

Optimization of hybrid-based Collaborative Filtering using Matrix Factorization, Feedforward Neural Network, and XGBoost

Filimantaptius Gulo*, Ronsen Purba, Muhammad Fermi Pasha

Abstract— Collaborative filtering recommendation systems are widely used in digital applications; however, they still face challenges such as cold-start and first-rater problems, as well as limited accuracy due to their inability to capture complex user-item relationships. This study proposes a hybrid recommendation model that integrates Matrix Factorization, MLP-based Feedforward Neural Network (MLP) and Extreme Gradient Boosting (XGBoost). Experiments were conducted on two real-world datasets, namely MovieLens (movies) and PT XYZ (hotels), to validate the effectiveness of the proposed approach. The results indicate that the hybrid model consistently outperforms baseline methods such as SGD-based Matrix factorization, Matrix factorization +MLP, and user/item-based Collaborative filtering. Specifically, the integration of nonlinear learning through MLP and feature enhancement via XGBoost significantly improves prediction accuracy while mitigating cold-start and first-rater issues. These findings suggest that hybrid machine learning-based approaches can advance the development of more adaptive, accurate, and personalized recommendation systems.

Index Terms—Collaborative Filtering, Feedforward Neural Networks, Matrix Factorization, Recommender Systems, XGBoost.

I. INTRODUCTION

RECOMMENDATION systems have become an important research topic due to their ability to assist users in discovering relevant items from large-scale data. These systems aim to automatically suggest products or services to users based on their preferences and behaviors, thereby reducing the need to manually explore large volumes of data [1]. Recommendations have become an integral part of daily life and are widely used in various applications [2]. Collaborative filtering is widely recognized as one of the most well-known recommendation algorithms [1][3], primarily designed to perform predictions and provide personalized recommendations [4]. The cold-start problem arises when a user has no prior rating history, making it difficult to generate recommendations for such users. Another major challenge is the first-rater problem, where a new item cannot be recommended until it receives initial ratings from users [5]. Therefore, addressing the cold-start issue remains an important area for improvement [6].

Collaborative filtering is one of the most widely used recommendation algorithms [7]. Much research has been done in the field of collaborative filtering for recommendations [8]. Therefore, one of the challenges in

research is the use of machine learning algorithms to improve the accuracy of recommendations [9]. A hybrid-based approach in Collaborative Filtering has been used to personalize the model [8]. Prior studies show that Matrix Factorization effectively addresses sparsity by learning latent user-item representations, yet its linear interaction assumption limits modeling complex preferences [1], [2], [10], [11]. Among these models, the MLP method has been widely applied in recommendation.

Hybrid recommendation systems that integrate Matrix Factorization with other learning algorithms have been widely explored to improve recommendation performance [2]. In particular, tree-based ensemble methods such as XGBoost have been applied to collaborative filtering to enhance prediction accuracy through feature-based residual learning, enabling more personalized recommendations [3], [6], [10], [20], [21]. The promising performance of XGBoost has motivated further research, including the development of the hybrid-based architecture proposed in this study [13].

Unlike prior Matrix Factorization with XGBoost or Matrix Factorization with MLP approaches, this study introduces a unified three-stage Matrix Factorization $\rightarrow$ MLP $\rightarrow$ XGBoost framework enhanced with handcrafted global, user, and item statistical features. This combination has not been previously explored and is validated on two highly sparse real-world datasets, demonstrating consistent accuracy improvements.

II. RELATED WORK

This section discusses related work encompassing Matrix Factorization, MLP, and XGBoost.

A. Matrix Factorization

Matrix factorization is widely used to learn latent user-item representations in collaborative filtering; however, its linear interaction assumption limits its ability to capture complex user preferences and cold-start scenarios [1][2]. Most conventional matrix factorization methods are based on matrix decomposition, where users and items are represented by latent factors derived from the user-item rating matrix [1][4].

This model is employed in recommendation tasks, such as matrix decomposition and clustering models [14]. Matrix factorization technique [10] works by decomposing the user-item rating matrix into the product of two lower-dimensional rectangular matrices, while assigning different regularization weights to the latent factors to improve the prediction accuracy [15].

Filimantaptius Gulo, Ronsen Purba and Muhammad Fermi Pasha are with the Information Technology Program, Universitas Mikroskil, Medan 20238, Indonesia. *Correspondence: filimantapgullo@gmail.com

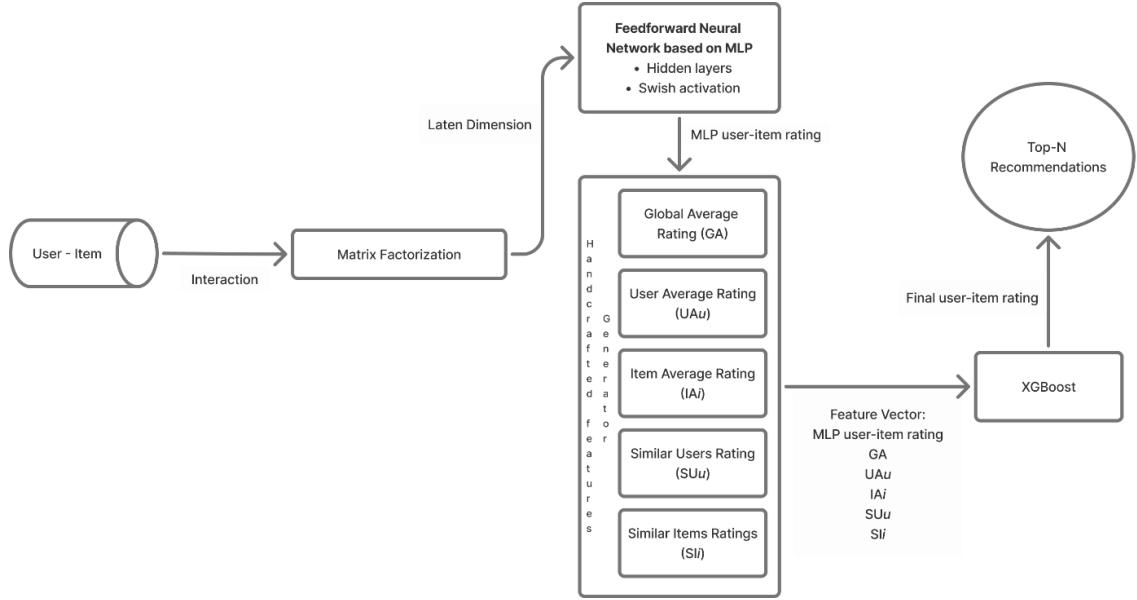


Fig. 1. Block diagram of the hybrid collaborative filtering model using matrix factorization, feedforward neural networks, and xgboost.

