ISSN 2723-6471

SiMoI New Method to Solve the Sparsity Problem in Collaborative Filtering

Hendra Kurniawan[1], Sri Lestari[2,*], Sushanty Saleh[3], Rafli Banu Satrio[4]

[1,2,3,4]*Darmajaya Institute of Informatics and Business, Lampung, Indonesia*

(Received: June 1, 2025; Revised: August 5, 2025; Accepted: November 7, 2025; Available online: December 1, 2025)

Abstract

Sparsity data is a major challenge in collaborative recommendation systems, characterized by the predominance of missing values within the user-item matrix. When a substantial portion of data is unavailable, the estimation process becomes hindered, and prediction accuracy declines due to limited usable information. To address this issue, this study introduces a novel method called SiMoI (Similarity, Mode, and Minimum Imputation), which is adaptively designed to handle high levels of sparsity. The SiMoI method combines user similarity with imputation strategies based on mode and minimum values. By leveraging subsets of the most informative users and items, the method efficiently fills missing entries while maintaining prediction stability. Evaluation was conducted using both real and synthetic datasets with varying sizes and degrees of sparsity, including an extreme scenario with 93.7% missing data. Experimental results show that SiMoI consistently produces more accurate predictions than baseline methods. Under high-sparsity conditions, SiMoI achieved an RMSE as low as 0.823, outperforming KNNI (0.947) and MEAN (1.021). Moreover, SiMoI demonstrated resilience across different data scales and sparsity distributions, indicating its flexibility and scalability in diverse contexts. These findings suggest that SiMoI is an effective and stable approach for addressing sparsity and holds strong potential for implementation in user-based recommendation systems, particularly in real-world scenarios where data availability is frequently limited.

Keywords: Sparsity, Imputation, KNNI, MEAN, SiMoI, Recommendation System, Collaborative Filtering

1. Introduction

Recommendation systems are widely used in various fields to improve the quality of personal services, such as E-Learning, E-Government, E-Business, E-Library, E-Tourism, and E-Commerce [1], [2]. For organizations, information-based services play a vital role in supporting operations and enhancing the quality of user interactions. Therefore, the implementation of recommendation systems has become one of the main strategies, as they are capable of delivering information that is relevant to user interests and preferences, thereby optimizing service effectiveness [1]. The methods that have been widely used include content-based filtering [3], [4], demographic [5], [6], collaborative [7], [8], and hybrid [9], [10], [11].

Collaborative Filtering (CF) functions by analyzing rating data patterns to make predictions [12] and produce quality recommendations. However, this method faces significant problems with cold start [13], sparsity [14], and scalability [15], [16]. Sparsity is a condition of data scarcity caused by the fact that most users do not provide ratings for products. For example, the MovieLens 100K dataset has a sparsity level of 93.7%, meaning that only 6.3% (100,000 ratings) were provided by 943 users for 1,682 movies. The resulting matrix contains very few values, with most of its elements remaining empty. This condition leads to low similarity values and the inability of the system to generate accurate recommendations [17].

Sparsity causes incomplete information, making the prediction results less accurate or biased. Additionally, it causes difficulty for system to understand user preferences, affecting the quality of the recommendations, user satisfaction, and trust in the system. To address this problem, previous research had been carried out such as Ahmadian et al. [14] proposed RSTRC (Recommender System based on Temporal Reliability and Confidence), a recommendation system that leverages temporal reliability and rating confidence to dynamically update user preferences, thereby producing

***Corresponding author: Sri Lestari (srilestari@darmajaya.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i1.1015

 ISSN 2723-6471

more accurate and reliable recommendation. Satvika et al. [18] enhanced the performance of the K Nearest Neighbor (KNN) algorithm using the PFI (Permutation-based Feature Importance) method to forecast students' on-time graduation. The evaluation results showed that the integration of PFI could enhance the performance of the KNN model with better accuracy results. Therefore, this research aimed to propose a new SiMoI (Similarity, Mode, and Min Imputation) method to solve the sparsity problem. Imputation was performed on empty data (NaN) by incorporating the Euclidean Distance method to gauge similarity, mode, and mean functions. The results were in the form of values used for imputation, and the SiMoI method showed the potential to improve data quality to determine user preferences as well as produce more accurate recommendations.

2. Literature Review

2.1. K Nearest Neighbors Imputation (KNNI)

KNNI is a popular method used for solving missing value problems [19]. When there are many missing values, the data matrix will become sparse or sparsity. KNNI is a variant of the KNN method; therefore, it uses the basic concept of KNN by calculating the closest distance between data containing missing values in the test data and complete data in the training data. For data containing missing values, the process is not carried out in the distance calculation [20].

