

A Multi-Criteria Recommendation Framework using Adaptive Linear Neuron

Mohammed Hassan¹

Department of Software Engineering
Bayero University Kano
Kano, Nigeria

Mohamed Hamada²

Graduate School of
Computer Science and Engineering
University of Aizu
Aizuwakamatsu-city, Fukushima, Japan

Saratu Yusuf Ilu³

Department of Software Engineering
Bayero University Kano
Kano, Nigeria

Abstract—Recent developments in the field of recommender systems have led to a renewed interest in employing some of the sophisticated machine learning algorithms to combine multiple characteristics of items during the process of making recommendations. Considerable number of research papers have been published on multi-criteria recommendation techniques. Most of these studies have focused only on using some basic statistical methods or simply by extending the similarity computation of the traditional heuristic-based techniques to model the system. Researchers have not treated the uncertainty that exists about the relationship between multi-criteria modelling approaches and effectiveness of some of the complex and powerful machine learning techniques; in fact, no previous study has investigated the role of artificial neural networks to design and develop the system using aggregation function approach. This paper seeks to remedy these challenges by analysing the performance of multi-criteria recommender systems, modelled by integrating an adaptive linear neuron that was trained using delta rule, and asymmetric singular value decomposition algorithms. The proposed model was implemented, trained and tested using a multi-criteria dataset for recommending movies to users based on action, story, direction, and visual effects of movies. Taken together, the empirical results of the study suggested that there is a strong association between artificial neural networks and the modelling approaches of multi-criteria recommendation technique.

Keywords—Multi-criteria recommender systems; adaptive linear neuron; artificial neural network; singular value decomposition; prediction accuracy

I. INTRODUCTION

Web-based services are growing expeditiously and producing considerable amount of data, which make it more challenging for users to find items that might be relevant to their preferences [1]. Recommender systems (RSs) are intelligent decision support systems that have been employed by many popular websites to assist users by recommending interesting items that might match their choices [2]. For example, Amazon is a popular online shop that analyzes the transaction history of their customers and the similarities between users to predict whether a user will be interested in some new/unseen items. In addition to the area of e-commerce, RSs have recently become among the exceptionally important systems, and are employed in a variety of web-based applications: some of the popular application areas include technology-enhanced learning, tourism guides, online news, hotel and restaurant guides, and more generally, in the area of social networking where people will be recommended to other people for friendships [3] [4], [5].

Usually, traditional RSs provide a list of recommendations through either a content-based filtering, a collaborative filtering, or a hybrid technique that integrates the two techniques in some ways. The content-based RSs predict ratings that a user might give to items based on their descriptions and historical records of user's preferences. Collaborative filtering-based recommender systems are generally based on the users' behaviour and their similarities with other users. The hybrid that combines the two techniques is considered in many cases to be more efficient than any of the single techniques [6], [7].

While those techniques have been successfully applied and their efficiency has been tested and are improved continuously over the past several years [8], one major problem of this kind of application that was recently discovered is the use of a single rating to determine users' preferences to items [9]. This is because several items' characteristics can play important roles in deciding whether to like an item or not. For example, in learning object RSs, a user may decide to read a book or a research paper based on either the author, the publisher, or just the quality of the contents of the learning object. Therefore, collecting additional information from the user that are related to various items' characteristics can by far improve the recommendation accuracy [10].

However, extending any of the traditional techniques to accommodate several conflicting criteria requires a new technique to effectively combine the multiple ratings for improving the accuracy of the systems [11]. Multi-criteria recommendation technique has been proposed to incorporate the criteria rating information and produces more accurate predictions than the existing single rating techniques. In addition, the issue has grown in importance in the light of improving the accuracy of both the traditional and the multi-criteria techniques. The accuracy of multi-criteria techniques has been subject to the approaches and algorithms used in combining the criteria ratings. One major approach is the aggregation function technique that focused on the mutual relationships between the criteria ratings to produce an overall rating, which represents the final preference of the user. Moreover, despite the efficiency of the aggregation function approach, little research has been able to draw on any systematic research into modeling the system using some machine learning techniques such as support vector regression [12], fuzzy-based algorithms [11], [13], and so on. In fact, no previous study has investigated the performance of the aggregation function approach using an adaptive linear neuron [9], [14]. This paper proposed a simple

neural network-based model integrated with an asymmetric singular value decomposition (AsymSVD) to examine the performance of the multi-criteria RSs. The experiment was conducted using a multi-criteria rating dataset that measures users' preferences on the basis of four characteristics of the items. The empirical results of the study were analyzed and compared with conventional single rating AsymSVD.

This paper first gives a brief overview of the related background in Dection II. Section III contains the experimental methodology while Dection IV gives the analysis of our findings, and finally, Dection V concludes the paper and proposed possible future research directions.

II. RELATED BACKGROUND

A. Asymmetric Singular Value Decomposition

SVD in the context of RSs is a matrix factorization model where items and users are represented by vectors in a latent factor space. The latent factor is a low dimensional space for comparing users and items and estimating the ratings between them as an inner product of their vectors. Asymmetric SVD (AsymSVD) is a powerful matrix factorization technique among the family of SVDs that is proved to be more efficient than the ordinary SVD technique [15]. It represents users as a combination of items' features to enable the system to quickly make predictions for new users [16].

Ordinarily, every user u in SVD is associated with an n -dimensional latent vector $V_u \in R^n$ and every item i is associated with a vector $V_i \in R^n$. The predictor of a rating \hat{r}_{ui} between u and i is given as:

$$\hat{r}_{ui} = b_{ui} + V_u^T V_i \quad (1)$$

where b_{ui} is a baseline predictor for normalizing the \hat{r}_{ui} by removing cases where some items might receive higher ratings and a tendency that some users may give higher ratings than others. The value of b_{ui} between u and i is computed using the overall average rating μ , the average rating for u μ_u , and the average rating for i μ_i as:

$$b_{ui} = \mu + (\mu_u - \mu) + (\mu_i - \mu) \quad (2)$$

Now, returning to the AsymSVD, it requires additional information to predict the value of \hat{r}_{ui} . Let $|N(u)|$ be the number of items on which u provides implicit feedback, and $|I(u)|$ be the total number of items rated by u , then the prediction rule for AsymSVD is given below as presented by [17]:

$$\hat{r}_{ui} = b_{ui} + V_i^T |I(u)|^{-\frac{1}{2}} \sum_{k \in I(u)} (r_{uk} - b_{uk}) x_k + |N(u)|^{-\frac{1}{2}} \sum_{k \in N(u)} y_k \quad (3)$$

where $V_i, x_i, y_i \in R^n$ are three n -dimensional factor vectors that each i is associated with. This technique offers many benefits that overcame some of the limitations of memory-based collaborative filtering techniques. It can handle new user problems since it does not parameterize users. Other benefits include expandability, efficient aggregation of implicit feedback, and it typically requires fewer parameters [17].

