A Comparison of BAT and Firefly Algorithm in Neighborhood based Collaborative Filtering

Hartatik[1], Bayu Permana Sejati[2]
Faculty of Computer Science
Universitas Amikom Yogyakarta
Yogyakarta, Indonesia

Hamdani Hamdani[3]*
Department of Informatics
Universitas Mulawarman
Samarinda, Indonesia

Andri Syafrianto[4]
Department of Informatics
STMIK El Rahma
Yogyakarta, Indonesia

***Abstract*—The recommender system is a knowledge-based filtering system that predicts the users' rating and preference for what they might desire. Simultaneously, the neighborhood method is a promising approach to perform predictions, resulting in a high accuracy based on the common items. This method, furthermore, could affect the resulting accuracy value because when each user provides limited data and sparsity, the accuracy of value might be narrow down as a consequence. In this research, we use the Swarm Intelligent (SI) technique in the recommender system to overcome this problem, whereby SI will train each feature to optimal weight. This technique's main objective is to form better groups of similar users and improve recommendations' accuracy. The intelligent swarm technique used to compare its accuracy to help provide recommendations is the Firefly and Bat Algorithm. The results show that the Firefly Algorithm has slightly better performance than the Bat Algorithm, with a difference in the mean absolute error of 0.02013333. The significance test using the independent t-test method states that no statistically significant difference between Bat and Firefly algorithm.**

***Keywords*—Bat algorithm; firefly algorithm; collaborative filtering; recommender system; swarm intelligent**

I. Introduction

The recommender system is research that is most popular in the business world. The recommender system is developed on several algorithms to find the best pattern from data and provide a recommendation by filtering a preference-based on the user's interests or needs [1], weighted update [2]. For example, one of the recommender system's prominent implementations is the amazon online store to offer similar books according to customer search history when visiting the online store.

Plenty of techniques can develop recommendation systems such as Content-Based Filtering, Collaborative Filtering, and Knowledge-Based Filtering. Content-Based Filtering, moreover, is a user modelling process in which users' interest is inferred from the items the user interacts with [3][4][5]. The items refer to usually textual, for example, email or web pages. In Content-based Filtering, the most defining features are used to model items and users. Meanwhile, the most discriminatory parts are identified and stored as vectors containing the components and their weights. The user model usually consists of user-item features. User models and recommendation candidates are compared to generate recommendations, for example, using the vector space model and the cosine similarity coefficient.

The second technique is Collaborative Filtering (CF). This technique will inform the user based on the feedback from other users who have relatively similar attributes [6][7]. The semblance of two users' tastes is calculated based on the history presented rating [8]. There are two CF approaches, namely the neighborhood-based method and the latent factor model - matrix factorization [8].

Neighborhood-based methods provide a study about the relationships between items or between users. The item approach is based on the user's preference for an object based on the same users' ranking of similar items [9]. Another case with the latent factor model - matrix factorization, which converts items and users to the same latent factor space [9]. The latent space is then used to explain the ranking by characterizing the product and user in terms of automatically generated factors from user feedback.

The neighborhood-based method is popular enough because of its simplicity, efficiency, and ability to produce accurate and personalized recommendations [8]. Neighborhood-based methods make predictions based on common items that help the accuracy value when each user provides limited data, and the spread, the resulting accuracy value will be small-scale [9]. Some researchers have tried adding techniques such as clustering [10] and intelligent swarm [11]. The addition of the SI uses the recommender technique to learn the optimal weight of each feature. Thus, it can form better groups of similar users and improve recommendations.

This research tried to combine IS techniques (BA and FA) with the neighborhood-based method to solve the sparsity problem and improve the accuracy of the recommendation system. First, IS methods are trained to get each feature's optimal weight, which is then used by the neighborhood-based method to get the rating prediction. At the end of the experiment, MAE and RMSE values will be obtained, which show the comparison of the accuracy of the BA and FA in providing recommendations.

II. Related Work

In the last few decades, metaheuristic techniques have experienced rapid development due to their increased search efficiency. Researchers have widely used the SI metaheuristic technique to find the most optimal solution search space

*Corresponding Author: hamdani@unmul.ac.id

phenomenon [11]. In System recommendations, the intelligent swarm is used to learn each feature's optimal weight to get a group of similar users who can be called active users [12].

