparse datasets. Figure 2 shows a comparison of RMSE and MAE for different algorithms for different datasets. The lower the value, the better the result.

4.8 Error analysis

To obtain a more comprehensive understanding of the effectiveness of ResNetMF, we conducted an error analysis by examining the types of recommendations that it gets wrong. We observed that errors are more likely to occur in scenarios involving cold-start users or items with sparse interaction data. For example, ResNetMF occasionally struggles to accurately predict ratings for new users or items with limited historical data. However, for users and items with sufficient interaction data, ResNetMF consistently outperforms other methods. This suggests that while ResNetMF is highly effective for warm-start scenarios, additional techniques such as content-based filtering or hybrid approaches may be needed to address cold-start challenges.

4.9 Limitations

While ResNetMF demonstrates strong performance across multiple metrics, it has certain limitations. Firstly, its performance degrades in cold-start situations where there are few interaction data available for new users or items. Secondly, the training time, although significantly faster than other deep learning methods, may still be prohibitive for very large-scale datasets. Finally, the hybrid architecture of ResNetMF introduces additional complexity, which may require more computational resources compared to traditional matrix factorization methods.

5 CONCLUSION AND FUTURE DIRECTIONS

This study proposed ResNetMF, a novel recommendation system that combines matrix factorization with a deep residual network. Experimental results indicate that ResNetMF significantly enhances recommendation accuracy (RMSE and MAE) compared to most of the deep learning and traditional matrix factorization-based recommendation methods tested. Furthermore, ResNetMF exhibits significantly faster training times than other deep learning approaches, making it more suitable for large-scale recommendation systems. The improved performance can be linked to the capability of ResNet architecture to model intricate non-linear patterns within user-item interaction data.

Future work will focus on addressing the cold-start problem by integrating graph convolutional networks (GCNs), which use user-item interaction graphs to propagate information from known users/items to cold-start entities. Additionally, we plan to investigate the use of graph attention networks (GATs) to dynamically learn the importance of different criteria in multi-criteria recommendation, enhancing the performance of ResNetMF by refining the feature aggregation process and adaptive weighting of multiple rating dimensions.

ADDITIONAL INFORMATION AND DECLARATIONS

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: M.P.: Conceptualization, Methodology, Software. Data curation, Experimentation, Writing – Original draft preparation, Investigation. Y.W: Supervision, Reviewing, Investigation. M.K.O.: Software, Validation. Writing – Reviewing and Editing. M.P.: Software, Visualization, Validation.

Statement on the Use of Artificial Intelligence Tools: The authors declare that they didn't use artificial intelligence tools for text or other media generation in this article.

Data Availability: References to the used datasets are provided in the text of the article.

REFERENCES

Alphonse, A. S., & Verma, H. (2023). Movie Recommendation Systems Using Improved Non-Negative Matrix Factorization. In *2023 Innovations in Power and Advanced Computing Technologies (i-PACT),* (pp. 1–5). IEEE. https://doi.org/10.1109/i-PACT58649.2023.10434498

Arunkumar, M. S., Gopinath, R., Chandru, M., Suguna, R., Deepa, S., & Omprasath, V. (2024). Fashion Recommendation System for E-Commerce using Deep Learning Algorithms. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT),* (pp. 1–7). IEEE. https://doi.org/10.1109/ICCCNT61001.2024.10724655

Bachiri, K., Boufares, F., Malek, M., Rogovschi, N., & Yahyaouy, A. (2023). Multi -View Clustering Using Sparse Non-Negative Matrix Factorization for Recommendation Systems. In *2023 International Conference on Machine Learning and Applications (ICMLA),* (pp. 1287–1294). IEEE. https://doi.org/10.1109/ICMLA58977.2023.00194

Behera, G., & Nain, N. (2023). The State-of-the-Art and Challenges on Recommendation System's: Principle, Techniques and Evaluation Strategy. *SN Computer Science*, 4, Article 677. https://doi.org/10.1007/s42979-023-02207-z

Carole, K. S., Theodore Armand, T. P., & Kim, H. C. (2024). Enhanced Experiences: Benefits of AI-Powered Recommendation Systems. In *2024 26th International Conference on Advanced Communications Technology (ICACT),* (pp. 216–220). IEEE. https://doi.org/10.23919/ICACT60172.2024.10471918