B. Adaptive Linear Neurons (Adaline)

Artificial Neural Network (ANN) is one of the biologically inspired algorithms that tend to imitate the manner of the decision process and functions of biological nervous systems like the brain [18]. ANN is a computing system consisting of a number of highly interconnected neurons to solve computational problems. ANNs have been successfully applied to address several real-life problems [19]. Over the past decades, research has shown an increased interest in using various kinds of ANNs due to its practical applications. Some of the areas of its applications include the area of physical science and engineering [?], medicine [20], business [21], education [22], and almost all areas of our daily activities.

To understand the basic structure of ANNs, Fig. 1 contains a simple neural network consisting of a single neuron. Although there are several internal computations performed by neurons in ANN, the figure can enable us to gain an understanding of the structure and some its basic functionalities. Though, the ANN presented in Fig. 1 contains a single neuron, but generally, ANN is typically organized in layers, and each layer is made up of neuron(s). The neurons contain activation functions for the network to learn and understand something complicated. It can be seen from the figure that the single neuron contains an activation function f (see Eq. 4) that receives the weighted sum of the inputs.

$$f\left(\sum_i x_i \omega_i\right) = \sum_i x_i \omega_i \quad (4)$$

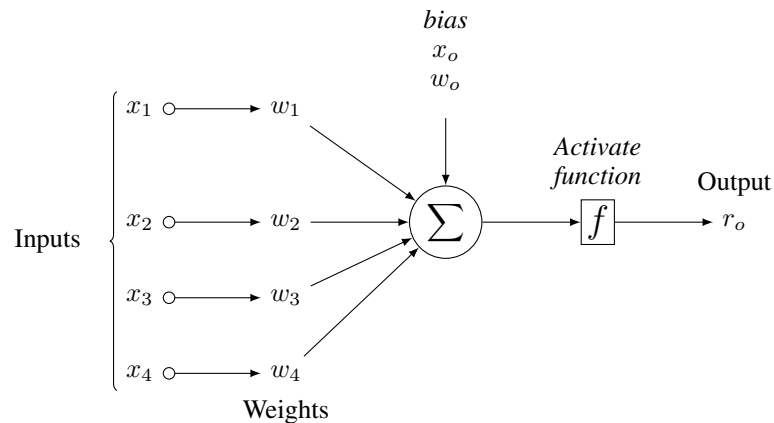


Fig. 1. Single layer artificial neural network

Although the general concept of ANNs has been formulated long ago by McCulloch and Pitts [23], the idea of an adaptive linear neuron (Adaline) was originally developed in later years by Widrow et al. [24] for designing adaptive switching circuits. Adaline is a network with exactly one neuron, having synaptic weights ω_i , a summation function, and a bias (x_o) similar to Fig. 1. Adaline uses continuous predictive model to learn the synaptic weights of the model. The synaptic weights are adjusted according to the value of $\sum_i \omega_i x_i$, (x_i is the i th input). To formalize its learning process, let σ be a positive real number called learning rate, which determines the rate of convergence of the network, and let r and \hat{r} be the target output and the actual output of the model respectively.

Then the weight $\omega_i(k+1)$ of i th input at $(k+1)$ th iteration is updated as given in Eq. (5).

$$\omega_i(k+1) = \omega_i(k) + \sigma \times E(k) \times x_i \times f' \left(\sum \right) \quad (5)$$

where k refers to k th iteration, f' is the derivative of the activation function, and E is the mean square error measured from the entire training data during k th iteration (see Eq. (6), where N is the size of the training data). The formula in Eq. (5) is called the delta rule, which was developed to consider the nonlinearity and the derivatives of the activation function [?].

$$E(k) = \frac{1}{N} \sum_{j=1}^N (r_j(k) - \hat{r}_j(k))^2 \quad (6)$$

C. Multi-Criteria Recommender Systems(MCRSSs)

In order to explain the concept of MCRSSs, it is important to briefly highlight the general concept of a collaborative filtering (CF) technique. CF is considered to be the simplest and the most commonly used recommendation technique [25]. The aim of CF is to predict ratings of items by active users based on their rating pattern. This technique is basically further subdivided into: *memory-based* techniques that use heuristics for rating predictions, and *model-based* techniques, which build some predictive models to learn about users' behaviors and make predictions. The AsymSVD explained in section II-A is a perfect example of a model-based CF technique, and therefore, the rest of the explanation will focus on memory-based techniques. The memory-based CF, also referred to as a neighborhood-based technique is one of the oldest and the most commonly used techniques that have been used in developing most of the existing RSs. It is mainly based on similarities between users (user-based) and/or between items purchased by the same user (item-based). Therefore, the two basic principles used to describe memory-based techniques are the user-based and the item-based CF techniques which used ratings of similar users or ratings of items rated in a similar fashion by the same user to make recommendations [26]. Altogether, the utility function f of single-rating RSs predicts the rating \hat{r}_{ui} of item i by user u as:

$$f : u \times i \mapsto \hat{r}_{ui} \quad (7)$$

It is necessary here to clarify exactly how the function f produces $\hat{r}_{ui} \forall u \in U$ and $\forall i \in I$, where U and I are the domain of users and items respectively. Although several heuristics have been formulated in different literature, the central idea of how the prediction function works in memory-based systems is the use of similarity values $sim(u, v)$ or $sim(i, j)$ between two users u and v or between two items i and j respectively to predict \hat{r}_{ui} . For example, Aggarwal [26] uses Eq. (8) to explain how to estimate \hat{r}_{ui} where \bar{u} is the mean rating of u (see Eq. (9)) and $sim(u, v)$ can be obtained using any of the various similarity measures such as Pearson correlation coefficient (see Eq. (10)), and $\rho_u(i)$ is a domain of users who are strongly similar to u and provide ratings to i .