Generally, the imputation of missing values using KNNI consists of six stages. The first stage is to determine the value of "K", as the number of closest observations to be used. The second stage is to calculate the distance between observations with missing values on variable j and those without missing values on the corresponding variables, using Euclidean distance, as expressed in Equation 1.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2} \quad (1)$$

$d(x_i, x_j)$= distance from i to cluster center j, x_i= training data and x_j= test data, while for n= number of attributes and k = attribute, x_{ik}= data i on attribute k, and x_{jk}= data j on attribute k. The third stage is to find the K shortest observations by determining the smallest distance value, which is used for imputation. In the fourth stage, the weight for all k shortest observations is calculated, where the closest receives the highest score. The fifth stage is to calculate mean value of the k shortest observations without missing values using Equation 2.

$$X_j = \frac{1}{k}\sum_{k=1}^{k} V_{kj} \quad (2)$$

X_j= weighted mean, where V_{kj}= value in complete data on variables without missing values, based on k as the closest observation. The sixth stage is to carry out the imputation process on observations with missing values using mean data obtained in the fifth stage.

2.2. MEAN imputation

Mean imputation method is often used to solve missing values [21]. This method applies mean gained from the values in the same column to complete the missing values [22]. The process is carried out by avoiding data loss and maintaining the size of the entire dataset using Equation 3.

$$Mean = \frac{\sum_{i=1}^{N} x_i}{N} \quad (3)$$

N is the number of existing values (excluding missing values), while xi is the existing values.

3. Methodology

3.1.Similarity, Mode, and Min Imputation (SiMoI)

SiMoI is a new method offered to overcome the sparsity problem by incorporating similarity, mode, and min for imputation of missing values. The stages in the SiMoI method start with reading the dataset of a matrix (user x movie), followed by sorting the movie data based on the number of ratings in descending order and taking Top-M (100movies). This is continued by calculating the similarity between users through Euclidean Distance, with the provision that users as the initial state have the most ratings. After calculating Euclidean Distance, sorting is carried out in ascending order and takes Top-U (100users). The results of Top-U and Top-M formed a matrix, namely 100userx100movie. The

 ISSN 2723-6471

imputation process in the SiMoI method is governed by three conditions. First, if a column contains a single mode (i.e., the rating value with the highest frequency), that value is used to replace missing entries. Second, if multiple modes occur with equal frequency, the smallest rating among them is selected. Third, if a column consists entirely of missing or zero values, all entries are replaced with a default value of 1. The resulting imputed matrix serves as a reference for subsequent iterations until all missing values in the main matrix are filled. The use of mode and min functions is context-dependent: mode is applied as the primary strategy, while min is used only to resolve ties between multiple modes.

Pseudocode of the SiMoI method

1. Data loading and initialization
$$R \leftarrow LOAD(DataFrame)$$
2. Calculate the number of ratings per movie.
For each movie $m_j \in M$, count the number of users who provided a valid rating.
$$c_j = \sum_{u_i \in U} I(r_{u_i,m_j} \neq NaN)$$
Using I(.) as the indicator function.
Then, sort the movies based on c_j in descending order
$$M_{sorted} = sort(M, c_j, descending)$$
3. Select the top movies (in this case, 100)
Retrieve the 100 movies with the highest number of ratings.
$$M_{100} = \{m_1, m_2, \dots, m_{100}\} \subset M_{sorted}$$
4. Compute user similarity
For each pair of users (u_i, u_j):
   a. Select the reference user with the highest number of ratings:
$$u_{ref} = \arg max_{u \in \{u_i, u_j\}} \sum_m I(r_{u,m} \neq NaN)$$
   b. Compute the Euclidean distance based on the top 100 movies.
$$d(u_i, u_j) = \sqrt{\sum_{m \in M_{100}} (r_{u_i,m} - r_{u_j,m})^2}$$
   (Only non-NaN elements are included in the computation)
   c. Sort users based on $d(u_i, u_j)$ in ascending order.
   d. Select the 100 nearest users for each user."
$$U_{100}(u_i) = argmin^{100}_{u_j \in U, j \neq i} d(u_i, u_j)$$
5. Construct a 100×100 submatrix
Form the submatrix S consisting of the 100 nearest users and the top 100 movies:
$$S = [r_{u,m}]u \in U_{100}, m \in M_{100}$$
6. Missing value imputation
For each column m_j in the submatrix S:
   a. Extract valid values (non-NaN and non-zero):
$$V_{m_j} = \{r_{u,m_j} | r_{u,m_j} \neq NaN, r_{u,m_j} \neq 0\}$$
   b. Count the frequency of each rating value:
$$f_r = |\{u \in U_{100} | r_{u,m_j} = r\}|$$
   For each r ∈ {1,2,3,4,5}
   c. Determine the mode value and handle special conditions:
$$M_{mode} = \{r | f_r = \max(f_1, f_2, f_3, f_4, f_5)\}$$
   The replacement value is determined as follows:
$$\hat{r}_{m_j} = \begin{cases} 1, & if\ |V_{m_j}| = 0 \\ \min(M_{mode}), & if |M_{mode}| > 1 \\ M_{mode}[1], & if |M_{mode}| = 1 \end{cases}$$
   d. Replace missing values with the designated replacement value:

 ISSN 2723-6471

$$r^*_{u,m_j} = \begin{cases} r_{u,m_j}, & if r_{u,m_j} \neq NaN \\ \hat{r}_{m_j}, & if r_{u,m_j} = NaN \end{cases}$$

After all columns have been imputed, the resulting matrix is:

$$S^* = [r^*_{u,m_j}]U_{100}xM_{100}$$

7. Update the main matrix
   Insert the imputed values into the main matrix:

$$R[u,m] = \begin{cases} S * [u,m], & if\ (u,m) \in U_{100}xM_{100} \\ R[u,m], & others \end{cases}$$

8. Iterate until no missing values remain
   The process (Steps 2–7) is repeated until all elements in matrix R are filled.

$$while\ \exists(u,m): r_{u,m} = NaN \Longrightarrow Repeat\ steps\ 2\ to\ 7.$$

   The final result is a complete matrix

$$R^* = R$$

Output

$$R^* = [r^*_{u,m}]UxM$$

It is a fully imputed rating matrix with no missing values.

