Computers, Materials & Continua
DOI: 10.32604/cmc.2022.025658
Article

Tech Science Press

Modified Bat Algorithm for Optimal VM's in Cloud Computing

Amit Sundas[1], Sumit Badotra[1,*], Youseef Alotaibi[2], Saleh Alghamdi[3] and Osamah Ibrahim Khalaf[4]

[1]Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, 144411, India
[2]Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah, 21955, Saudi Arabia
[3]Department of Information Technology, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia
[4]Al-Nahrain Nano-Renewable Energy Research Center, Al-Nahrain University, Baghdad, 10072, Iraq
*Corresponding Author: Sumit Badotra. Email: sumit.26152@lpu.co.in
Received: 01 December 2021; Accepted: 09 February 2022

Abstract: All task scheduling applications need to ensure that resources are optimally used, performance is enhanced, and costs are minimized. The purpose of this paper is to discuss how to Fitness Calculate Values (FCVs) to provide application software with a reliable solution during the initial stages of load balancing. The cloud computing environment is the subject of this study. It consists of both physical and logical components (most notably cloud infrastructure and cloud storage) (in particular cloud services and cloud platforms). This intricate structure is interconnected to provide services to users and improve the overall system's performance. This case study is one of the most important segments of cloud computing, i.e., Load Balancing. This paper aims to introduce a new approach to balance the load among Virtual Machines (VM's) of the cloud computing environment. The proposed method led to the proposal and implementation of an algorithm inspired by the Bat Algorithm (BA). This proposed Modified Bat Algorithm (MBA) allows balancing the load among virtual machines. The proposed algorithm works in two variants: MBA with Overloaded Optimal Virtual Machine (MBA-OOVM) and Modified Bat Algorithm with Balanced Virtual Machine (MBA-BVM). MBA generates cost-effective solutions and the strengths of MBA are finally validated by comparing it with Bat Algorithm.

Keywords: Bat algorithm; cloud computing; fitness value calculation; load balancing; modified bat algorithm

1 Introduction

Optimization is the process of obtaining the best possible solution for any particular problem while satisfying underlying constraints [1,2]. While solving any problem and obtaining its solution, the optimization of the solution is considered as one of the major concerns [3,4]. This concern becomes more important when the solution to any combinatorial problem is to be obtained [5–7]. To obtain the optimal result for combinatorial optimization problems, the finite sets of results are selected

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

among all feasible results, which satisfy certain constraints. Generally, the problems are related to finding the optimal path in vehicular routing, for the job and task scheduling, like preparing the timetable for trains in railways, solving knapsack problems, or any other engineering problem [8,9]. Different techniques were taken so far to provide the solution to these problems, which include Iterative Improvement, Simulated Annealing, Evolutionary Computation, Variable Neighborhood Search, Iterative Local Search, Meta-Heuristic Techniques, and Heuristic Techniques [10–12]. Currently, there are a variety of optimization techniques available to solve different kinds of problems [13–15]. Bat Algorithm (BA) is a meta-heuristic technique, used for optimization and has gained popularity in past years. It has proven to be beneficial and result-oriented in different fields. It was formulated in 2010 and worked based on echolocation [16,17]. In BA, artificial bats emit a pulse in different directions and wait for the echo to receive. Once the echo of all emitted pulses is received, artificial bats compare the computed fitness value [18]. The solution, which has minimum/maximum fitness value, is considered the best possible solution [19,20]. The selection of minimum or maximum fitness value depends on the problem and choice of either maximizing or minimizing the solution [21,22].

Cloud Computing gained popularity due to its pay-per-use facility and enabled users to access different services. There are three main users of cloud computing: Service Provider, Cloud Broker, and Cloud User [23]. Services offered by any cloud provider include offering infrastructure, platform, and software, to the users, as a service [24,25]. Nowadays, some cloud providers offer storage as a service and have implemented different platforms for cloud security. Cloud consists of virtual machines, which enables cloud users to use cloud services [26–28]. Whenever a user requests for the execution of any task/job, a virtual machine is assigned. The decision regarding which virtual machine will process the request depends upon the load balancer. The load balancing algorithm's task is to select the most appropriate virtual machine for the execution of the task [29,30].

Different optimization techniques have been used so far for efficient load balancing [31,32]. The BA has proved its applicability in this area as well. Work in this field of research, the nature-inspired bat algorithm is used to maintain the balance distribution among available virtual machines in the cloud. To this end, an improved/modified version of the bat algorithm is deployed [33–35]. This modified version of the bat algorithm works by considering distance as a fitness value [36–42].

The main advantages or objectives of doing the research is to:

- It provides a more optimized and Modified Bat Algorithm.
- The research provides a better technique classification which results in enhanced performance and reduces the execution cost.
- The study provides a Modified Bat Algorithm (MBA) which allows balancing the load among virtual machines.

This work has been divided into various sections. The second section highlights existing studies conducted by numerous cloud computing and BA, researchers. The third section describes the Changed Bat load balancing algorithm in cloud computing. The fourth section focuses on conducting the Performance analysis of the Proposed Algorithm, followed in the final portion by the final and future work research work.

2 Related Work

Rout et al. [1] carried out a review and analysis of existing load balancing techniques and the associated challenges. The author outlined the benefits and disadvantages of several methods for load balancing. Furthermore, the author categorized these methods by replication, speed, heterogeneity,

SPOF, overhead network, geographical dispersion, implementation complexity, and fault tolerance. Veeraiah et al. [2] suggested algorithm for resource programming with dynamic load balancing characteristics. The method used data processing, data transmission, and virtual machine latency to identify the optimum virtual machine. A method for balancing load was compared to a random resource scheduling approach. The load-balance-based approach beat the random method in terms of average task response time, throughput, and efficiency.

Suryanarayana et al. [3] mentioned different cloud computing research areas, which need researchers immediate attention to enhance the productivity of cloud. A researcher balancing load in the cloud has difficulties, the author says. The issues were automated service provisioning, virtual machine migration, virtual machine geographic distribution, energy management, and algorithm complexity.

Afzal et al. [4] classified load unbalancing as multi variant and multi constraint problem of cloud computing. The paper offered insight into the factors affecting load-balancing problem. These factors included the dynamic nature of jobs, traffic flow towards cloud service provider, lack of efficient scheduling algorithm, uneven distribution of jobs across available computing resources and lack of robust mapper function for mapping jobs to appropriate resources.