$$\hat{r}_{ui} = \bar{u} + \frac{\sum_{v \in \rho_u(i)} sim(u, v)(\hat{r}_{vi} - \bar{v})}{\sum_{v \in \rho_u(i)} |sim(u, v)|} \quad (8)$$

$$\bar{u} = \frac{\sum_{k \in I_u} \hat{r}_{uk}}{|I_u|} \quad (9)$$

TABLE I. RATING MATRIX FOR MULTI-CRITERIA RSS

item/User	i_1	i_2	i_3	i_4
u_1	4 _{3,1,4,5}	5 _{5,1,5,5}	3 _{3,1,4,2}	2 _{1,4,2,1}
u_2	4 _{1,5,1,1}	5 _{1,5,2,1}	3 _{5,3,1,1}	1 _{2,1,5,2}
u_3	5 _{5,2,5,5}	4.5 _{4,3,4,5}	3 _{3,2,4,1}	1 _{1,3,1,1}
u_4	5 _{3,5,4,2}	2 _{4,1,4,3}	3 _{3,1,3,3}	4 _{1,5,2,2}

$$sim(u, v) = \frac{\sum_{k \in I_u \cap I_v} (\hat{r}_{uk} - \bar{u})(\hat{r}_{vk} - \bar{v})}{\sqrt{\sum_{k \in I_u \cap I_v} (\hat{r}_{uk} - \bar{u})^2} \sqrt{\sum_{k \in I_u \cap I_v} (\hat{r}_{vk} - \bar{v})^2}} \quad (10)$$

In MCRSSs, the value of \hat{r}_{ui} is specified by the user on the basis of multiple criteria. In contrast to the traditional single-rating techniques, MCRSSs require users to provide ratings to several items' attributes. Each rating represents a particular preference of the user on a specific attribute. For example, in movie RSs, the attributes can be the action, the story, the direction, and the visual effect of the movie. Table I shows examples of a multi-criteria recommendation problem just like that of the movie, where a user u_k provides ratings to four attributes of an item i_k for $k = 1, 2, 3, 4$ and an overall rating r_o (the bolded numbers) which is similar to \hat{r}_{ui} in the previous equations. It is interesting to note that from the rating information displayed in the table, it is somehow ambiguous or difficult to determine the correct similarity between users based on r_o . For instance, one may think that u_1 and u_2 are similar since they all give the same ratings to i_1, i_2 , and i_3 , and also they have almost the same opinions on the last item. But on the other hand, looking critically at their criteria ratings, the two users have entirely contradicting opinions; because, unlike u_2 , u_1 did not care about the influence of the rating given to the second criteria of each item [27]. Moreover, since MCRSSs recommendation problems require multiple ratings, then the utility function presented in Eq. (7) cannot be applied directly to solve these kinds of problems. Therefore, their utility function f extends that of the single-rating problems to account for all the criteria ratings as well as the overall rating (see Eq. (11)) [12], where r_o is similar to \hat{r}_{ui} in the single-rating function.

$$f : u \times i \mapsto r_o \times r_1 \times r_2 \times \dots \times r_n \quad (11)$$

Research on the MCRSSs has been mostly restricted to combining the criteria ratings to efficiently utilize this technique for improving the prediction accuracy of RSs. An aggregation function approach is one of the model-based approaches which assumes a relationship between the overall rating and the criteria ratings (see Eq. (12)) to predict users' preferences [9]. As indicated in Fig. 2, the framework of the aggregation approach requires a combination of a single rating CF and a learning model that learns the function in Eq. (12) to compute r_o .

$$r_o = g(r_1, r_2, \dots, r_n) \quad (12)$$

III. EXPERIMENT

To establish the predictive performance of the proposed technique, a multi-criteria dataset was collected for this study,

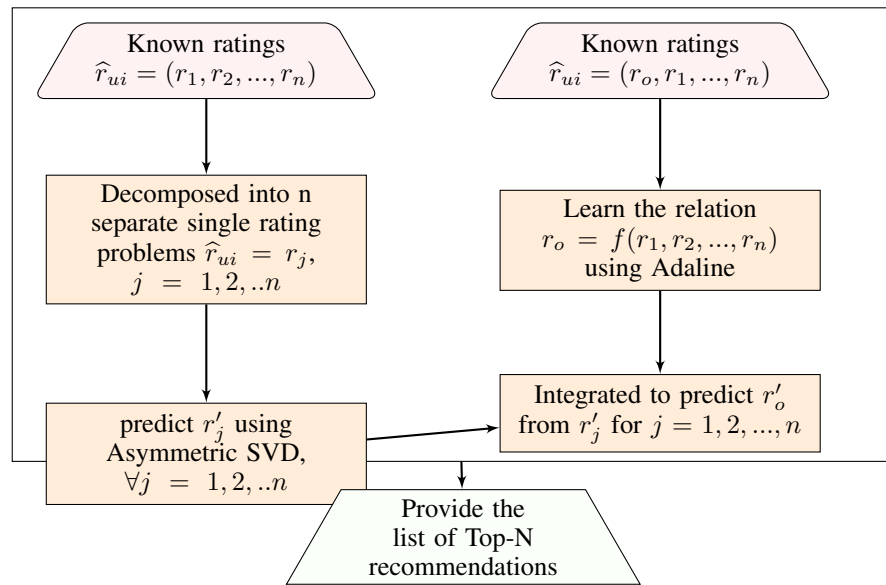


Fig. 2. Aggregation Function Framework

TABLE II. SAMPLE OF YAHOO MOVIE DATASET

UserID	MovieID	Direction(r ₁)	Action(r ₂)	Story(r ₃)	Visual(r ₄)	Overall(r _o)
101	1	13	6	5	8	5
101	3	9	10	10	11	9
101	5	8	11	10	10	11
103	26	6	6	8	12	8
103	2	9	11	10	9	10
103	61	6	10	9	8	7

and different evaluation metrics have been applied to analyse the effectiveness of the Adaline-based MCRSs. Therefore, the rest of this section is dedicated to describing the nature of the dataset and explanation of the evaluation metrics.