SiMoI (Similarity, Mode, and Min Imputation) offers a mathematically grounded approach for addressing extreme sparsity in user rating data, such as the MovieLens 100K dataset, which contains 97.3% missing entries. By utilizing a dense 100×100 submatrix as the model's core, SiMoI integrates four key principles: selecting users with similar rating patterns based on Euclidean distance, imputing missing values using the most frequent rating (mode) to reduce extreme bias, applying a three-layer condition to ensure that even fully empty columns can be imputed, and performing iterative, partial updates to maintain computational efficiency and minimize noise from infrequent user–item interactions. The result is a fully populated rating matrix that is both stable and representative, capable of propagating learned behavioral patterns across the entire dataset in a gradual and efficient manner.

In the pseudocode, 100 users and 100 movies were selected based on a top-N strategy, where entities with the highest number of ratings were prioritized. This selection aims to construct a dense and representative User-Movie matrix that serves as a reliable reference for the imputation process. By focusing on the most active users and movies, the matrix contains sufficient rating density to support robust evaluation and method development. The value of 100 is adjustable and may be modified according to analytical objectives or computational constraints.

This research used the MovieLens 100K dataset, accessed from https://grouplens.org/datasets/movielens/100k/. It contains 100,000 ratings from 943 users on 1,682 movies, with each user having rated at least 20 movies. Ratings are given on a scale from 1 to 5 [8], [23]. However, this dataset contained a sparsity of 93.7% as calculated using Equation 4.

$$Sparsity = 1 - \frac{Total\ existing\ ratings}{Total\ Possible\ ratings} \quad (4)$$

The total existing ratings is 100,000, while the total possible ratings are 943 users x 1682 movies

$$Sparsity = 1 - \frac{100.000}{943\ x\ 1682} = 1 - \ 0{,}063 = 0{,}937\ or\ 93{,}7\%$$

The occurrence of sparsity due to several missing data can be overcome with two strategies, namely ignoring missing value and imputation missing value. The first strategy is generally performed by eliminating cases that contain missing data causing a significant impact on the size of dataset. This strategy will be appropriate when applied to data with few missing values. Meanwhile, data with high sparsity values can be overcome using the second strategy, namely imputation missing value [24].

This research performed missing value imputation using a new method, namely SiMoI. The method is compared with K Nearest Neighbor Imputation (KNNI) and the MEAN function which is widely used to overcome sparsity problem, by observing the evaluation results using Root Mean Squared Error (RMSE).

Figure 1 the flowchart illustrates the stages of the research process, starting from data preparation to the analysis of evaluation results. The process begins with the Start stage, followed by Dataset Setup, which involves preparing the data used in the experiment. At this stage, the study utilizes the MovieLens dataset with a size of 943 users × 1682

 ISSN 2723-6471

movies, as well as several synthetic datasets with sizes of 800×800, 700×700, and 500×500. The synthetic datasets are used to test the performance of the proposed method under different data conditions, both in terms of dataset size and data density. The next stage is Sparsity Conditions, which functions to set the level of data sparsity or the percentage of missing values in the dataset. The sparsity conditions include the original data with 93.7% sparsity, high sparsity (80%), low sparsity (20%), and a condition with no sparsity information. This setup aims to determine how effectively each imputation method can perform under various levels of data sparsity. Subsequently, in the Imputation Methods stage, the missing values are filled using three different approaches: SiMoI, KNNI (K-Nearest Neighbor Imputation), and MEAN. The SiMoI method is a newly proposed approach in this study, while KNNI and MEAN serve as comparative baselines. This stage aims to evaluate the ability of each method to accurately estimate the missing rating values. After the imputation process, the Evaluation Metric stage is conducted to assess the imputation results using the RMSE (Root Mean Square Error) metric. The RMSE value measures the magnitude of error between the predicted and actual values, where a smaller RMSE indicates better imputation performance. The final stage is Result Comparison, which involves analyzing and comparing the results. In this stage, the RMSE values of each method are analyzed to determine which method provides the most accurate results in handling data sparsity. The process concludes with the End stage, marking the completion of the entire experimental procedure.