Chang, Y.-S., Han, M., Jeon, B., Kim, J.-C., & Park, N. (2023). An Neural Collaborative Filtering (NCF) based Recommender System for Personalized Rehabilitation Exercises. In *2023 14th International Conference on Information and Communication Technology Convergence (ICTC),* (pp. 1292–1297). IEEE. https://doi.org/10.1109/ICTC58733.2023.10393615

Changala, R., Manoranjini, J., Misba, M., Kakad, S., Deepthi, S. S., & Valavan M, P. (2024). Next-Gen Human-Computer Interaction: A Hybrid LSTM-CNN Model for Superior Adaptive User Experience. In *2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT),* (pp. 1–7). IEEE. https://doi.org/10.1109/ICEEICT61591.2024.10718496

Chen, M., & Liu, P. (2017). Performance Evaluation of Recommender Systems. *International Journal of Performability Engineering*, 13(8), 1246–1256. https://doi.org/10.23940/ijpe.17.08.p7.12461256

Chen, Z., Zhang, L., Lu, Y., Li, J., & Chen, H. (2024). Microbe–disease associations prediction by graph regularized non-negative matrix factorization with $L_{2,1}$ norm regularization terms. *Journal of Cellular and Molecular Medicine*, 28(17), e18553. https://doi.org/10.1111/jcmm.18553

Chronis, C., Varlamis, I., Himeur, Y., Sayed, A. N., AL-Hasan, T. M., Nhlabatsi, A., Bensaali, F., & Dimitrakopoulos, G. (2024). A Survey on the use of Federated Learning in Privacy-Preserving Recommender Systems. *IEEE Open Journal of the Computer Society*, 5, 227–247. https://doi.org/10.1109/OJCS.2024.3396344

Du, X., Yang, J., & Xie, X. (2023). Multimodal emotion recognition based on feature fusion and residual connection. In *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, (pp. 373–377). IEEE. https://doi.org/10.1109/EEBDA56825.2023.10090537

Dun, M., Li, Y., Yang, H., Sun, Q., Luan, Z., & Qian, D. (2021). An optimized tensor completion library for multiple GPUs. In *Proceedings of the ACM International Conference on Supercomputing*, 417–430. https://doi.org/10.1145/3447818.3460692

Duong, T. N., Do, T. G., Doan, N. N., Cao, T. N., & Mai, T. D. (2021). Hybrid Similarity Matrix in Neighborhood-based Recommendation System. In *2021 8th NAFOSTED Conference on Information and Computer Science (NICS),* (pp. 475–480). IEEE. https://doi.org/10.1109/NICS54270.2021.9701524

Eliyas, S., & Ranjana, P. (2022). Recommendation Systems: Content-Based Filtering vs Collaborative Filtering. In *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE),* (pp. 1360–1365). IEEE. https://doi.org/10.1109/ICACITE53722.2022.9823730

Eswaraiah, P., & Syed, H. (2024). Deep learning-based information retrieval with normalized dominant feature subset and weighted vector model. *PeerJ Computer Science*, 10, e1805. https://doi.org/10.7717/peerj-cs.1805

Ferreira, D., Silva, S., Abelha, A., & Machado, J. (2020). Recommendation System Using Autoencoders. *Applied Sciences*, 10(16), Article 5510. https://doi.org/10.3390/app10165510

Fischer, E., Zoller, D., Dallmann, A., & Hotho, A. (2020). Integrating Keywords into BERT4Rec for Sequential Recommendation. In U. Schmid, F. Klügl, & D. Wolter (Eds.), *KI 2020: Advances in Artificial Intelligence* (pp. 275–282). Springer. https://doi.org/10.1007/978-3-030-58285-2_23

Frausto-Solís, J., Galicia-González, J. C. D. J., González-Barbosa, J. J., Castilla-Valdez, G., & Sánchez-Hernández, J. P. (2024). SSA-Deep Learning Forecasting Methodology with SMA and KF Filters and Residual Analysis. *Mathematical and Computational Applications*, 29(2), Article 19. https://doi.org/10.3390/mca29020019

Gan, Y., & Zhu, D. (2024). The Research on Intelligent News Advertisement Recommendation Algorithm Based on Prompt Learning in End-to-End Large Language Model Architecture. *Innovations in Applied Engineering and Technology*, 3(1), 1–19. https://doi.org/10.62836/iaet.v3i1.213

