The Journal of Supercomputing (2020) 76:3374–3390
https://doi.org/10.1007/s11227-018-2583-3

Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC–BA)

K. Karthikeyan[1] · R. Sunder[2] · K. Shankar[3] · S. K. Lakshmanaprabu[4] · V. Vijayakumar[5] · Mohamed Elhoseny[6] · Gunasekaran Manogaran[7]

Published online: 5 September 2018

Abstract
A cloud data center consumes more energy for computation and switching servers between modes. Virtual Machine (VM) migration enhances the execution of cloud server farm in terms of energy proficiency, adaptation to internal failure, and accessibility. Cloud suppliers, be that as it may, ought to likewise enhance for amounts like energy consumption and administrations costs and in this manner, trying to have all the Virtual Machines with the least measures of physical equipment machines conceivable. The part of virtualization is critical and its execution is subjected to VM migration and machine allotment. A greater amount of the energy is caught up in the cloud; consequently, the use of various calculations is required for sparing energy and productivity upgradation in the proposed work. In the proposed work, the Naive Bayes classifier with hybrid optimization using Artificial Bee Colony–Bat Algorithm (ABC–BA) was implemented to reduce the energy consumption in VM migration. The proposed method was evaluated in CloudSim and the performances were compared using performance index such as success &failure rate, and energy consumption. It is observed from the implementation results that the proposed method reduces energy consumption compared to other existing methods. From the implementation outcomes of the proposed work, it was understood that the model was able to achieve the minimum energy consumption and failure rate i.e., 1000–1200 kWh, 0.2 with maximum success rate and accuracy of 1 and 97.77%.

Keywords Cloud computing · Virtual Machine · Failure detection · Prediction · Optimization · And energy consumption

✉ V. Vijayakumar
vijayakumar.v@vit.ac.in; viji_06@hotmail.com

Extended author information available on the last page of the article

1 Introduction

Cloud computing enables a few associations among the clients to utilize an assortment of users without introducing or getting their own records on any compact computer connected with web [1]. ‘Data focuses’ utilize virtualization and host an arrangement of Virtual Machines (VMs) onto an arrangement of Physical Machines (PMs) thus by enabling effective asset usage. Productive asset usage has been one of the fundamental drivers amid the server farm development [2]. For instance, VMs can be moved to start with one PM followed by the next keeping in mind the end goal to kill the source PM and spare energy. A VM migration is straightforward to applications and present-day virtualization innovation invariantly bolsters it [3]. The virtualization innovation presents the programming deliberation layer called as Virtual Machine Monitor or hypervisor. Essentially, two virtualization approaches are utilized in general [4]. VM migration is relied upon, since it is quick and VM benefit debasement ought to be low. Live migration occurs in a quick manner and straightforward in order to keep consistent administration. The current procedure utilizes pre-duplicate approach for the migration process [5].

Virtualization wound up the mainstream for installed frameworks to achieve better asset usage, yet the adaptation to non-critical failure is another viewpoint to be considered here [6]. The materialness of VM migration, as an adaptation to internal failure system, is inspected. To enhance the dependability (progression for revise benefit), migration is executed as administration retrieval because of equipment faults [6] that keep the framework inside VM to comply with the utilitarian or potentially-continuous detail [7]. The energy sparing impact of the long haul perspective and the execution of the disappointment rate of migration occasions are confirmed. Further, other two trials were performed to make a relative parameter [8] which was tried and the ideal power administration arrangement was acquired, which adds in adjusting the energy cost and punishment cost because of execution infringement while at the same time, it also limits the aggregate cost and least energy consumption [9]. Any failure, likewise, brings about investing more opportunities for finishing the cloud administration and increasing the expansion time for recouping the disappointment both of which additionally implies more energy consumption. Along these lines, energy consumption is additionally a subsequent characteristic of dependability [10]. The current state data is to be classified either as typical or anomalous according to the probabilities identified with dominant part class. An information point would be identified as class 2 if its likelihood of having a place with class 1 would be too low [11].

