International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

NIEF-CS: Nature Inspired Energy Efficient Framework bases on BAT Algorithms for Solving the Cloud Selection Problem

[1]Om Prakash, [2]Muzaffar Azim, [3]S. M. K. Quadri

Submitted: 12/09/2023 **Revised**: 22/10/2023 **Accepted**: 09/11/2023

Abstract: Nowadays, Due to the rapid increase of cloud enabled services, selecting a dependable cloud provider has become extremely complex. A complete review of cloud services from several perspectives needs a precise decision-making process. In view of the vast complexity and limitations of present approaches, which damage the trust of the energy saving cloud selection procedure, further research is necessary to deliver more genuine decision making results. The purpose of this work is for tackling the Cloud Selection problem, a nature-inspired Energy Efficient Framework based on BAT-NN algorithms is used: We provide a new method for forecasting Energy Efficient Cloud Selection (NIEF-CS) in this research, where we have applied our suggested model to compute and predict numerous risk variables. In order to achieve energy saving Cloud selection, and we have compared the model to existing ML techniques like BAT-MM, Genetic-NN, and PSO-NN. We have proposed Nature inspired Energy Efficient Framework bases on BAT-NN algorithms. The UCI ML Repository was selected to collect information on cloud QWS dataset is widely used for study, research and verification of hybrid learning approach with optimal set of concrete services. Results: Our proposed model NIEF-CS achieved the best prediction among ML techniques.

Keywords: *NIEF-CS, BAT-NN, ML techniques, Efficient*

Introduction

Cloud computing has grown in popularity over the past decade due to its ability to scale, reliability, and low cost compared to traditional computer models that rely on specialized in-house infrastructure [1]. The word cloud computing paradigm is gotten from various revolutionary technologies and is often confused with other computing paradigms. However, it provides higher savings and dependability compared to the traditional approach [2]. Recent technological advancements have resulted in the rapid growth of cloud enabled technologies. As a result, the different services provided by numerous cloud service providers(CSPs) has expanded significantly [3]. However, the availability of multiple service providers offering a wide range of comparable services with varying features makes it challenging for cloud consumers to select the proper service that meets their needs. Furthermore, cloud clients are unaware of how their demands can be optimized and evaluated [4].

The main components that make it possible to identify and assess different cloud service providers in this context are Quality of Service (QoS) criteria. Performance, reaction time, security, and dependability are only a few of the functional and non-functional features of cloud services that are covered by QoS criteria [5]. A decision-maker or cloud customer assesses the various cloud services being offered using QoS standards in a service selection conundrum. Because of this, the choice of cloud services is a multi-criteria decision [6].Finding the right cloud service for prospective consumers is the largest problem in the selection process for cloud services. A significant challenge [7] is making an effective and precise choice of the appropriate cloud service. A variety of strategies have been put out in the literature to address this problem, including machine learning-based strategies, analytical hierarchical process-based strategies, and multi-criteria decision-making strategies. These approaches use different methods to analyze and evaluate the QoS parameters of cloud services and provide recommendations for cloud service selection [8]. However, there are still some challenges associated with the selection of cloud services, such as the lack of transparency in service-level agreements (SLAs), limited knowledge about cloud service providers, and the need for efficient decision-making methods. Thus, the development of efficient and effective approaches for cloud service selection is still an ongoing research area [9]. Nevertheless, the potential advantages of cloud computing are reduced costs, improved scalability, and enhanced reliability, make it a highly attractive option for businesses and organizations of all sizes. A significant issue with cloud service selection is identifying essential factors that denote whether the cloud service offered fits the commercial and technological expectations of cloud clients [8].

[1]*Research Scholar, FTK-Centre for IT, Jamia Millia Islamia,New Delhi, India*
[2]*System Analyst, FTK-Centre for IT, Jamia Millia Islamia, New Delhi, India*
[3]*Professor, Department of Computer Science, Jamia Millia Islamia, New Delhi, India*
opkushwaha2013@gmail.com, mazim@jmi.ac.in, quadrismk@jmi.ac.in

As seen in Figure 1, a cloud client has two sorts of requirements: essential and non-essential. In brief, a essential need denotes a service's unique behaviour, whereas a non-essential requirement denotes a service's performance. Choosing adequate cloud QoS criteria is a tough undertaking due to the complication of cloud based services and the less common measurements [9].

In [10], researchers expended significant effort on cloud service choosing. The authors established the cloud storage measuring index (SMI), which divides cloud QoS requirements into seven core areas and is frequently used to select cloud services.