A. Dataset Description

The experiment was conducted using a Yahoo!Movie dataset [28] for recommending movies to users on the basis of four different criteria of movies. The four criteria are the action, direction, story, and visual effects of the movies. Furthermore, in addition to the four criteria, the dataset contains an overall rating that indicates whether a user is finally interested in a movie or not. The ratings are presented using scaled ratings from 1 to 13, initially collected in the form of letters from A^+ to F representing the highest and lowest preferences respectively. However, to work with only numerical data, the ratings were later transformed into positive integers from 13 to 1, representing the original values from A^+ to F respectively. For example, ratings like A^- , B^+ , C^- , A , and F were respectively changed to 11, 10, 5, 12, and 1. Also to avoid cases of missing ratings or cases of users who rated few movies, the dataset was further filtered so that only users with ratings of at least five movies would be considered. Finally, the dataset contains 62,156 ratings for 6,078 users on 976 movies, which shows that every user has rated an average of approximately 10 movies. Furthermore, the correlations between each criterion and the overall rating were measured to be 86.5%, 90.5%, 91.1%, and 83.4% for direction,

action, story, and visual respectively. Table II shows samples of the numerical dataset used in the experiment. The first two columns of the table contain the identification numbers for users and movies respectively, while the remaining five columns contain the criteria ratings and their corresponding overall rating.

B. Evaluation Metrics

Several evaluation metrics have been proposed in various research works to find out the efficiency of RSs with regard to a particular evaluation criterion such as prediction accuracy, systems' response time, user satisfaction, serendipity, and so on. However, as mentioned in Section ??, the aim of this study was to analyze the prediction accuracy of the proposed Adaline-based MCRSs, and compare its performance with that of the corresponding single-rating traditional technique. Therefore, we used some of the more powerful evaluation metrics for measuring the prediction accuracy, because accuracy is the most important property of RSs based on the assumption that a system that provides good prediction accuracy will obviously be preferred by users [14]. The three basic categories of prediction accuracy measures are: the rating prediction accuracy measures, the usefulness of the prediction measures, and the measure of the ranking accuracy of the predicted items. In measuring the rating prediction accuracy, we used two commonly used metrics: the mean absolute error (MAE) that estimates the deviation between predicted ratings and actual ratings (see Eq. (13)), and the root mean square error (RMSE)

in Eq. (14) which also computes the deviation as in MAE, but gives more emphasis to larger errors; where \hat{r}_k and r_k are the k th predicted and actual ratings respectively.

$$MAE = \frac{1}{n} \sum_{k=1}^n |\hat{r}_k - r_k| \quad (13)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{k=1}^n (\hat{r}_k - r_k)^2} \quad (14)$$

Furthermore, to determine whether the proposed technique recommends items which are predicted to be good, we applied the concept of precision and recall to measure the exact fraction of relevant recommendations out of all N recommended items and to determine the fraction of relevant items recommended out of all relevant items respectively. To mathematically define precision and recall, and some other evaluation metrics that will be explained in the next paragraph, let $\#tp$ be the number of relevant items out of the top- N items recommended to a user u , $\#fp$ be the number of irrelevant items out of the top- N items, $\#fn$ be the number of good items that are not recommended, and $\#tn$ be the number of irrelevant items that are not in the recommendation. Then the precision and recall are estimated using Eqs. (15) and (16) respectively. F_1 in Eq. (17) combines the precision and recall to compute the accuracy in measuring useful predictions [29].

$$precision = \frac{|relevant \ items \ recommended|}{|all \ recommendation|} = \frac{\#tp}{N} \quad (15)$$

$$recall = \frac{|relevant \ items \ recommended|}{|all \ relevant \ items|} = \frac{\#tp}{\#tp + \#fn} \quad (16)$$

$$F_1 = \frac{2precision \times recall}{precision + recall} \quad (17)$$

$$specificity = \frac{\#tn}{\#tn + \#fp} \quad (18)$$

Additionally, as mentioned earlier, relevant recommendations are more useful when they happen to appear at the topmost position of the recommendation lists (sorted in decreasing order of relevance), as items that appear beneath the list may be overlooked by users. Therefore, the experiment employed some ranking metrics that extend precision and recall to account for the position (rank) of relevant items in the ranked list. Such metrics include measuring the area under the curve (AUC) of a receiver operating characteristics (ROC) curve in Eq. (19), which measures how accurate the algorithms separate predictions into relevant and irrelevant by finding the area under the curve of the sensitivity rate (recall) against the specificity in Eq. (18). The mean reciprocal rank (MRR) given in Eq. (20) measures the response of RSs to a sample of a query, where $rank_{ui}^+$ is the position of relevant recommendation of i to u , and $rank_k$ is a position of the k th item among the N recommended items.

$$AUC_u = \frac{1}{N} \left[\left(\sum_{i=1}^{I_u^+} rank_{ui}^+ \right) + \binom{I_u^+ + 1}{2} \right] \quad (19)$$

$$MRR = \frac{1}{N} \sum_{k=1}^N \frac{1}{rank_k} \quad (20)$$

Moreover, other important ranking metrics for comparing prediction algorithms like the mean average precision (MAP, Eq. (21)) and a normalized discounted cumulative gain (NDCG, see Eq. (22)) were used, where rel_k is a binary number that determines whether the recommended item at rank k is relevant or not ($rel_k = 1$ if the item at position k is relevant and 0 otherwise). A fraction of concordant pair (FCP) (see Eq. (23)) was used to estimate the proportion of item pairs that are well ranked [30], where n_c is the total number of concordant pairs for all users in the test dataset that are ranked correctly for items i and j with their predicted ratings \hat{r}_{ui} and \hat{r}_{uj} , and the corresponding actual ratings from the dataset are r_{ui} and r_{uj} respectively, given by: $n_c = \sum_{u \in U} |(i, j)| \{ \hat{r}_{ui} > \hat{r}_{uj} \Rightarrow r_{ui} > r_{uj} \}$ and n_d is the corresponding sum for discordant pairs calculated as: $n_d = \sum_{u \in U} |(i, j)| \{ \hat{r}_{ui} > \hat{r}_{uj} \Rightarrow r_{ui} \leq r_{uj} \}$.

$$MAP = \frac{1}{m} \sum_{j=1}^m \sum_{k=1}^N \frac{precision@k}{I_u^+} \quad (21)$$

$$NDCG = \frac{DCG}{IDCG} = \frac{\left(rel_1 + \sum_2^N \frac{rel_k}{\log_2 k} \right)}{\left(rel_1 + \sum_2^{I_u^+ - 1} \frac{rel_k}{\log_2 k} \right)} \quad (22)$$

$$FCP = \frac{n_c}{n_c + n_d} \quad (23)$$

Finally, we used the Pearson correlation measure (Pc) in Eq. (24) to find the percentage correlation between predicted ratings and the corresponding actual ratings from the dataset.

$$Pc = \frac{\sum (\hat{r}^k - \bar{\hat{r}})(r^k - \bar{r})}{\sqrt{\sum (\hat{r}^k - \bar{\hat{r}})^2} \sqrt{\sum (r^k - \bar{r})^2}} \quad (24)$$

C. Experimental Settings

For the single rating AsymSVD and the ANN-based MCRSs proposed in this paper, there are certain important parameters controlling the output of the model. SVD algorithms generally worked based on two meta parameters called the learning rate γ (gamma) and the regularization λ (lambda) that prevents overfitting. They are set to default values of 2×10^{-3} and 4×10^{-2} respectively as they have been used in several SVD-based experiments [31]. Furthermore, working with an ANN equally require setting the learning rate σ (sigma) that controls the rate of convergence of the network, and other training parameters such as the maximum number of iterations, and target error. The value of σ was obtained to be 0.007 after trying several real numbers between 10^{-1} and 10^{-3} .

