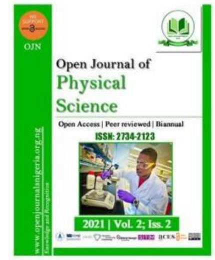
Open Journal of Physical Science (OJPS)
ISSN: 2734-2123
Article Details:
DOI: 10.52417/ojps.v6i1.842
Article Ref. No.: OJPS0601004-770
Volume: 6; Issue: 1, Pages: 48 - 62 (2025)
Accepted Date: 10th April, 2025

Open Journals Nigeria (OJN)
Open Access | Bi-annual | Peer-reviewed
www.openjournalsnigeria.org.ng
editorial@openjournalsnigeria.org.ng



RESEARCH ARTICLE

OJPS0601004-770

RECOMMENDATION MODELS FOR MENTORSHIP MATCHING: A COMPARATIVE STUDY OF PARTICLE SWARM OPTIMIZATION AND CUCKOO SEARCH ALGORITHMS.

Oguntuase, R. A.

Department of Computer Science, School of Computing, The Federal University of Technology, P.M.B. 704, Akure, Nigeria

***Corresponding Author Email:** rianatoguntuase@gmail.com*

ABSTRACT

Effective mentorship is vital for personal and professional growth, particularly in academic and professional settings. However, finding the right mentor from a large number of academic researchers available today can be a challenging task, particularly for newcomers to the field or for research institutions seeking to facilitate mentorship matches. Scholarly recommender systems (SRSs) have been identified as efficient tools in academic and research settings, but they also pose a significant challenge due to their high-dimensional search spaces, a challenge that metaheuristic algorithms have emerged to tackle with efficiency. This approach leverages profile and publication data from the Academic Family Tree (AFT) database and employs Particle Swarm Optimization (PSO) and Cuckoo Search (CS) algorithms to optimize mentorship matching. Data mining methodology consisting of data acquisition, pre-processing, training, and testing was used in this study. Experimental results revealed superior performance, with PSO achieving precision, recall, and accuracy of 1.00, alongside a mean reciprocal rank (MRR) of 0.80. Notably, PSO outperformed CS, which yielded a precision of 0.94, recall of 0.83, accuracy of 0.90, and an MRR of 0.80 at 10 recommendations. These findings underscore the potential of PSO in developing reliable mentorship matching systems.

Keywords: *Academic Mentorship, CS Algorithm, Machine Learning, Metaheuristic Algorithms, PSO, Scholarly Recommender Systems, TF-IDF.*

INTRODUCTION

Scholarly recommender systems (SRSs) have emerged as a vital tool to alleviate the information overload faced by academic researchers (Wang *et al*., 2019; Zhang *et al*., 2023). Typically, SRSs focus on recommending useful items to support the needs of scholars, researchers, and academics. These systems have proven effective in recommending various scholarly items, such as publications (Ghosal *et al*., 2019; Magara *et al.*, 2018), publication venues (Adebayo and Ojokoh, 2024; Alshareef *et al*., 2019), and collaborators (Zhu and Yaseen, 2022; Averchenkova *et al.*, 2020). Mentorship is a vital component of collaborator recommendations, as it enables the formation of meaningful relationships that foster growth, learning, and success. Effective mentorship can bridge the gap between junior and senior professionals, supporting the transmission of expertise, development of talents, and acceleration of career growth. In academic and professional settings, effective mentorship is crucial for fostering personal and professional growth. A good mentor can provide valuable guidance, support, and networking opportunities, leading to improved career outcomes and increased job satisfaction (Oguntuase *et al*., 2024). However, finding the right mentor can be a challenging task, particularly for newcomers to the field or for research institutions seeking to facilitate mentorship matches. This is where Mentorship Matching Recommender Systems (MMRSs) come into play. MMRSs aim to bridge the gap between mentors and mentees by providing personalized recommendations. These systems utilize a variety of data sources, such as user profiles, expertise, interests, and preferences, to suggest suitable mentor-mentee pairings. By facilitating meaningful connections, MMRSs have the potential to enhance career development, improve job satisfaction, and promote a culture of knowledge sharing and collaboration. However, SRSs pose a significant challenge due to their high-dimensional search spaces, stemming from the vast number of users, items (such as publications and experts), and features (such as titles and keywords) involved. Effectively navigating these complex spaces is crucial for optimal recommendations.

Fortunately, metaheuristic algorithms have emerged as a highly efficient approach to tackle this challenge (Gad, 2022), offering a promising solution for optimizing SRSs. This study focuses on the development and comparison of two optimized recommendation models for mentorship matching, utilizing two popular metaheuristic algorithms, namely Particle Swarm Optimization (PSO) and Cuckoo Search (CS) algorithms. PSO, inspired by the collective behavior of bird flocks, has been effectively utilized in solving a range of optimization problems (Oguntuase, 2024, Cao *et al.,* 2019). CS, on the other hand, mimics the brood parasitism of cuckoo birds, utilizing Levy flights to efficiently search for optimal solutions (Yang and Deb, 2009). The analysis of these algorithms aims to identify the most effective approach for facilitating successful mentor-mentee relationships. This research aims to inform the design of optimized mentorship matching systems, tailored to individual needs, which will in turn enhance research productivity, boost career fulfillment, and foster a more vibrant and collaborative academic environment.