Source / Feature	Safety Regulation/ Regulatory Guide	IEEE Industry Standarts	Configuration Management Process Area
Function	- Specific requirement on protection system - Endorse IEEE industry standarts - Reflect to standarts review plan	- Specific SCM activities - Change management - Supplier control - Tools, techniques and methodologies - Provide some samples plan	- Specific and generic goal - Specific and generic practice - Refer to other process area
Non-function	- Quality - Reliability - Quality assurance - Manage configuration activities	- Traceability - Consistency - No specific descriptions for safety system	- Integrity - Support other process area - No specific descriptions for safety system

Fig 1: functional and non-functional

This study's key focus includes:

- Intense studies and thoughtful evaluation of cutting-edge meta-heuristics oriented cloud task scheduling techniques to determine their applicability kinds, scheduling goals, and constraints.
- A dynamic BAT-based meta-heuristic cloud selection strategy that minimises number of iterations and enhances resource usage and performance is recommended.
- NIEF-CS experimental performance was evaluated in comparison to those of its counterparts.

The rest of the paper is organized as follows: Part 2 explains each aspect of the relevant work, and Section 3 describes the proposed model solution . This section discusses the BAT algorithms, Framework, parameters involved, suggested method, and proposed methodology. Part 4 contains details on the experimental design setup, dataset information, findings, and discussions. The conclusion is discussed in Section 5.

1. Related Work

This section presents an overview of cutting-edge meta-heuristic task scheduling algorithms developed in recent research projects. In task scheduling, meta-heuristic algorithms are commonly utilized due to their effectiveness and ability to find near-optimal solutions. However, finding the best meta-heuristic algorithm for task scheduling depends on the specific application domain, problem complexity, and evaluation criteria.

One of the proposed approaches is the hybrid load-balancing technique, presented in [11], that combines the advantages of the Effective Instructional Based Optimized (TLBO) approach with the Grey Wolf Optimization (GWO) technique. This approach effectively balances workload based on time and cost and reduces task queue wait time. However, it does not consider throughput, which is a critical parameter in some applications.

In [12], the author proposes a target-aware work scheduling technique for Cloud that utilizes the Genetic Algorithm (GA) method to optimise execution time and resource cost. However, the GA algorithm has scalability issues when dealing with vast and intricate topics such as task scheduling.

The Look-Ahead-GA is a improved version of GA, proposed by [13] and is well-suited for large-scale platforms like Grid computing. This method determines task ordering based on resource completion times in each generation and selects the resource with the lowest failure rate during the mutations stage. The scheduling goals of this approach are dependability and task failure rate, but it disregards makespan and performance as rescheduling criteria.

In [14], a PSO-based Task-focused Load Balancing (TBSLB-PSO) technique is proposed, which enhances load balancing in cloud by using the task migration method to move tasks from overloaded cloud virtual machines (VMs) to underloaded VMs. This methodology reduces load balancing duties by transferring work without halting overburdened VMs.

In [15], the authors provide a project scheduling solution based on adaptive particle swarm optimization (APSO) for capacity resource restrictions. By taking use of the activities' in-degree and out-degree in the directed acyclic graph, the Vertex Priority Generation (VPG) converts invalid particles into valid particles. Making use of Makespan rather than performance and ARUR (Average Resource Utilisation Ratio) allows for a more accurate assessment of APSO's performance.

In [16], authors proposed a modified PSO (PSO-BOOST) meta-heuristic technique based cloud work scheduling system. Cost, Time, acceptance ratio, and throughput are evaluated using this approach. However, this approach does not explicitly specify the function of inertia weight or its selection criterion. Additionally, ARUR is not regarded as an assessment parameter, which is crucial for determining how effective a scheduling method is.

In [17], the author proposes an upgraded version of PSO based on an adaptive inertia weight technique. The proposed method was tested for solution correctness and fast convergence, but makespan and performance are not evaluated as parameters. The method was also tested on real-world engineering challenges, which demonstrated its effectiveness and applicability in practical scenarios.

In [18], the author provides an overview of several inertia weight systems used by various scientists in their research, categorizing them into three major groups: time varying, stable, and dynamic persistence weights. However, performance and ARUR are not taken into account in this overview, which is crucial when selecting an inertia weight system for task scheduling.