A semi-supervised approach would build a model in light of the lion’s share class [12]. During true situations, the execution of a cloud administration might get hindered due to different disappointments, especially VM disappointments and server disappointments in cloud computing conditions [13]. This suggests that the fulfillment time of the cloud benefit has a considerable association with unwavering quality variables which in addition also implies that it might be anything, but a steady yet an arbitrary esteem [14].One productive strategy to enhance the usage of the assets and decrease the energy consumption is dynamic combination of VM empowered by ‘virtualization innovation.’ This reduces the general energy consumption by the framework [15]. Its gauge and the optimization are the imperative parts of planning in the energy design undertaken. Whenever required, it is reactivated to suit new VMs or VMs that are migrated [16, 17]. In a perfect world, the energy consumption of all server hubs is corresponding to the aggregate asset utilization of all VMs. VM union permits specialist co-ops to kill extreme energy consumption that are not utilized for clients' calculation [18]. In the viewpoint of the discussions made above, the optimization of energy consumption in a processing plant should be made possible by actualizing [13] and the adjusting the line voltage through equally discarding the line conductors and by limiting the heap irregular characteristics.

The rest of the paper is discussed as follows. The second section discussed about the recent literatures regarding Virtual Machine migration and energy consumption with optimization process whereas third section shows the preliminaries of the current proposed model. The detailed discussion of the proposed methodology is presented under Sect. 4 whereas Sect. 5 discusses about the implementation results and comparative study of proposed method which is finally concluded in the sixth section.

2 Literature review

A joined, gauge-based Virtual Machine migration in cloud server farms was proposed by Paulraj et al. in the year 2018 [19].They proposed the consolidated anticipating method to foresee the asset necessity of any Virtual Machine. In view of the current and the anticipated asset usage, live migration was performed by Combined Forecast Load-Aware strategy. The analyses were completed to assess the execution of the proposed procedure in live VM migration. The results demonstrated that the proposed approach has limited quantity of migrations, energy use, and the message overhead when compared with the current condition of crafts manship strategy.

Live migration of Virtual Machines (VMs) is a fundamental element of virtualization since it permits VMs to move from one area and then onto the next without suspending VMs, according to the study by Mostafa Noshy et al. in 2018 [20]. It provides a superior understanding of live migration of Virtual Machines and its fundamental methodologies. In particular, it focuses around exploring the state-of-the-workmanship optimization procedures that are committed to boost live VM migration as per memory migration. It surveys, talks about, breaks down and looks at these procedures to understand their optimization and their difficulties. This work additionally featured open research issues that required encouragement and examination to optimize the procedure of live migration in case of Virtual Machines.

Anita Choudhary et al. classified the types of Live VM migration (single, various and half and a half) in 2017 [21]. They arranged VM migration systems in view of the duplication instruments (replication, de-duplication, repetition, and pressure), paid attention to the setting (reliance, delicate page, grimy page, and page blame) and assessed different live VM migration methods. In this study, the authors discussed about varieties of execution measurements like application service downtime, total migration time and amount of information exchanged. The core theme of this work was that it showed the foundation of live VM migration procedures and an inside-out survey which is useful for cloud experts and specialists to additionally investigate the difficulties and to provide optimal solutions.

In 2015, Inderjit Singh Dhanoa et al. [22] conducted a study in which it was discussed that live Virtual Machine migration in server farms can possibly diminish energy consumption up to a certain level of utilization. Regardless of this, VM size and system bandwidth greatly affect the energy consumption of sub-frameworks amid VM live migration. In order to consider this study, the paper was segregated and the effect of migration time on energy consumption of sub-frameworks amid visitor VM Live Migration was taken out.

Virtualization has increased in a recognizable size at most extreme statures in each division of the business according to Sadia Ansari et al. [23]. They proposed a system to recognize the hypervisor assaults in Virtual Machines. Here the study utilized a Bayesian classifier on freely-accessible dataset. They portrayed vulnerabilities of two Hypervisors such as XEN and VMware, in view of constant values. Three characteristics such as authentication, respectability effect, and confidentiality affect were considered as information including vector. When experimented, the authors figured the back probability of weakness which shows its level, being a hypervisor assault.