B. Feedforward Neural Network

The MLP-based Feedforward Neural Network models have been adopted in collaborative filtering to capture nonlinear user-item interactions beyond matrix factorization limitations, enabling more expressive preference modeling through multi-layer nonlinear transformations [9][12][16][17].

The advantages of MLP in representing features efficiently and flexibly have encouraged researchers to develop various deep learning-based collaborative filtering models that utilize this architecture [18]. By leveraging the ability of MLP to model nonlinearity, this algorithm can compute more accurate interactions between users and items, thereby improving the quality of predictions in recommendation systems [19].

C. XGBoost

XGBoost is employed in this study as a feature-based refinement model that learns residual errors from Matrix Factorization – MLP predictions through gradient-boosted decision trees, enabling accurate rating estimation from structured handcrafted features [5][20][21].

As a classifier, XGBoost has been widely adopted in various machine learning applications due to its efficiency and superior performance [22]. In this study, XGBoost is employed to predict the final assessment scores assigned by users to an item. The algorithm applies an integrated learning approach by leveraging multi-classification methods and constructing hundreds of decision trees to accelerate the training process, thereby producing more accurate and stable predictions [10][3].

Existing studies combine only two components (e.g., Matrix Factorization with XGBoost or Matrix Factorization with MLP) and do not integrate all three stages with handcrafted features. No prior work evaluates such a hybrid model across multiple sparse domains, leaving a gap that this study addresses.

III. THE PROPOSED RECOMMENDATION MODEL

The user-item rating matrix is first decomposed using

Matrix Factorization to learn latent representations of users and items. These latent vectors are concatenated and forwarded to the MLP-based Feedforward Neural Networks to capture nonlinear user-item interactions. Subsequently, the MLP output is combined with handcrafted features, including the global average rating, user and item average ratings, as well as similar-user and similar-item ratings, to form a unified feature vector. This feature vector is then processed by the XGBoost model to produce the final rating prediction. The explicit data flow and feature integration shown in Figure 1 clarify the dependency between the learning stages.

Unlike existing hybrid collaborative filtering frameworks that typically combine only two learning components, the proposed model sequentially integrates Matrix Factorization, MLP, and XGBoost in a unified three-stage pipeline. Matrix Factorization learns latent user-item representations, MLP captures nonlinear interaction patterns, and XGBoost refines predictions through feature-based learning, resulting in a more comprehensive hybrid architecture.

The following describes the technical specifications of the proposed hybrid recommendation model based on the view in Fig. 1.

A. Matrix factorization model

Matrix factorization is a technique that decomposes the original matrix into two low-rank matrices, which can be interpreted as the latent factors of users and items, as follows [10].

$$R_{in} \approx U V^T \quad (1)$$

where $U \in R_{in}^{M \times F}$ and $V \in R_{in}^{F \times N}$ are the user-item features of factor F , such that $F \ll (M, N)$.

B. Feedforward Neural Network

The MLP-based Feedforward Neural Network consists of an input layer L_{in} which is given input in the form of latent features resulting from Matrix Factorization, several hidden layers designed to learn nonlinear interactions L_k and an output layer L_{out} [23].

The L_{in} layer serves as the input to the neural network, consisting of latent features derived from matrix

factorization and is represented in vector form as follows:

$$y_0 = P_u \otimes Q_i \quad (2)$$

where $\otimes$ is the dot product operator, P_u represents the user latent vector and Q_i represents the item latent vector. The L_k layer consists of several hidden layers within the neural network. It begins with the first hidden layer L_1 which receives input from y_0 and output vector y_1 of this hidden layer is computed as follows:

$$y_1 = \rho_1(w_1 y_0 + a_1) \quad (3)$$

where w_1 is the set of weights contained in a matrix between the input layer and the first hidden layer L_1 , a_1 are biases in L_1 layer, and ρ_1 is the activation function that enables the nonlinear property of the neural network. In this study, we utilize the Swish activation function, which was proposed by the Google Brain. Swish activation function is defined as follows:

$$\text{Swish}(x) = x \cdot \text{sigmoid}(x) = \frac{x}{1 + e^{-x}} \quad (4)$$

where x is the input variable, the sigmoid function is defined as $\text{sigmoid}(x) = \frac{x}{1 + e^{-x}}$, and (x) is the result of Swish activation. In general, the outputs of hidden layers L_k are denoted by y_k and computed as follows:

$$y_k = \rho_k(w_k(\rho_{k-1}(\dots \rho_2(w_2 y_1 + a_2) \dots)) + a_k) \quad (5)$$

where ρ_k , w_k , and a_k denote the activation function, weight matrix, and bias vector of the neurons in the hidden layer L_k , respectively. In the output layer L_{out} of the MLP, the output prediction y_{out} is computed in a straightforward manner as follows:

$$y_{\text{out}} = \rho_{\text{out}}(w_{\text{out}} y_k + a_{\text{out}}) \quad (6)$$

where ρ_{out} denotes the output layer activation function, which adopts a Swish like nonlinearity similar to the one employed in the hidden layers.

C. Handcrafted Features

Handcrafted features refer to manually engineered knowledge incorporated into the model parameters [21]. These features are then selectively combined using a filtering mechanism to enhance the adaptability of the method [24]. By generating handcrafted features such as global averages as defined in (7), user-item averages ratings presented in (8) and (9), respectively. [10]. Additionally, features such as user similarity and similar items [7][10] as defined in (10) and (12), respectively. These features are subsequently utilized as additional inputs in the XGBoost model [10]. Therefore, the handcrafted features are written as follows:

$$\text{Global}_{avg} = \frac{\sum r_{ji}^a}{\sum N_{nz}} \quad (7)$$

where N_{nz} denotes the number of non-zero entries in the rating matrix.

$$\text{User}_{Avg_i} = \frac{\sum r_{ij}}{N_i} \quad (8)$$

$$\text{Item}_{Avg_j} = \frac{\sum r_{ij}}{M_j} \quad (9)$$

$$\text{cosine}_{sim}(u_1, u_2) = \frac{\sum u_1 \cdot u_2}{\|u_1\| \|u_2\|} \quad (10)$$

$$SU_u(i) = \frac{\sum_{v \in \mathcal{N}_u} \text{cosine}_{sim}(u, v) \hat{r}_{vi}^{MLP}}{\sum_{v \in \mathcal{N}_u} \text{cosine}_{sim}(u, v)} \quad (11)$$

where $\mathcal{N}_u$ denotes the set of top- k most similar users to user u , identified using cosine similarity over latent user vectors obtained from Matrix Factorization. The term $\hat{r}_{vi}^{MLP}$ represents the predicted rating generated by the MLP model for user v on item i . This formulation produces a similarity-weighted average rating, which serves as an informative signal for users with sparse interaction histories.