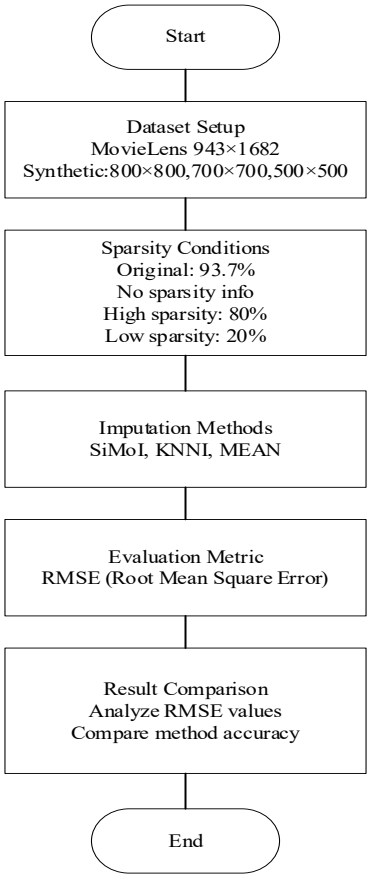

Figure 1. Experimental Workflow for Matrix Imputation and RMSE Evaluation

3.2.Root Mean Squared Error (RMSE)

Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as "3.5-inch disk drive."

RMSE is used to evaluate the effectiveness of imputation model in predicting missing values. It is a statistical measure that calculates the magnitude of deviation between predicted and actual values, as expressed in Equation 5 [23].

$$RMSE = \sqrt{\frac{1}{|E|}\sum_{i=1}^{|E|}\left|P_{ut,i} - r_{ui}\right|} \quad (5)$$

 ISSN 2723-6471

The original rating given by users for an item is denoted by r_{ui}. The projected rating of user for the same item is denoted by $P_{ut,i}$, while the size of the test set is represented by $|E|$.

4. Results and Discussion

This research was conducted with three scenarios (figure 1), namely the first scenario using a full-sized dataset of 943 users × 1682 movies with a sparsity condition of 93.7%. The second scenario uses data with different sizes and without considering the spatial conditions. Continued with the third scenario using data with different sizes of 800 users × 800 movies, 700 users × 700 movies, and 500 users × 500 movies. The datasets were taken randomly and conditioned at a sparsity level of 80% and 20%. Furthermore, eaFch dataset was imputed using three methods, namely SiMoI (Similarity, Mode, and Mean Imputation), KNNI (K-Nearest Neighbor Imputation), and the column average method (MEAN). Evaluation was carried out using the RMSE (Root Mean Square Error) metric. The RMSE calculation follows the formulation presented in Equation 5, which quantifies the average squared difference between predicted and actual values. The evaluation results for the first scenario are shown in figure 2, while for the second and third scenarios, the evaluation results can be seen in figure 3, figure 4, figure 5.

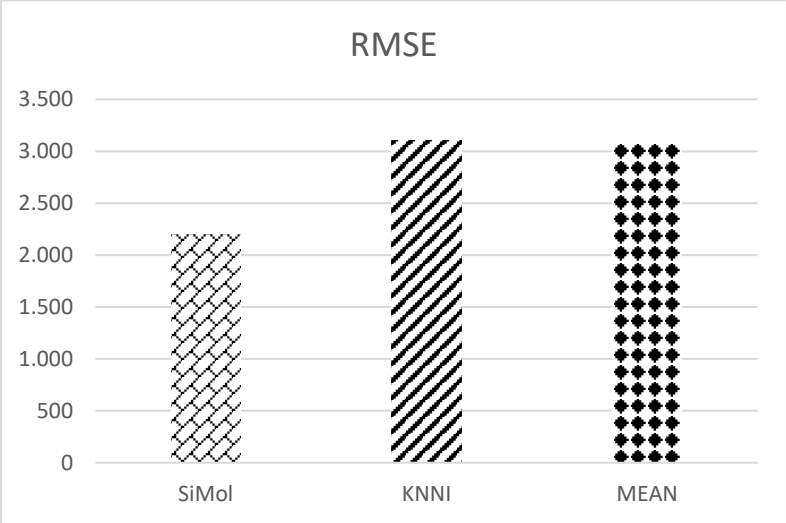

Figure 2. RMSE Evaluation Results on Imputation Method with Dataset 943x1682 with Sparsity 93.7%

Initial experiments were conducted on a full dataset of 943 users × 1682 movies and a sparsity level of 93.7%. Based on the RMSE evaluation results in figure 2, the SiMoI method showed the best performance with an RMSE value of 2.200, lower than KNNI (3.106) and MEAN (3.065). This indicates that the SiMoI method is more effective in handling very sparse data conditions (extreme sparsity) in large dataset sizes.

Figure 3 shows the results of the RMSE evaluation of the three methods, SiMoI, KNNI, and MEAN, on data conditions of various sizes, namely 800 users × 800 films, 700 users × 700 films, and 500 users × 500 films, in the SiMoI method, with RMSE values respectively: 2.984, 3.045, and 3.040. The KNNI method with RMSE values , respectively, namely 3.321, 3.349, and 3.252. The MEAN method with RMSE values , respectively, namely 3.501, 3.528, and 3.383. The average RMSE value for the SiMoI method is 3.023, which is lower than that of KNNI (3.307) and MEAN (3.471). The average RMSE difference between SiMoI and KNNI is 0.284, reflecting an accuracy improvement of approximately 8.6%, while the difference between SiMoI and MEAN is 0.448, indicating an improvement of around 12.9%. The RMSE values for SiMoI remain relatively stable across different data sizes, with a maximum variation of only 0.061, suggesting consistent performance despite changes in matrix scale.