In 2018, FAN Chengli et al. [24] added the variable neighborhood seek factor to arrangement look condition which has improved the nearby inquiry capacity and the populace-assorted variety. Furthermore, the memory system was made advanced through the motivation attained by the neuroscience examination of genuine bumblebees. In this method, the fake honey bees can be expected to recall their past successful encounters and further guide the ensuing foraging conduct.

3 Preliminaries to the current work

Failure tolerance is a noteworthy issue to ensure accessibility and dependability of basic services and further the execution of application. Adaptation to internal failure fills comes as a compelling attention to address unwavering quality concerns. Adaptation to non-critical failure implies that the framework should keep continuing with the operations under blame nearness. In these circumstances, the current study administration module must be set up for all conditions. The conditions of the hosts are changed progressively in a continuous manner in the light of the comparing workloads. Nonetheless, it is by accomplishing more accurate arrangement of the present issue window, it can more be drawn out toward the term energy sparing optimization impact. Responsive adaptations to internal failure approaches decrease the impact of disappointments on application execution when the disappointment viably happens. From existing optimization strategies that had not delivered accurate energy consumption arrangement of the present issue window, it is more drawn out toward the term energy sparing optimization impact.

4 Methodology

VMM is performed starting with one physical machine and then onto the other machine [25]. It is utilized for stack adjusting and physical machine blame tolerant. Migration of a VM tries to make sensibility, execution and adaptation better in a non-critical

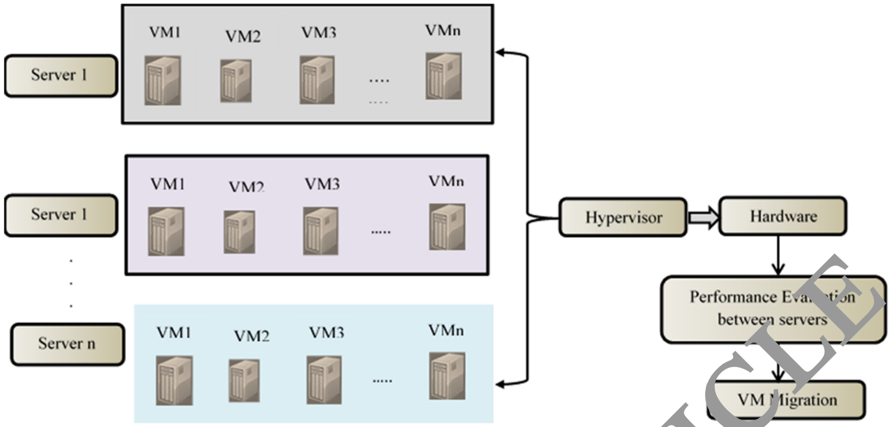

Fig. 1 VMM model

failure of a framework. To be precise, it substantiates the heap-adjusting highlight of a datacenter' through migrating VMs out of the over-stacked server to an under-stacked server or physical machine. This work i.e., the proposed work that concentrates on VMM cloud data centers, energy sparing and machine selection process utilize inventive model. The prepared units in cloud conditions (VM) are utilized to fabricate the foundation of a cloud by interconnecting virtualized data centers of expansive scales whereas the assets used for computing are delivered to client over web as on-demand service. This methodology, at first considers the assets, data unit and the number of jobs (machines) to the prediction procedure so as to discover failure and non-failure VM in the cloud framework. This investigation was performed by utilizing Minkowski Distance in which there is a trouble in classifying the covered hyper-box, which was settled by additionally presenting a base separation-based classifier. In addition, the failure rate and the success rate of failed VMs are investigated utilizing swarm based optimization i.e., ABC with Bat algorithm. From this usage, the result are produced as energy sparing and optimal machine selection of cloud data units. A detailed clarification of the current proposed methodology is depicted in upcoming sections.