$$\text{cosine}_{sim}(v_1, v_2) = \frac{\sum v_1 \cdot v_2}{\|v_1\| \|v_2\|} \quad (12)$$

$$SI_i(u) = \frac{\sum_{j \in \mathcal{N}_i} \text{cosine}_{sim}(i, j) \hat{r}_{uj}^{MLP}}{\sum_{j \in \mathcal{N}_i} |\text{cosine}_{sim}(i, j)|} \quad (13)$$

where $\mathcal{N}_i$ denotes the set of top- k most similar items to item i , computed based on cosine similarity between item latent vectors. The term $\hat{r}_{uj}^{MLP}$ is the MLP-predicted rating of user u on item j . This feature captures neighborhood-based item relevance and provides meaningful estimates for first-rater items.

Subsequently, the average ratings for users and items are computed using the output of the MLP model. The user average represents the mean predicted rating for a specific user across all items, whereas the item average corresponds to the mean predicted rating for a particular item aggregated across all users.

To clarify the construction of handcrafted features reported in Tables III and IV, this subsection provides a detailed explanation of how each feature is computed and normalized prior to being used as input for the XGBoost model.

All handcrafted features, including GA, UA, IA, SU, and SI, are normalized using min-max normalization to ensure consistent numerical scales before being processed by the XGBoost model. The normalization is defined as:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (14)$$

where $x_{\min}$ and $x_{\max}$ are computed from the training set only. This normalization step prevents feature dominance and improves the stability of gradient-boosted tree learning.

D. XGBoost

XGBoost is utilized to predict the final rating score given by users to a particular item. This algorithm is an ensemble learning technique that leverages multi-class classification and builds hundreds of decision trees to accelerate the training process, thereby producing more accurate predictions [10].

XGBoost is an ensemble technique consisting of multiple base learners. Let $F = \{F_1, F_2, F_3, \dots, F_m\}$ be a set of base learners. The primary objective is to select a function that minimizes the overall loss for a given sample $X = \{X_1, X_2, X_3, \dots, X_n\}$. The objective function is defined as follows in (15).

$$\theta = \sum J(y_i, \hat{y}_i^{t-1} + F_t(x_i)) + \omega(F_t) \quad (15)$$

where J is the loss function, y_i is the true label, $\hat{y}_i^{t-1}$ is the previous prediction, $F_t(x_i)$ is the prediction from the new tree to be added, and $\omega(F_t)$ is the regularization term that controls the complexity of the tree (e.g., number of leaves, magnitude of leaf weights) to help prevent overfitting [21].

The objective function can be expanded using the Taylor series. Therefore, (16) can be rewritten as follows.

$$\theta^t = \sum_{j=1} \theta(r_j, \hat{r}_j^{t-1}) + \frac{\partial \theta(r_j, \hat{r}_j^{t-1})}{\partial \hat{r}_j^{t-1}} F_t(x_j) + \frac{\partial^2 \theta(r_j, \hat{r}_j^{t-1})}{\partial \hat{r}_j^{t-1}} F_t(x_j)^2 \quad (16)$$

In addition, (16) can be simplified as follows.

$$\theta^t = \sum C + g_j F_t(x_j) + h_j F_t(x_j)^2 + \omega(F_t) \quad (17)$$

where $C = \theta(r_j, \hat{r}_j^{t-1})$ is a constant independent of any function, g_j is the first derivative of the loss function at the previous stage and h_j is the second-order derivative of the loss function with respect to the prediction at the previous iteration. We now select F_t such that θ_t is the minimum. The constant term is then removed, as it does not depend on any function (17). Consequently, the loss function can be rewritten as follows.

$$\theta^t = \sum g_j F_t(x_j) + h_j F_t(x_j)^2 + \omega(F_t) \quad (18)$$

Let F_t denote the set of K leaf nodes, I_j represent the set of instances assigned to node j and W_j be the predicted score for that node. Accordingly, for any instance i belonging to I_j , $F_t = W_j$ and regularization term used to measure and control the complexity of the model. It can be written in its substituting as follows.

$$\Omega(F_t) = \gamma K + \frac{1}{2} \lambda \sum_j w_j^2 \quad (19)$$

Substituting F_t into (17) and setting for each leaf j , $\frac{\partial \theta}{\partial w_j} = 0$, yields the optimal weight is defined as follows.

$$w_j^* = \frac{-\sum g_j}{\sum h_j + \lambda} \quad (20)$$

Finally, by substituting the optimal weight back into the objective, the minimum loss is defined as follows.

$$\theta^t = -\frac{1}{2} \sum \frac{(\sum g_j)^2}{\sum (h_j + \lambda)} + \gamma K \quad (21)$$

E. Hybrid Integration Pipeline

The proposed recommendation model applies a sequential hybrid learning framework that integrates Matrix Factorization, an MLP-based Feedforward Neural Network, and XGBoost. Each component is explicitly connected in a pipeline to ensure that the output of one stage serves as the input to the next stage.

1) Matrix Factorization Stage

Matrix Factorization is first applied to decompose the user-item rating matrix into latent user and item representations. The predicted rating generated by Matrix Factorization is defined as:

$$\hat{r}_{ui}^{MF} = \mu + b_u + b_i + p_u^T q_i \quad (22)$$

where μ represents the global average rating, b_u and b_i denote the user and item bias terms, and p_u and q_i are the latent feature vectors of users and items, respectively.

Instead of directly using the predicted rating as the final output, the latent vectors are concatenated to form a joint latent interaction vector:

$$z_{ui} = [p_u || q_i] \quad (23)$$

This latent interaction vector is forwarded to the next learning stage.

2) MLP-based Stage

The concatenated latent vector z_{ui} is used as the input to the MLP-based Feedforward Neural Network. This network is designed to capture nonlinear interactions between users and items that cannot be modeled by Matrix Factorization alone. The MLP output is expressed as:

$$\hat{r}_{ui}^{MLP} = f_{MLP}(z_{ui}) \quad (24)$$

where $f_{MLP}(\cdot)$ denotes the nonlinear transformation learned by the MLP using Swish activation functions. The resulting prediction reflects a refined nonlinear estimation of user preferences.