Hsieh et al. [5] proposed a three-phase dynamic data replication algorithm, which was validated on Hadoop, based cloud environment. This algorithm used the concept of virtualization and replication methods to reduce computing costs. It was proven to be an efficient algorithm in both densely and loosely coupled environments.

Jyoti et al. [6] reviewed existing load balancing and service brokering techniques of cloud computing environment. In addition, the author explained the factors affecting the performance of load balancing techniques/algorithms. Further, parameters affecting the performance of service brokers were also explored. Research challenges of both load balancing and service brokering were also highlighted.

Panda et al. [7] proposed a probabilistic load-balancing method. It used a 2-approximation method with temporal complexity O (MN), where M and N are tasks and virtual machines (VM). The results were validated using the standard deviation of VM loads, minimum, maximum, and zero loads. Dey et al. [8] presented a comprehensive study of virtual machine migration techniques, along with their challenges and advantages. The author also explored mathematical models of different task scheduling techniques and identified parameters to be improved. These parameters included meantime before shut down, meantime before migration, service level agreement performance degradation, energy consumption and so on. Pan et al. [9] developed an online load balancing algorithm to improve overall user experience and system performance. Here, task-level scheduling was done, and the task was considered as the graph's vertices and job was considered the graph itself. The author used CloudSim to simulate the functionality of online algorithm.

Alotaibi et al. [10] proposed layered architecture to balance the load and resource allocation in the cloud computing environment. It consisted of six layers, namely, end users (at top most layer), application layer, request dispatcher, load balancer, resource allocator and resource registration table, and resources (at bottom layer). This layered architecture was suitable for all size companies and worked on the client server model.

Chawla et al. [11] proposed a novel package-based load balancing algorithm in the cloud computing environment. Here, resources were grouped as a package in each virtual machine. Only the VM containing the requested package responded, reducing the overall cost of the service provider

and user's execution. CloudSim was used to simulate the algorithm and results indicate reduction in execution time and waiting time.

Ghomi et al. [12] categorized load-balancing techniques broadly into six categories. These categories were Agent-based, Hadoop Map Reduce, workflow specific, General, application-oriented, Natural Phenomena-based, and network-aware. Bharany et al. [13] proposed an improvised bat algorithm, and it worked on the concept of min-min, max-min, and alpha-beta pruning algorithm, for task scheduling in cloud computing. To generate the population of artificial bats, three algorithms were used and to determine the sequence in which virtual machines would execute tasks would execute tasks, bat algorithm was implemented.

Jha et al. [14] reviewed the existing work done by numerous researchers who had developed different types of nature-inspired algorithms. Few researchers enhanced the performance of existing algorithms, either by hybridizing with other existing techniques or by introducing key features. Furthermore, author highlighted the key areas of future work, i.e., how integrating biological characteristics of bat can lead to the development of novel bat algorithm variant. Ghose et al. [15] elaborated pursuit strategies adopted by bats while capturing preys. The author mentioned two strategies: constant bearing (CB), and constant absolute and target detection (CATD). Considering the movement of prey, whether it is moving in predictable fashion or moving erratically, CB or CATD strategy, was applied, respectively.

Chen et al. [16] proposed a novel variant of bat algorithm, i.e., guidable bat algorithm which integrated doppler effect. Moreover, conflict theory was used in evolutionary computing to improve the efficiency of search discovery. Mirjalili et al. [17] developed a binary bat algorithm, as a standard bat algorithm only optimizes continuous problems. The mapping of the algorithm was done using V-shaped transfer functions. The result validation was done over total of 24 unimodal, multimodal, and composite benchmark functions. Compared with the results achieved were superior than current particulate swarm optimization and genetic algorithm.

Li et al. [18] proposed a variant of bat algorithm, i.e., complex valued bat algorithm. In the complex valued algorithm, real and imaginary values were updated separately, which enhanced the diversity of the population. It was proven as a flexible and effective algorithm, when tested over 14 benchmark functions. Yang [19] developed a meta-heuristic approach for optimizing solutions, which was inspired by bats and named as bat algorithm. Due to its automatic zooming, and parameter tuning, the bat algorithm proved to be more efficient than particle swarm optimization and genetic algorithm. Kurdi et al. [20] proposed a locust inspired cloud computing scheduling method. Here, a decentralized approach was followed by servers for allocating jobs to virtual machines. It saved energy consumption in data centers. The findings were confirmed for a dynamic voltage-frequency scaling, workload-aware planning and a static threshold with minimal usage policy.

Khoda et al. [21] proposed a system for intelligent computational offloading for 5G mobile devices. This system's motive was to reduce the energy consumption and latency. Its motive was to reduce the latency and reduce energy consumption while offloading data from cloud servers to mobile devices. The results were computed considering four factors: execution or implementation time, prediction accuracy, energy consumption and execution saving time. Sayantani et al. [22] used a bio-inspired cognitive model to schedule the tasks for IoT applications in a heterogeneous cloud computing environment. Here, the genetic algorithm (GA) was hybridized with ant colony optimization algorithm (ACO) and proved beneficial as it reduced makespan compared to ACO and GA used separately.

Asma et al. [23] proposed a mobility-aware optimal resource allocation architecture, namely, Mobi-Het for big data task execution in a mobile cloud computing environment. This architecture

had three main components, namely master cloud, cloudlet, and mobile device. This architecture was evaluated through test-bed implementations in a dynamic environment, and results were computed in terms of workload distribution, success percentage, and execution time.

Jodayree et al. [27] proposed a rule-based workload dynamic balancing algorithm based on predictions of incoming jobs and named Cicada. The motive of Cicada was to allocate cloud services more efficiently in comparison to static allocation algorithms. Further, the author also proposed C-rule algorithm for predicting incoming workload.

This work aims to apply the biological characteristics of bats to enhance the functionality of BA. Therefore, this research paper explores the flight behavior of bats. Bats adopt either the Constant Bearing strategy or the Constant Absolute Target Detection strategy. In this work, the main focus has been to balance the load among virtual machines of the Cloud Computing environment using CATD strategy. To this end, a new variant of the Bat Algorithm, which is inspired by CATD strategy of bats, is proposed and applied to solve load balancing. A new technique of determining fitness value has been deployed here, i.e., fitness value is computed considering 'range between target and bat' as a parameter. A detailed explanation of the proposed algorithm is given in the subsequent sections.