4.1 Virtual Machine migration

VM strategy is one of the generally utilized strategies in cloud computing which is utilized for actualizing these three services alongside VM migration procedures utilized for supporting virtualized cloud computing data centers and it is shown in Fig. 1. The reduction of energy consumption of data focus is the fundamental research zone to keep the nature of service. Virtual Machine, at the source host, is stopped and then all the conditions of the source host were exchanged with the objective or goal host followed by which, the working of virtual machine was resumed at the objective host [26]. This migration procedure has two imperative parameters which are discussed below.

Down Time Downtime alludes to the time during which the service of the VM is not accessible. Migration makes feasible for VMs without much extensive downtime.

Migration Time It alludes to the total amount of time required to exchange a Virtual Machine at the source to goal hub without influencing its availability. Virtualization is the real idea of cloud computing which is gaining importance in mainstream in cloud computing environments.

Correspondingly, a few scientists have been contemplating this topic in which live VM migration innovation occurs and then VMs are moved, keeping in mind the end goal to satisfy the prerequisite of execution and workload confinement while limiting the energy drawn by a cloud data focus. At a specific point, when the VM just started migrating from one machine to another, the mapping between the physical machine and the VM ought to be reflected in the VM's state in an effective manner before it continues on the host system.

4.2 Failure detection in cloud system

This is a typical approach to execute fault tolerant ability with the help of essential and optional reinforcement systems. The optional reinforcement is always present if the essential server fizzles. The condition of the auxiliary server ought to be same as the primary server. In previous times, failure prediction approach managed VM failures proactively as opposed to waiting for failures to happen and then respond to it. Nonetheless, the proactive approach requires a failure to be unsurprising. The dynamic cloud condition is done [27] by the clouds stage through powerfully-setting off a scope of 'close-down occasions' or 'failure occasions' of physical hosts during the interim time of the machine selection and energy consumption process. Different sorts of failure cloud system in VM process which generally has components such as follows.

Responsive Failure Tolerance It diminishes the impact of failures on the execution of the application when failure successfully occurs. There are different systems which depend on these strategies like Checkpoint/Restart, Replay and Retry and so on.

Proactive Failure tolerance A proactive Fault Tolerance Arrangement is to evade recuperation from faults, blunders, and failures by foreseeing them and proactively overcoming it through presumed parts by other working segments. A portion of the systems which depend on these approaches is Preemptive Migration, Software Rejuvenation and so forth.

From the proposed show, the failures on 'n' energy consumption and the number of invalid VM migration occasions are recognized. The number of failure in VM migration occasions will increase in ascending rate of failure of hosts. Failures in constructed VMs and hosting servers are dealt as various types of failures in the current unwavering quality model as they has distinctive repair activities.

4.3 Failure prediction model

Prediction is a method in machine learning which can be utilized to foresee the gathering participation of various information occurrences. It plays the assignment using

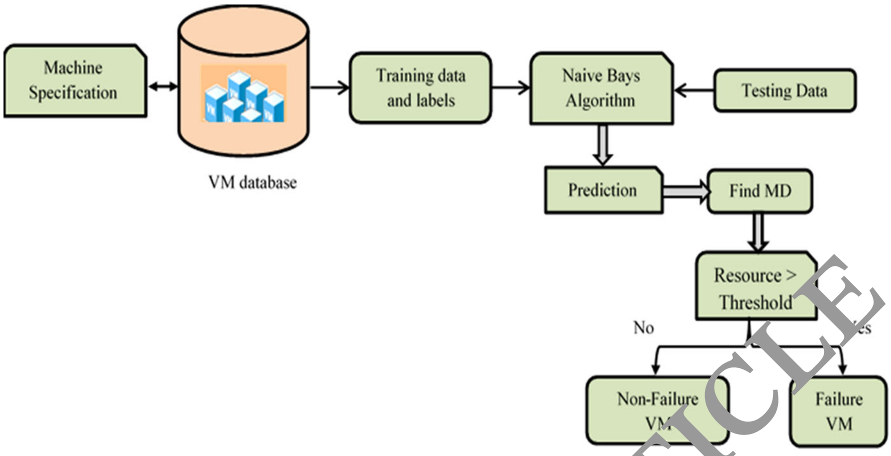

Fig. 2 NB for failure prediction

which it sums the outstanding structure to apply it upon new information. The proposed classifier (Naive Bayes) is a failure prediction model for logical work processes in light of proactive adaptation for non-critical failure. The failure impact of work process assignments on the Cloud assets and execution had been reduced by machine learning strategies. The failure VMs are relocated to fresh area to spare the assets.