3) Handcrafted Features and XGBoost Stage

To further enhance prediction accuracy and address cold-start and first-rater problems, handcrafted statistical features are incorporated into the model. These features include the global average rating, user average rating, item average rating, similar users rating, and similar items rating. The final feature vector is constructed as:

$$x_{ui} = [\hat{r}_{ui}^{MLP}, GA, UA_u, IA_i, SU_u, SI_i] \quad (25)$$

This feature vector is used as the input to the XGBoost model, which produces the final rating prediction:

$$\hat{r}_{ui}^{final} = f_{XGB}(x_{ui}) \quad (26)$$

XGBoost refines the prediction by learning residual errors and nonlinear decision boundaries through an ensemble of gradient-boosted decision trees.

From a mathematical perspective, the proposed hybrid framework follows a sequential derivation pipeline. Matrix Factorization first produces latent user-item vectors, which are concatenated to form the joint interaction vector z_{ui} (Eq. 23). This vector serves as the direct input to the MLP, yielding a nonlinear rating estimation $\hat{r}_{ui}^{MLP}$ (Eq. 24). The MLP output is then combined with handcrafted statistical features to construct the final feature vector x_{ui} (Eq. 25), which is subsequently used by XGBoost to generate the final prediction $\hat{r}_{ui}^{final}$ (Eq. 26). This formulation explicitly connects the mathematical components of Matrix Factorization, MLP, and XGBoost into a unified learning pipeline rather than independent models.

The detailed steps of the proposed Matrix Factorization, MLP and XGBoost hybrid recommendation algorithm are presented in Algorithm as follows.

Input:
User-item rating matrix R
Latent dimension F
MLP parameters Θ_{MLP}
XGBoost parameters Θ_{XGB}
Output:
Final predicted rating $\hat{r}_{ui}^{final}$
1: Initialize Matrix Factorization parameters (P, Q, b_u, b_i)
2: Train SGD-based Matrix Factorization on observed ratings in R

```

3: For each observed user-item pair  $(u, i) \in R$  do
4:   Construct latent interaction vector  $z_{ui} = [p_u \parallel q_i]$ 
5:   Predict nonlinear rating  $\hat{r}_{ui}^{MLP} = \text{MLP}(z_{ui})$ 
6: End For
7: Compute handcrafted statistical features using MLP predictions:
   Global average rating (GA)
   User average rating (UAu)
   Item average rating (IAi)
   Similar users rating (SUu)
   Similar items rating (SIi)

8: Construct final feature vector:
    $x_{ui} = [\hat{r}_{ui}^{MLP}, \text{GA}, \text{UA}_u, \text{IA}_i, \text{SU}_u, \text{SI}_i]$ 

9: Train XGBoost model using feature vector  $x_{ui}$ 
10: Predict final rating  $\hat{r}_{ui}^{\text{final}} = \text{XGBoost}(x_{ui})$ 

Return  $\hat{r}_{ui}^{\text{final}}$ 

```

IV. EXPERIMENTS, DATASETS, EVALUATION AND RESULTS

In this section, we briefly present the experimental setup, the real-world datasets used, the evaluation metrics employed to assess model performance, and the validation of the simulation results.

A. Experiments setup

The experimental setup is summarized in Table I. All experiments for the proposed hybrid Collaborative Filtering technique were conducted on a system equipped with an AMD Ryzen 9 3900X central processing unit (3.80 GHz) and 64 GB of memory. The hybrid Collaborative Filtering technique combines Matrix Factorization, MLP and XGBoost for generating recommendations.

TABLE I. EXPERIMENTAL ENVIRONMENT

System resource	Description
Programming language	Python
Operating system	Windows 11
CPU	AMD Ryzen 9 3900X 12-Core Processor (3.80 GHz)
Memory	64 Gb
Algorithm	Matrix Factorization, MLP, XGBoost
Recommendation technique	Collaborative Filtering

To ensure experimental reproducibility, all datasets were split into training and testing sets using a fixed random seed ($random_state = 42$), with a 90:10 ratio. Hyperparameter tuning was conducted exclusively on the training set, and the best configurations were selected based on MAE and RMSE performance. All reported results were obtained using fixed hyperparameter settings summarized in Tables II and III.

TABLE II. OPTIMAL HYPERPARAMETER CONFIGURATIONS FOR MATRIX FACTORIZATION AND MLP

Model	Dataset	MAE
Matrix Factorization	Movies	Latent factors = 64, learning rate = 0.03, regularization = 0.03, epochs = 35
	Hotels	Latent factors = 64, learning rate = 0.04, regularization = 0.04, epochs = 40
MLP	Movies	Hidden layers = [128, 64, 32, 16], learning rate = 0.003, batch size = 64, epochs = 50, patience = 15
	Hotels	Hidden layers = [128, 64, 32], learning rate = 0.005, batch size = 64, epochs = 55, patience = 15

TABLE III. OPTIMAL HYPERPARAMETER CONFIGURATION FOR XGBOOST

Dataset	MAE
Movies	$n_estimators = 900$, $learning_rate = 0.5$, $max_depth = 9$, $min_child_weight = 5$, $subsample = 0.8$, $colsample_bytree = 0.8$, $gamma = 0.1$, $reg_alpha = 0.01$, $reg_lambda = 1.0$, $random_state = 42$
Hotels	$n_estimators = 1000$, $learning_rate = 0.5$, $max_depth = 10$, $min_child_weight = 5$, $subsample = 0.8$, $colsample_bytree = 0.8$, $gamma = 0.1$, $reg_alpha = 0.01$, $reg_lambda = 1.0$, $random_state = 42$

TABLE IV. COMPUTATIONAL COST AND TRAINING TIME OF THE PROPOSED MODELS

Model	Datasets	Training Time (hours)	Model Complexity
Matrix Factorization	Movies	~1.03 h	Latent factors (F = 64)
Matrix Factorization + MLP	Movies	~2.14 h	Matrix Factorization + 4 hidden layers
Matrix Factorization + MLP + XGBoost	Movies	~3.42 h	Matrix Factorization + MLP + 900 trees
Matrix Factorization	Hotels	~20.78 h	Latent factors (F = 64)
Matrix Factorization + MLP	Hotels	~41.89 h	Matrix Factorization + 3 hidden layers
Matrix Factorization + MLP + XGBoost	Hotels	~44.52 h	Matrix Factorization + MLP + 1000 trees

The proposed hybrid recommendation model incurs additional computational cost due to the sequential integration of Matrix Factorization, MLP, and XGBoost. As shown in Table IV, training time increases progressively with each added component, with Matrix Factorization being the least expensive. Although the complete hybrid model requires longer training time, the cost remains acceptable for offline recommendation scenarios and is justified by the consistent performance improvements. The progressive comparison of Matrix Factorization, Matrix Factorization + MLP, and Matrix Factorization + MLP + XGBoost provides implicit ablation evidence of each component's contribution to the final accuracy.