3 Modified Bat Algorithms for Load Balancing

This section describes the proposed algorithm for selecting the optimal virtual machine for the execution of tasks in the Cloud Computing environment while balancing the load. In order to find the best solution, the distance between artificial bats and solutions is calculated in magical way. The advancements introduced in standard bat algorithm, revolve around either parameter initialization or parameter update or hybridization with other techniques or mapping continuous space problems to binary space problems. In these advancements to BA, very few authors have adopted different strategies to compute distance. In this research paper, a different way of computing distance has been suggested, which is equivalent to the strategy adopted by real bats while targeting their target (prey). Further, the modified bat algorithm has been applied for balancing the load in cloud by using CATD inspired Bat Algorithm. The steps for balancing the load in Cloud Computing using Modified Bat Algorithm are described in pseudo-code. There are four main phases of Modified Bat Algorithm, which include- Initialization of Parameters, Computation of Fitness Value, Selection of Optimal VM and Ensuring the optimal VM is not overloaded.

4 Performance Analysis of the Proposed Algorithm

To assess updated bat algorithm performance while selecting best suited virtual machine (optimal virtual machine), results are contrasted with the results of standard bat algorithm, when applied for balancing the load in cloud computing environment. Here, jobs correspond to bats and targets/preys correspond to Virtual Machine. The goal is that the bats should target those preys which have more energy level. More energy level corresponds to more fitness value. Lesser the value of distance, more is the fitness value. So, jobs will be sent to those virtual machines which are closer to client location (where jobs are dispatched to carry out). The number of jobs/tasks to be assigned to available virtual machines is varied over [10,15,20] and a number of artificial bats of bat population are varied over [10,15,20] for minimum 10 iterations. The value of parameters is defined below in Tab. 1:

Table 1: Parameter description

Parameter	Description	Value
Fmin	Frequency lower bound	0
Fmax	Frequency upper bound	2
A	Constant	0.9
B	Constant	0.9
A	Loudness	0>A<1
R	Pulse emission rate	0>R<1

Furthermore, results are compared using standard deviation, mean, median, best and worst values of the results obtained both a modified conventional bat algorithm. The factors considered for evaluation include execution time and cost. The results are then evaluated in two aspects. Firstly, a modified bat method is used for load balancing across the virtual machines available. During this evaluation, it is noticed that the optimal virtual machine may get overloaded, while other virtual machines are under loaded [33–35]. This variant of modified bat algorithm is named as Overloaded Optimal Virtual Machine (OOVM). To preserve the equilibrium between overloaded and underloaded virtual machines, an advancement is introduced in modified bat algorithm, where the selected optimal virtual machine is not assigned any task for evaluation, if its existing task count exceeds the threshold value. This variant of modified bat algorithm is named as Balanced Virtual Machine (BVM) [36]. The results of modified bat algorithm for an overloaded optimal virtual machine, on the basis of cost and implementation time are shown in Tabs. 2 and 3, respectively.

Table 2: Performance evaluation of MBA-OOVM on the basis of implementation time

On the basis of execution time		Standard bat algorithm			Modified bat algorithm (OOVM)		
VM	Parameters	Bat population					
		10	15	20	10	15	20
10	Best	4.030	4.503	4.823	3.976	4.250	4.633
	Median	4.070	4.379	4.832	4.058	4.337	4.728
	Worst	4.223	4.982	4.901	4.123	4.407	4.804
	Mean	4.079	4.632	4.862	4.057	4.337	4.727
	Standard deviation	0.040	0.043	0.047	0.039	0.037	0.038
15	Best	4.151	4.343	4.386	4.187	4.552	4.684
	Median	4.241	4.432	4.993	4.273	4.645	4.780
	Worst	4.349	4.423	4.999	4.342	4.720	4.857
	Mean	4.213	4.446	4.343	4.272	4.644	4.779
	Standard deviation	0.041	0.040	0.048	0.041	0.046	0.046
20	Best	4.233	4.777	4.983	4.234	4.889	5.142
	Median	4.124	4.982	4.783	4.321	4.989	5.248
	Worst	4.381	5.012	5.012	4.391	5.070	5.333
	Mean	4.230	4.992	4.784	4.320	4.988	5.247
	Standard Deviation	0.042	0.049	0.049	0.042	0.046	0.047

Table 3: Performance evaluation of MBA-OOVM on the basis of cost

On the basis of cost		Standard bat algorithm			Modified bat algorithm (OOVM)		
VM	Parameters	Bat population					
		10	15	20	10	15	20
10	Best	155.263	165.977	180.914	152.263	162.977	177.914
	Median	156.746	167.111	182.150	153.746	164.111	179.150
	Worst	158.147	168.097	183.225	155.147	165.097	180.225
	Mean	156.757	167.093	182.131	153.757	164.093	179.131
	Standard deviation	0.100	0.082	0.070	0.092	0.089	0.077
15	Best	163.492	177.761	182.905	160.502	174.771	179.915
	Median	165.053	178.975	184.154	162.063	175.985	181.164
	Worst	166.528	180.031	185.241	163.538	177.041	182.251
	Mean	165.065	178.956	184.135	162.075	175.966	181.145
	Standard deviation	1.048	0.068	0.070	0.099	0.067	0.070
20	Best	165.356	190.915	200.815	161.516	187.075	196.975
	Median	166.934	192.220	202.187	163.094	188.380	198.347
	Worst	168.426	193.354	203.380	164.586	189.514	199.540
	Mean	166.946	192.199	202.166	163.106	188.359	198.326
	Standard deviation	1.060	0.074	0.077	1.060	0.071	0.067

The results of modified bat algorithm are shown in algorithm number 1 for a balanced virtual machine on the basis of cost and implementation time are shown in Tabs. 4 and 5, respectively. It is noticed that while applying standard bat algorithm for balancing the load of VM's, for 10 VM's, if lesser number of bats are deployed, still an optimal result could be obtained with lesser cost, but the standard deviation of the obtained results is high [37]. But, when 15 VM's are considered and 10 bats are deployed, it is observed that standard deviation was high, but the cost is less. If 15 bats are deployed for 15 VM's, the standard deviation is reduced but the cost increased, whereas using 20 bats for 15 VM' is not suitable due to the increase in cost and standard deviation. So, for 15 VM's, 15 bats should be deployed for optimal result.