IV. RESULTS AND DISCUSSION

To analyse the result of the experiment, the experimental dataset was divided into training and test data using k -fold cross validation. This technique works by dividing the dataset randomly into k groups of approximately the same size. One part out of the k -groups was taken to be the test set after using the $k - 1$ groups for training the model. The same process was repeated k -times for training and testing, and calculate the average results of the evaluation metrics. Throughout this study, two different values of k (5 and 10) were used with the varying values of N (10 and 20) for

TABLE III. EVALUATION RESULT FOR 5-FOLD CROSS-VALIDATION. COLUMN TWO AND THREE (MCRSSs IN COLUMN TWO AND ASYMSVD IN COLUMN THREE) REPRESENT THE EXPERIMENTAL RESULT FOR 5-FOLD AND TOP-10, WHILE COLUMN FOUR AND FIVE ARE THE RESULTS FOR 5-FOLD AND TOP-20

	TOP-10		TOP-20	
	<i>MCRSSs</i>	<i>AsymSVD</i>	<i>MCRSSs</i>	<i>AsymSVD</i>
<i>MAE</i>	0.6931	2.4343	0.6925	2.4463
<i>RMSE</i>	1.1563	3.1322	1.1735	3.1482
<i>Precision</i>	0.7376	0.7111	0.7366	0.7100
<i>Recall</i>	0.9004	0.8573	0.8999	0.8570
<i>F₁</i>	0.8109	0.7773	0.8101	0.7766
<i>FCP</i>	0.9433	0.7054	0.9388	0.6949
<i>NDCG</i>	0.9935	0.9273	0.9939	0.9277
<i>MAP</i>	1.2149	0.0155	1.2404	0.0204
<i>MRR</i>	0.0474	0.0006	0.0243	0.0005
<i>AUC</i>	0.9405	0.6709	0.9458	0.6779

TABLE IV. EVALUATION RESULT FOR 10-FOLD CROSS-VALIDATION. COLUMN TWO AND THREE (MCRSSs IN COLUMN TWO AND ASYMSVD IN COLUMN THREE) REPRESENT THE EXPERIMENTAL RESULT FOR 10-FOLD AND TOP-10, WHILE COLUMN FOUR AND FIVE ARE THE RESULTS FOR 10-FOLD AND TOP-20

	TOP-10		TOP-20	
	<i>MCRSSs</i>	<i>AsymSVD</i>	<i>MCRSSs</i>	<i>AsymSVD</i>
<i>MAE</i>	0.7136	2.4318	0.6949	2.4216
<i>RMSE</i>	1.1819	3.1398	1.1561	3.1232
<i>Precision</i>	0.8336	0.8046	0.8434	0.8041
<i>Recall</i>	0.9016	0.8593	0.9018	0.8598
<i>F₁</i>	0.8663	0.8310	0.8716	0.8311
<i>FCP</i>	0.9407	0.7074	0.9364	0.7203
<i>NDCG</i>	0.9975	0.9344	0.9975	0.9337
<i>MAP</i>	1.0584	0.0229	1.3459	0.0414
<i>MRR</i>	0.0362	0.0007	0.0179	0.0005
<i>AUC</i>	0.9453	0.6990	0.9457	0.6966

the Top-N recommendation. This results in performing four different experiments: 5-fold and Top-10, 5-fold and Top-20, 10-fold and Top-10, and 10-fold and Top-20. The purpose of doing was to check whether in any of the four situations the performance of the single rating AsymSVD can outperform that of the proposed ANN-based MCRSSs. Table III shows the experimental results obtained from 5-fold cross validation and using Top-10 and 20 recommendation settings for the first two experiments. The table is divided into two parts as labelled TOP-10 and TOP-20 in the first row of the table. Each of the two experiments shows the performance of the proposed technique (MCRSSs) and the corresponding results of the single rating AsymSVD. Although the results of the two experiments did not show much significant changes, strong evidence of higher performance of the proposed technique was found in each of the corresponding values of the evaluation metrics. Similar results are shown in Table IV, where the value of k was changed to 10 (for 10-fold) so that their performance can be analysed when the size of training set was increased and the experiments were repeated 10 times instead of 5 times. The same values of N (TOP-10 and TOP-20) as in the previous table were maintained in order to see whether changing the value of k can have a great influence in their performance. Interestingly, the comparison of all the four results reveals no any case that favours single rating technique over the proposed ANN-based model. Moreover, for the purpose of making head to head comparison between the algorithms

and across all the evaluation metrics, Table V provides the average performance and the positive differences between the corresponding performance under each metric. To distinguish between these values in all the three tables, it is important to emphasize that except in the first two rows (MAE and RMSE) where smaller values show high prediction accuracy than the bigger values, the higher the value the more accurate is the algorithm. Furthermore, the last column of the table shows the average percentage of the accuracy improvement between the two techniques. For instance, the fraction of concordant pairs (FCP) has average values of 0.9398 and 0.7070 for MCRSSs and AsymSVD respectively, and a difference of 0.2328 (increase in accuracy), with percentage improvement of 32.93%.

This means, taking any two arbitrary predicted ratings from each of the techniques and taking their corresponding ratings from the dataset, the possibility of MCRSSs to satisfy the condition of been concordant pairs is 32.93% more than that of AsymSVD.