B. Dataset

The datasets used in the experiments are summarized in Table V. The MovieLens 100K dataset is a widely used benchmark dataset in recommender system research. It contains 100,000 explicit ratings on a 1–5 scale, provided by 943 users for 1,682 movies, where each user has rated at least 20 movies. On average, each user provides approximately 106 ratings, which is consistent with the official public specification of the dataset. The sparsity level is computed using the standard formulation based on the ratio between the number of observed ratings and the total possible user-item interactions. Similarly, the Hotel dataset used in this study is obtained from PT. XYZ and represents hotels across Indonesia. The dataset contains 16,292 hotels with 9,741 ratings provided by 7,298 users.

TABLE VI. A SNIPPET OF ALL THE MOVIE FEATURES

user_id	movie_id	actual_rating	predicted_rating	ga_r	ua_r	ia_r	su_r	si_r
1	1	0.0	5.0	2.7	3.3	3.9	3.9	3.3
1	2	0.0	4.0	2.7	3.3	3.4	3.4	3.3
1	3	0.0	4.0	2.7	3.3	3.1	3.1	3.3
...	...	...	...	...	...	...	...	...
610	193585	0.0	3.0	2.7	3.1	2.5	2.5	3.1
610	193587	0.0	3.0	2.7	3.1	2.5	2.5	3.1
610	193609	0.0	4.0	2.7	3.1	3.2	3.2	3.1

TABLE VII. A SNIPPET OF ALL THE HOTEL FEATURES

user_id	hotel_id	actual_rating	predicted_rating	ga_r	ua_r	ia_r	su_r	si_r
1	69602	0.0	3.0	1.5	1.4	1.7	1.7	1.4
1	69603	0.0	3.0	1.5	1.4	3.0	3.0	1.4
1	69603	0.0	1.0	1.5	1.4	1.1	1.1	1.1
...	...	...	...	...	...	...	...	...
8588	85891	0.0	1.0	1.5	1.2	1.1	1.1	1.2
8588	85892	0.0	1.0	1.5	1.2	1.5	1.5	1.2
8588	85893	0.0	1.0	1.5	1.2	1.5	1.5	1.2

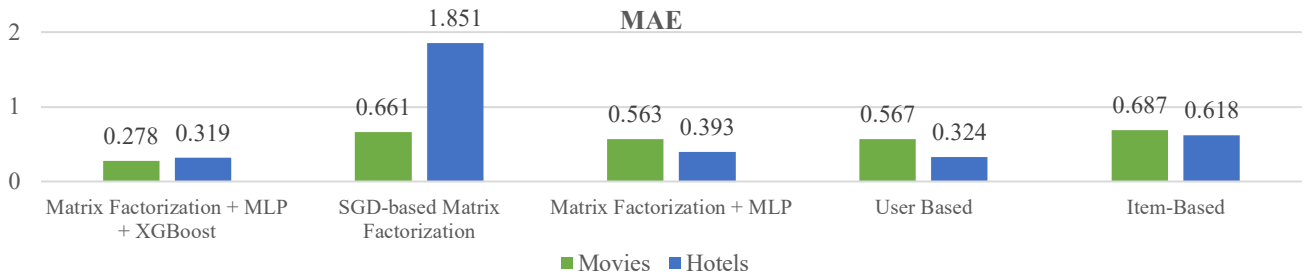


Fig. 2. MAE performances on movie and hotel

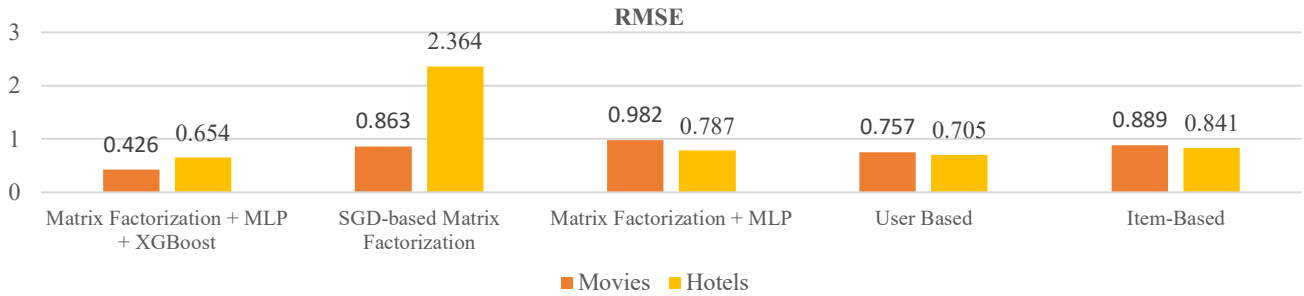


Fig. 3. RMSE performances on movie and hotel

TABLE V. DESCRIPTION OF THE DATASET

Datasets	Number of items	Number of users	% of sparsity
Movies	1682	943	98.33
Hotels	16292	7298	99.99

The level of sparsity in the dataset is quantified using the following formula.

$$\text{Sparsity} = 1 - \frac{R}{M \times N} \quad (27)$$

let R be the total number of ratings, and M and N denote the number of users and the number of items, respectively [10].

C. Evaluation metrics

Recommendations are used to predict the ratings of items for users. To evaluate the accuracy of these predictions, three evaluation measures are employed: Mean Absolute Error (MAE), as defined in (28), Root Mean Squared Error (RMSE), as defined in (29). These metrics quantify how closely the predicted ratings align

with the actual ratings. Low MAE and RMSE values reflect high predictive accuracy of the recommendation model. The MAE and RMSE [23] are defined as follows.

$$\text{MAE} = \frac{\sum_{i \in S_{rec}} |r_{ui} - \tilde{r}_{ui}|}{N} \quad (28)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i \in S_{rec}} (r_{ui} - \tilde{r}_{ui})^2}{N}} \quad (29)$$

where r_{ui} denotes the true rating given by user u to item i , and $\tilde{r}_{ui}$ represents the predicted rating of user u for item i ; N is the number of recommended items, and S_{rec} denotes the set of recommended items.