Table 1: Comparative analysis of various existing works

Paper	Method	Nature	Limitations	Advantages
[19]	Non-dominated sorting	Dynamic	Starvation risk for low-priority tasks, no consideration for bandwidth	Reduced execution time and increased throughput
[20]	Load balancing using mutation based PSO	Dynamic	Not suitable for dependent and heterogeneous tasks, large number of PSO iterations increase Makespan	Improved reliability, transmission time, execution time, cost, and round trip time
[21]	Enhanced load-balancing min-min	Static	Two service scheduling process lead to increased Make span time	Improved utilization of resources
[22]	Multi-objective PSO based framework	Dynamic	Not suitable for independent tasks, need to increase utilization of resources	Reduced energy dissipation, Makespan, and unsuccessful tasks
[23]	Priority-based cost algorithm	Static	Unable to solve dynamic cloud environments and related tasks	Reduced processing cost

[24]	Non-linear programming model	Dynamic	Take more time and cost to solve programmable models	Reduced Makespan and improved utilization ratio

2. Proposed Model

The process of choosing the best cloud service provider (CSP) from a variety of offered options in order to satisfy the particular needs of a given application or workload is referred to as cloud selection. During this procedure, a number of variables are taken into account, including cost, performance, dependability, security, and compliance. Identify the particular requirements of an application or workload that must be hosted in the cloud as the first step. Processing power, memory, storage, network bandwidth, and security are a few examples of these requirements.

Second step is to discover services, we have proposed nature inspired ML model for the cloud service filtering. We have chosen BAT algorithm as base learner and improve using neural network. Next step we have trained our model and test the model. In the last step we have compare our model with other models.

3.1. Dataset: The work presents a hybrid learning method was evaluated using a perfect selection of actual, verified services with the QWS dataset. The dataset, compiled by Eyhab Al-Masri from the University of Guelph, contains 2507 QoS parameters for actual services, including availability, response time, throughput, dependability, and success-ability [25]. This dataset has been widely used by researchers to investigate the structure of various cloud services [12][13].

Table 2: Detail description of dataset.

Quality Factor	Definition
Availability	The amount of time that the cloud service is operational and accessible to users under normal conditions.
Throughput	The rate at which a cloud service provider processes and delivers service requests to users, typically measured in terms of volume per unit of time [26].
Interoperability	The ability of a cloud service to interact and communicate effectively with other services, regardless of whether they are offered by the same or different providers.
Cost	The monetary expense associated with using specific cloud services or features.
Response Time	The time taken for a cloud service to acknowledge and answer to a user's request.
Stability	The consistency and predictability of a cloud service's performance, including its ability to maintain a certain level of service quality over time.
Adaptability	The capacity of a cloud service provider to adjust and customize service delivery based on user requests and changing conditions.
Accuracy	The degree to which a cloud service's computed or delivered results match the expected or advertised values [27].

Usability	A subjective measure of how easily and intuitively a user can access and utilize a cloud service's features and capabilities[28].
Scalability	The ability of a cloud service or system to handle varying levels of user demand and resource consumption without sacrificing performance or reliability [29][30].
Reliability	The extent to which a cloud service can function without errors, downtime, or other forms of disruption over a given period of time and in a particular environment [31][32].