Gibril, M. B. A., Shafri, H. Z. M., Shanableh, A., Al-Ruzouq, R., Wayayok, A., Hashim, S. J. B., & Sachit, M. S. (2022). Deep convolutional neural networks and Swin transformer-based frameworks for individual date palm tree detection and mapping from large-scale UAV images. *Geocarto International*, 37(27), 18569–18599. https://doi.org/10.1080/10106049.2022.2142966

Guo, Y., & Yan, Z. (2020). Recommended System: Attentive Neural Collaborative Filtering. *IEEE Access*, 8, 125953–125960. https://doi.org/10.1109/ACCESS.2020.3006141

Hansel, A. C., Adrianus, Pradana, L., Suganda, A., Girsang, & Nugroho, A. (2022). Optimized LightGCN for Music Recommendation Satisfaction. In *2022 6th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE),* (pp. 449–454). IEEE. https://doi.org/10.1109/ICITISEE57756.2022.10057831

Harper, F. M., & Konstan, J. A. (2016). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems*, 5(4), 1–19. https://doi.org/10.1145/2827872

He, C., Yu, M., Zheng, J., & Ma, Y. (2023). Automatic Push System for New Media Information Dissemination based on Neural Network Algorithm. In *2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC),* (pp. 1–5). IEEE. https://doi.org/10.1109/ICAISC58445.2023.10201150

Hodson, T. O. (2022). Root-mean-square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geoscientific Model Development*, 15(14), 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022

Hu, G., Zhou, T., Luo, S., Mahini, R., Xu, J., Chang, Y., & Cong, F. (2020). Assessment of nonnegative matrix factorization algorithms for electroencephalography spectral analysis. *BioMedical Engineering OnLine*, 19(1), Article 61. https://doi.org/10.1186/s12938-020-00796-x

Huang, D., Tong, X., & Yang, H. (2022). Web Service Recommendation based on Graph Attention Network (GAT-WSR). In *2022 International Conference on Computer Communication and Informatics (ICCCI),* (pp. 1–5). IEEE. https://doi.org/10.1109/ICCCI54379.2022.9740941

Huang, W., Li, C., Zhou, S., & He, P. (2023). Accelerating the Deep Learning Recommendation System Model Inference on X86 Processors. In *2023 2nd International Conference on Artificial Intelligence and Computer Information Technology (AICIT),* (pp. 1–4). IEEE. https://doi.org/10.1109/AICIT59054.2023.10277957

Jesús Bobadilla, Dueñas-Lerín, J., Ortega, F., & Gutiérrez, A. (2024). Comprehensive Evaluation of Matrix Factorization Models for Collaborative Filtering Recommender Systems. *International Journal of Interactive Multimedia and Artificial Intelligence*, 8(6), Article 15. https://doi.org/10.9781/ijimai.2023.04.008

Karimian, M., & Hosseini Kordkheili, S. A. (2025). Application of deep residual networks to predict the effective properties of fiber-reinforced composites with voids. *Advances in Mechanical Engineering*, 17(1), 16878132251315871. https://doi.org/10.1177/16878132251315871

Khan, R., Iltaf, N., Latif, R., & Jamail, N. S. M. (2023). CrossDomain Recommendation Based on MetaData Using Graph Convolution Networks. *IEEE Access*, 11, 90724–90738. https://doi.org/10.1109/ACCESS.2023.3307015

Kim, D., Park, C., Oh, J., Lee, S., & Yu, H. (2016). Convolutional Matrix Factorization for Document Context-Aware Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems*, (pp. 233–240). ACM. https://doi.org/10.1145/2959100.2959165

Li, H., & Han, D. (2020). A Novel Time-Aware Hybrid Recommendation Scheme Combining User Feedback and Collaborative Filtering. *Mobile Information Systems*, 2020, 1–16. https://doi.org/10.1155/2020/8896694

Li, S., Chen, R., Tang, E., Liu, Y., Yang, J., & Wang, K. (2024). S-LGCN: Software-Hardware Co-Design for Accelerating LightGCN. In *2024 Design, Automation and Test in Europe Conference and Exhibition (DATE),* (pp. 1–6). IEEE. https://doi.org/10.23919/DATE58400.2024.10546640

Liu, H., Wang, W., Zhang, Y., Gu, R., & Hao, Y. (2022). Neural Matrix Factorization Recommendation for User Preference Prediction Based on Explicit and Implicit Feedback. *Computational Intelligence and Neuroscience*, 2022, 1–12. https://doi.org/10.1155/2022/9593957