The research uses weight updates for group decision-making that have similar parameters to be used by decision-makers (DMs). This model involves stakeholders who may have the same or different parameters in choosing parameters so that they can accommodate the interests of all DMs to obtain alternative decisions [2]. Furthermore, S. Ujjin and P. J. Bentley, [13] used the Particle Swarm Optimization (PSO) technique to generate a set of weights for user features and used a modified Euclidean function to generate recommendations. Their approach shows a considerable improvement over the Pearson algorithm compared to the Genetic algorithm.

Meanwhile, R. Katarya and O. P. Verma, [14] used K-Means to provide initial parameters for the PSO algorithm. The PSO algorithm itself is used to optimize Fuzzy C Means Clustering. Experiments conducted on the MovieLens Dataset show that there are several improvements from the existing method.

Research conducted by J. Sobecki, [15] compared several SI algorithms, namely Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Intelligent Weed Optimization (IWO), Bee Colony Optimization (BCO), and Bat Algorithm (BA) [16], in recommendations of student courses. Prediction accuracy shows a difference of only 0.1 between the five algorithms.

Tried to improve the CF-based recommendation system's quality using two Swarm Intelligence methods, namely BA and ABC [17]. This research aims to improve the quality of the traditional recommendation system. The authors use PCC to find similarities between users in the utility matrix and Swarm Intelligent methods to find the best weight of items to get good neighbors called active users. The data used in this study is the Jester dataset by Ken Goldberg, from AUTO Lab, UC Berkeley. This study indicates that BA has a 6.9% better quality than ABC, obtained a lower root mean squared error (RMSE) score than ABC, and higher precision, recall, and F1 score. In another research, [17]. Used BA to improve the recommendation system's quality [9]. The difference with the first research was in using the trust-aware matrix technique to create a utility matrix, which is then optimized - using BA. The data used in this study is a dataset from MovieLens, Epinions, CiaoDVD, and Filmtrust. To measure the performance of BA, the writer compared it with Particle Swarm Optimization (PSO), where BA obtained better results, both in terms of measuring Mean Squared Error (MAE) as much as 3.84% better and BA reaching 85.54% compared to PSO of 85.54% compared to PSO of 81.85%.

Another research combined Fuzzy C Means (FCM) and Bat Optimization to solve CF's sparsity and scalability problem [18]. BA serves to find the most optimal number of clusters from FCM. This study uses Movie Lens film rating data to be compared with the number of different neighbors and other clustering methods such as K-Means and SOM Clusters. This study indicates that the MAE score obtained by FCM and BA is smaller than other methods in the sense that BA can improve the quality of the clustering-based recommendation system.

Improving CF's performance using swarm intelligence can be done with several other swarm intelligence methods such as PSO. M. Wasid and V. Kant, introduced a new strategy in the recommendation system by combining it with Fuzzy Features or FPSO-CF [19]. This study aims to improve CF's accuracy, where PSO is used to find user weights and represent user features so that the authors use fuzzy sets more efficiently. Experiments were carried out using a movie lens dataset, with 60 films and 497 users. This study indicates that FPSO-CF has higher accuracy and lower error than Pearson CF, Fuzzy CF, and Fuzzy Genetic CF.

Research conducted by [20] using the Firefly Algorithm improves the quality of a collaborative filtering-based recommendation system. The MovieLens Dataset is used and collected by the GroupLens research project at the University of Minnesota [20]. The study results indicate that the proposed method significantly improves the recommendations' accuracy and improves the recommendations' prediction quality and performance.

The above studies prove that the combination of SI and CF techniques helps achieve better personalized recommendations for users. Therefore, continuing this work further, in this paper, we have tried to compare the Bat (BA) [21] algorithm and firefly [16], [22] in the calculation of feature weights and the measurement of the Pearson Correlation Coefficient to find the neighbors of active users. Finally, top-N recommendations were made for finding active users.

III. Proposed Method and Algorithm

A. Recommender System

A recommendation system can also be called a decision support system that can direct users to personalize items of interest or be liked by users. The recommendation system can provide users with inner directions by finding items that match user preferences [23]. The recommendation system's basic form works with two methods, specifically user-item interaction, such as rating or buying behavior, and attribute information about users and items such as profiles or search keywords. The first method is user-item interaction called collaborative filtering, while the second method is called content-based [24]. The content-based method works by checking the attributes of the recommended item. For instance, if Netflix users have watched many movies with the cowboy genre, then the next film to recommend is a cowboy. Meanwhile, collaborative filtering recommends items based on similarities between users or between items. In other words, items recommended to users are likes by other similar users [25].