The rapid growth of academic publications and researchers has created an overwhelming information landscape, making it challenging for scholars to discover relevant collaborators, mentors, or peers (Wang *et al.*, 2019). Scholarly recommender systems seek to address this challenge by providing tailored suggestions that cater to individual needs and preferences. However, these systems often struggle with large search spaces, complex relationships, and dynamic user preferences. Traditional Collaborative Filtering (CF) (Zhu and Yaseen, 2022), Content-Based Filtering (CBF) (Pradhan and Pal, 2020), and hybrid (Husain *et al.*, 2021; Xu *et al*., 2019) approaches have been widely adopted in scholarly recommender systems (Zhang *et al.*, 2023). Nevertheless, these methods have inherent limitations when dealing with large-scale scholarly data. CF-based approaches rely on user-item interactions, which can be sparse in academic networks, leading to cold-start problems and suboptimal recommendations. CBF-based approaches, on the other hand, focus on item attributes but may fail to capture complex relationships and contextual information.

To tackle these challenges, metaheuristic algorithms have emerged as a viable solution, leveraging inspiration from natural phenomena and evolutionary processes to efficiently navigate complex search spaces (Kennedy and Eberhart, 1995; Yang and Deb, 2009). By integrating metaheuristic algorithms, scholarly recommender systems can transcend the limitations of traditional CF and CBF approaches, yielding more precise and personalized recommendations. A review of the current landscape of scholarly recommender systems, particularly in collaborator recommendation, underscores the necessity of metaheuristic algorithms in addressing the complexities of large-scale academic data.

The studies reviewed include Zhu and Yaseen (2022) who developed a CF research collaborator recommender system using Graph Neural Networks (GraphSAGE and Temporal Graph Networks). The system captures temporal interactions to predict future collaborations. Husain *et al*. (2021) developed a hybrid collaborator selection model for finding experts in Malaysian research universities. It considers factors like collaborators' profiles, publications, social/academic networks, human capital, social capital, and cultural capital to facilitate informed collaborator selection. Pradhan and Pal (2020) developed DRACoR, a CBF collaborator recommender system that suggests potential collaborators based on similar research interests and social accessibility. DRACoR uses publication metadata, topic modeling, and Doc2Vec to extract feature vectors from abstracts and titles, and cosine similarity to weigh author-to-author connections. A major challenge faced by these approaches is their inability to optimally navigate the high-dimensional search spaces inherent in scholarly recommender systems, characterized by sparse user-item interactions, cold-start problems, and scalability issues. Moreover, these approaches struggle to incorporate contextual information, leading to suboptimal recommendations and reduced system performance.

MATERIALS AND METHOD

This section outlines the methodology employed to develop the two proposed recommendation models based on PSO and CS algorithms. The architecture design of the proposed models is illustrated in Figure 1.