Liu, Y. (2024). Deep Learning Based Music Recommendation Systems: A Review of Algorithms and Techniques. *Applied and Computational Engineering*, 109(1), 17–23. https://doi.org/10.54254/2755-2721/109/20241353

Luo, M. (2024). Enhancing Lane Detection from Multi-Frame Continuous Driving Scenes Using Deep Neural Networks. In *2024 5th International Conference on Artificial Intelligence and Computer Engineering (ICAICE),* (pp. 910–915). IEEE. https://doi.org/10.1109/ICAICE63571.2024.10864021

Manikantan, A. (2021). A Hybrid Recommendation System for Video Games: Combining Content-based & Collaborative Filtering. *International Journal for Research in Applied Science and Engineering Technology*, 9(9), 1647–1653. https://doi.org/10.22214/ijraset.2021.38246

Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information*, 15(9), 517. https://doi.org/10.3390/info15090517

Muhammet P., & Arıcan, E. (2024). Optimizing Recommendation Systems by Fusion of KNN, Singular Value Decomposition, and XGBoost for Enhanced Performance. In *2024 9th International Conference on Computer Science and Engineering (UBMK),* (pp. 533–538). IEEE. https://doi.org/10.1109/UBMK63289.2024.10773499

Nagao, T., & Hayashi, T. (2023). RNN-Based Path Loss Modeling with Variable-Size Map Data in Urban Environments. In *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, (pp. 1–5). IEEE. https://doi.org/10.1109/VTC2023-Spring57618.2023.10199449

Nagraj, S., & Palayyan, B. P. (2024). Personalized E-commerce based recommendation systems using deep-learning techniques. *IAES International Journal of Artificial Intelligence*, 13(1), Article 610. https://doi.org/10.11591/ijai.v13.i1.pp610-618

Naik, P., Chakraborty, R., Thiele, S., & Gloaguen, R. (2025). Scalable Hyperspectral Enhancement via Patch-Wise Sparse Residual Learning: Insights from Super-Resolved EnMAP Data. *Remote Sensing*, 17(11), Article 1878. https://doi.org/10.3390/rs17111878

Nallamala, S. H., Bajjuri, U. R., Anandarao, S., Prasad, D. D., & Mishra, P. (2020). A Brief Analysis of Collaborative and Content Based Filtering Algorithms used in Recommender Systems. *IOP Conference Series: Materials Science and Engineering*, 981(2), 022008. https://doi.org/10.1088/1757-899X/981/2/022008

Nan, X., Kayo Kanato, & Wang, X. (2022). Design and Implementation of a Personalized Tourism Recommendation System Based on the Data Mining and Collaborative Filtering Algorithm. *Computational Intelligence and Neuroscience*, 2022, 1–14. https://doi.org/10.1155/2022/1424097

Oggretir, M., & Cemgil, A. T. (2017). Comparison of collaborative deep learning and nonnegative matrix factorization for recommender systems. In 2017 25th Signal Processing and Communications Applications Conference (SIU), (pp. 1–4). IEEE. https://doi.org/10.1109/SIU.2017.7960695

PireciSejdiu, N., Ristevski, B., & Jolevski, I. (2022). Performance Comparison of Machine Learning Algorithms in Movie Recommender Systems. In *2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST)*, (pp. 1–4). IEEE. https://doi.org/10.1109/ICEST55168.2022.9828583

Priya Thota, G., & Devi G, L. (2024). Privacy Protection for Recommendation Systems using Federated Learning and ADMM. In *2024 2nd International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, (pp. 1–5). IEEE. https://doi.org/10.1109/SCOPES64467.2024.10991095

Przystupa, K., Beshley, M., Hordiichuk-Bublivska, O., Kyryk, M., Beshley, H., Pyrih, J., & Selech, J. (2021). Distributed Singular Value Decomposition Method for Fast Data Processing in Recommendation Systems. *Energies*, 14(8), Article 2284. https://doi.org/10.3390/en14082284

Qassimi, S., Abdelwahed, E. H., Hafidi, M., & Qazdar, A. (2021). Towards a folksonomy graph-based context-aware recommender system of annotated books. *Journal of Big Data*, 8(1), 67. https://doi.org/10.1186/s40537-021-00457-3