Algorithm 1: Modified Bat Algorithm and Pseudocode of Proposed Algorithm

Data: Input Bats, N and Virtual Machine number, V.
Set min_freq, max_freq, velocity, pulse emission rate, loudness and position for entire bat population.

Result: Selection of best suited virtual machine

(Continued)

Algorithm 1: Continued

Begin

For i = 1 to V

Compute the fitness value of every available virtual machine V.

1. Deploy 'N' number of bats, where each bat is responsible for computing the fitness value of V which is present in search space.
2. Every bat will emit pulse and received echo for the computation of distance.
3. Detect the presence of any kind of obstacle.
   When the solution is present, delay, β and attenuation, α, as per following equation.
   $Echo_i = (Pulse\_Emitted_i * rand_1) + rand_2$
   otherwise
   $Echo_i = Pulse\_Emitted_i + rand_2$
4. Calculate the similarity between the generated sound $Pulse\_Emitted_i$ and $Echo_i$ using mathematical function, cross correlation and compute delay samples.
   [Correlation]= xcorr($Pulse\_Emitted_i$, $Echo_i$)
   DelaySample= Lags(find(Correlation==maximum(Correlation)))
5. Distance can be computed, using DelaySample and TimeSample.
6. Select solution as a best which is hving minimum fitness value.

end for

Select first virtual machine as the best local solution, having minimum fitness value.

for i = 1 to V

if VM_i == visited VM(i, :)

Increment variable and check for other VM's assigned load.

end if

To balance the load, the task is to be assigned to under loaded virtual machine, even in the presence of virtual machine which is having lesser fitness value than selected virtual machine

if count(i,:)>threshold

Compute the fitness value, again and select the optimal virtual machine.

end if

End

Table 4: Performance evaluation of MBA on the basis of implementation time

On the basis of execution time		Standard bat algorithm			Modified bat algorithm (BVM)		
VM	Parameters	Bat population					
		10	15	20	10	15	20
10	Best	4.030	4.503	4.823	4.045	4.351	4.723
	Median	4.070	4.379	4.832	4.078	4.391	4.794
	Worst	4.223	4.982	4.901	4.732	4. 412	4.801
	Mean	4.079	4.632	4.862	4.067	4.311	4.872
	Standard deviation	0.040	0.043	0.047	0.038	0.038	0.037

(Continued)

Table 4: Continued

On the basis of execution time		Standard bat algorithm			Modified bat algorithm (BVM)		
VM	Parameters	Bat population					
		10	15	20	10	15	20
15	Best	4.151	4.343	4.386	4.173	4.442	4.785
	Median	4.241	4.432	4.993	4.363	4.691	4.673
	Worst	4.349	4.423	4.999	4.413	4.812	4.866
	Mean	4.213	4.446	4.343	4.176	4.773	4.671
	Standard Deviation	0.041	0.040	0.048	0.043	0.043	0.043
20	Best	4.233	4.777	4.983	4.332	4.777	4.992
	Median	4.124	4.982	4.783	4.213	4.671	4.675
	Worst	4.381	5.012	5.012	4.399	5.016	5.031
	Mean	4.230	4.992	4.784	4.312	4.773	4.673
	Standard Deviation	0.042	0.049	0.049	0.048	0.046	0.047

Table 5: Performance evaluation of MBA on the basis of cost

On the basis of cost		Standard bat algorithm			Modified bat algorithm (BVM)		
VM	Parameters	Bat population					
		10	15	20	10	15	20
10	Best	155.263	165.977	180.914	149.293	159.157	174.944
	Median	156.746	167.111	182.150	150.776	160.291	176.180
	Worst	158.147	168.097	183.225	152.177	161.277	177.255
	Mean	156.757	167.093	182.131	150.787	160.273	176.161
	Standard deviation	0.100	0.082	0.070	0.954	0.815	0.647
15	Best	163.492	177.761	182.905	156.672	170.941	177.495
	Median	165.053	178.975	184.154	158.233	172.155	178.744
	Worst	166.528	180.031	185.241	159.708	173.211	179.831
	Mean	165.065	178.956	184.135	158.245	172.136	178.725
	Standard deviation	1.048	0.068	0.070	1.011	0.585	0.564

Considering 20 VM's with the deployment of 10 bats, there will be a trade-off between standard deviation and cost. Standard deviation will increase and the cost will decrease. If the number of bats is increased from 10 to 15 for 20 VM's, then the standard deviation will decrease and cost will increase. For 20 bats, standard deviation and cost both increases. So, for 20 VM's, 15 bats are suitable to obtain optimal results. While evaluating the results of modified bat algorithm-overloaded optimal virtual machine variant, it was observed that it had lesser cost and less variation in the values of standard deviation in comparison to the results obtained using standard bat algorithm for solving the same problem, if 10 bats are used for 10 virtual machines. In the case of 15 virtual machines, one should prefer the deployment of 15 bats, as it aims at lesser standard deviation values at a lesser cost. For 20 VM's, cost increases as the number of bats increases. So, an optimal solution can be obtained by

deploying 15 bats. As depicted in Tab. 3, the cost of selecting optimal virtual machine is lesser than the standard bat algorithm.

While evaluating the performance it is noticed on the basis of implementation time that the variation among the optimal results obtained reduced and the difference between the execution time of standard bat algorithm and modified bat algorithm-OOVM version became almost negligible up to three decimal points. The results show that the algorithm of the modified bat-OOVM produced more optimal results in comparison to the standard bat algorithm, while considering cost as the factor. The results which are computed on the basis of worst, best, standard deviation, mean, and median values by varying the number of bats present in bat population and number of virtual machines, are represented in graphs below. The results are given on a performance basis in Figs. 1–3. Fig. 1 represents the varying values of performance evaluation parameters for 10 virtual machines and varying bat population from [10,15,20]. Similarly, Figs. 2 and 3 represent the results for 15 and 20 virtual machines, for varying bat population, respectively. Further, with the inclusion of threshold value to limit the over utilization of the optimal virtual machine, BVM version of Modified Bat Algorithm has been introduced. The results obtained using a modified bat algorithm for balanced virtual machine are better in comparison to the results obtained using standard bat algorithm, as depicted in Tabs. 4 and 5.