 ISSN 2723-6471

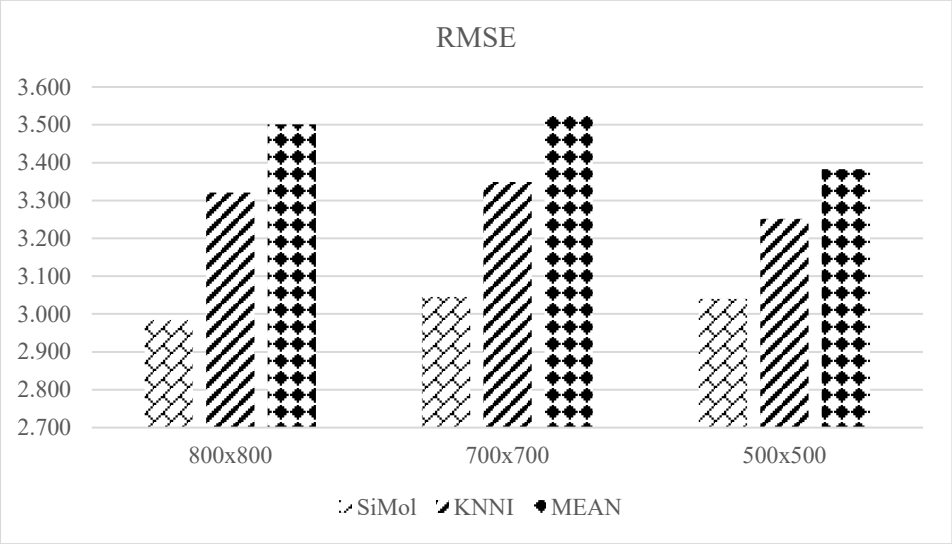

Figure 3. RMSE Evaluation Results on Imputation Methods with Different Matrix Sizes

In contrast, the KNNI and MEAN methods exhibit greater RMSE fluctuations. Specifically, MEAN shows a maximum variation of 0.145, indicating reduced stability on smaller data sizes. Overall, the SiMoI method demonstrates quantitatively superior and more stable performance compared to KNNI and MEAN under varying data conditions. The consistent reduction in RMSE values suggests that SiMoI is more adaptive to changes in data scale, making it a more reliable method for addressing high-sparsity scenarios.

Figure 4 presents the evaluation results on datasets of various sizes, each with a sparsity level of 80%. The data conditions include matrix sizes of 800 users × 800 movies, 700 users × 700 movies, and 500 users × 500 movies. The SiMoI method shows RMSE values of 2.819, 2.937, and 2.875, respectively. The KNNI method has RMSE values of 2.943, 2.974, and 2.863, respectively, while the MEAN method has RMSE values of 3.211, 3.244, and 3.041. SiMoI has an average RMSE of 2.877, which is lower than KNNI (2.927) and MEAN (3.165). The decrease in RMSE for SiMoI indicates consistent performance across varying data sizes. Although KNNI shows a slightly lower RMSE at the 500×500 matrix size (2.863 vs. 2.875), SiMoI remains superior in terms of overall average and stability. The average RMSE difference between SiMoI and MEAN reaches 0.288, reflecting an accuracy improvement of approximately ±9.1% compared to the MEAN method. SiMoI demonstrates quantitatively better and more consistent performance under high-sparsity conditions.

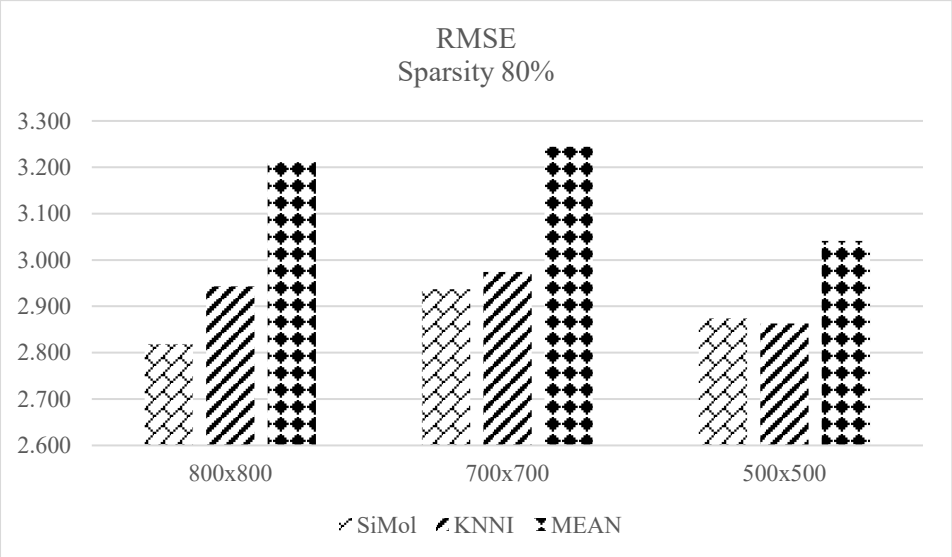

Figure 4. RMSE Evaluation Results on Imputation Method with Different Matrix Sizes and 80% Sparsity