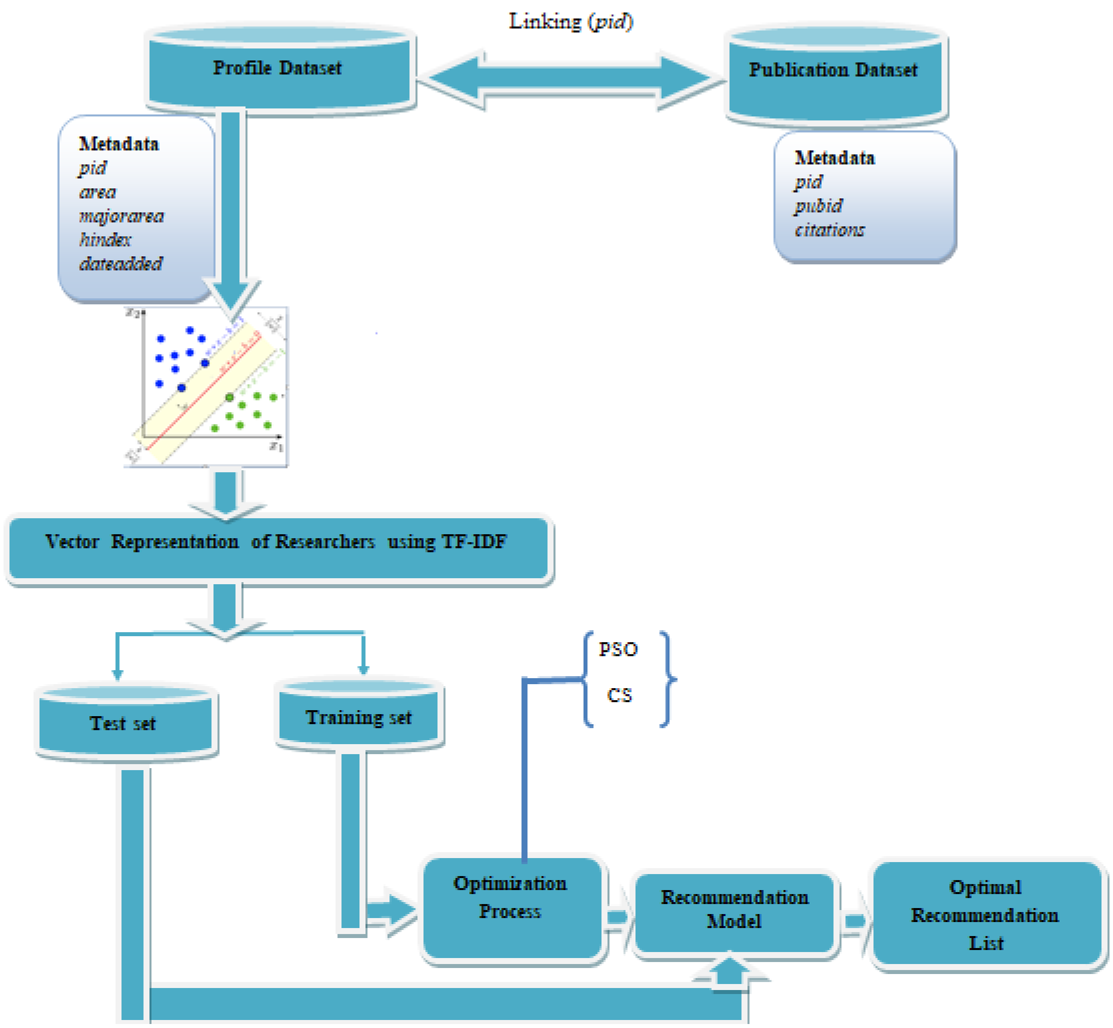


Figure 1: Architecture Design of the Optimized Models (Adapted from Oguntuase, 2024)

DATA DESCRIPTION AND PRE-PROCESSING

This study leverages two extensive datasets from the Academic Family Tree database, comprising profile and publication information. The datasets offer a comprehensive foundation for examining academic mentorship, with rigorous feature selection applied to optimize model performance. After the feature selection process, the profile dataset consists of 807,230 instances with five features, while the publication dataset contains 15,401,889 instances with three features. Integrating the datasets via a common identifier (pid) yielded a unified dataset with 15,371,421 instances and seven features.

Data quality assurance involved eliminating rows with missing data, resulting in a cleaned dataset of 4,306,451 instances. Further data aggregation was carried out, involving the counting of each researcher's publications, summing citations, selecting the minimum value for date-added, and removing duplicate data for h-index. The resulting dataset underwent feature renaming for clarity, with pubid, citations, h-index, and date-added being replaced with more descriptive names: num_publications, total_citations, h_index, and earlier_year, respectively. A classification process introduced a 'label' column, categorizing researchers as experienced or young based on publications, citations, h-index, and research experience. Data normalization using Min-Max Scaler facilitated modeling. The features "area" and "majorarea" were merged into "research_areas," which underwent TF-IDF transformation to generate numerical vectors. The vectorized dataset was then split into training and testing sets (80-20) for evaluation.

ALGORITHM IMPLEMENTATION

During recommendation optimization, each proposed model based on metaheuristic algorithms was trained on the labeled and vectorized data obtained from the output of the data preprocessing. The two metaheuristic-based algorithms, PSO and CS were employed to enhance and optimize the recommendation process. Each of the proposed recommendation models performs two tasks, namely recommending mentors to mentees and recommending mentees to mentors.

Fitness Function

The key to solving an optimization problem lies in formulating an appropriate fitness function that accurately captures the problem's requirements. The fitness function optimizes the population's search for the best recommendations. This work utilizes Average Precision (AP) as the fitness function. By using AP as the fitness function, the mentorship matching algorithm is optimized to maximize the number of successful matches. This optimization process involves defining a scoring function to calculate a similarity score between each mentee-mentor pair. The scoring function is then used to generate a ranked list of potential mentors/mentees for each mentee/mentor. The AP of the ranked list is calculated for each mentee/mentor, considering the top-N recommended mentors/mentees. Finally, the AP scores are used as the fitness function to assess the effectiveness of the mentorship matches and optimize the models using PSO and CS algorithms. By maximizing the AP scores, the quality of the mentorship matches is improved, increasing the number of successful matches.

PSO Algorithm

The PSO algorithm is modeled after the social behavior of certain animals, including bees, birds, and fish, where individuals interact, learn, and cooperate to achieve common goals (Shami *et al*., 2022, Gad, 2022). Each particle in PSO corresponds to a candidate solution for the optimization problem. The proposed PSO-based recommendation model comprises the following steps:

Step 1: Initialization **-** A swarm (population) of particles (researchers) is generated from the dataset. The PSO algorithm commences with two crucial initialization phases as common to metaheuristic algorithms, namely parameters initialization and population initialization. In the parameters initialization phase, algorithm parameters – including population size, dimension, inertia weight, cognitive coefficient, social coefficient, maximum number of iterations, and so on, are defined. The purpose of these parameters is to control and fine-tune the algorithm's behaviour to optimize its performance and achieve better solutions. In the population initialization phase, particles are randomly distributed across the search space. Particles are randomly assigned initial positions (mentor-mentee pairs) and velocities (directions for exploring potential pairs) using equations (1) and (2).

$$x_i^{(t)} = x_{min} + r(x_{max} - x_{min}) \quad (1)$$

$$v_i^{(t)} = v_{min} + r(v_{max} - v_{min}) \quad (2)$$

Where r represents a randomly generated number between 0.0 and 1.0, $x_i^{(t)}$ and $v_i^{(t)}$ are the initial position and initial velocity respectively

Step 2: Fitness Evaluation - The fitness function assesses the suitability of each particle (researcher) for mentor-mentee matching based on their expertise. The fitness function assigns a score to each researcher, reflecting their potential for effective mentorship. The researcher with the highest fitness score is deemed the most suitable candidate within the swarm.