B. Collaborative Filtering

Collaborative filtering (CF) is the most popular method of finding recommendations. CF works based on predictions and ratings or the behavior of other users in the system. The fundamentals behind this method are opinions of other users can be selected and aggregated in such a way as to provide a reasonable prediction of the preferences of the active user. It

can be intuitively assumed that if users agree about some items' quality or relevance, they will probably agree about other items. Another example, if a group of users likes the same things as Mary, then Mary is likely to like the things they like [26]. This approach is based on a simple idea; users will prefer items recommended by others who have something in the standard [27].

An example is like giving recommendations for places to eat that we like to our friends. What distinguishes other approaches is that CF only considers the utility matrix. CF is a stand-alone method because it does not know the item except the user [27].

C. Neighborhood-Based Collaborative Filtering

The neighborhood-based method, or it can be called memory-based, is the earliest method developed for collaborative filtering. This method is based on the fact that the users form a similar rating pattern on similar items [24], unlike the model-based approach, which is difficult to provide recommendations for films that do not match the film's information. As a result, a model-based recommendation system recommends films that are not according to user preferences [28]. Meanwhile, the neighborhood-based method can address those weaknesses [28].

D. User-Based Collaborative Filtering

This method is designed to find similarities from users who have similar rating patterns to other users and rated the item in question [27]. For instance, if Alice and Bob have rated films the same way in the past, other users could use Alice's ratings in film A to predict Bob's non-rated ratings on film A. In general, most similar users to Bob can be used to make ranking predictions for Bob. The similarity function is calculated between rating rows to find similar users. Pearson Correlation Coefficient and Cosine Similarity could be used to calculate similarity [29].

Pearson Correlation Coefficient (PCC) is a correlation search method developed by Karl Pearson. Meanwhile, correlation is a measurement technique that determines how close the relationship between the two variables is. The measurement results of the PCC can be either positive or negative. A positive relationship shows that the two variables have a parallel (linear) increase in value. Meanwhile, a negative relationship shows that the two variables have a parallel (linear) decrease in value. A parallel is an increase or decrease in value that follows between two variables. Equation (1) is used to calculate the similarity between users or items [24].

$$sim(u,v) = \frac{\sum_{i \in C}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in C}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in C}(r_{v,i} - \bar{r}_v)^2}} \quad (1)$$

Where, $sim(u,v)$ is the similarity value between user u and user v, $r_{u,i}$ and $r_{v,i}$ are the values of user u and user v against the item, and $\bar{r}_u$ and $\bar{r}_v$ are the mean values of user u and user v against the item.

E. BAT Algorithm

Bat Algorithm (BA) is a metaheuristic method based on swarm intelligence proposed by Xin-She [30] and BA the proposed work is formulated as a non-linear optimization problem [16]. BA was inspired by small bats (microbats) that use echolocation/sonar to detect prey [12]. Most of the bat species are insectivorous. Besides using echolocation to catch food, bats also use it to avoid obstacles and find perches in the dark. Echolocation works by emitting sound, and then when the sound hits the object, the sound will return to the source [30]. Bats fly randomly and have position x_i, velocity v_i, frequency f_i, pulse rate r_i and loudness A_i, to find the optimal solution. The bat moves each iteration to come up with a new solution that may be more optimal [12]. The following are the steps of the BA algorithm. A New Discretization of Bat algorithm in Equation (2), (3), and (4).

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \quad (2)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_{best})f_i^{(t)} \quad (3)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (4)$$

Where, β is the random vector between $[0,1]$, and $t+1$ is a number of iteration.

Update the best position of *x* best using (5).

$$x_{new} = x_{old} + \in A^t \quad (5)$$

Where, $\in$ determines the measurement and helps in getting the convergence velocity from BA.

Save the best solutions. This step save the best current solution as well as update the loudness A_i and pulse rate r_i Equation (6) and (7).

$$A_i^{t+1} = \alpha A_i^t \quad (6)$$

$$r_i^{t+1} = r_1^0(1 - e^{(-\gamma)}) \quad (7)$$

F. Firefly Algorithm

Firefly algorithm is a SI method inspired by nature, namely the firefly lifestyle. Xin-She developed this method in 2007 [31] [24]. As with other SI methods, the Firefly algorithm aims to solve optimization problems like the firefly lifestyle. Fireflies produce short, rhythmic lights that are different from one another. Firefly populations each have their lighting characteristics. In this method, the firefly is compared, and the less bright fireflies move towards, the brighter fireflies [32]. The firefly chosen as the most attractive is the optimal response to the problem [20], dynamic adaptive [22], and hybrid firefly algorithm [16].