 ISSN 2723-6471

Figure 5 displays the results evaluation of the SiMoI, KNNI, and MEAN methods was conducted on datasets with low sparsity (20%) and varying matrix sizes: 800×800, 700×700, and 500×500. The SiMoI method has RMSE values of 1.467, 1.468, and 1.369, respectively. The RMSE values for the KNNI method are 1.421, 1.426, and 1.294, respectively. Meanwhile, the MEAN method has RMSE values of 1.489, 1.493, and 1.374, respectively. All methods showed a decrease in RMSE values, indicating improved accuracy under dense data conditions. The average RMSE values for each method were 1.380 for KNNI, 1.435 for SiMoI, and 1.452 for MEAN.

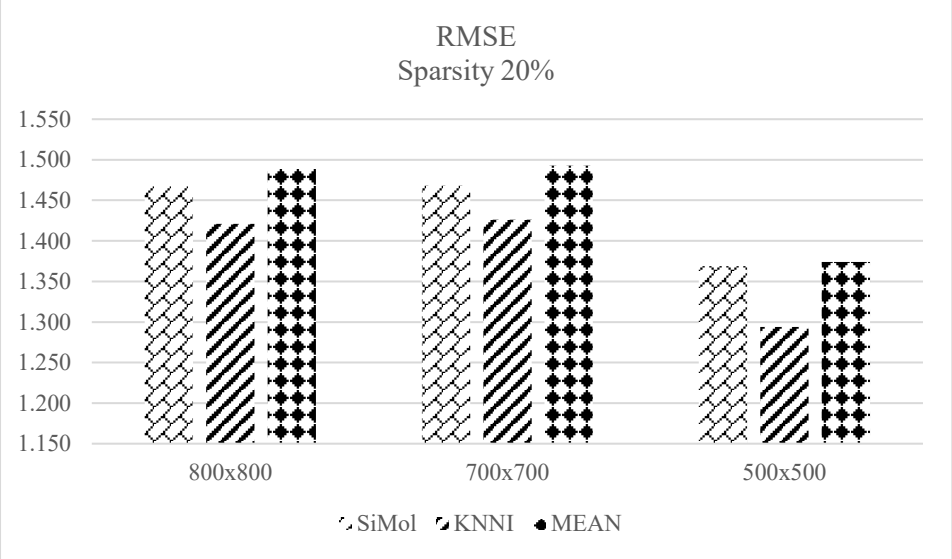

Figure 5. RMSE Evaluation Results on Imputation Method with Different Matrix Sizes and 20% Sparsity

The average RMSE difference between KNNI and SiMoI was 0.055, reflecting an accuracy improvement of approximately 3.8%, while the difference between KNNI and MEAN was 0.072, indicating an improvement of around 5.0%. SiMoI remained competitive, with a 0.017 difference compared to MEAN, or about 1.2% higher in accuracy.

In terms of stability, SiMoI exhibited smaller RMSE fluctuations (maximum difference of 0.099) compared to MEAN (0.119), suggesting more consistent performance across different matrix sizes. Overall, KNNI proved to be the most optimal method under dense data conditions, while SiMoI maintained stable and adaptive performance, particularly in higher sparsity scenarios.

KNNI's superior performance under low sparsity is attributed to the availability of more complete data, which enables more accurate neighbor identification. In contrast, SiMoI, designed to handle high sparsity, was less optimal under dense conditions. This implies that further development of SiMoI should consider adaptive strategies, such as integrating neighbor-based techniques or adjusting component weights, to remain competitive across varying levels of sparsity.

Based on the quantitative evaluation results presented in figure 2, figure 3, figure 4, figure 5, the performance of imputation methods is strongly influenced by the level of sparsity and the size of the dataset. Under extreme sparsity conditions (93.7%), the SiMoI method demonstrated the best performance, achieving the lowest RMSE value (2.200), outperforming KNNI (3.106) and MEAN (3.065). This indicates that SiMoI is more effective in handling highly sparse data within large-scale datasets.

At moderate sparsity (80%), SiMoI maintained consistent performance, with stable and competitive RMSE values compared to KNNI and MEAN. In contrast, under low sparsity conditions (20%), KNNI achieved the lowest average RMSE (1.380), followed by SiMoI (1.435) and MEAN (1.452). The RMSE difference between KNNI and SiMoI (0.055) reflects an accuracy improvement of approximately 3.8%, while the difference between KNNI and MEAN (0.072) indicates an improvement of around 5.0%. Although SiMoI was not the top performer in dense data conditions, it remained competitive, with a 0.017 advantage over MEAN, or about 1.2% higher in accuracy.

 ISSN 2723-6471

In terms of stability, SiMoI exhibited smaller RMSE fluctuations (maximum difference of 0.099) compared to MEAN (0.119), indicating consistent performance across varying matrix sizes. KNNI's superior performance under low sparsity is attributed to the availability of more complete data, which enables more accurate neighbor identification. Conversely, SiMoI, designed to address high sparsity, was less optimal in dense data scenarios.