4.3.1 Naive Bayes classifier

Naive Bayesian classifier is a measurable classifier according to Bayes' Theorem which is the most extreme posterior theory. A more spellbinding term for the fundamental probability model would be 'free distance-based model.' Naive Bayes classifiers accept that the nearness or nonattendance of a VM data like time, memory utilization and size to foresee the failure and non-failure machine as illustrated in Fig. 2. So, to clarify naive Bayesian classifier [24, 28], let $C_n = \{M_1, M_2, \ldots, M_n\}$ be a case from a dataset, whose component esteems were made on an arrangement of 'n' characteristics. To decrease the calculation cost, an extra supposition was made so that the traits are indistinguishably conveyed. This implies that the probability of experiencing a particular word is autonomous of the particular word position.

Probability Function In the current study situation, three scores are separately considered without including the rest of the scores. The probability display of a predictor is a contingent model which can be defined as follows

$$\mathrm{Proab}(M_1, M_2, \ldots, M_n) \tag{1}$$

According to Bayes Theorem,

$$\mathrm{Proab}(M_1, M_2, \ldots, M_n) = \frac{1}{md}\mathrm{proab} \prod\nolimits_{i=1}^{n} \mathrm{proab}(M_i) \tag{2}$$

where *md* is the distance which stands as a scaling factor. The naive Bayes classifier consolidated this model with a choice run the show. One basic manage is to pick the theory which is most plausible and this is known as the greatest posteriori or choice run the show.

Minkowski Distance Minkowski distance, the most utilized uniqueness measure, is assessed by summing the differences between two information focuses as far as all machines are concerned. The distance between two focuses like settled edge and resilience esteems which is explained in the equation below.

$$md(i, j) = \sum \left(\left|m_{ij} - m_{jk}\right|\right)^{1/q} \tag{3}$$

The objective to pick Eq. (3) is to make the proposed system adaptable since the naïve Bayesian classifier is one of the fastest predictors which was already mentioned before in this section. The reason behind the choice of NB classifier is that, it has higher prediction accuracy with reduced calculation time and so it has to be utilized in the cloud framework which basically requires early prediction of VM failure.

4.4 Optimization model for migration process

In a standard point of view, the optimization process can be described to find the best solution of the function from the system within constraints [29–31]. As of late, minimizing the energy consumption in server farms has turned into a principle focal point for cloud suppliers in failed Virtual Machines. The heuristic optimization i.e., ABC is built with BA with a consideration to settle unique workload designation choices and considering energy use when choosing the position of VMs. The proposed way to deal with sparing energy and the most extreme execution in migrant VMs are moved into comparison of ideal target with a specific edge. The system which underscores the execution metric may not meet the ideal target of sparing energy consumption.

4.4.1 Artificial Bee Colony Optimization (ABC)

ABC algorithm is a populace and predicated heuristic algorithm that reproduces searching conduct according to honey bee swarms model. This procedure is anything but difficult to actualize and make it successful. In this method, the food focuses are changed or adjusted through moving conduct of honey bees (artificial bees) over the span of time. ABC optimization process contains three stages such as employed bees, onlooker bees and the scout bees. Three types of bees share the data of food sources with each other.

The food sources need to be found out and the information regarding the data of food sources need to be conveyed to onlooker bees in the hive. The obligation of utilized bees is to abuse the food sources. But the onlooker bees choose whether to misuse the food sources or not, as indicated by the data shared by the utilized bees. Scout bees attempt to locate another food source through random looking. One food source in ABC stands for a conceivable answer for the optimization issue.