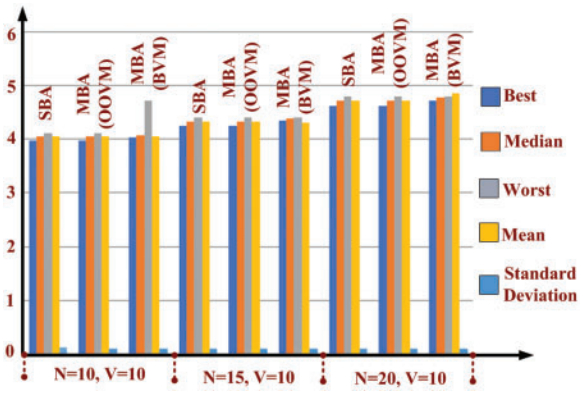

Figure 1: Comparison result graph on the basis of execution time of SBA, MBA-OOVM and MBA-BVM for V = 10 and N varying between

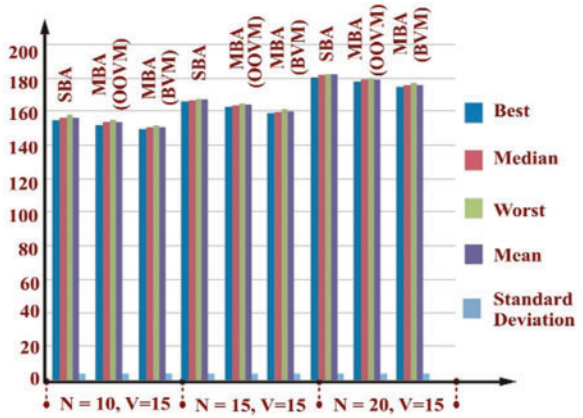

Figure 2: Comparison result graph on the basis of execution time of SBA, MBA-OOVM and MBA-BVM for V = 15 and N varying between

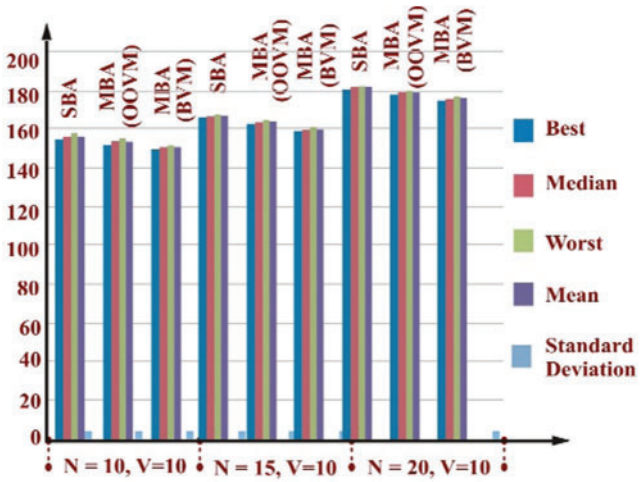

Figure 3: Comparison result graph on the basis of execution time of SBA, MBA-OOVM and MBA-BVM for V = 20 and N varying between

For 10 VM's, if the number of bats deployed is increased, the time to obtain the optimal result also increases, but the standard deviation among the feasible solution decreases. For 15 VM's, deployment of 15 bats will be suitable while applying standard bat algorithm and modified bat algorithm-BVM. For 20 VM's, deployment of 15 bats will obtain optimal results and balance the load amongst all of the cloud-based virtual computers.

The outcomes were evaluated based on the cost of the whole procedure, are depicted in Figs. 4–6. Fig. 4 represents the varying values of performance evaluation parameters for 10 virtual machines and varying bat population from [10, 15 and 20]. Similarly, Figs. 5 and 6 represent the results for 15 and 20 virtual machines for varying bat population, respectively [10,15,20]. Fig. 4 displays the comparing findings graphical representation of the cost required to select the optimal virtual machine for the execution of jobs/tasks by appointing 10 bats for 10 virtual machines. Results prove that deployment of a greater number of bats will not improve the results and selection of an optimal virtual machine can be done by deploying only 10 bats.

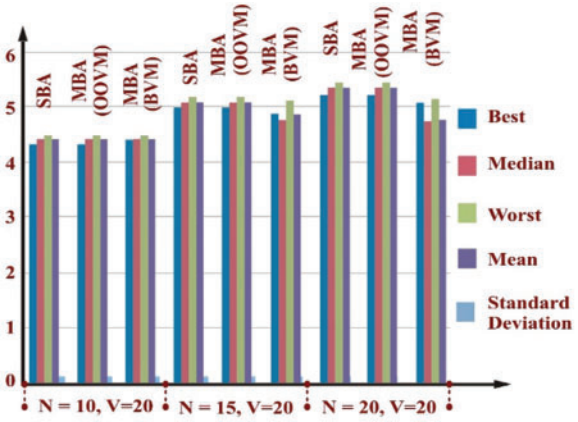

Figure 4: Comparison result graph on the basis of cost of SBA, MBA-OOVM and MBA-BVM for V = 10 and N varying between

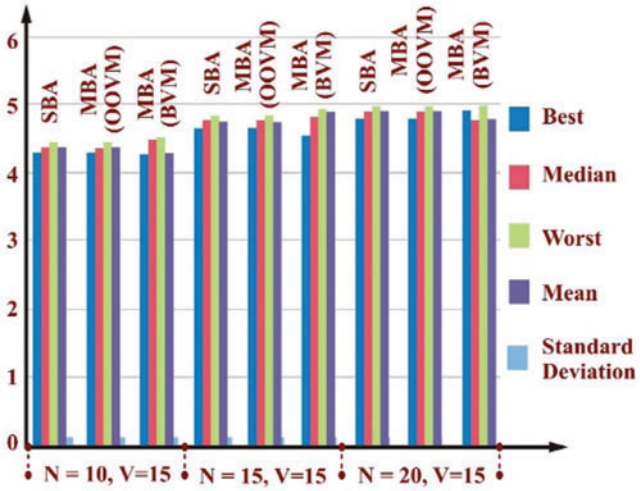

Figure 5: Comparison result graph on the basis of cost of SBA, MBA-OOVM and MBA-BVM for V = 15 and N varying between