These findings suggest that the selection of imputation methods should be aligned with the characteristics of sparsity and dataset scale. To enhance SiMoI's flexibility, future development may consider adaptive strategies, such as integrating neighbor-based techniques or adjusting component weights, to ensure competitiveness across diverse data conditions.

5. Discussion

The primary objective of this study is to address the problem of sparsity, a condition in which the majority of values within a data matrix are missing—an issue commonly encountered in collaborative recommendation systems. High levels of sparsity can hinder the prediction process due to the limited information available for identifying patterns or similarities among users. Therefore, this research is specifically designed to evaluate the effectiveness of imputation methods in filling missing values across varying levels of sparsity and dataset scales.

Experimental results demonstrate that the SiMoI method consistently delivers the best performance under high-sparsity conditions, including extreme scenarios where the proportion of missing values reaches 93.7%. In such cases, other methods such as KNNI and MEAN experience significant drops in accuracy, whereas SiMoI remains capable of producing predictions with low and stable RMSE values. This indicates that SiMoI successfully addresses the core challenge of the study, developing an imputation approach that remains effective even when available information is severely limited.

The strength of SiMoI lies in its mechanism, which not only relies on user similarity but also incorporates mode and minimum values as adaptive imputation strategies suited to sparsely populated data structures. This approach enables SiMoI to function optimally even when neighbor-based methods like KNNI lose effectiveness due to insufficient comparative data.

In conclusion, SiMoI proves to be superior not only in terms of quantitative accuracy but also conceptually, by fulfilling the central aim of the research: providing a reliable and stable solution to the sparsity problem in user-based recommendation systems. These findings reinforce SiMoI's position as a relevant and promising method for real-world applications that frequently face data limitations.

6. Conclusion

One of the most significant challenges in developing collaborative recommendation systems is addressing the issue of sparsity, a condition in which the majority of values within the data matrix are missing. Extreme sparsity can severely hinder the estimation process and reduce prediction accuracy, particularly when the available information is insufficient to identify meaningful patterns or user similarities. This problem serves as the central focus of the present study, which aims to evaluate and develop an imputation approach capable of performing effectively across varying levels of data sparsity and dataset scales.

Through a series of systematic experiments, the SiMoI method demonstrated superior and consistent performance, especially under high-sparsity conditions. SiMoI was able to produce predictions with low and stable error rates, even in extreme scenarios where the proportion of missing data reached 93.7%. This advantage is not only quantitative but also conceptual, as SiMoI is designed with an adaptive mechanism that combines user similarity with imputation strategies based on mode and minimum values. This structure enables SiMoI to remain effective even when the available information is severely limited.

In summary, the SiMoI method successfully addresses the primary objective of this research by providing a reliable and stable solution to the sparsity problem in user-based recommendation systems. These findings reinforce SiMoI's position as a relevant and promising method for real-world applications, particularly in environments that frequently encounter data limitations.

7. Declarations

7.1. Author Contributions

Author Contributions: Conceptualization: H.K., S.L., and R.B.S.; Methodology: S.L.; Software: H.K.; Validation: H.K., S.L., and R.B.S.; Formal Analysis: H.K., S.L., and R.B.S.; Investigation: H.K.; Resources: S.L.; Data Curation: S.L.; Writing—Original Draft Preparation: H.K., S.L., and R.B.S.; Writing—Review and Editing: S.L., H.K., and R.B.S.; Visualization: H.K.; All authors have read and agreed to the published version of the manuscript.

7.2. Data Availability Statement

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

7.3. Funding

The authors are grateful to the Indonesian Ministry of Education, Culture, Research, and Technology, and Higher Education (Kemendikbud-Ristek DIKTI) for funding this research under Decree Number: 0459/E5/PG.02.00/2024. This financial support is essential in streamlining various stages of the research from data collection to final analysis and documentation. Additionally, the authors are grateful to the Darmajaya Institute of Informatics and Business, particularly the Faculty of Computer Science, for the continuous support and encouragement in this research.

7.4. Institutional Review Board Statement

Not applicable.

7.5. Informed Consent Statement

Not applicable.

7.6. Declaration of Competing Interest

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

References

[1] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis Support Syst*, vol. 74, pp. 12–32, Jun. 2015, doi: 10.1016/j.dss.2015.03.008.

[2] B. Patel, P. Desai, and U. Panchal, "Methods of Recommender System: A Review," in *International Conference on Innovations in information Embedded and Communication Systems (ICIIECS)*, 2017, pp. 1–4. doi: https://doi.org/10.1109/ICIIECS.2017.8275856.

[3] L. Yao, Q. Z. Sheng, A. H. H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based web service recommendation," *IEEE Trans Serv Comput*, vol. 8, no. 3, pp. 453–466, May 2015, doi: 10.1109/TSC.2014.2355842.

[4] Y. Xu and J. Yin, "Collaborative recommendation with user generated content," *Eng Appl Artif Intell*, vol. 45, pp. 281–294, Oct. 2015, doi: 10.1016/j.engappai.2015.07.012.

[5] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System," *Int J Comput Appl*, vol. 110, no. 4, pp. 975–8887, 2015, doi: https://doi.org/10.5120/19308-0760.