4.4.2 Bat Algorithm (BA)

In nature, bats are entrancing creatures. Small-scale bats use a kind of sonar called echolocation to identify the prey, maintain a strategic distance from hindrances, and find their perching hole oblivious. When flying and chasing, bats transmit some short ultrasonic heartbeats [32] to the earth and proceed according to their echoes. Studies demonstrate that the data from echoes empowers bats to construct an exact picture of their environment and take decision according to the distance, shapes and the prey's area.

4.4.3 Proposed optimization: hybrid ABC with BA

In the canonical Artificial Bee Colony algorithm, the new food source is created utilizing an annoyance that originated from a randomly chosen bee in a random measurement. In ABC optimization scout bee stage, the failure machines and particulars are selected randomly and so the issue arise i.e., not achieving ideal energy consumption. So, the researchers considered BAT optimization which is a refreshing method at this stage while a flowchart of crossbreed optimization strategy is illustrated in Fig. 3.

4.4.4 Generally, this proposed model has the following procedures

1. Initialize the failure VM specification
2. Generally, the population helps in the initialization process
3. Employed bee Phase
4. Onlooker bee Phase
5. Scout Bee phase
   Bat updating
6. Memorize the best food source
7. Until

Steps The equation given below mentions that the placement selection is initialized with the number of failure VM, time and jobs and the colony size in addition to the number of employed or onlooker bees. Let represent the $F_{\mathrm{VM}n}$ food source,

$$\mathrm{IS} = \{F_{\mathrm{VM}1}, F_{\mathrm{VM}2}, \ldots\ldots, F_{\mathrm{VM}n}\} \tag{4}$$

Objective of the proposed model

The energy consumption is the standard energy consumed through hubs that contribute in message which is then passed on from essential hub to the objective hub. The objective of the utility capacity is to boost the VM arrangement by minimizing the energy consumption discussed in the equation below. Lower estimations of energy consumption help in decreased usage. This denotes that lower the measure of energy devoured, better the task will be.

$$\mathrm{EC}_i = \min\{\mathrm{energy}_{\text{ideal machine, } i} + \mathrm{energy}_{\text{running machine, } i}\} \tag{5}$$