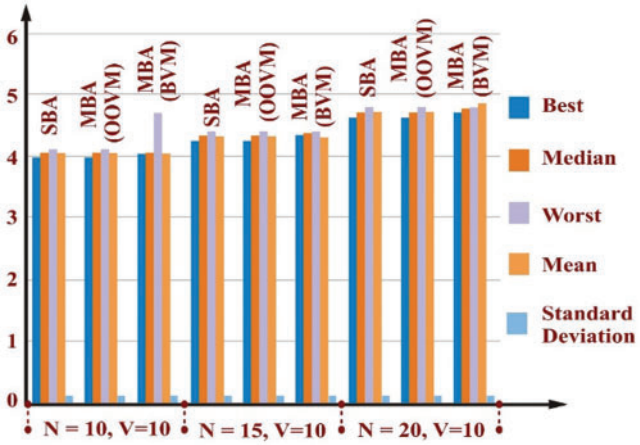

Figure 6: Comparison result graph on the basis of cost of SBA, MBA-OOVM and MBA-BVM for V = 20 and N varying between

Fig. 5 depicts that for 15 virtual machines, deployment of 15 bats will serve the purpose. If we deploy a greater number of bats, it will not improve the performance, but will lead to an increase in cost. Fig. 6 depicts that for 20 virtual machines, 15 bats are sufficient to select the optimal virtual machine [10,15,20]. In earlier related literature works, main focus was to ensure the optimal the use and balance of the load between these computers by virtual machines, in Cloud Computing environment. Consequently, number of jobs submitted to best/optimal VM's, may overburden them. In order to avoid such situation and to ensure that the jobs are assigned to all virtual machines in a balanced way, two new variants of Bat Algorithm are designed. It is clearly evident from the results that the best mean value obtained using Modified Bat Algorithm-BVM had outperformed Modified Bat Algorithm-OOVM and Standard Bat Algorithm. Except in the case of 15 bats and 15 virtual machines, the results of Standard Bat Algorithm and Modified Bat Algorithm-BVM are quite similar.

5 Performance Evaluation Using Mathematical Benchmark Functions

In this section, the proposed algorithm performance of optimization is evaluated with respect to various mathematical benchmark optimization functions. The proposed algorithm of this research work is tested on 13 mathematical benchmark functions, with various properties (unimodal and multimodal). In order to investigate proposed algorithm performance of this research work, 13 benchmark functions apply to the proposed variant of Bat Algorithm. The parameters considered for evaluation of proposed algorithm includes best, mean, median, standard deviation and worst. The run is successfully executed for 500 iterations and for bat population equal to 50. The results of comparison are depicted in Tab. 6.

Table 6: Comparison of proposed algorithm results for F1 to F13 benchmark functions

Function	Parameters	Bounds	Optimal value	Modified BA
Rosenbrock	Best	[−30, 30]D	0	7.60E−02
	Median			9.54E−02
	Worst			4.20E−08
	Mean			1.39E−01
	SD			2.26E−01
Sphere	Best	[−100, 100]D	0	6.96E−02
	Median			8.47E−02
	Worst			6.62E−07
	Mean			1.02E−01
	SD			1.18E−01
Rastrigin	Best	[−5.12, 5.12]D	0	5.25E−02
	Median			1.45E−01
	Worst			6.30E−07
	Mean			1.89E−01
	SD			2.59E−01
Griwank	Best	[−600, 600]D	0	1.00E+00
	Median			1.00E+00
	Worst			4.20E−08
	Mean			1.00E+00
	SD			1.05E+00
Schaffer	Best	[−100, 100]D	0	1.10E−04
	Median			1.79E−04
	Worst			4.20E−08
	Mean			2.98E−04
	SD			3.99E−04
B2	Best	[−100, 100]D	0	1.24E−01
	Median			2.76E−01
	Worst			4.13E−01
	Mean			2.85E−01
	SD			3.43E−01

(Continued)

Table 6: Continued

Function	Parameters	Bounds	Optimal value	Modified BA
Zakharov	Best	[−10, 10]D	0	1.79E−02
	Median			2.42E−02
	Worst			4.20E−08
	Mean			3.67E−02
	SD			5.64E−02
Goldstein and Price	Best	[−2, 2]D	3	7.96E−01
	Median			7.41E−01
	Worst			4.20E−11
	Mean			7.59E−01
	SD			8.04E−01
Ackley	Best	[−32, 32]D	0	1.40E+00
	Median			9.26E−01
	Worst			3.56E−06
	Mean			8.69E−01
	SD			1.14E+00
Branin	Best	[−10, 10]D	0.397887	8.77E−01
	Median			1.04E−01
	Worst			4.20E−08
	Mean			3.29E−01
	SD			5.28E−01
Easom	Best	[−100, 100]D	−1	−2.33E−05
	Median			−2.37E−05
	Worst			4.20E−08
	Mean			−3.21E−02
	SD			1.06E−01
Hartmann	Best	[0, 1]D	−3.86278	−3.14E+00
	Median			−3.15E+00
	Worst			4.20E−08
	Mean			−3.13E+00
	SD			3.30E+00
Shubert	Best	[−10, 10]D	−186.7309	9.64E−02
	Median			9.70E−02
	Worst			4.20E−08
	Mean			1.22E−01
	SD			1.70E−01

In order to obtain best optimal solution, Hartmann function can be used as benchmark function as it offers minimum and optimal solution. In case of maximization function, B2 function can be preferred over other functions as it offers maximum value for 50 bat population over 500 iterations. On the basis of Mean and Median, Shubert function can be used for benchmarking as it offers minimum value over 500 iterations and for 50 bats. But in order to reduce gaps between solutions obtained in search space, Schaffer function offers best results with respect to standard deviation parameter. Even

though, it does not offer best optimal solution but can be used in those scenarios where motive is to optimize the solutions present in search space. Fig. 7 depicts values of different parameters like best, median, mean, standard deviation and worst, for 13 different functions. If results of all benchmark functions are considered and compared, then based on parameters of interest, different functions will be preferred for different scenarios.