Fireflies' attraction is their brightness, the light intensity at a certain distance from the light source as an inverse-square law. That means that the light intensity will decrease with increasing distance. The fireflies' attractiveness is directly

proportional to the nearby fireflies' light intensity and is measured depending on the length of the fireflies from one another. The equation of variation of attractiveness β with distance r is defined (8) [33].

$$\beta = \beta_0 . e^{-\gamma . r^2} \tag{8}$$

Where β_0 is the value of the firefly attractiveness when $r = 0$ and γ is the light absorption coefficient. The position shift of the firefly i attracted to the firefly j is determined by (9) [20].

$$x_i = x_i + \beta_0 . e^{-\gamma . r_{i,j}^2 (x_j - x_i)} + \alpha . (rand - 0.5) \tag{9}$$

G. Rating Prediction

Collaborative filtering aims to predict an empty rating of the utility matrix. There are various methods to do this, and one of those is the k-Nearest Neighbor (k-NN). k-NN approach is one of the data mining methods considered among the top 10 data mining techniques [20]. The k-NN approach uses the well-known concept of *Cicero pares cum paribus facility congregant* (birds of a feather flock together or equivalent to equals easily associated). It attempts to identify an unknown sample based on the known classification of its neighbors. k-NN is used to predict the consumer's rating, following the k-NN method [14](10).

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} sim(u,v) . r_{vi}}{\sum_{v \in N_i^k(u)} sim(u,v)} \tag{10}$$

Where $sim(u,v)$ is the similarity value between user u and user v, $\hat{r}_{ui}$ is predicted values of user u for the items i and r_{vi} are the values of user v for the item i.

H. Evaluation

The recommendation engine's output is also calculated as a rating Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE), calculating the delta between real and expected ratings. For more significant errors, the RMSE metric is used, while MAE benefits from simple understanding.

Whenever a new known rating is captured in the system, the deployed recommendation engine's accuracy must be continuously measured. The new actual rating is paired against the recommender's previous prediction [27]. MAE and RMSE formula can be seen at (11) and (12), respectively.

$$MAE = \frac{\sum_{i=1}^{n} | p_i - a_i |}{n} \tag{11}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} | p_i - a_i |^2}{n}} \tag{12}$$

To determine whether the differences between the two proposed models are statistically significant or not, we applied an independent t-test for an unpaired sample [16], [20]. The Independent sample t-test is defined by (13).

$$t = \frac{(\bar{x}_a - \bar{x}_b)}{\sqrt{\frac{S_a^{\,2}}{n_a} + \frac{S_b^{\,2}}{n_b}}} \tag{13}$$

This study used data from an open dataset at movielens.org, which contained 100,000 users who rated the film [27]. The distribution of the dataset can be seen in Table I.

TABLE I. DATA DISTRIBUTION IN DATASET

Data	Ratings	Users	Movies
Training	72456	943	1639
Testing	18114		
Totals	90570		

These stages begin with data collection from the MovieLens 100K site. A preprocessing process consists of features extraction to obtain movies id, users id, ratings, and timestamp. Furthermore, the data is transformed into a utility matrix to look for similarities using the Pearson Correlation Coefficient (PCC). The similarity matrix results are optimized using SI to obtain each user's weight. After that, data is divided into training data and testing data. The rating prediction was calculated using k-NN method by considering the user weight. The steps above could see in Fig. 1.

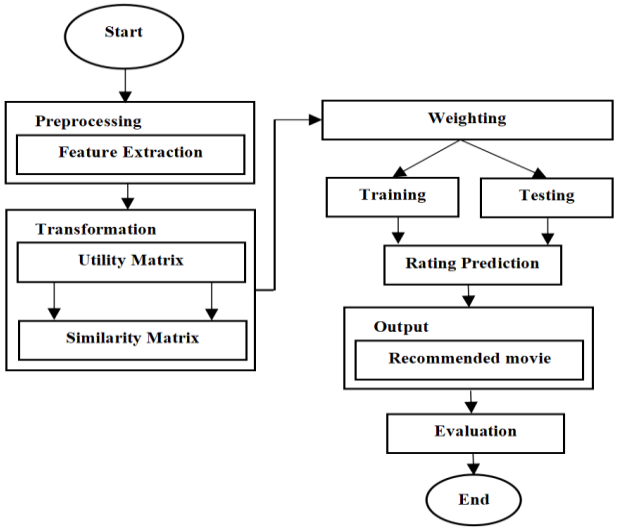

Fig. 1. Our Proposed Approach.