In addition to MAE and RMSE, we evaluate the ranking quality of the recommendation models using Top-k metrics, namely Recall@k and Precision@k. For each user, the model generates a ranked list of top-10 recommended items based on predicted ratings. An item is considered relevant if its ground-truth rating is greater than or equal to 4.0. These ranking-based metrics assess the ability of the model to place relevant items at higher positions in the recommendation list, which is crucial for

practical recommender system performance. Based on these quantities, Recall@k and Precision@k are defined as:

$$\text{Recall@k} = \frac{|R_u^k \cap T_u|}{|T_u|} \quad (30)$$

$$\text{Precision@k} = \frac{|R_u^k \cap T_u|}{k} \quad (31)$$

D. Results and Discussion

To evaluate the effectiveness of the hybrid collaborative filtering model method compared to the baseline technique on the reference dataset, the dataset was first divided into training and testing subsets using a 90:10 ratio. Further, matrix factorization was applied to decompose the user-item rating matrix, the output of which was used as input to an MLP to learn nonlinear relationship patterns between entities through hidden layers. The learned representations were subsequently incorporated as additional features into the XGBoost model. Additionally, several handcrafted features were generated, including the global average (ga_r), user average rating (ua_r), item average rating (ia_r), similar users rating (su_r), and similar items rating (si_r). Table VI presents a snapshot of all features used for movie rating prediction, while Table VII provides a snapshot of all features used for hotel rating prediction.

TABLE VIII. COMPARISON OF HYBRID-BASED COLLABORATIVE METHODS AND BASIC METHODS ON DATASETS

Model	Dataset	MAE	RMSE	Recall@k	Precision@k
Matrix Factorization + MLP + XGBoost	Movies	0.278	0.426	0.948	0.935
SGD-based Matrix Factorization	Hotels	0.319	0.654	0.876	0.975
Matrix Factorization + MLP	Movies	0.661	0.863	0.685	0.913
Matrix Factorization + MLP	Hotels	1.851	2.364	0.620	0.401
User-Based	Movies	0.563	0.982	0.758	0.877
User-Based	Hotels	0.393	0.787	0.832	0.962
Item-Based	Movies	0.567	0.757	0.917	0.903
Item-Based	Hotels	0.324	0.705	0.682	0.990
	Movies	0.687	0.889	0.928	0.858
	Hotels	0.618	0.841	0.794	0.904

Table VIII summarizes the comparative performance of all evaluated models on the Movie and Hotel datasets. The proposed Matrix Factorization + MLP + XGBoost model consistently achieves the lowest MAE and RMSE values, while also yielding higher Recall@10 and Precision@10 scores compared to all baseline approaches. These results indicate that the proposed model not only improves rating prediction accuracy but also more effectively ranks relevant items at top positions, which is essential for practical recommender system deployment.

The observed performance gains stem from the complementary roles of each component in the hybrid architecture. Matrix Factorization learns latent user-item representations but is limited by linear interaction assumptions. The MLP stage captures nonlinear interaction patterns beyond Matrix Factorization, while XGBoost further refines predictions using handcrafted statistical and similarity-based features, which are particularly beneficial for sparse users and first-rater items. The progressive improvement from Matrix Factorization to Matrix Factorization + MLP and ultimately to Matrix

Factorization + MLP + XGBoost provides implicit ablation evidence that each component contributes to the overall model performance. To verify whether the observed performance improvements are statistically significant, a paired t-test was conducted between the proposed hybrid model and each baseline method using the same test set. The results indicate that the improvements are statistically significant at a significance level of 0.05. Consistent with these findings, figures 4 and 5 provide qualitative evidence of the model's prediction behavior by illustrating the alignment between actual and predicted ratings. The concentration of data points around the diagonal line indicates that large deviations are relatively rare, while most predictions fall within a reasonable range of the ground truth values. This visual pattern supports the quantitative results reported in terms of MAE and RMSE, confirming that the proposed model produces stable and consistent predictions across both datasets.

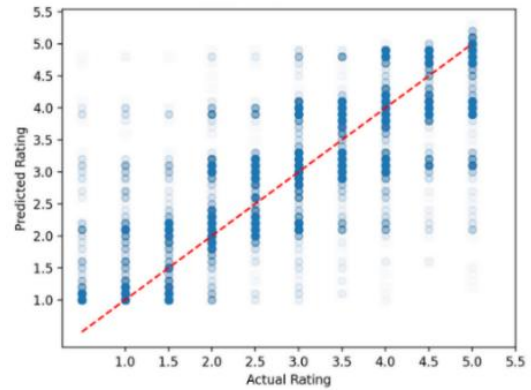


Fig. 4. Scatter plot of actual versus predicted ratings for the movie dataset, providing qualitative evidence of prediction alignment.

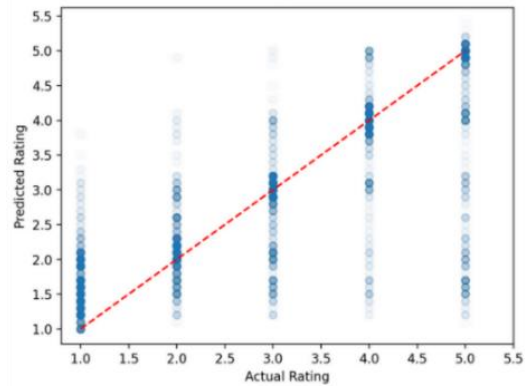


Fig. 5. Scatter plot of actual versus predicted ratings for the hotel dataset, illustrating the consistency of predictions with ground truth ratings.

The reported results were further validated using multiple random train-test splits. Across all splits, the proposed model consistently outperformed baseline methods, and paired t-test results confirmed that the improvements are statistically significant ($p < 0.05$). This confirms that the observed gains are robust and practically meaningful rather than artifacts of a single split.

The superior performance of the proposed hybrid model arises from the complementary roles of its components. Matrix Factorization captures latent user-item representations but is limited by linear interaction assumptions. The integration of an MLP enables modeling of nonlinear interactions, leading to improved accuracy, while XGBoost further refines predictions using handcrafted statistical and similarity-based features, which are particularly beneficial for sparse users and first-rater

items. The progressive improvements from Matrix Factorization to Matrix Factorization + MLP and Matrix Factorization + MLP + XGBoost provide implicit ablation evidence that each component contributes meaningfully to the overall effectiveness of the proposed model.

To clarify the apparent differences in RMSE values across baseline methods in Table VIII, we confirm that all models, including SGD-based Matrix Factorization, Matrix Factorization + MLP, and user/item-based collaborative filtering, were evaluated on the same original rating scale without additional normalization during error computation. The reported RMSE values are therefore directly comparable across methods. The relatively higher RMSE observed in certain baselines, particularly SGD-based Matrix Factorization on the hotel dataset, is attributed to extreme data sparsity and unbounded predictions rather than differences in evaluation scale or normalization.