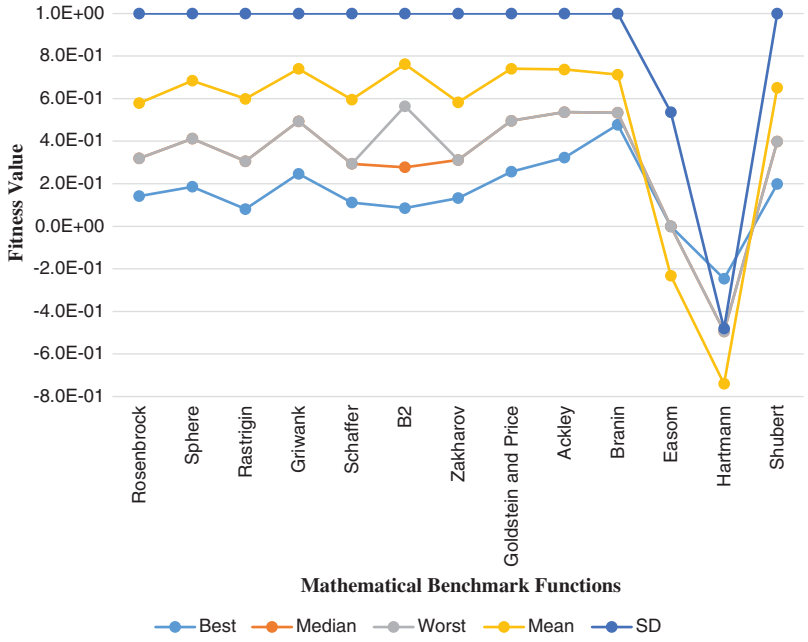

Figure 7: Comparison of modified BA over mathematical benchmark functions

6 Conclusions and Future Work

A novel variant of the bat algorithm, which is inspired by the bat's flight behavior, has been designed for solving combinatorial problems. The different flight behavior adopted by Microchiroptera bats has been studied and modeled mathematically. The motive of this research is to provide an efficient technique for balancing the load of virtual machines available on the Cloud and ensuring that no optimal virtual machine should be overloaded. A BAT algorithm is used for QOS, energy management, resource scheduling, and load balancing in cloud computing. Most academics are working to enhance the BAT algorithm's ability to allocate load across different VMs. A network modification system is implemented through infrastructure layer. To ensure the same, the standard bat algorithm is modified by incorporating the strategy that real bats use while estimating the distance between themselves and their prey and while capturing the target. It improved the performance of the algorithm and the altered bat algorithm was applied to solve the problem of balancing the cloud computing load. The results computed have proven the applicability of a modified bat algorithm to balance the load on cloud and generated more optimal results. Further, to enhance the algorithm performance and increase its applicability to other fields, the standard bat algorithm can be hybridized with other newly developed meta-heuristic techniques. Real Bats can jam the pulse emitted by other bats or receive the echo and target the prey of other real bats. This astonishing feature also motivates to development of another bat algorithm variant. Moreover, real bats adopt different pursuit strategies depending on the movement of prey, depending on the prey to capture, and many additional factors.

One can research these areas to propose a new bat algorithm with improved performance. There can also be potential research to develop other variants of standard bat algorithm or to enhance the proposed algorithm performance, other biological features of the bat can also be explored.

Acknowledgement: We deeply acknowledge Taif University for supporting this study through Taif University Researchers Supporting Project Number (TURSP-2020/313), Taif University, Taif, Saudi Arabia.

Funding Statement: This research is funded by Taif University, TURSP-2020/313.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Rout, P. Parida, Y. Alotaibi, S. Alghamdi and O. I. Khalaf, "Skin lesion extraction using multiscale morphological local variance reconstruction-based watershed transform and fast fuzzy c-means clustering," *Symmetry*, vol. 13, no. 11, pp. 2085, 2021.

[2] Badotra S. and S. Narayan Panda "Evaluation and comparison of OpenDayLight and open networking operating system in software-defined networking," *Cluster Computing*, vol. 23, no. 2, pp. 1281–1291, 2020.

[3] G. Suryanarayana, K. Chandran, O. I. Khalaf, Y. Alotaibi, A. Alsufyani *et al.,* "Accurate magnetic resonance image super-resolution using deep networks and Gaussian filtering in the stationary wavelet domain," *IEEE Access*, vol. 9, pp. 71406–71417, 2021.

[4] S. Afzal and G. Kavitha, "Load balancing in cloud computing–A hierarchical taxonomical classification," *Journal of Cloud Computing*, vol. 8, no. 1, pp. 22, 2019.

[5] H. C. Hsieh and M. L. Chiang. "The incremental load balance cloud algorithm by using dynamic data deployment," *Journal of Grid Computing*, vol. 17, no. 3, pp. 553–575, 2019.

[6] A. Jyoti, M. Shrimali, S. Tiwari and H. P. Singh "Cloud computing using load balancing and service broker policy for IT service: A taxonomy and survey," *Journal of Ambient Intelligent Humanized Computer*, vol. 11, no. 11, pp. 4785–814, 2020.

[7] S. K. Panda and P. K. Jana, "Load balanced task scheduling for cloud computing: A probabilistic approach," *Knowledge and Information Systems*, vol. 61, no. 3, pp. 1607–1631, 2019.

[8] N. S. Dey and T. Gunasekhar, "A comprehensive survey of load balancing strategies using hadoop queue scheduling and virtual machine migration," *IEEE Access*, vol. 7, pp. 92259–92284, 2019.

[9] J. Pan, P. Ren and L. Tang, "Research on heuristic-based load balancing algorithms in cloud computing," *Intelligent Data Analysis and Applications*, vol. 370, pp. 417–426, 2015.

[10] Y. Alotaibi, M. N. Malik, H. H. Khan, A. Batool, S. U. Islam *et al.,* "Suggestion mining from opinionated text of big social media data," *Computers, Materials & Continua*, vol. 68, no. 3, pp. 3323–3338, 2021.

[11] A. Chawla and N. S. Ghumman, "Package-based approach for load balancing in cloud computing," *Big Data Analytics*, vol. 654, pp. 71–77, 2018.

[12] E. J. Ghomi, A. M. Rahmani and N. N. Qader, "Load-balancing algorithms in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 50–71, 2017.