IV. PROPOSED METHOD

This research used data in the MovieLens 100K dataset, consisting of 943 users and 1680 films with a spread rate of 99.1%. The first stage is the data pre-processing. The purpose of data pre-processing is to extract features from the rating data in order to get useful features before making predictions. The next step is to carry out each user's weighting process using the

BA and Firefly algorithm. The BA and Firefly algorithm do the process repeatedly to get the best solution.

In the BA implementation, the initial alpha value is 0.95, and the gamma is 0.05. With a bat's population is 50, times the number of dimensions in the similarity matrix is 943, and a generation is initiated by multiplying population and dimension (50*943). Different from BA, there is no initial dimension at parameter initiation in the Firefly algorithm.

Parameter alpha in the Firefly algorithm was set at 1.0, beta at 1, and gamma at 0.95. The generation and the number of dimensions in the Firefly similarity matrix algorithm were set the same with BA in 943. Furthermore the sis initiation parameters can be seen in Table II.

TABLE II. PARAMETER INITIATION OF EACH SIS

Parameters	Bat Algorithm	Firefly Algorithm
Dimension	943	-
Population	50	943
Generation	50*943	943
Alpha	0.95	1.0
Beta	-	1
Gamma	0.95	0.01

When a new active user enters the system whose suggestions are to be made, BA and Firefly iteratively optimize the feature (item) weight by searching in the search space dimension m (item total). The results of this calculation can be seen in Table III.

TABLE III. WEIGHTING RESULT COMPARISON OF BAT AND FIREFLY

Methods	Max Weights	Min Weights	Avg Weights	Std Weights
BA	4.600450	-4.731380	-0.048353	1.695925
Firefly	0.257814	0.001194	0.006614	0.017754

After the feature weighting has been studied, active user neighbours are then formed using PCC to measure the distance between two similar users using (1). PCC is modified by multiplying the actual rank by the weight calculated by the BA and Firefly algorithms.

V. RESULT AND DISCUSSION

This section describes the results of the experiments that have been carried out and the work's findings. Several previous studies have stated that adding SI to traditional methods such as PCC can provide better predictive values. Therefore, in this experiment, we also add the conventional method's experimental results without SI modification.

An active user that was generated with BA and Firefly picked up from the highest weight. In this sense, we compare the error calculation results between the models trained using 50,100 and 200 active users in this test. The mean absolute error and RMSE values obtained using PCC, BA, and Firefly algorithms are shown in Table IV. The results show the active users generated by SI can improve the recommendation system's quality by 20%.

TABLE IV. RESULT OBTAINED USING BA AND FIREFLY ON MOVIELENS DATA SET

Methods	Active Users	MAE	RMSE	Time (s)
PCC	50	0.7331	1.0111	2.17
	100	0.7438	1.0392	5.25
	200	0.7231	1.0014	20.5
BA	50	0.6033	0.8669	10.8
	100	0.6287	0.8839	4.53
	200	0.6506	0.9010	17.7
Firefly	50	0.6010	0.8680	6.69
	100	0.6049	0.8695	4.9
	200	0.6163	0.8727	11.7

Fig. 2 and Fig. 3 graphically show the MAE variation for different numbers of active users randomly selected using both algorithms.

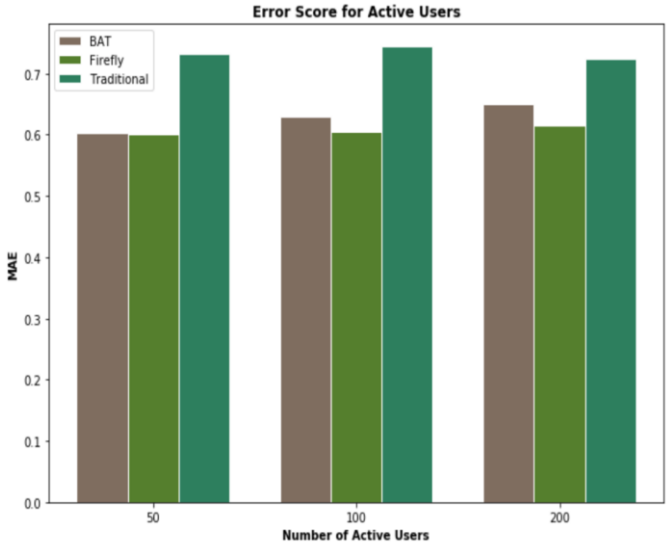

Fig. 2. Comparison MAE of user Active and non user Active.