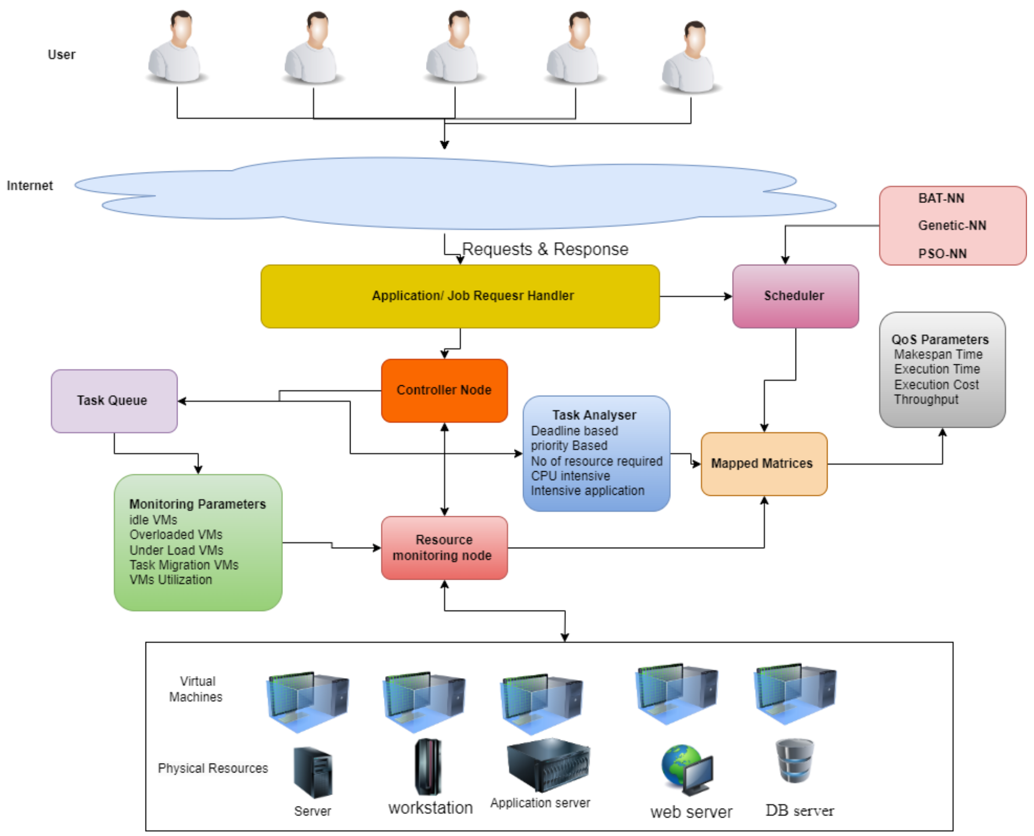

Fig 2: Proposed Model Framework

3.2. Bat Algorithm (BA) : The Bat Algorithm (BA) involves several steps, beginning with the initialization of various parameters. These parameters are represented as:

M: size of the bats population

N: the maximum number of rounds

global_best: the current best location found by the algorithm

In addition, each individual bat in the population is represented by a set of parameters, including:

- *x_i* - location of the *ith* bat
- *v_i* - speed of the *ith* bat
- *f_i* - pulse frequency of *ith* bat, which ranges between { *f_min* and f_*max* }
- *fitness_i* - fitness function for the ith bat
- A_i: the loudness of the ith bat, which is updated using a constant constraint in the range [0, 1]
- *r_i* - pulse rate of the ith bat, which is also updated using a constant parameter in the given range [0, 1]

In the algorithm, each bat's pulse frequency is adjusted based on its fitness, with fitter bats having higher pulse frequencies. This is represented as:

$$f\_i = f\_min + (f\_max - f\_min) * rand() \;\ldots\ldots\ldots\ldots (1)$$

where, rand() is a random value in the range [0, 1].

The loudness of each bat is updated using the formula:

$$A\_i = \alpha * A\_i \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where, alpha is a constant value in the range [0, 1].

Similarly, the pulse rate of each bat is updated using the formula:

$$r\_i = r\_i * (1 - exp(-\gamma * t)) \ldots\ldots\ldots\ldots (3)$$

Where, is a constant parameter and t is the current iteration number.

By iteratively updating these parameters and adjusting the pulse frequencies, the algorithm seeks to optimize a given objective function.

Step 2. Update the ith bat's global_best position, velocity, pulse frequency, and location as follows:

$$v\_i(t+1) = v\_i(t) + (x\_i(t) - global\_best) * f\_i \; x\_i(t+1) = x\_i(t) + v\_i(t+1) \ldots\ldots\ldots (4)$$

where *v_i(t)* and *x_i(t)* represent the bat's velocity and location at time t, global_best represents the bat's current global best position, and rand_1 and rand_2 represent random values between 0 and 1.

Step 3. The following equation produces a new output for the bat if the random number is bigger than r_i:

$$x\_i(t+1) = global\_best + rand() * (x\_j(t) - x\_k(t)) \ldots\ldots\ldots\ldots (5)$$

where x_j(t) represents the location of another bat and x_k(t) denotes the average location of all bats at time a given time t.

Step 4. If (random_number < A_i) and fitness_i is improved, the new solution is accepted.

In next step, update the value of A_i and r_i, as follows:

$$A\_i(t+1) = \alpha * A\_i(t) \; r\_i(t+1) = r\_i(t) * (1 - exp(-\gamma * t)) \ldots\ldots\ldots\ldots (6)$$

where alpha α is a constant parameter in the range [0, 1], gamma is a constant parameter, and fitness_i(t+1) and fitness_i(t) represent the fitness of the bat at times t+1 and t, respectively. As t increases, A_i and r_i decrease.

Step 5. Find the current best option by sorting the bats according to their fitness.

Step 6. Input the globally optimal solution after sorting the bats according to fitness and going back to Step 2 as many times as necessary. Identify the present ideal solution.

3.3. Algorithm :

Procedure LCA

Step 1: Initialization

i. Assign random position of (x) and velocity (v) for each bat.

ii. Give each bat a unique frequency (f), heart rate (r), and loudness (l) value.

Step 2: Formation of New Solutions

Produce new solutions at time step t by:

Update velocity: $v\_t = v\_t + (x\_t - x\_best) * f$

Update position: *x_t = x_t + v_t*

Adjusting frequency: *f = f_min + rand(0,1) * (f_max - f_min)*

Step 3: Produce a Local Solution

If *rand(0,1) > r_i*, choose the best among available solutions and create a temporary solution for it.

Step 4: Populate a New Random Solution

$If\ rand(0,1) < A\_i\ and\ f\_i < f\_min$, accept the new solution.