Step 3: Generation of New Solutions **-** Based on the fitness evaluation, each particle updates its velocity to move towards better positions (more suitable mentor-mentee pairs), and a new set of particles is generated using equations (3) and (4) (Shi and Eberhart, 1998).

$$v_i^{(t+1)} = wv_i^{(t)} + \mu_1\theta_1\left(x_{pb} - x_i^{(t)}\right) + \mu_2\theta_2\left(x_{gb} - x_i^{(t)}\right) \quad (3)$$

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

Where w represents the inertia weight, μ_1 and μ_2 are the parameters that control particle attraction to x_{pb} and x_{gb} respectively, $\theta_1$and θ_2 represent randomly generated numbers between 0.0 and 1.0. The particle's personal best position and the swarm's global best position are denoted by x_{pb} and x_{gb} respectively. $x_i^{(t)}$ is the current position of a particular particle i, while $v_i^{(t)}$ is the velocity of particle i.

The velocity update is influenced by the particle's current velocity, that is, the best position found by the particle so far (personal best -x_{pb}) and the best position found by the entire swarm (global best -x_{gb}). Using the updated velocity, each particle moves to a new position, exploring potential mentor-mentee pairs. The position update is based on the particle's current position and velocity. This process simulates the exploration of potential mentor-mentee pairs and the convergence towards optimal matches.

Step 4: Iteration **-** Continue repeating steps 2 and 3 until the algorithm terminates when it reaches the maximum iteration count or converges to a stable solution.

Step 5: Termination - The best particle is presented as the most optimal solution.

Finally, the model based on CS algorithm recommends the researcher(s) with the optimal fitness as the best match(es).

CS Algorithm

The breeding habits of cuckoo birds, particularly their parasitic behaviour, influenced the development of the CS algorithm. Each particle in CS also corresponds to a candidate solution for the optimization problem. CS algorithm uses a combination of Levy flights and abandonment to efficiently search for the optimal solution. The proposed CS-based recommendation model comprises the following steps:

Step 1: Initialization – In CS, the initial step involves setting up parameters and generating the population as well. A population of nests (researchers) is generated from the dataset. The algorithm parameters are defined, and nest positions are also initialized using equation 1 (as applicable to PSO). Unlike PSO, velocity initialization is not required in CS. Instead, CS employs a random walk mechanism to search for optimal solutions.

Step 2: Fitness Evaluation - The fitness function assesses the suitability of each nest (researcher) for mentor-mentee matching based on their expertise. The nest with the highest fitness score is considered the best nest (mentor-mentee pair) in the population.

Step 3: Generation of New Solutions (Egg Laying) **-** A nest to lay an egg (new solution) is selected using Lévy flight. Searching for the new population of nests (mentor-mentee pairs) using Lévy flight can be achieved using equation 5.

$$x_i^{(t+1)} = x_i^{(t)} + \beta * L(s, \mu) \tag{5}$$

Where β is the step size scaling factor, s is the levy exponent, μ is the standard deviation, $*$ represents the entry-wise product of two vectors, and $L(s, \mu)$ is a Lévy flight parameter.

Step 4: Abandonment and Replacement **-** The worst nests (least suitable mentor- mentee pairs) are replaced by new nests generated through egg laying. This process simulates the selection of optimal mentor-mentee pairs.

Step 5: Iteration - Continue repeating steps 2 to 4 until the algorithm terminates when it reaches the maximum iteration count or converges to a stable solution.

Step 6: Termination **-** The best nest is presented as the most optimal solution.

Finally, the model based on CS algorithm recommends the researcher(s) with the optimal fitness as the best match(es).

MODEL CONFIGURATION

To optimize model performance, thorough parameter tuning was performed experimentally, involving systematic testing of various parameter values to yield the best outcomes. Notably, in the CS algorithm, a discovery rate (pa) of 0.35 or higher yielded subpar precision, recall, and accuracy, but improved Mean Reciprocal Rank (MRR). Conversely, pa values between 0.25 and 0.325 produced optimal outcomes. The optimal parameter settings for each of the two proposed models are detailed in Tables 1 and 2.