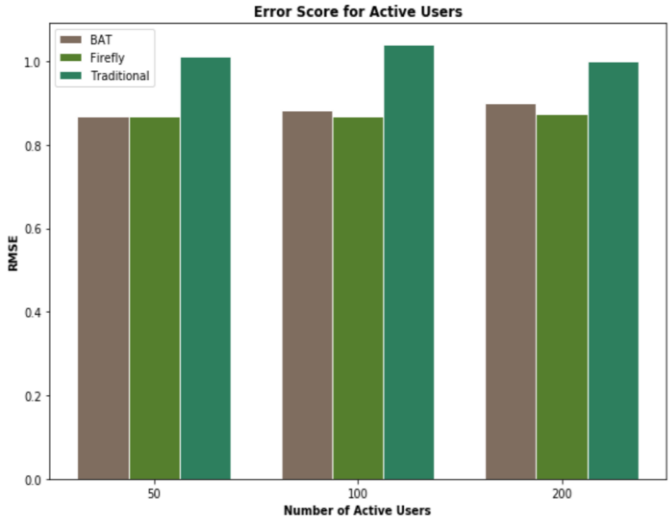

Fig. 3. Comparison RMSE of user Active and non user Active.

Thereafter, based on the experiment results using the active user, the recommender system that uses a certain optimal swarm intelligence weight has decreased in the error calculation. For MAE and RMSE scores, the Firefly Algorithm has a slightly better score than the BA. Meanwhile, the weightless recommendation system shows that the MAE and RMSE results are still relatively high. It concludes using the weightiest users affects the quality of the recommendation system. However, the execution time on the weightless recommendation system has a shorter time than the weighted ones. It could see in Fig. 2 and 3 sections 50 users that there is a significant difference in execution time, although, at 200 users, the time from the traditional recommendation system has increased. It concludes the more users there are, the more execution time is needed and does not affect the increase or decrease in the error calculation result.

In this research, we applied an independent sample t-test to saw the significance of the BAT and Firefly algorithm. The results of the independent t-test can be seen in Table V.

TABLE V. THE RESULT OF INDEPENDENT T-TEST ON MOVIELENS DATA SET

User Active	**MAE**	
	BAT	**Firefly**
50	0.6033	0.601
100	0.6287	0.6049
200	0.6506	0.6163
Mean	0.62753333	0.6074
Standard deviation	0.023671572	0.007950472
α	0.05	
Degree of freedom	4	
t-Value calculated using (3)	1.140233717	

Tests to determine the null hypothesis are $H_0 : \mu_1 = \mu_2$. it means that there is no difference between BAT (μ_1) and Firefly algorithm (μ_2). Meanwhile, the alternative hypothesis was $H_1 : \mu_1 \neq \mu_2$ which is explains there is a difference between BAT (μ_1) and Firefly algorithm (μ_2).

Refer to the test results in Table IV, the t-value = 1.140233717 and df = 4. Using the Two Tails T Distribution Table, the t-table value = 2.776 for the error rate of 5% and 4.604 for the error rate of 1%. Since the t-value is in the area of acceptance for the H_0 hypothesis at an error level of 5% or 1%, it can be concluded that there is no statistically significant difference between BAT and Firefly algorithms.

VI. CONCLUSION

The neighborhood method is a promising approach to perform predictions, resulting in a high accuracy based on the common items. This method, furthermore, could affect the resulting accuracy value because when each user provides limited data and sparsity, the accuracy of value might be narrow down as a consequence. The recommender system can use SI technique to overcome this problem. In this work, we proposed an approach to giving weights to the items in the user-item rating matrix to find the active user's better neighborhood using the BA and firefly algorithm. This method helped provide personalized recommendations to all users as it generated a different set of weights for each user.