Procedure LCA Enhance *r_i* and decrease *A_i*. **Step 5:** Repeat Step 2 to 4 for each bat. **Step 6:** Give ranking to the Bats and obtain the available best solution Sort and rank each bats on the basis of the fitness values and find the best solution. **Step 7:** Repeat Steps 2 to 6 until the maximum rounds are reached. Step 8: Output: Print the final solution *x_best.*

3. Result

The experiments were carried out in the Anaconda 3 environment, on an Intel(R) i7-1165G7 @2.8 GHz processor with 16 GB RAM and a 64-bit Windows OS, in order to show the efficiency of the NIEF-CS model proposed in this work. The proposed technique entails training multiple models and combining their predictions to improve accuracy. The QWS dataset is divided into two parts: training and testing. For the evaluation of our model, we used two parameters.

4.1. Accuracy: Accuracy is a well-liked measure for evaluating the efficiency of classification methods. It determines the proportion of examples in the dataset that were properly classified out of all the instances[13]. The following is the equation for accuracy.

$$Accuracy = \frac{(Number\ of\ Accurate\ Predictions)}{(Total\ Number\ of\ Predictions)} \qquad (1)$$

MSE stands for Mean Squared Error, and it is a common metric used to calculate the performance of the regression model.

The equation for MSE is as follows:

$$MSE\ =\ (1/n)\ *\ \Sigma\ (yi\ -\ \hat{y}i)^2 \qquad (2)$$

In the given dataset, n represents the overall number of observations, yi is the dependent variable's value as it was observed, and i is the predicted value for that observation.

We have studied and evaluated the performance of our suggested approach through a battery of experiments on a QWS dataset. Bioinspired models are developed from the ground up and trained (BAT-MM, Genetic-NN, and PSO-NN). Table 1 and Figure 3, 4, 5 depicts the training MSE, Training and test accuracy of BAT-MM, Genetic-NN, and PSO-NN.

Table 2: Test Accuracy of bio-Inspired models

Model	MSE	Train Accuracy	Test Accuracy
BAT-NN (Proposed Model)	0.1734	91.43	93.33
Genetic-NN	0.2285	85.71	76.67
PSO-NN	0.1626	91.43	86.67
[3]	NA	NA	90.11
[34]	NA	NA	90.68