[13] S. Bharany, S. Sharma, S. Badotra, O. I. Khalaf, Y. Alotaibi *et al.,* "Energy-efficient clustering scheme for flying Ad-hoc networks using an optimized LEACH protocol," *Energies*, vol. 14, no. 19, pp. 6016, 2021.

[14] B. Sumit and S. Narayan Panda"SNORT based early DDoS detection system using Opendaylight and open networking operating system in software defined networking," *Cluster Computing*, vol. 24, no. 1, pp. 501–513, 2021.

[15] K. Ghose, T. K. Horiuchi, P. S. Krishnaprasad and C. F. Moss, "Echolocating bats use a nearly time-optimal strategy to intercept prey," *PLoS Biology*, vol. 4, no. 5, pp. 108, 2006.

[16] B. Sumit and J. Singh "Creating firewall in transport layer and application layer using software defined networking," In *Innovations in Computer Science and Engineering*, pp. 95–103. Springer, Singapore, 2019.
[17] S. Mirjalili, S. M. Mirjalili and X. S. Yang, "Binary bat algorithm," *Neural Computing and Applications*, vol. 25, no. 3, pp. 663–81, 2014.
[18] L. Li and Y. Zhou, "A novel complex-valued bat algorithm," *Neural Computing and Applications*, vol. 25, no. 6, pp. 1369–81, 2014.
[19] X. S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization*, vol. 284, pp. 65–74, 2010.
[20] H. A. Kurdi, S. M. Alismail and M. M. Hassan, "LACE: A locust-inspired scheduling algorithm to reduce energy consumption in cloud datacenters," *IEEE Access*, vol. 6, pp. 35435–35448, 2018.
[21] M. E. Khoda, M. A. Razzaque, A. Almogren, M. M. Hassan, A. Alamri *et al.,* "Efficient computation offloading decision in mobile cloud computing over 5G network," *Mobile Networks and Applications*, vol. 21, no. 5, pp. 777–792, 2016.
[22] B. Sayantani, M. Karuppiah, K. Selvakumar, K. C. Li, S. K. H. Islam *et al.,* "An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment," *Future Generation Computer System*, vol. 88, pp. 254–261, 2018.
[23] B. Sumit and A. Sundas, "A systematic review on security of E-commerce systems," *International Journal of Applied Science and Engineering*, vol. 18, no. 2, pp. 1–19, 2021.
[24] M. F. Shirjini, A. Nikanjam and M. A. Shoorehdeli, "Stability analysis of the particle dynamics in bat algorithm: Standard and modified versions," *Engineering with Computers*, vol. 18, pp. 78–88, 2020.
[25] W. Lin, G. Peng, X. Bian, S. Xu, V. Chang and Y. Li, "Scheduling algorithms for heterogeneous cloud environment: Main resource load balancing algorithm and time balancing algorithm," *Journal of Grid Computing*, vol. 17, no. 4, pp. 699–726, 2019.
[26] P. G. Gopinath and S. K. Vasudevan, "An in-depth analysis and study of load balancing techniques in the cloud computing environment," *Procedia Computer Science*, vol. 50, pp. 427–432, 2015.
[27] M. Jodayree, M. Abaza and Q. Tan, "A predictive workload balancing algorithm in cloud services," *Procedia Computer Science*, vol. 159, pp. 902–912, 2019.
[28] M. Kumar and S. C. Sharma, "Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing," *Procedia Computer Science*, vol. 115, pp. 322–329, 2017.
[29] M. Kumar, K. Dubey and S. C. Sharma, "Elastic and flexible deadline constraint load balancing algorithm for cloud computing," *Procedia Computer Science*, vol. 125, pp. 717–724, 2018.
[30] Y. Alotaibi, "A new secured e-government efficiency model for sustainable services provision," *Journal of Information Security and Cybercrimes Research*, vol. 3, no. 1, pp. 75–96, 2020.
[31] S. Palanisamy, B. Thangaraju, O. I. Khalaf, Y. Alotaibi and S. Alghamdi, "Design and synthesis of multi-mode bandpass filter for wireless applications," *Electronics*, vol. 10, no. 22, pp. 2853, 2021.
[32] V. Priya, C. S. Kumar and R. Kannan, "Resource scheduling algorithm with load balancing for cloud service provisioning," *Applied Soft Computing*, vol. 76, pp. 416–424, 2019.
[33] B. Sumit and S. Narayan Panda, "A survey on software defined wide area network," *International Journal of Applied Science and Engineering*, vol. 17, no. 1, pp. 59–73, 2020.
[34] X. Cai, J. Zhang, H. Liang, L. Wang and Q. Wu, "An ensemble bat algorithm for large-scale optimization," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 11, pp. 3099–3113, 2019.
[35] Q. Jiao and D. Xu, "A discrete bat algorithm for disassembly sequence planning," *Journal of Shanghai Jiaotong University*, vol. 23, no. 2, pp. 276–285, 2018.
[36] W. A. Neto, M. F. Pinto, A. L. Marcato, I. C. da Silva and D. D. A. Fernandes, "Mobile robot localization based on the novel leader-based bat algorithm," *Journal of Control, Automation and Electrical Systems*, vol. 30, no. 3, pp. 337–346, 2019.
[37] Y. Alotaibi, "A new database intrusion detection approach based on hybrid meta-heuristics," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1879–1895, 2021.
[38] N. Talbi, "Design of fuzzy controller rule base using bat algorithm," *Energy Procedia*, vol. 162, pp. 241–250, 2019.

[39] X. Yue and H. Zhang, "Improved hybrid bat algorithm with invasive weed and its application in image segmentation," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9221–9234, 2019.
[40] N. Subramani, P. Mohan, Y. Alotaibi, S. Alghamdi and O. I. Khalaf, "An efficient metaheuristic-based clustering with routing protocol for underwater wireless sensor networks," *Sensors*, vol. 22, no. 2, pp. 1–16, 2022.
[41] S. Badotra and S. N. Panda, "A review on software-defined networking enabled IoT cloud computing," *IIUM Engineering Journal*, vol. 20, no. 2, pp. 105–126, 2019.
[42] T. Sarvesh, S. Badotra, M. Gupta and A. Rana, "Efficient and secure multiple digital signature to prevent forgery based on ECC," *International Journal of Applied Science and Engineering*, vol. 18, no. 5, pp. 1–17, 2021.