The MAE value shows that SI can improve the recommender system's quality by 20%. For MAE scores, the Firefly Algorithm has a slightly better score than the BA. BA had MAE score 0.6033, 0.6287 and 0.6506 for k=50, 100 and 200 respectively. Meanwhile, Firefly Algorithm had an MAE score of 0.601, 0.6049, and 0.6163 for k=50, 100, and 200, respectively.

The test result using the independent t-test method got t-value=1.140233717. Since the t-value accepted the null hypothesis at an error level of 5% or 1%, the conclusion was there is no statistically significant difference between BA and Firefly algorithms.

VII. FUTURE WORK

In our future work, we would like to add implicit feedback as an additional feature in IS. This thinking stems from the assumption that users' preferences and preferences may change over time. Thus, the addition of the time parameter may make the resulting prediction more relevant and follow the user's interests.

In addition, it is necessary to think of a way to reduce the scalability of the recommendation system in the future. It is a typical case that adding IS slows down the recommendation calculation process. Therefore, it is necessary to think of a way to reduce this problem. The application of the clustering method at the beginning is probably one way that can do.

REFERENCES

[1] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” Egypt. Informatics J., vol. 16, no. 3, pp. 261–273, 2015, doi: 10.1016/j.eij.2015.06.005.

[2] H. Hamdani, R. Wardoyo, and K. Mustofa, “A method of weight update in group decision-making to accommodate the interests of all the decision makers,” Int. J. Intell. Syst. Appl., vol. 9, no. 8, 2017, doi: 10.5815/ijisa.2017.08.01.

[3] Y. Afoudi, M. Lazaar, and M. Al Achhab, “Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network,” Simul. Model. Pract. Theory, vol. 113, no. July, p. 102375, 2021, doi: 10.1016/j.simpat.2021.102375.

[4] B. Walek, V. Fojtik, “A hybrid recommender system for recommending relevant movies using an expert system”, Expert Systems with Applications 158, 113452, 2020.

[5] E. Faizal and H. Hamdani, “Weighted Minkowski similarity method with CBR for diagnosing cardiovascular disease,” Int. J. Adv. Comput. Sci. Appl., vol. 9, no. 12, 2018, doi: 10.14569/IJACSA.2018.091244.

[6] H. Khojamli and J. Razmara, “Survey of similarity functions on neighborhood-based collaborative filtering,” Expert Syst. Appl., vol. 185, no. June 2020, p. 115482, 2021, doi: 10.1016/j.eswa.2021.115482.

[7] X. Yuan, L. Han, S. Qian, G. Xu, H. Yan, “Singular value decomposition based recommendation using imputed data”, Knowledge-Based Systems, Volume 163, 1 January 2019, Pages 485-494.

[8] R. Zhang, Q. D. Liu, Chun-Gui, J. X. Wei, and Huiyi-Ma, “Collaborative Filtering for Recommender Systems,” Proc. - 2014 2nd Int. Conf. Adv. Cloud Big Data, CBD 2014, pp. 301–308, 2015, doi: 10.1109/CBD.2014.47.

[9] F. Ricci, L. Rokach, B. Shapira, P. B. Kantor, and F. Ricci, Recommender Systems Handbook. 2011.

[10] L. Xiaojun, "An improved clustering-based collaborative filtering recommendation algorithm," Cluster Comput., vol. 20, no. 2, pp. 1281–1288, 2017, doi: 10.1007/s10586-017-0807-6.

[11] G. Guo, J. Zhang, and N. Yorke-Smith, "A Novel Recommendation Model Regularized with User Trust and Item Ratings," IEEE Trans. Knowl. Data Eng., vol. 28, no. 7, pp. 1607–1620, 2016, doi: 10.1109/TKDE.2016.2528249.

[12] S. Yadav, Vikesh, Shreyam, and S. Nagpal, "An Improved Collaborative Filtering Based Recommender System using Bat Algorithm," Procedia Comput. Sci., vol. 132, pp. 1795–1803, 2018, doi: 10.1016/j.procs.2018.05.155.

[13] S. Ujjin and P. J. Bentley, "Particle swarm optimization recommender system," 2003 IEEE Swarm Intell. Symp. SIS 2003 - Proc., pp. 124–131, 2003, doi: 10.1109/SIS.2003.1202257.