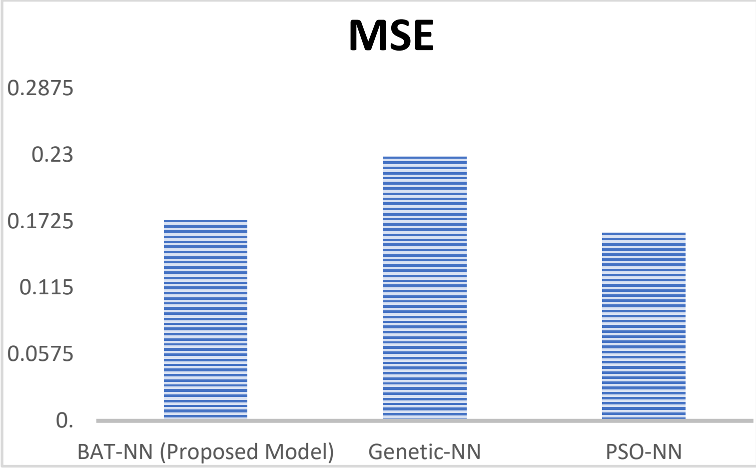

Fig 3: MSE of bio-Inspired models

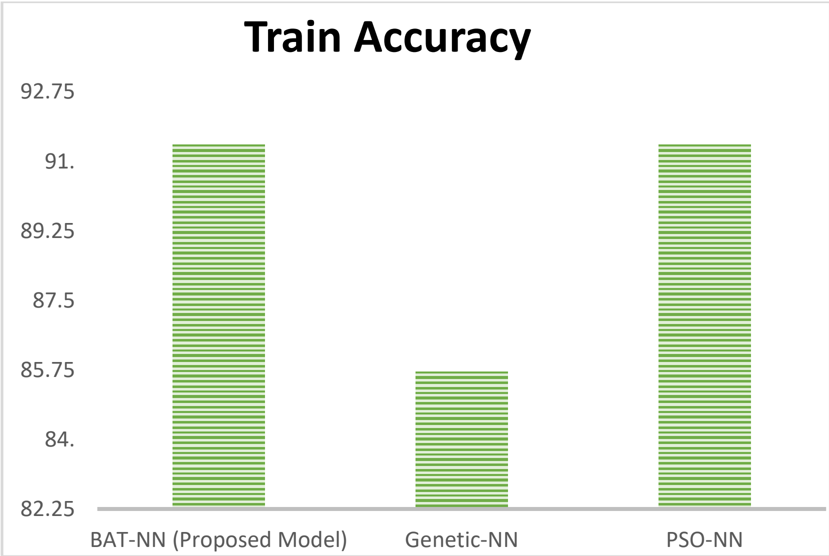

Fig 4: Training Accuracy of bio-Inspired models

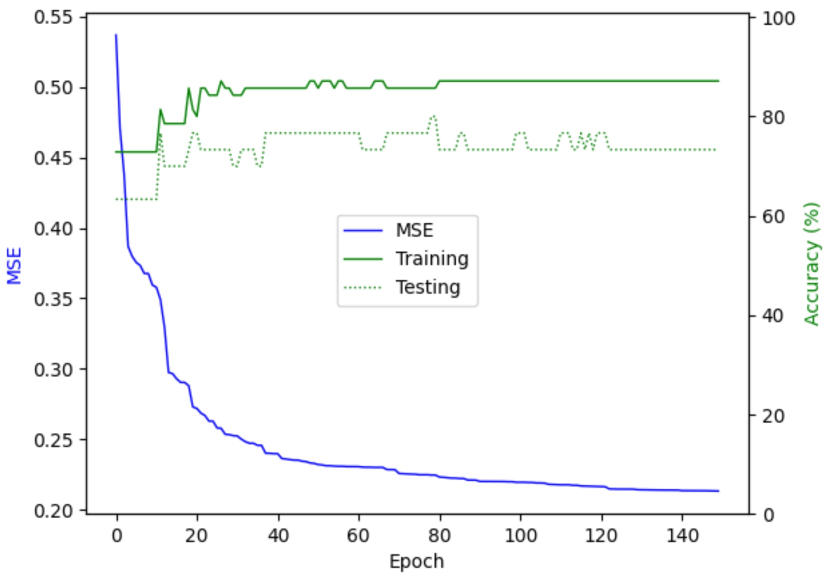

Fig 5: Plot for Genetic-NN

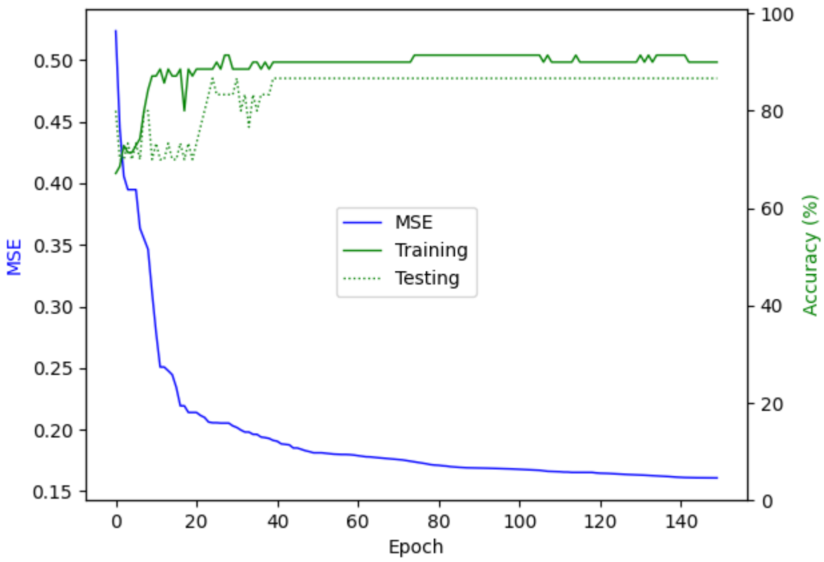

Fig 6: Plot for PSO-NN

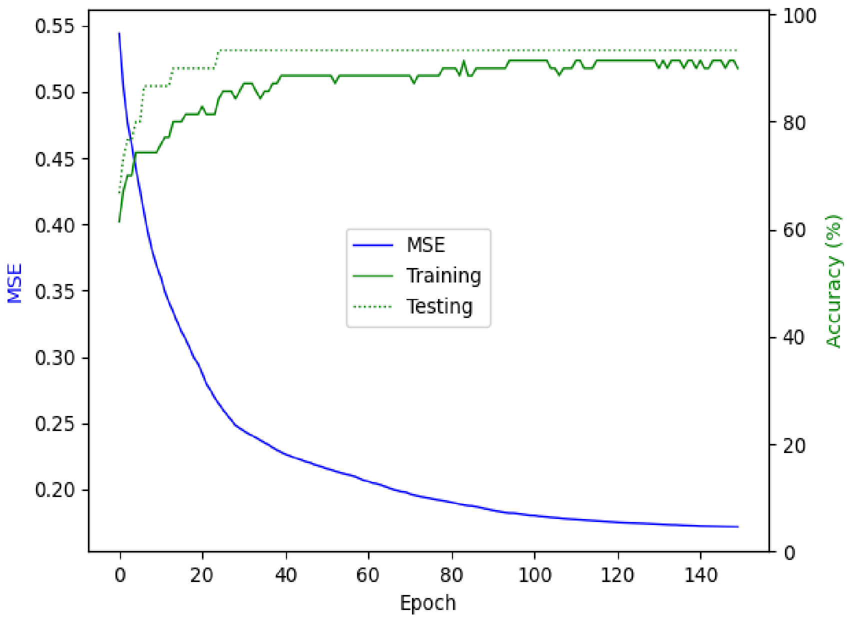

Fig 7: Plot for BAT-NN

Figure 5, 6, 7 depicts the plots MSE, Train, Test accuracy for the bio inspired models. Our proposed model outperform [3], [34]other models and state of art model on QWS dataset. BAT-NN achieved 93.33% test accuracy. Which is better than genetic-NN, and PSO-NN.

4. Conclusion

In this research, we have focused on cloud service selection strategy in multicriteria decision-making circumstances. We introduced the NIEF-CS cloud selection framework, which is based on a nature-inspired algorithm. By summarizing and analyzing the existing research, we compare numerous strategies. Many real-world examples of current approaches' applicability are offered. As a result, NIEF-CS has a significant impact and relevance in multi-criteria decision situations. Hence, the work reviewed some of the benefits and drawbacks, and also illustrated numerous implementations of the NIEF-CS framework in selecting cloud services. Furthermore, multicriteria applied approaches are comprehensively described and assembled for use in various study disciplines. Moreover, the NIEF-CS framework and its diverse qualities are described and contrasted, which can assist furure researchers in determining future objectives. We believe that the research might be expanded to include intercloud service options.

References

[1] S. K. Garg, S. Versteeg, and R. Buyya, “Smicloud: A framework for comparing and ranking cloud services,” in 2011 Fourth IEEE International Conference on Utility and Cloud Computing, 2011, pp. 210–218.

[2] V. Narayan, A. K. Daniel, and A. K. Rai, “Energy efficient two tier cluster based protocol for wireless sensor network,” in 2020 international conference on electrical and electronics engineering (ICE3), 2020, pp. 574–579.