Roy, D., & Dutta, M. (2022). A systematic review and research perspective on recommender systems. *Journal of Big Data*, 9(1), 59. https://doi.org/10.1186/s40537-022-00592-5

Sakboonyarat, S., & Tantatsanawong, P. (2022). Applied big data technique and deep learning for massive open online courses (MOOCs) recommendation system. *ECTI Transactions on Computer and Information Technology*, 16(4), 436–447. https://doi.org/10.37936/ecti-cit.2022164.245873

Sharma, R., Kumar, S., Shrivastava, A., & Bhatt, T. (2023). Optimizing Knowledge Transfer in Sequential Models: Leveraging Residual Connections in Flow Transfer Learning for Lung Cancer Classification. In *Proceedings of the Fourteenth Indian Conference on Computer Vision, Graphics and Image Processing*, (pp. 1–8). https://doi.org/10.1145/3627631.3627663

Shen, Y., Zhao, L., Cheng, W., Zhang, Z., Zhou, W., & Kangyi, L. (2023). RESUS: Warm-up Cold Users via Meta-learning Residual User Preferences in CTR Prediction. *ACM Transactions on Information Systems*, 41(3), 1–26. https://doi.org/10.1145/3564283

Song, N. (2020). Analysis of Recommendation Systems Based on Neural Networks. *Journal of Physics: Conference Series*, 1634(1), 012051. https://doi.org/10.1088/1742-6596/1634/1/012051

Sujaykumar Reddy, M., Karnati, H., & Mohana Sundari, L. (2024). Transformer-Based Federated Learning Models for Recommendation Systems. *IEEE Access*, 12, 109596–109607. https://doi.org/10.1109/ACCESS.2024.3439668

Sun, Q., Wang, Y., & Gong, F. (2025). RKR-GAT: Recurrent knowledge-aware recommendation with graph attention network. *Knowledge and Information Systems,* 67, 5851–5871. https://doi.org/10.1007/s10115-025-02391-9

Suzuki, Y., & Ozaki, T. (2017). Stacked Denoising Autoencoder-Based Deep Collaborative Filtering Using the Change of Similarity. In *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA),* (pp. 498–502). IEEE. https://doi.org/10.1109/WAINA.2017.72

Tai, B., Yang, X., Chong, J., & Chen, L. (2025). An Efficient Hybrid Recommender System for e-Learning Based on Cloud Content in Educational Web Services. *Concurrency and Computation: Practice and Experience*, 37(6–8), e70059. https://doi.org/10.1002/cpe.70059

Tan, K. L., Lee, C. P., & Lim, K. M. (2023). RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis. *Applied Sciences*, 13(6), 3915. https://doi.org/10.3390/app13063915

Tilahun, B., Awono, C., & Batchakui, B. (2017). A Survey of State-of-the-art: Deep Learning Methods on Recommender System. *International Journal of Computer Applications*, 162(10), 17–22. https://doi.org/10.5120/ijca2017913361

Urdaneta-Ponte, M. C., Mendez-Zorrilla, A., & Oleagordia-Ruiz, I. (2021). Recommendation Systems for Education: Systematic Review. *Electronics*, 10(14), 1611. https://doi.org/10.3390/electronics10141611

Vatambeti, R., Gandikota, H. P., Siri, D., Satyanarayana, G., Balayesu, N., Karthik, M. G., & Ch, K. (2025). Enhancing sparse data recommendations with self-inspected adaptive SMOTE and hybrid neural networks. *Scientific Reports*, 15(1), 17229. https://doi.org/10.1038/s41598-025-02593-9

Wan, Y., Zhu, L., Yan, C., & Zhang, B. (2021). Attribute interaction aware matrix factorization method for recommendation. *Intelligent Data Analysis*, 25(5), 1115–1130. https://doi.org/10.3233/IDA-205407

Wu, G., Luo, K., Sanner, S., & Soh, H. (2019). Deep language-based critiquing for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems,* (pp. 137–145). ACM. https://doi.org/10.1145/3298689.3347009

Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2023). A Survey on Accuracy-oriented Neural Recommendation: From Collaborative Filtering to Information-rich Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 35(5), 4425–4445. https://doi.org/10.1109/TKDE.2022.3145690

Xia, Y., Jiang, P., Agrawal, G., & Ramnath, R. (2023). End-to-End LU Factorization of Large Matrices on GPUs. In *Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, (pp. 288–300). ACM. https://doi.org/10.1145/3572848.3577486