Table 1: Parameter tuning in PSO-based Model

Parameter	Value	Description
N	500	Number of particles in the swarm
D	A number of vectorized features	Optimization space dimension
$Max-it$	200	Iteration limit
μ_1	1.495	The cognitive acceleration coefficient - it denotes the constant that controls how much a particle is attracted towards its personal best (x_{pb}) position. It is usually set between 1.4 and 2.0.
μ_2	1.495	The social acceleration coefficient- it denotes the constant that controls how much a particle is attracted to the global best (x_{gb}) position found by the entire swarm. It is usually set between 1.4 and 2.0.
w	0.729	The Inertia weight- it is the parameter that regulates the balance between exploration and exploitation. It is usually set between 0.0 and 1.0
$Top-N$	5, 10	Top N recommendations

Table 2: Parameter tuning in CS-based Model

Parameter	Value	Description
N	500	Number of host nests
D	A number of vectorized features	Optimization space dimension.
$Max-it$	200	Iteration limit
s	1.5	The levy exponent- it controls the distribution of step sizes. It is usually set between 1.0 and 2.0. It is typically set between 1.0 and 3.0
pa	0.325	Probability of abandoning a nest. This value implies there is a 32.5% chance of abandoning a nest and replacing it with a new one. It is usually set between 0 and 1.
β	1.5	The step size scaling factor- it controls the scale of the step sizes. It is usually set between 1.0 and 2.0
$Top-N$	5, 10	Top N recommendations

EVALUATION METRICS

This sub-section outlines the performance metrics used to assess the models. The trained model's performance is evaluated on a 20% test set, utilizing standard metrics for ranking-based recommender systems, including precision, recall, accuracy, and MRR using equations (6) to (9).

$$precision = \frac{W}{W+X} \quad (6)$$

$$recall = \frac{W}{W+Y} \quad (7)$$

$$accuracy = \frac{W+Z}{W+X+Y+Z} \quad (8)$$

Where W is the number of correct mentor-mentee matches, X is the number of incorrect mentor-mentee matches, Y is the number of missed potential mentor-mentee matches, and Z is the number of correct non-match.

$$MRR = \frac{1}{n}\sum_{i=1}^{n}[1/Rank(i)] \quad (9)$$

Where n is the number of users (researchers) and $Rank(i)$ is the rank of the first relevant mentor-mentee match for the ith user.

IMPLEMENTATION DETAILS

This sub-section provides an in-depth overview of the implementation details of the proposed mentorship recommendation models. The system leverages the strengths of PSO and CS algorithms to recommend mentors to mentees and vice versa based on expertise similarities. The implementation was carried out on a machine equipped with an Intel Core i5 processor, utilizing Python version 3.12 as the programming language. The Spyder Integrated Development Environment (IDE) was employed to write, debug, and execute the code. The implementation commenced with the loading of labeled data, comprising information on two distinct categories of researchers, namely experienced researchers and young researchers. The labeled data served as the foundation for training and evaluating the mentorship recommendation models. The PSO-based and CS-based models were optimized using the average precision as the fitness function. The optimization process involved iteratively refining the model parameters to maximize the average precision, which measured the accuracy of the recommendations.

The optimized models were evaluated using a comprehensive set of metrics, including precision, recall, accuracy, and Mean Reciprocal Rank (MRR). These metrics provided a thorough assessment of the models' performance, enabling the identification of strengths and weaknesses. The trained models generated recommendations for two tasks: recommending mentors to mentees and mentees to mentors. The recommendations were based on expertise similarities and were provided for two different scenarios: top-5 and top-10 recommendations for both tasks. The model provided recommendations for experienced researchers under the mentor-mentees task and young researchers under the mentee-mentors task. The recommendations were tailored to each scenario, ensuring that the suggested mentors or mentees possessed the requisite expertise to foster productive relationships. By utilizing these tools and technologies, we were able to efficiently develop, train, and evaluate the machine-learning models.

RESULTS AND DISCUSSION

This section presents the key findings from experimentation conducted on the AFT datasets, evaluating the performance of the proposed models in recommending suitable experienced researchers to young researchers and vice versa.

COMPARISON METRICS

These sub-sections describe the metrics used to compare the performance of the PSO and CS algorithms.

Precision

Precision measures the proportion of recommended mentors that are actually relevant and suitable for the mentees. The precisions @5 and @10 for the two optimized recommendation models are summarized in Table 3.

Table 3: Precision results for optimized recommendation models

Model	Precision @5	Precision@10
PSO	1.00	1.00
CS	1.00	0.94

PSO demonstrates exceptional performance, with a precision score of 1.0 at both @5 and @10. This indicates that all recommended mentors/mentees are relevant and suitable for the top 5 and top 10 recommendations. CS achieves a perfect precision score at @5, but its performance drops slightly at @10, with a precision score of 0.94. The results indicate that PSO may be a better choice when high precision is crucial, while CS may be more suitable when a larger pool of recommendations is desired, albeit with slightly lower precision.

Recall

Recall measures the proportion of all relevant recommended mentors that are actually recommended to the mentees. The recalls @5 and @10 for the two optimized recommendation models are presented in Table 4.

Table 4: Recall results for optimized recommendation models

Model	Recall@5	Recall@10
PSO	1.00	1.00
CS	1.00	0.83

PSO demonstrates exceptional performance, with a recall score of 1.0 at both @5 and @10. This indicates that PSO is able to identify all relevant mentors/mentees within the top 5 and top 10 recommendations. CS achieves a perfect recall score at @5, but its performance drops to 0.83 at @10. The results indicate that PSO may be a better choice when high recall is crucial, while CS may be more suitable when a larger pool of recommendations is desired, albeit with slightly lower recall.