The high RMSE of the SGD-based Matrix Factorization baseline on the hotel dataset is mainly caused by extreme sparsity and unbounded predictions, whereas the proposed hybrid model yields more stable estimates through MLP and XGBoost.

The proposed hybrid collaborative filtering model integrating Matrix Factorization, MLP, and XGBoost consistently outperforms baseline methods across all evaluated datasets in terms of MAE and RMSE. This improvement is driven by the combination of nonlinear interaction modeling via MLP and feature-based refinement using XGBoost, complemented by latent factor learning and handcrafted user-item statistical features.

E. Cold-Start and First-Rater Evaluation

In this study, k is set to 3, where users or items with fewer than three historical interactions are considered cold-start entities. To explicitly validate the capability of the proposed hybrid model in addressing cold-start and first-rater problems. Additional experiments were conducted under controlled sparse scenarios to explicitly evaluate cold-start and first-rater performance.

In the cold-start user setting, users with fewer than k historical ratings were isolated, and all their interactions were excluded from the training set. The models were then evaluated based on their ability to predict ratings for these unseen users.

Similarly, the first-rater item scenario was simulated by selecting items with extremely limited interaction history and removing their ratings from the training data. The task was to predict initial ratings for these new items.

The proposed hybrid model was compared against SGD-based Matrix Factorization, Matrix Factorization + MLP, and traditional user-based and item-based collaborative filtering approaches. Performance was evaluated using MAE and RMSE metrics.

TABLE IX. PERFORMANCE COMPARISON UNDER COLD-START USER AND FIRST-RATER ITEM SCENARIOS

Model	Scenario	Scenario	MAE	RMSE
Matrix Factorization (SGD)	Cold-start and	Movies	0.661	0.863
	First-rater	Hotels	1.851	2.364
Matrix Factorization + MLP	Cold-start and	Movies	0.563	0.982
	First-rater	Hotels	0.393	0.787
Matrix Factorization + MLP + XGBoost	Cold-start and	Movies	0.278	0.426
	First-rater	Hotels	0.319	0.654

Table IX presents the MAE and RMSE results under cold-start user and first-rater item scenarios.

The experimental results demonstrate that the proposed Matrix Factorization + MLP + XGBoost model consistently outperforms baseline methods under both cold-start user and first-rater item scenarios. The performance gain is primarily attributed to the incorporation of handcrafted features, such as global average, user and item averages, and similarity-based ratings, which provide informative signals even in the absence of sufficient historical interactions. These findings empirically confirm the model's robustness in sparse recommendation settings.

V. CONCLUSION

This study proposes a hybrid collaborative filtering recommendation system that integrates Matrix Factorization, MLP-based Feedforward Neural Network, and XGBoost to address sparsity, cold-start, and first-rater problems. Unlike existing hybrid collaborative filtering frameworks that typically combine only two learning components, the proposed model introduces a unified three-stage pipeline in which Matrix Factorization learns latent user and item representations, MLP captures nonlinear user and item interactions, and XGBoost refines predictions using handcrafted statistical features. While the proposed method is built upon established learning models, its contribution lies in the integration strategy, learning dependency, and empirical validation across multiple sparse domains, which together provide new insights into designing robust hybrid recommender systems.

For future research directions, the proposed method can be extended by incorporating user opinion analysis through review mining or sentiment analysis derived from user activities. Textual information in the form of comments or reviews can be transformed into sentiment scores using sentiment lexicons, thereby converting linguistic expressions that reflect user attitudes into numerical values representing the level of appreciation or satisfaction with an item. Furthermore, the topics or subjects embedded in such comments can be correlated with other items potentially relevant to user preferences, thereby enriching the information representation within the recommendation system.

In addition, the proposed method can be further enhanced by considering additional dimensions of user and item features, as well as by adopting deep learning approaches and advanced optimization algorithms to accelerate the model training process. Future research may also focus on integrating both explicit and implicit user feedback, as well as evaluating the implementation in real-world scenarios to ensure the external validity and practical applicability of the developed recommendation system.

DATA AVAILABILITY

The datasets used in this study are publicly available, and the source code can be accessed via the following Git repository: <https://github.com/filmangullo/Collaborative-Filtering---Matrix-Factorization---Feedforward-Neural-Network---XGBoost>

ACKNOWLEDGMENT

The authors would like to thank Erwin Setiawan Panjaitan for his valuable insights and feedback, and Mr. Hardy, Rector of Universitas Mikroskil, for his support. The authors also acknowledge the faculty and staff of the Information Technology Program at Universitas Mikroskil, as well as the broader academic community, for their contributions and support.