The percentage improvements were measured by dividing the positive difference by the corresponding value of AsymSVD under each metric and multiply the result by 100 (i.e, % improvement = $\frac{|MCRSSs - AsymSVD|}{AsymSVD} \times 100$, where $|MCRSSs - AsymSVD|$ is the absolute value of the difference in performance between the proposed ANN-based MCRSSs and the existing AsymSVD technique)

Nevertheless, to support these experimental findings with more evidence, the predicted values of the two algorithms were collected and filtered the inner joints of the ratings between each user×item pair. This was done in order to measure the strength of the existence of the linear relationship between the actual and the two predicted ratings. Which means, increase or decrease in actual rating will cause a corresponding increase or decrease in the predicted rating. Table VI displays the resulting correlation between all the three categories of ratings. The values were calculated using the correlation formula presented in Eq. (24) from the 5-fold and top-10 experiment. The single most striking observation to emerge from the data comparison was that the predictions of the proposed ANN-based technique are much closer to the actual ratings than to the AsymSVD. The result reported that the correlations were found to be approximately 95% between actual and predicted ratings of MCRSSs, and 78.3% between actual and predicted ratings of AsymSVD. The correlation between MCRSSs and AsymSVD (80.2%) is also interesting because it reflects the relationship between each of them and the actual rating. This means that the relationship that exists between the predictions of the AsymSVD and those of the other two ratings (actual and MCRSSs ratings) are almost the same, or in other words, the difference between the predictions of AsymSVD and the actual ratings is almost the same as that between the predictions of AsymSVD and that of the proposed ANN-based technique.

However, it is important to keep in mind that the data presented in the table measured only the strength of the linearity between the ratings. In particular, the questions now are: how do we determine if both ratings are increasing monotonically (That is, when the curve of the actual rating increases/decreases, so also that of the predicted rating) and what is the rate of increase or decrease of each of the predicted ratings with respect to the actual ratings? Those are

TABLE V. AVERAGE PERFORMANCE AND THE PERCENTAGE IMPROVEMENTS. THE AVERAGE WAS TAKEN FROM THE FOUR EXPERIMENTS (5-FOLD TOP-10, 5-FOLD TOP-20, 10-FOLD TOP-10, AND 10-FOLD TOP-20)

	<i>MCRSs</i>	<i>AsymSVD</i>	<i>differences</i>	<i>Improvement</i>
<i>MAE</i>	0.6985	2.4335	1.7350	71.30%
<i>RMSE</i>	1.1670	3.1359	1.9689	62.79%
<i>Precision</i>	0.7878	0.7574	0.0304	4.00%
<i>Recall</i>	0.9009	0.8584	0.0425	4.95%
<i>F₁</i>	0.8397	0.8040	0.0357	4.44%
<i>FCP</i>	0.9398	0.7070	0.2328	32.93%
<i>NDCG</i>	0.9922	0.9358	0.0564	6.03%
<i>MAP</i>	1.2149	0.0251	1.1898	4740.24%
<i>MRR</i>	0.0315	0.0006	0.0309	5150%
<i>AUC</i>	0.9443	0.6861	0.2582	37.63%

TABLE VI. CORRELATIONS BETWEEN PREDICTED AND ACTUAL RATINGS

	<i>Actual</i>	<i>MCRSs</i>	<i>AsymSVD</i>
<i>Actual</i>	1.000	0.950	0.783
<i>MCRSs</i>	0.950	1.000	0.802
<i>AsymSVD</i>	0.783	0.802	1.000

the questions that can not be answered directly from Table VI. Therefore, to address these questions, another method is required to clearly display the actual behaviour of each technique based on their predictions. To achieve this, we plotted graphs of some arbitrary corresponding actual ratings from the two algorithms and their corresponding actual ratings. Three graphs are plotted to show the strength of the monotonicity between: i. The actual and AsymSVD (see Fig. 3). ii. The actual and the predictions of the proposed ANN-based MCRSs (see Fig. 4), and iii. The curves of the combination of all the three ratings (see Fig. 5). The monotonicity is shown in the graphs by comparing the curve of the actual rating and that of the corresponding predicted values.