[6] L. Jia Yin, N. Zuraidin Mohd Safar, and F. Sains Komputer dan Teknologi Maklumat, "Research on the Demographic Filtering Machine Learning in Movie Recommendation System," *Applied Information Technology And Computer Science*, vol. 4, no. 1, pp. 290–307, 2023, doi: 10.30880/aitcs.2023.04.01.018.

[7] 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, doi: 10.1016/j.mlwa.2023.100495.

[8] G. Behera and N. Nain, "Collaborative Filtering with Temporal Features for Movie Recommendation System," in *Procedia Computer Science,* Elsevier B.V., 2022, pp. 1366–1373. doi: 10.1016/j.procs.2023.01.115.

[9] Z. Ren, B. Peng, T. K. Schleyer, and X. Ning, "Hybrid collaborative filtering methods for recommending search terms to clinicians," *J Biomed Inform*, vol. 113, Jan. 2021, doi: 10.1016/j.jbi.2020.103635.

[10] K. Kobyshev, N. Voinov, and I. Nikiforov, "Hybrid image recommendation algorithm combining content and collaborative filtering approaches," in *Procedia Computer Science*, Elsevier B.V., 2021, pp. 200–209. doi: 10.1016/j.procs.2021.10.020.

[11] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network," *Simul Model Pract Theory*, vol. 113, Dec. 2021, doi: 10.1016/j.simpat.2021.102375.

[12] F. E. Alsaadi, Z. Wang, N. S. Alharbi, Y. Liu, and N. D. Alotaibi, "A new framework for collaborative filtering with p-moment-based similarity measure: Algorithm, optimization and application," *Knowl Based Syst*, vol. 248, Jul. 2022, doi: 10.1016/j.knosys.2022.108874.

[13] K. Vahidy Rodpysh, S. J. Mirabedini, and T. Banirostam, "Resolving cold start and sparse data challenge in recommender systems using multi-level singular value decomposition," *Computers and Electrical Engineering*, vol. 94, Sep. 2021, doi: 10.1016/j.compeleceng.2021.107361.

[14] S. Ahmadian, N. Joorabloo, M. Jalili, and M. Ahmadian, "Alleviating data sparsity problem in time-aware recommender systems using a reliable rating profile enrichment approach," *Expert Syst Appl*, vol. 187, Jan. 2022, doi: 10.1016/j.eswa.2021.115849.

[15] Mohamad Fahmi Hafidz and Sri Lestari, "Solution to Scalability and Sparsity Problems in Collaborative Filtering using K-Means Clustering and Weight Point Rank (WP-Rank)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi*), vol. 7, no. 4, pp. 743–750, Aug. 2023, doi: 10.29207/resti.v7i4.4543.

[16] M. Singh, "Scalability and sparsity issues in recommender datasets: a survey," *Knowl Inf Syst*, vol. 62, no. 1, pp. 1–43, Jan. 2020, doi: 10.1007/s10115-018-1254-2.

[17] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," Nov. 01, 2015, *Elsevier B.V*. doi: 10.1016/j.eij.2015.06.005.

[18] G. A. J. Satvika, I. N. Sukajaya, and I. G. A. Gunadi, "Improving k-nearest neighbor performance using permutation feature importance to predict student success in study," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 35, no. 3, pp. 1835–1844, Sep. 2024, doi: 10.11591/ijeecs.v35.i3.pp1835-1844.

[19] A. Fadlil, Herman, and D. Praseptian M, "K Nearest Neighbor Imputation Performance on Missing Value Data Graduate User Satisfaction," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi),* vol. 6, no. 4, pp. 570–576, Aug. 2022, doi: 10.29207/resti.v6i4.4173.

[20] A. Fadlil, Herman, and M. Dikky Praseptian, "Single Imputation Using Statistics-Based and K Nearest Neighbor Methods for Numerical Datasets," *Ingenierie des Systemes d'Information*, vol. 28, no. 2, pp. 451–459, Apr. 2023, doi: 10.18280/isi.280221.

[21] H. Shi, P. Wang, X. Yang, and H. Yu, "An Improved Mean Imputation Clustering Algorithm for Incomplete Data," *Neural Process Lett*, vol. 54, no. 5, pp. 3537–3550, Oct. 2022, doi: 10.1007/s11063-020-10298-5.

[22] M. Wolbers, A. Noci, P. Delmar, C. Gower-Page, S. Yiu, and J. W. Bartlett, "Standard and reference-based conditional mean imputation," *Pharm Stat*, vol. 21, no. 6, pp. 1246–1257, Nov. 2022, doi: 10.1002/pst.2234.

[23] G. Jain, T. Mahara, A. Kumar, and S. C. Sharma, "Time-Aware Based Recommendation System using Gower's Coefficients: Enhancing Personalized Recommendation," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 3379–3388. doi: 10.1016/j.procs.2024.04.318.

[24] A. Jadhav, D. Pramod, and K. Ramanathan, "Comparison of Performance of Data Imputation Methods for Numeric Dataset," *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913–933, Aug. 2019, doi: 10.1080/08839514.2019.1637138.