[3] Faiz, M., Fatima, N., Sandhu, R., Kaur, M., & Narayan, V. (2022). Improved Homomorphic Encryption for Security in Cloud using Particle Swarm Optimization. Journal of Pharmaceutical Negative Results, 4761-4771.

[4] V. Narayan and A. K. Daniel, “RBCHS: Region-based cluster head selection protocol in wireless sensor network,” in Proceedings of Integrated Intelligence Enable Networks and Computing, Springer, 2021, pp. 863–869.

[5] P. K. Mall, R. K. Yadav, A. K. Rai, V. Narayan, and S. Srivastava, “Early Warning Signs Of Parkinson’s Disease Prediction Using Machine Learning Technique,” J. Pharm. Negat. Results, pp. 2607–2615, 2023.

[6] G. Somani, M. S. Gaur, D. Sanghi, M. Conti, and R. Buyya, “DDoS attacks in cloud computing: Issues, taxonomy, and future directions,” Comput. Commun., vol. 107, pp. 30–48, 2017.

[7] S. Srivastava and S. Sharma, “Analysis of Cyber Related Issues by Implementing Data Mining Algorithm,” in 2019 9th International Conference on Cloud Computing, Data Science Engineering (Confluence), 2019, pp. 606–610, doi: 10.1109/CONFLUENCE.2019.8776980.

[8] V. Narayan and A. K. Daniel, “Design Consideration and Issues in Wireless Sensor Network Deployment.”,” Invertis J. Sci. \& Technol., p. 101, 2020.

[9] Narayan, Vipul, and A. K. Daniel. "CHOP: Maximum coverage optimization and resolve hole healing problem using sleep and wake-up technique for WSN." *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal* 11.2 (2022): 159-178.

[10] Narayan, Vipul, A. K. Daniel, and Pooja Chaturvedi. "E-FEERP: Enhanced Fuzzy based Energy Efficient Routing Protocol for Wireless Sensor Network." *Wireless Personal Communications* (2023): 1-28.

[11] S. Mousavi, A. Mosavi, and A. R. Varkonyi-Koczy, “A load balancing algorithm for resource allocation in cloud computing,” in Recent Advances in Technology Research and Education: Proceedings of the 16th International Conference on Global Research and Education Inter-Academia 2017 16, 2018, pp. 289–296.

[12] J. Meena, M. Kumar, and M. Vardhan, “Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint,” IEEE Access, vol. 4, pp. 5065–5082, 2016.