REFERENCES

- [1] M. Ibrahim, I. S. Bajwa, N. Sarwar, F. Hajjaj and H. A. Sakr, "An Intelligent Hybrid Neural Collaborative Filtering Approach for True Recommendations," *IEEE Access*, Vol. 11, p. 64831, 2023, doi: 10.1109/ACCESS.2023.3289751.
- [2] K. K. Yadav, H. K. Soni, G. Y. Yadav and M. Sharman, "Collaborative Filtering Based Hybrid Recommendation System using Neural Network and Matrix Factorization Techniques," *International Journal of Intelligent Systems and Applications in Engineering*, Vol. 12, pp. 695-701, 2023. [Online]. Available: <https://ijisae.org/index.php/IJISAE/article/view/4307>
- [3] Y. Li, J. Xu and M. Yang, "Collaborative Filtering Recommendation Algorithm Based on KNN and Xgboost Hybrid," *Journal of Physics: Conference Series*, p. 1-6, 2021, doi: 10.1088/1742-6596/1748/3/032041.
- [4] M. Ibrahim, I. S. Bajwa, N. Sarwar, H. A. Waheed, M. Z. Hasanand and M. Z. Hussain, "Improved Hybrid Deep Collaborative Filtering Approach for True Recommendations," *Computers, Materials & Continua*, Vol. 74 pp. 5301-5317, 2022, doi: 10.32604/cmc.2023.032856.
- [5] S. R. Mandalapu, B. Narayanan and S. Putheti, "A Hybrid Collaborative Filtering Mechanism for Product Recommendation System," *Multimedia Tools and Applications*, Vol. 83, pp. 12776-12798, 2024. [Online]. Available: <https://link.springer.com/article/10.1007/s11042-023-16056-8>
- [6] C. Lai, D. Liu and K. Lien, "A hybrid of XGBoost and Aspect-based Review Mining with Attention Neural Network for User Preference Prediction," *International Journal of Machine Learning and Cybernetics*, Vol. 12, pp. 1203-1217, 2021. [Online]. Available: <https://link.springer.com/article/10.1007/s13042-020-01229-w>
- [7] Y. Qingmin, C. Xingyu, "Research on Collaborative Filtering Recommendation Algorithm," *2022 IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, p. 741-743, 2022, doi: 10.1109/ICPECA53709.2022.9719134.
- [8] K.K Jena, S. K. Bhoi, C. Mallick, S. R. Jena, R. Kumar, H. V. Long and N. T. K. Son, "Neural Model Based Collaborative Filtering for Movie Recommendation System", *International Journal of Information Technology*, Vol. 14, pp. 2067-2077, 2022. [Online]. Available: <https://link.springer.com/article/10.1007/s41870-022-00858-4>
- [9] R. Rismala, N. U. Maulidevi and K. Surendro, "Personalized Neural Network-based Aggregation Function in Multi-criteria Collaborative Filtering," *Journal of King Saud University - Computer and Information Sciences*, Vol. 36, pp. 1-14, 2024, doi: 10.1016/j.jksuci.2024.101922.
- [10] G. Behera, S. K. Panda, M. Hsieh and K. Li, "Hybrid Collaborative Filtering using Matrix Factorization and XGBoost for Movie Recommendation," *Computer Standards & Interfaces*, Vol. 90, pp. 1-8, 2023, doi: 10.1016/j.csi.2024.103847.
- [11] S. Rendle, W. Krichene, L. Zhang and John Anderson, "Neural Collaborative Filtering vs. Matrix Factorization Revisited," *Computer Science and Engineering*, pp. 240-248, 2020, doi: 10.1145/3383313.3412488.
- [12] F. Ullah, B. Zhang, R. U. Khan, T. Chung, M. Attique, K. Khan, S. E. Khediri and S. Jan, "Deep Edu: A Deep Neural Collaborative Filtering for Educational Services Recommendation," *IEEE Access*, Vol. 8, pp. 110915-110926, 2020, doi: 10.1109/ACCESS.2020.3002544.
- [13] K. Singh, S. Dhawan and N. Bali, "An Ensemble Learning Hybrid Recommendation System using Content-Based, Collaborative Filtering, Supervised Learning and Boosting Algorithms," *Automatic Control and Computer Sciences*, Vol. 58, pp. 491-505, 2023, doi: 10.3103/S0146411624700615.
- [14] J. Chen, C. Zhao, Ulji and L. Chen, "Collaborative Filtering Recommendation Algorithm Based on User Correlation and Evolutionary Clustering," *Complex & Intelligent Systems*, Vol. 6, pp. 147-156, 2020, doi: 10.1007/s40747-019-00123-5.
- [15] R. Lara-Cabrera, A. González-Prieto and F. Ortega, "Deep Matrix Factorization Approach for Collaborative Filtering Recommender Systems," *Applied Sciences*, Vol. 10, pp. 1 - 14, 2020, doi: 10.3390/app10144926.
- [16] P. Li, S. A. M. Noah and H. M. Sarim, "A Survey on Deep Neural Networks in Collaborative Filtering Recommendation Systems," 2024. [Online]. Available: <https://arxiv.org/abs/2412.01378>.
- [17] L. Zou, L. Xia, Y. Gu, X. Zhao, W. Liu, J. X. Huang and D. Yin, "Neural Interactive Collaborative Filtering," *SIGIR '20: The 43rd International ACM SIGIR conference on research and development in Information Retrieval*, p. 749 - 758, 2020, doi: 10.1145/3397271.3401181.
- [18] H. Chen, F. Qian, C. Liu, Y. Zhang, H. Su, and S. Zhao, "Training Robust Deep Collaborative Filtering Models via Adversarial Noise Propagation," *ACM Transactions on Information Systems*, vol. 42, pp 1-27, 2023, doi: 10.1145/3589000.
- [19] W. Lin, M. Zhu, X. Zhou, R. Zhang, X. Zhao, S. Shen, and L. Sun, "A Deep Neural Collaborative Filtering Based Service Recommendation Method with Multi-Source Data for Smart Cloud-Edge Collaboration Applications," *Tsinghua Science and Technology*, vol. 29, pp. 897-910, 2024, doi: 10.26599/TST.2023.9010050.
- [20] Z. Shahbazi, D. Hazra, S. Park and Y. C. Byun, "Toward Improving the Prediction Accuracy of Product Recommendation System Using Extreme Gradient Boosting and Encoding Approaches," *Symmetry*, Vol. 12, p. 1 - 20, 2020, doi: 10.3390/sym12091566.
- [21] M. Nalluri, M. Pentela and N. R. Eluri, "A Scalable Tree Boosting System: XG Boost," *International Journal of Research Studies in Science, Engineering and Technology*, Vol. 7, p. 36 - 51, 2020, doi: 10.22259/2349-476X.0712005.
- [22] D. Deenadayalan and A. Kangaammal, "User Feature Similarity Supported Collaborative Filtering for Page Recommendation Using Hybrid Shuffled Frog Leaping Algorithm," *International Journal of Intelligent Engineering and Systems*, vol.16, pp. 301-313, 2023. doi: 10.22266/ijies2023.0228.27.
- [23] A. N. Ngaffo and Z. Choukair, "A Deep Neural Network-based Collaborative Filtering using a Matrix Factorization with a Twofold Regularization", *Neural Computing and Applications*, Vol 34, pp. 6991-7003, 2021, doi: 10.1007/s00521-021-06831-9.
- [24] Y. Mao and C. Ye, "User Identification Across Online Social Networks Based on Gated Multi-feature Extraction," *Alexandria Engineering Journal*, pp. 211 - 221, 2024, doi: 10.1016/j.aej.2024.06.059.



Filimantaptius Gulo has a degree in the Information Technology program at the Faculty of Informatics, Universitas Mikroskil, Medan, Indonesia. His current research focuses on the optimization of a hybrid collaborative filtering model, utilizing Matrix Factorization, Feedforward Neural Network, and XGBoost. (email: filimantapgullo@gmail.com).



Ronsen Purba is a Lecturer in the Master of Information Technology (MTI) Program at the Faculty of Informatics, at Universitas Mikroskil, Medan, Indonesia. (email: ronsen@mikroskil.ac.id).



Muhammad Fermi Pasha is a Lecturer in the Master of Information Technology (MTI) Program at the Faculty of Informatics, at Universitas Mikroskil, Medan, Indonesia. (email: muhammad.pasha@mikroskil.ac.id).