Accuracy

Accuracy measures the proportion of correct recommendations (both positive and negative) out of all recommendations made. The accuracy @5 and @10 for the two optimized recommendation models are summarized in Table 5.

Table 5: Accuracy results for optimized recommendation models

Model	Accuracy@5	Accuracy@10
PSO	1.00	1.00
CS	1.00	0.90

PSO demonstrates exceptional performance, with an accuracy score of 1.0 at both @5 and @10. This indicates that PSO's recommendations are entirely accurate, with no errors or misclassifications. CS achieves a perfect accuracy score at @5, but its performance drops slightly to 0.94 at @10. The results indicate that PSO may be a better choice when high accuracy is crucial, while CS may be more suitable when a larger pool of recommendations is desired, albeit with slightly lower accuracy.

MRR

Mean Reciprocal Rank (MRR) evaluates how well the recommendation algorithm ranks relevant mentors or mentees. It considers the position of the first relevant recommendation in the list. The MRR @5 and @10 for the two optimized recommendation models are shown in Table 6.

Table 6: MRR results for optimized recommendation models

Model	**MRR@5**	**MRR@10**
PSO	0.87	0.80
CS	0.87	0.80

PSO demonstrates consistent performance, with an MRR score of 0.87 at both @5 and @10. This indicates that PSO's recommendations are relatively stable in terms of relevance ranking. CS shows consistent performance, with an MRR score of 0.80 at both @5 and @10. This suggests that CS's recommendations are also relatively stable in terms of relevance ranking. PSO's MRR scores are higher than CS's at both @5 and @10, indicating that PSO's recommendations are generally more relevant and better ranked.

COMPARATIVE RESULTS

These sub-sections present the results of the comparison, including the statistical analysis and visualizations used to illustrate the differences between the algorithms. The comparison of results of the two optimized models at recommendation of 10 researchers (mentors/mentees) is presented in Table 7.

Table 7: Evaluation Results of the optimized models

Model	**Precision**	**Recall**	**Accuracy**	**MRR**
PSO	**1.00**	**1.00**	**1.00**	**0.80**
CS	0.94	0.83	0.90	**0.80**

A comparative analysis of the performance evaluation results (Table 7) reveals that the PSO-based model (Precision: 1.00, Recall: 1.00, Accuracy: 1.00, and MRR: 0.80) outperformed the CS-based model across all metrics @ Top 10 recommendations except in MRR where it ties with CS. This improvement is attributed to the model's ability to effectively handle sparse data and capture intricate user-item relationships. A comparative analysis of the accuracy results reveals that the PSO-based model surpasses the CS-based model by 10%. Likewise the precision and recall results show that the PSO-based model surpasses the CS-based model by 6% and 17% respectively.

The empirical results underscore the efficiency of metaheuristic algorithms in enhancing recommendation model precision, inclusiveness, and accuracy. The PSO-based model emerges as the most precise, comprehensive, and accurate out of the two algorithms. These findings encourage further exploration of optimization methods to refine recommendation models. Figure 2 illustrates comparison results of the two optimized models (PSO and CS).

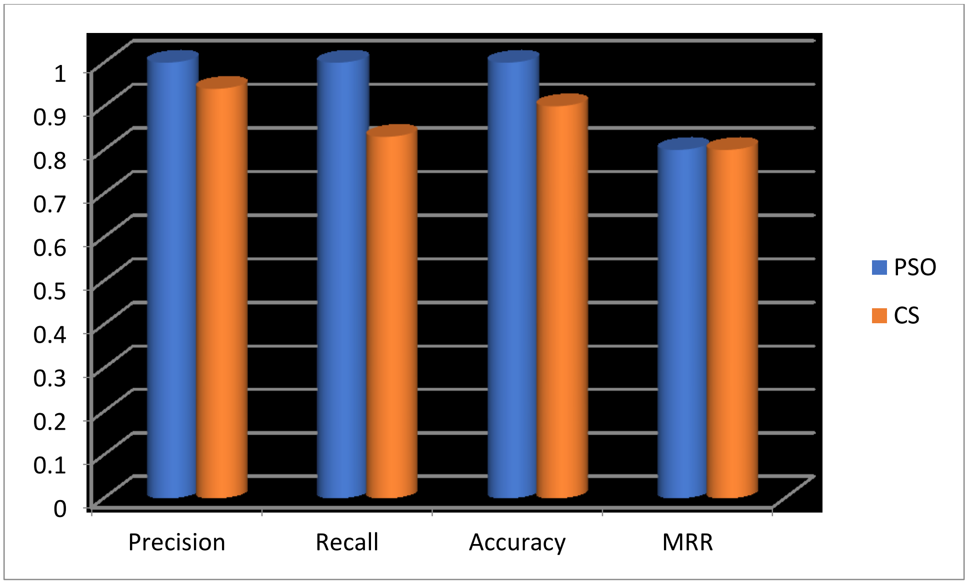


Figure 2: Comparison results of PSO and CS Optimized Models

The performance of the two models is further assessed using precision-recall curve analysis. This metric assesses the models' accuracy in detecting true positives while limiting false positives, with precision-recall curves illustrating the trade-off between these two metrics. The precision-recall curves for PSO-based and CS-based models are presented in Figures 3 and 4, respectively.