[13] X. Wang, C. S. Yeo, R. Buyya, and J. Su, “Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm,” Futur. Gener. Comput. Syst., vol. 27, no. 8, pp. 1124–1134, 2011.

[14] F. Ramezani, J. Lu, and F. K. Hussain, “Task-based system load balancing in cloud computing using particle swarm optimization,” Int. J. Parallel Program., vol. 42, pp. 739–754, 2014.

[15] N. Kumar and D. P. Vidyarthi, “A model for resource-constrained project scheduling using adaptive PSO,” Soft Comput., vol. 20, no. 4, pp. 1565–1580, 2016.

[16] M. Kumar and S. C. Sharma, “PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing,” Neural Comput. Appl., vol. 32, pp. 12103–12126, 2020.

[17] A. Agrawal and S. Tripathi, “Particle swarm optimization with adaptive inertia weight based on cumulative binomial probability,” Evol. Intell., vol. 14, pp. 305–313, 2021.

[18] X. Huang, C. Li, H. Chen, and D. An, “Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies,” Cluster Comput., vol. 23, pp. 1137–1147, 2020.

[19] P. Zhang and M. Zhou, “Dynamic cloud task scheduling based on a two-stage strategy,” IEEE Trans. Autom. Sci. Eng., vol. 15, no. 2, pp. 772–783, 2017.

[20] A. I. Awad, N. A. El-Hefnawy, and H. M. Abdel_kader, “Enhanced particle swarm optimization for task scheduling in cloud computing environments,” Procedia Comput. Sci., vol. 65, pp. 920–929, 2015.

[21] A. V. Lakra and D. K. Yadav, “Multi-objective tasks scheduling algorithm for cloud computing throughput optimization,” Procedia Comput. Sci., vol. 48, pp. 107–113, 2015.

[22] R. K. Jena, “Multi objective task scheduling in cloud environment using nested PSO framework,” Procedia Comput. Sci., vol. 57, pp. 1219–1227, 2015.

[23] S. Selvarani and G. S. Sadhasivam, “Improved cost-based algorithm for task scheduling in cloud computing,” in 2010 IEEE International Conference on Computational Intelligence and Computing Research, 2010, pp. 1–5, doi: 10.1109/ICCIC.2010.5705847.

[24] W. Lin, C. Liang, J. Z. Wang, and R. Buyya, “Bandwidth-aware divisible task scheduling for cloud computing,” Softw. Pract. Exp., vol. 44, no. 2, pp. 163–174, 2014.

[25] Narayan, V., Awasthi, S., Fatima, N., Faiz, M., Bordoloi, D., Sandhu, R., & Srivastava, S. (2023, May). Severity of Lumpy Disease detection based on Deep Learning Technique. In 2023 International Conference on Disruptive Technologies (ICDT) (pp. 507-512). IEEE.

[26] Narayan, Vipul, and A. K. Daniel. "Energy Efficient Protocol for Lifetime Prediction of Wireless Sensor Network using Multivariate Polynomial Regression Model." (2022).

[27] Narayan, Vipul, and A. K. Daniel. "CHHP: coverage optimization and hole healing protocol using sleep and wake-up concept for wireless sensor network." *International Journal of System Assurance* Engineering and Management 13.Suppl 1 (2022): 546-556.

[28] Narayan, Vipul, and A. K. Daniel. "CHHP: coverage optimization and hole healing protocol using sleep and wake-up concept for wireless sensor network." International Journal of System Assurance Engineering and Management 13.Suppl 1 (2022): 546-556.

[29] Awasthi, Shashank, et al., eds. AI and IoT-based Intelligent Health Care & Sanitation. Bentham Science Publishers, 2023.

[30] Awasthi, Shashank, et al. "An epidemic model for the investigation of multi-malware attack in wireless sensor network." IET Communications (2023).

[31] Awasthi, Shashank, et al., eds. Artificial intelligence for a sustainable industry 4.0. Springer International Publishing, 2021.

[32] Awasthi, Shashank, Naresh Kumar, and Pramod Kumar Srivastava. "An epidemic model to analyze the dynamics of malware propagation in rechargeable wireless sensor network." Journal of Discrete Mathematical Sciences and Cryptography 24.5 (2021): 1529-1543.

[33] Dr. Sandip Kadam. (2014). An Experimental Analysis on performance of Content Management Tools in an Organization. International Journal of New Practices in Management and Engineering, 3(02), 01 - 07. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/27

[34] Faris, W. F. . (2020). Cataract Eye Detection Using Deep Learning Based Feature Extraction with Classification. Research Journal of Computer Systems and Engineering, 1(2), 20:25. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/7