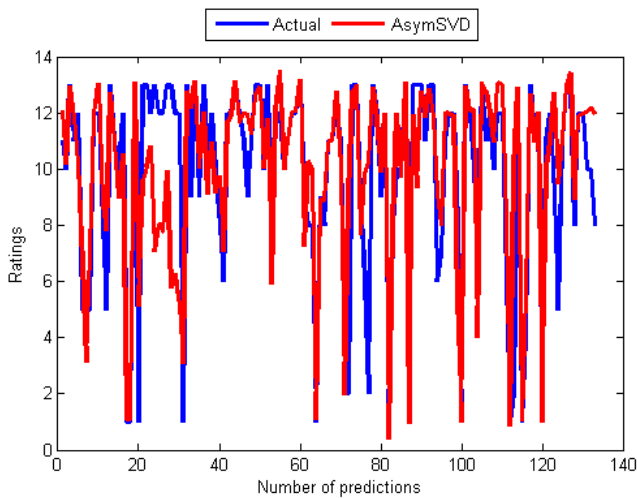


Fig. 3. Curves of actual ratings and predicted values for AsymSVD

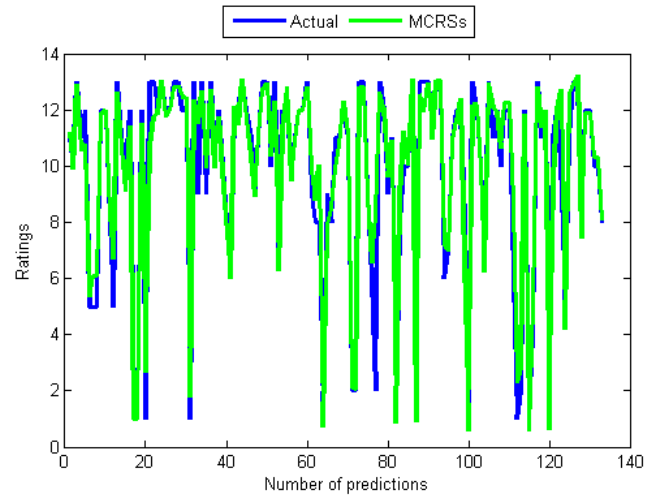


Fig. 4. Curves of actual and predicted values for MCRSs

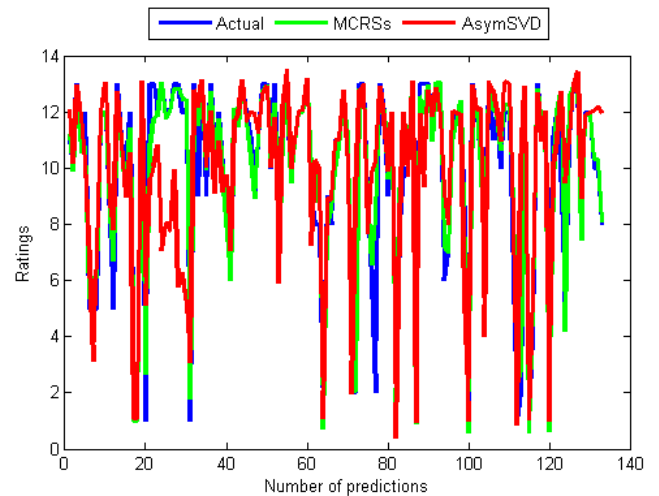


Fig. 5. Curves of actual and predicted values for MCRSs and AsymSVD

Considering the two curves in Fig. 3, the observed correlation between the actual and the predicted ratings might be explained in this way. In several occasions, the predicted ratings of AsymSVD are far away from the expected ratings from the test data. For example, from 20 to 40 along the number of predictions line (x-axis), almost all the ratings were predicted not near to the actual ratings, in fact, some are almost opposite to the expected ratings. That is, when the actual is high then the predicted value will be low and vice versa. As an example of such cases, the AsymSVD predicted 5.7 instead of 13, and 10.3 instead of 2. However, while the result is

TABLE VII. COMPARISON OF PERCENTAGE IMPROVEMENTS WITH OTHER RELATED WORKS

Model	MAE	RMSE
<i>MCRSSs</i>	71.30%	62.79%
Jannach et al [12]	44.44%	32.60%
Lakiotaki et al [28]	50.79%	48.13%
Jannach et al[32]	—	29.62%
Fan et al[33]	16.00%	—
Sahoo et al[34]	49.64%	—

not generally bad, these discrepancies are also attributed to the problems of prediction accuracy of the AsymSVD. On the other hand, the correlation between the predictions of the proposed ANN-based model and the actual ratings presented in Fig. 4 is interesting because the two curves moved together in almost all the 130 cases plotted in the figure. Furthermore, except for just one case close to point 78 along the x-axis where the proposed model predicted a higher value of 7.2 instead of 2, the generality of the correlation is extremely good. Finally, all the three curves were harmonized in Fig. 5 to produce a pleasing visual combination of predictions of the two techniques, which will make the comparative analysis more easier. The figures could serve as additional evidence to support our findings shown in the previous tables. Furthermore, this figure has pointed out some of the inconsistencies of the single rating technique as on many occasions, its predictions vary significantly from those of the actual and the proposed ANN-based model.

Nevertheless, to conclude this section, this study produced results which corroborate the findings of a great deal of the previous work in this field. We are aware that direct comparison of the results of the current study with the similar findings reported in other literature may be difficult since the datasets may not be identical, and the single rating techniques used to model the systems may entirely be different. The easier way to make the comparison was followed by taking the percentage of improvements in their studies similar to the method we used in Table V. Moreover, it was also observed that not all the evaluation metrics used in this study were applied in their work, but nevertheless, almost all of them used MAE and/or RMSE to analyse the prediction accuracy of their models. Therefore, Table VII contains the percentage decrease in errors between their models and the corresponding single rating techniques. For those of them that performed several experiments by changing some experimental parameters such value of N for top-N recommendation, or used several models by changing single rating technique as in the work of Jannach et al., where they used Slope One and SVD-based single ratings techniques, we considered taking the average of all the experiments conducted with varying value of N and taking the best performance improvement in the case of more than one technique. Interestingly, the proposed ANN-based MCRSSs in the first row was observed to produce the highest improvement over the previous works. The ones with the minus (-) sign mean the corresponding metric was not applied in their studies.

V. CONCLUSION AND FUTURE WORK

Several methods such as the of support vector machine [12], utilités additives algorithm (UTA) [28], probabilistic

methods such as Bayesian method [34], and so on, have been applied to user modeling for improving the prediction accuracy of multi-criteria recommendation technique as reported in the recent literature [9]. However, while these studies have contributed tremendously to the field of recommender systems, Adomavicius et al. [9] [14] have pointed out the need to explore some of the powerful machine learning techniques such as artificial neural networks into user modelling in multi-criteria recommendation using aggregation function approach and analyse the usefulness of such approach. According to recent reports, no research exists that used artificial neural networks to model this kind of multi-dimensional rating problem. The purpose of the current study was to design a neural network-based model that followed aggregation function approach to predict users' preferences in multi-criteria recommendation systems.

The proposed approach has employed an asymmetric singular value decomposition (AsymSVD) that was considered to be among the most accurate single rating techniques to model the system. Several experiments have been conducted and different evaluation metrics have been applied to evaluate and compare the accuracy of the proposed ANN-based model and the AsymSVD technique. The relevance of this approach to improve the prediction accuracy of MCRSSs is clearly supported by the current findings. The results of multiple evaluation metrics revealed that the ANN-based model is by far, better than the existing single rating technique. Moreover, the most interesting finding to emerge from this study is that the proposed model produced more accurate rating prediction accuracy than the previous works mentioned above. This was confirmed by the summary of their results in Table VII, where the percentage decrease in prediction errors are presented.

The following conclusions can be drawn from the present study. The present study provides additional evidence with respect to the effectiveness of using multiple ratings instead of just a single rating to predict users' preferences [9]. The findings of this study also indicate that using powerful machine learning algorithms especially ANNs can further enhance the prediction accuracy of MCRSSs. Together, this work contributes to existing knowledge of aggregation function approaches by providing the results of the predictive performance of one of the classical examples of the most powerful machine learning algorithms.

Apart from the work of Jannach that applied support vector regression [12], the current study is among the second attempts to apply powerful machine learning algorithms to solve multi-criteria recommendation problems using an aggregation function approach [9]. Collectively, the two studies explored only one component of soft computing. Other components of soft computing such as Fuzzy logic, evolutionary algorithms such as genetic algorithms, metaheuristics and swarm intelligence, and so on have not been experimented. It is recommended that further research be undertaken to analyze the performance of these algorithms towards improving the prediction accuracy of MCRSSs. Although AsymSVD was used in several works of literature and its efficiency has been proved to be good, a greater focus on more powerful single rating techniques could also produce interesting findings. The choice of the kind of ANN to be used in this research follows the recent study that established the superiority of the performance of single

layer network trained with delta rule over a multi-layered network trained using a back propagation algorithm [3], another possible area of future research would be to investigate the possibility of training the multi-layered networks using more powerful training algorithms such as simulated annealing algorithms, genetic algorithms, and more precisely, the issue of introducing deep learning into this domain is an intriguing one which could be usefully explored in further research.