[14] R. Katarya and O. P. Verma, "A collaborative recommender system enhanced with particle swarm optimization technique," Multimed. Tools Appl., vol. 75, no. 15, pp. 9225–9239, 2016, doi: 10.1007/s11042-016-3481-4.

[15] J. Sobecki, "Comparison of selected swarm intelligence algorithms in student courses recommendation application," Int. J. Softw. Eng. Knowl. Eng., vol. 24, no. 1, pp. 91–109, 2014, doi: 10.1142/S0218194014500041.

[16] Z. Cheng, H. Song, J. Wang, H. Zhang, T. Chang, and M. Zhang, "Hybrid firefly algorithm with grouping attraction for constrained optimization problem," Knowledge-Based Syst., vol. 220, p. 106937, 2021, doi: 10.1016/j.knosys.2021.106937.

[17] S. Yadav, V. Kumar, S. Sinha, and S. Nagpal, "Trust aware recommender system using swarm intelligence," J. Comput. Sci., vol. 28, pp. 180–192, 2018, doi: 10.1016/j.jocs.2018.09.007.

[18] V. Vellaichamy and V. Kalimuthu, "Hybrid collaborative movie recommender system using clustering and bat optimization," Int. J. Intell. Eng. Syst., vol. 10, no. 5, pp. 38–47, 2017, doi: 10.22266/ijies2017.1031.05.

[19] M. Wasid and V. Kant, "A Particle Swarm Approach to Collaborative Filtering based Recommender Systems through Fuzzy Features," Procedia Comput. Sci., vol. 54, pp. 440–448, 2015, doi: 10.1016/j.procs.2015.06.051.

[20] F. Shomalnasab, M. Sadeghzadeh, and M. Esmaeilpour, "An Optimal Similarity Measure for Collaborative Filtering Using Firefly Algorithm," J. Adv. Comput. Res., vol. 5, no. August, pp. 101–111, 2015.

[21] Y. Saji and M. Barkatou, "A discrete bat algorithm based on Lévy flights for Euclidean traveling salesman problem," Expert Syst. Appl., vol. 172, no. January, p. 114639, 2021, doi: 10.1016/j.eswa.2021.114639.

[22] J. Liu, Y. Mao, X. Liu, and Y. Li, "A dynamic adaptive firefly algorithm with globally orientation," Math. Comput. Simul., vol. 174, pp. 76–101, 2020, doi: 10.1016/j.matcom.2020.02.020.

[23] A. Felfernig, L. Boratto, and W. Gan, "Group Recommender Systems," Springer, 2018, pp. 1–176.

[24] L. Luo, H. Xie, Y. Rao, F.L. Wang, "Personalized Recommendation by Matrix Co-Factorization with Tags and Time Information", Expert Systems With Applications, Volume 119, 1 April 2019, Pages 311-321.

[25] J. Leskovec, A. Rajaraman, and J. D. Ullman, "Mining of Massive Datasets," Min. Massive Datasets, 2020, doi: 10.1017/9781108684163.

[26] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative filtering recommender systems," Found. Trends Human-Computer Interact., vol. 4, no. 2, pp. 81–173, 2010, doi: 10.1561/1100000009.

[27] V. Kotu and B. Deshpande, Recommendation Engines. 2019.

[28] C. Desrosiers and G. Karypis, "Recommender Systems Handbook," Recomm. Syst. Handb., pp. 107–108, 2011, doi: 10.1007/978-0-387-85820-3.

[29] Z. Tan and L. He, "An Efficient Similarity Measure for User-Based Collaborative Filtering Recommender Systems Inspired by the Physical Resonance Principle," IEEE Access, vol. 5, pp. 27211–27228, 2017, doi: 10.1109/ACCESS.2017.2778424.

[30] X. S. Yang, "A new metaheuristic Bat-inspired Algorithm," Stud. Comput. Intell., vol. 284, pp. 65–74, 2010, doi: 10.1007/978-3-642-12538-6_6.

[31] X. Yang, "Metaheuristic Optimization : Nature-Inspired," Artif. Intell., Evol. Comput. Metaheuristics, pp. 405–420, 2013.

[32] H. Wang et al., "Firefly algorithm with neighborhood attraction," Inf. Sci. (Ny)., vol. 382–383, pp. 374–387, 2017, doi: 10.1016/j.ins.2016.12.024.

[33] A. B. Downey, "Think Stats: Probability and Statistics for Programmers," Psychol. Bull., vol. 70, no. 2, p. 140, 2011.