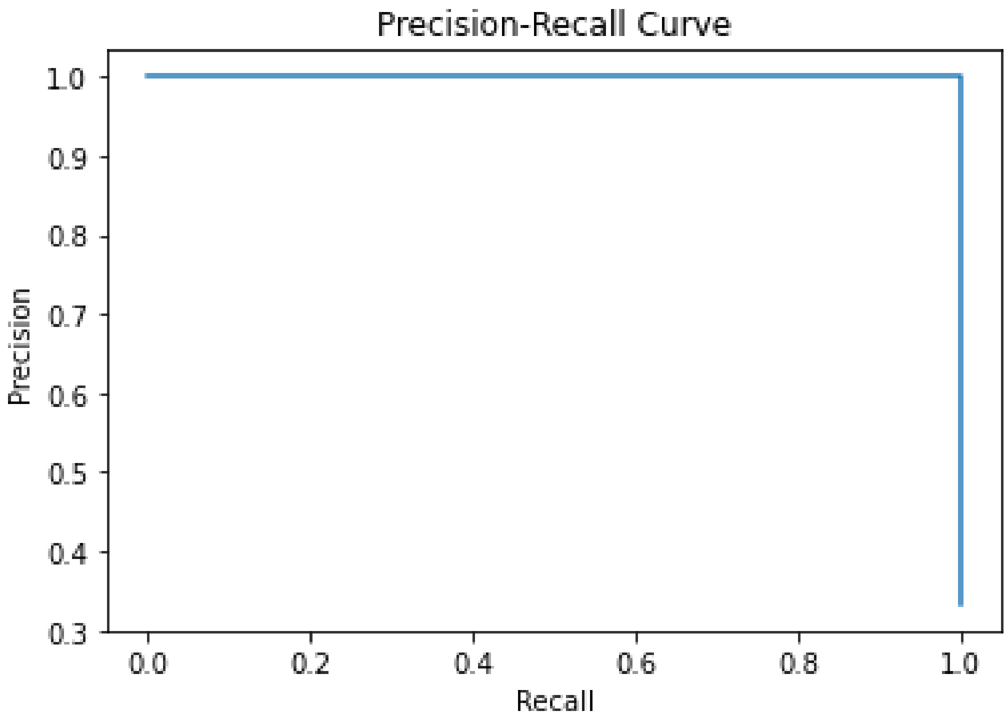


Figure 3: Precision-Recall Curve of PSO-Based Optimized Recommendation Model

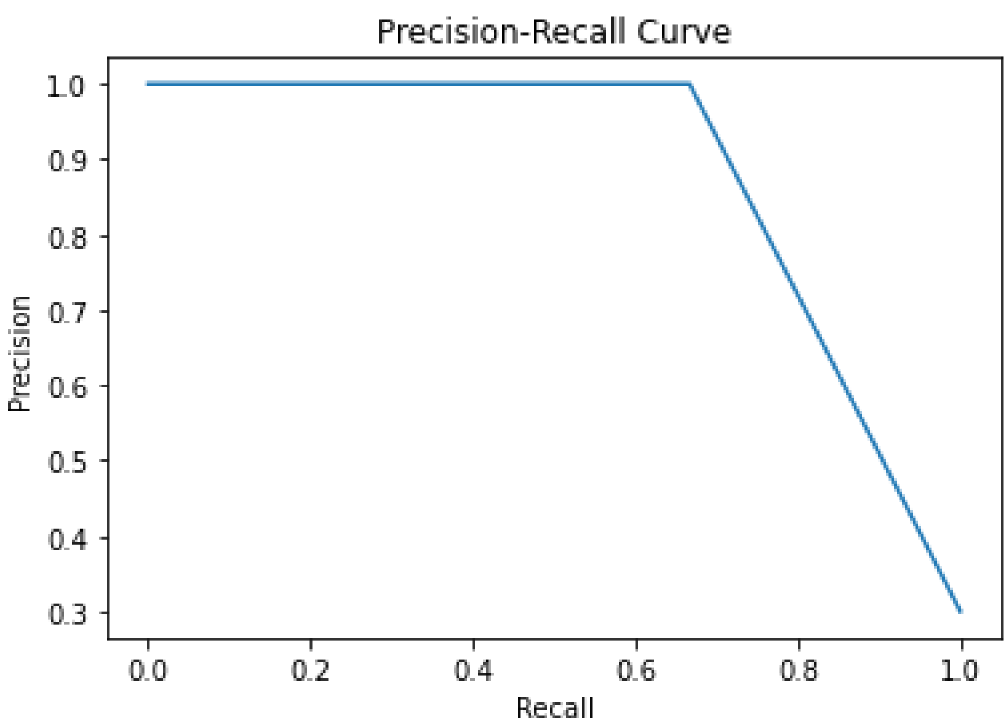


Figure 4: Precision-Recall Curve of CS-Based Optimized Recommendation Model

Precision-recall curves demonstrate that both optimized recommendation models achieve a good balance between precision and recall, providing effective, robust, and high-quality recommendations. These results demonstrate that the optimized models are well-suited for real-world mentorship recommendation applications, where they can effectively handle complex relationships and large datasets. Notably, the PSO-based model produced the best results, highlighting its potential for practical applications.