REFERENCES

- [1] B. P. Knijnenburg and M. C. Willemsen, "Evaluating recommender systems with user experiments," in *Recommender Systems Handbook*. Springer, 2015, pp. 309–352.
- [2] M. Hamada and M. Hassan, "Artificial neural networks and particle swarm optimization algorithms for preference prediction in multi-criteria recommender systems," in *Informatics*, vol. 5, no. 2. Multidisciplinary Digital Publishing Institute, 2018, p. 25.
- [3] M. Hassan and M. Hamada, "Performance comparison of featured neural network trained with backpropagation and delta rule techniques for movie rating prediction in multi-criteria recommender systems," *Informatica*, vol. 40, no. 4, 2016.
- [4] —, "Performance comparison of feed-forward neural networks trained with different learning algorithms for recommender systems," *Computation*, vol. 5, no. 3, p. 40, 2017.
- [5] —, "Performance analysis of neural networks-based multi-criteria recommender systems," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. IEEE, 2017, pp. 490–494.
- [6] M. Hassan, "New machine learning methods for modeling a multi-criteria recommender system," Ph.D. dissertation, The University of Aizu, 2018.
- [7] M. HASSAN and M. HAMADA, "A neural network recommendation approach for improving accuracy of multi-criteria collaborative filtering," *IEICE technical report*, vol. 117, no. 75, pp. 17–20, 2017.
- [8] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.
- [9] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-criteria recommender systems," in *Recommender systems handbook*. Springer, 2015, pp. 854–887.
- [10] H. Mohamed, L. Abdulsalam, and H. Mohammed, "Adaptive genetic algorithm for improving prediction accuracy of a multi-criteria recommender system," in *2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*. IEEE, 2018, pp. 79–86.
- [11] M. Hamada, N. B. Odu, and M. Hassan, "A fuzzy-based approach for modelling preferences of users in multi-criteria recommender systems," in *2018 IEEE 12th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*. IEEE, 2018, pp. 87–94.
- [12] D. Jannach, Z. Karakaya, and F. Gedikli, "Accuracy improvements for multi-criteria recommender systems," in *Proceedings of the 13th ACM conference on electronic commerce*, 2012, pp. 674–689.
- [13] M. Nilashi, O. bin Ibrahim, and N. Ithnin, "Hybrid recommendation approaches for multi-criteria collaborative filtering," *Expert Systems with Applications*, vol. 41, no. 8, pp. 3879–3900, 2014.
- [14] G. Adomavicius, N. Manouselis, and Y. Kwon, "Multi-criteria recommender systems," in *Recommender systems handbook*. Springer, 2011, pp. 769–803.
- [15] P. Cremonesi, F. Garzotto, and R. Turrin, "Investigating the persuasion potential of recommender systems from a quality perspective: An empirical study," *ACM Transactions on Interactive Intelligent Systems (TiIS)*, vol. 2, no. 2, pp. 1–41, 2012.
- [16] P. Cremonesi, F. Garzotto, S. Negro, A. V. Papadopoulos, and R. Turrin, "Looking for "good" recommendations: A comparative evaluation of recommender systems," in *IFIP Conference on Human-Computer Interaction*. Springer, 2011, pp. 152–168.
- [17] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 426–434.
- [18] F. Cardinaux, M. Enekl, F. Giron, T. Kemp, and S. Uhlich, "Method, system and artificial neural network," Feb. 18 2020, uS Patent 10,564,923.
- [19] M. W. Craven and J. W. Shavlik, "Learning symbolic rules using artificial neural networks," in *Proceedings of the Tenth International Conference on Machine Learning*, 2014, pp. 73–80.
- [20] O. Er, O. Cetin, M. S. Bascil, and F. Temurtas, "A comparative study on parkinson's disease diagnosis using neural networks and artificial immune system," *Journal of Medical Imaging and Health Informatics*, vol. 6, no. 1, pp. 264–268, 2016.
- [21] M. Tkáč and R. Verner, "Artificial neural networks in business: Two decades of research," *Applied Soft Computing*, vol. 38, pp. 788–804, 2016.
- [22] J. H. Rodríguez, M. J. R. Conde, and F. J. C. Fraile, "Artificial neural networks applications in computer aided diagnosis: system design and use as an educational tool," in *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2016, pp. 1201–1208.
- [23] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [24] B. Widrow and M. E. Hoff, "Adaptive switching circuits," Stanford Univ Ca Stanford Electronics Labs, Tech. Rep., 1960.
- [25] G. Adomavicius and Y. Kwon, "New recommendation techniques for multicriteria rating systems," *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 48–55, 2007.
- [26] C. C. Aggarwal, "Neighborhood-based collaborative filtering," in *Recommender systems*. Springer, 2016, pp. 29–70.
- [27] M. Hassan and M. Hamada, "Evaluating the performance of a neural network-based multi-criteria recommender system," *International Journal of Spatio-Temporal Data Science*, vol. 1, no. 1, pp. 54–69, 2019.
- [28] K. Lakiotaki, N. F. Matsatsinis, and A. Tsoukias, "Multicriteria user modeling in recommender systems," *IEEE Intelligent Systems*, vol. 26, no. 2, pp. 64–76, 2011.
- [29] M. Hassan and M. Hamada, "A computational model for improving the accuracy of multi-criteria recommender systems," in *2017 IEEE 11th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*. IEEE, 2017, pp. 114–119.
- [30] Y. Koren and J. Sill, "Collaborative filtering on ordinal user feedback," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.
- [31] D. Jannach, L. Lerche, F. Gedikli, and G. Bonnin, "What recommenders recommend—an analysis of accuracy, popularity, and sales diversity effects," in *International conference on user modeling, adaptation, and personalization*. Springer, 2013, pp. 25–37.
- [32] D. Jannach, F. Gedikli, Z. Karakaya, O. Juwig *et al.*, "Recommending hotels based on multi-dimensional customer ratings," in *ENTER*, 2012, pp. 320–331.
- [33] J. Fan and L. Xu, "A robust multi-criteria recommendation approach with preference-based similarity and support vector machine," in *International Symposium on Neural Networks*. Springer, 2013, pp. 385–394.
- [34] N. Sahoo, R. Krishnan, G. Duncan, and J. Callan, "Research note—the halo effect in multicomponent ratings and its implications for recommender systems: The case of yahoo! movies," *Information Systems Research*, vol. 23, no. 1, pp. 231–246, 2012.