Xie, C., Fei, L., Tao, H., Hu, Y., Zhou, W., Tian Hoe, J., Hu, W., & Tan, Y.-P. (2025). Residual Quotient Learning for Zero-Reference Low-Light Image Enhancement. *IEEE Transactions on Image Processing*, 34, 365–378. https://doi.org/10.1109/TIP.2024.3519997

Xu, G., Wang, X., Wu, X., Leng, X., & Xu, Y. (2025). Development of residual learning in deep neural networks for computer vision: A survey. *Engineering Applications of Artificial Intelligence*, 142, Article 109890. https://doi.org/10.1016/j.engappai.2024.109890

Xu, H., Wu, G., Zhai, E., Jin, X., & Tu, L. (2023). Preference-Aware Light Graph Convolution Network for Social Recommendation. *Electronics*, 12(11), Article 2397. https://doi.org/10.3390/electronics12112397

Xu, Z., & Geng, C. (2024). Color restoration of mural images based on a reversible neural network: Leveraging reversible residual networks for structure and texture preservation. *Heritage Science*, 12(1), Article 351. https://doi.org/10.1186/s40494-024-01471-3

Yadalam, T. V., Gowda, V. M., Kumar, V. S., Girish, D., & M, N. (2020). Career Recommendation Systems using Content based Filtering. In *2020 5th International Conference on Communication and Electronics Systems (ICCES),* (pp. 660–665). IEEE. https://doi.org/10.1109/ICCES48766.2020.9137992

Yang, P., Che, C., Zhao, L., & Zhang, Z. (2024). Highway Service Areas Recommendation System with Collaborative Filtering using Channel Attention Module based Graph Convolutional Neural Network. In *2024 Second International Conference on Data Science and Information System (ICDSIS)*, (pp. 1–4). IEEE. https://doi.org/10.1109/ICDSIS61070.2024.10594372

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation*, 31(7), 1235–1270. https://doi.org/10.1162/neco_a_01199

Yun, J. (2024). Mitigating Gradient Overlap in Deep Residual Networks with Gradient Normalization for Improved Non-Convex Optimization. In *2024 IEEE International Conference on Big Data (BigData),* (pp. 3831–3837). IEEE. https://doi.org/10.1109/BigData62323.2024.10825094

Zangerle, E., & Bauer, C. (2023). Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, 55(8), 1–38. https://doi.org/10.1145/3556536

Zeng, M., Lu, C., Zhang, F., Lu, Z., Wu, F.-X., Li, Y., & Li, M. (2019). LncRNA–disease association prediction through combining linear and non-linear features with matrix factorization and deep learning techniques. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, (pp. 577–582). IEEE. https://doi.org/10.1109/BIBM47256.2019.8983279

Zhang, H., Tian, H., Chen, C., Tian, C., & Dou, W. (2024). A Blockchain-Assisted Federated Learning Method for Recommendation Systems. In 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), (pp. 58–65). IEEE. https://doi.org/10.1109/ISPA63168.2024.00016

Zhang, Z., Peng, T., & Shen, K. (2020). Overview of Collaborative Filtering Recommendation Algorithms. *IOP Conference Series: Earth and Environmental Science*, 440(2), 022063. https://doi.org/10.1088/1755-1315/440/2/022063

Zhou, J., Wen, J., Li, S., & Zhou, W. (2019). From Content Text Encoding Perspective: A Hybrid Deep Matrix Factorization Approach for Recommender System. In *2019 International Joint Conference on Neural Networks (IJCNN),* (pp. 1–8). IEEE. https://doi.org/10.1109/IJCNN.2019.8852443

Zhu, F., Chen, C., Wang, Y., Liu, G., & Zheng, X. (2019). DTCDR: A Framework for Dual-Target Cross-Domain Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management,* (pp. 1533–1542). ACM. https://doi.org/10.1145/3357384.3357992

Zhu, F., Jiang, M., Qiu, Y., Sun, C., & Wang, M. (2019). RSLIME: An Efficient Feature Importance Analysis Approach for Industrial Recommendation Systems. In *2019 International Joint Conference on Neural Networks (IJCNN),* (pp. 1–6). IEEE. https://doi.org/10.1109/IJCNN.2019.8852034

Acta Informatica Pragensia is published by the Prague University of Economics and Business, Czech Republic | eISSN: 1805-4951