CONCLUSION

The experimental results demonstrated the effectiveness of the proposed models, with the PSO-based model achieving 100% accuracy and the CS-based model achieving 90% accuracy. These findings suggest that the proposed models can provide accurate and reliable mentorship recommendations, facilitating productive collaborations and knowledge transfer among researchers. Future research could involve leveraging additional data sources, exploring other metaheuristic algorithms and evaluating the proposed models in real-world settings.

CONFLICTS OF INTEREST

No conflict of interest to be disclosed by the author.

ACKNOWLEDGEMENT

The author would like to acknowledge the emotional support provided by her spouse, Engineer Musa Oguntuase, whose unwavering encouragement and sacrifice made this research possible.

REFERENCES

Adebayo, T. A., & Ojokoh, B. A. (2024). Prescriptive Analytics for Publication Venue Recommendation. *International Journal of Computer Applications*, **186**(7): 1-7.

Ahangar, H. G., Siamian, H., & Yaminfirooz, M. (2014). Evaluation of the scientific outputs of researchers with similar h index: A critical approach. *Acta Informatica Medica*, **22**(4): 255.

Alshareef, A. M., Alhamid, M. F., & Saddik, A. E. (2019) Academic venue recommendations based on similarity learning of an extended nearby citation network. *IEEE Access*, **7**: 38813–38825.

Averchenkova, A., Akhmetzyanova, A., Sudarikov, K., Petrov, S. Makarov, I., Pendiukhov, M., & Zhukov, L. E. (2020). Collaborator Recommender System. *Network Algorithms, Data Mining, and Applications: NET,* **8**: 101-119.

Cao, L., Xu, L., & Goodman, E. D. (2019). A Collaborative-based particle swarm optimizer with History-guided Estimation for Optimization in dynamic environments. *Expert Systems with Applications,* **120**:1-13.

Gad, A. G. (2022). Particle Swarm Optimization Algorithm and its Applications: A Systematic Review. *Archives of Computational Methods in Engineering,* **29**: 2531-2561.

Ghosal, T., Chakraborty, A., Sonam, R., Ekbal, A., Saha, S., & Bhattacharyya, P. (2019) Incorporating full text and bibliographic features to improve scholarly journal recommendation. *In: 2019 ACM/IEEE joint conference on digital libraries (JCDL)*, 374–375.

Hirsch, J. E. (2005). An Index to quantify an Individual's scientific research output. *PNAS*, **102** (46): 16569-16572.

Husain, O., Salim , N., Alias, R. A., Abdelsalam, S., Thurasamy, R., Faisal, H. M. & Hamzah, M. (2021). Modeling Academic Research Collaborator Selection Using an Integrated Model. *IEEE Access*, **9** (2021): 102397-102421.

Kennedy, J. & Eberhart, R. (1995). Particle Swarm Optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Perth, Australia, 1942-1945.

Magara, M. B., Ojo, S. O., & Zuva, T. (2018). Towards a serendipitous research paper recommender system using bisociative information networks (Bisonets). *In: 2018 International conference on advances in big data, computing and data communication systems*, 1-6.

Oguntuase, R. A., Ojokoh, B. A., Oluwadare, S. A., & Afolayan, A. H. (2024). Investigating academic researchers' perceptions of a recommender system for mentor-to- mentee matching. *Global Journal of Engineering and Technology Advances*, **19**(2): 59-68.

Oguntuase, R. A. (2024). Optimizing researcher mentorship matching: A particle swarm optimization-based recommendation model. *World Journal of Advanced Engineering Technology and Sciences,* **13**(2): 698-707.

Pradhan, T., & Pal, S. (2020). A multi-level fusion based decision support system for academic collaborator recommendation. *Knowledge-Based Systems*, **197**: 105784

Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., & Mirjalili, S. (2022). Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access*, **10**: 10031-10061.

Shi, Y. H., & Eberhart, R. C. (1998). A modified particle swarm optimizer. In: 1998 *IEEE International Conference on Evolutionary Computation proceedings. IEEE world congress on computational intelligence,* 69-73.

Wang, W., Liu, J., Yang, Z., Kong, X., & Xia, F. (2019). Sustainable Collaborator Recommendation Based on Conference Closure. *IEEE Transactions on Computational Social Systems*, **6**(2): 1-12.

Xu, Z., Yuan, Y., Wei, H., & Wan, L. (2019) A serendipity-biased deepwalk for collaborators recommendation. *PeerJ Computer Science,* **5** (178): 1-19.

Yang, X. S., & Deb, S. (2009). Cuckoo search via Lévy flights. *In: 2009 world congress on nature & biologically inspired computing*, 210–214.

Zhang, Z., Patra, B. G., Yaseen, A., Zhu, J., Sabharwal, R., Roberts, K., Cao, T. & Wu, H. (2023). Scholarly recommendation system: a literature survey. *Knowledge and Information Systems*, **65** (11): 4433-4478.

Zhu, J. & Yaseen, A. (2022). A Recommender for Research Collaborators Using Graph Neural Networks. *Frontier in Artificial Intelligence,* **5**: 881704. doi: 10.3389/frai.2022