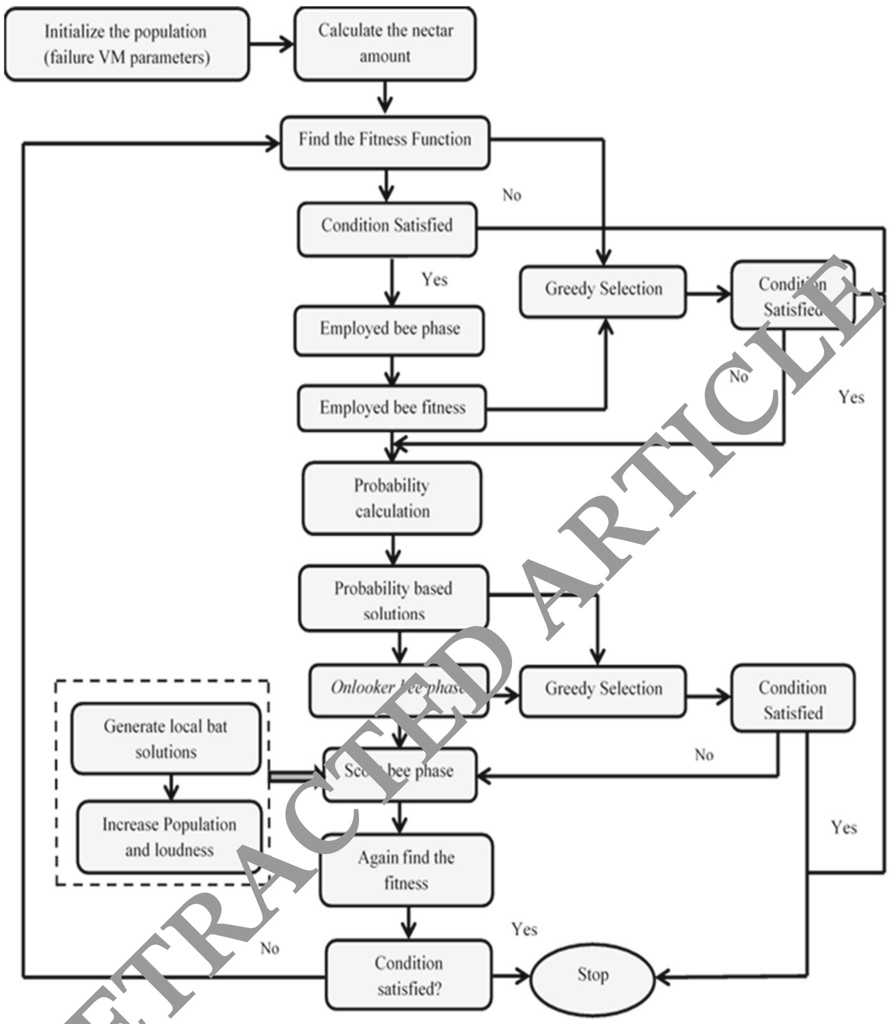

Fig. 3 Hybrid ABC-BA

(1) ABC phase: Employed bee

At this stage, each employed bee looks around the given arrangement. Employed bee flies to a food source and finds another food source inside the area of the food source. High amount of food source was chosen. The food source data was put away by employed bee was imparted to onlooker bees. Employed bee is at where the new arrangement is produced under the given condition.

$$\text{Employed be for EC} = F_{ij} + \alpha_{ij}(F_{ij} + F_{kj}) \tag{6}$$

In the above condition, α_{ij} runs from -1 to 1 which shows the file position and k was decided randomly $k \in$ {1, 2…size (i)} with the centrality limitation in this procedure being $k \neq I$. Then, a ravenous determination system is connected among F_i and F_k to choose the superior arrangement. After all the employed bees finish their pursuits, they share the arrangement data to the onlooker bees.

(2) Onlooker bee phase

Onlooker bees watch the waggle move in the move zone and ascertain the productivity of the food sources and then a higher food source was selected randomly [23]. After that, the onlooker bees undergo a randomly look in the area of food source.

(3) Probability function

In onlooker bee stage, the random determination methodology was accustomed to search for nearby optimization esteem in the area of food source and the highest probability arrangement was picked by onlooker bees.

$$\text{Proab} = \frac{\text{EC}_i}{\sum_{i=1}^{\text{NS}} \text{EC}_i} \tag{7}$$

The purpose behind the probability rate of achieved wellness is after the eager determination among the employed and onlooker bees' stage. After finishing the probability figuring process, the achieved arrangement was found to be as per the request that the wellness has brought down the probability need rate.

(4) Scout Bee phase

The Scout bee occurs when the left arrangements happen and it is rehashed consistently until the point when as far as possible, the reason is prompted for the random arrangement which has happened already. This arrangement is assumed to be abandoned and another arrangement is delivered randomly in the hunting space to supplant the abandoned one. Here the movement of virtual bats, loudness and heartbeat rate are considered in view of the recurrence shape bat optimization process.

Movement of virtual bats, Loudness and pulse rate

Keeping in mind the end goal i.e., enhancement of nearby hunt capacity of the algorithm, Yang has made a complete structure altogether in which the bat can enhance the arrangement close to the received one. The tumult a beat emanation rate refreshed as a bat draws nearer to its objective, in particular its prey.

$$F_{\text{new}} = F_{\text{old}} + \gamma K^t \tag{8}$$

where F_{old} is a high quality solution chosen by some mechanism, K^t denotes the average loudness of all the bats at tth time step and ε is a randomly-generated value that ranges from − 1 to 1. As the tumult, more often than not, diminishes, once a bat has found its prey, the rate of heartbeat discharges increases. A bat has recently found the prey and the din due to which the heartbeat discharge rate is refreshing which is given by the formula.

$$K_j = \beta * K_j^t \tag{9}$$

The standard bat algorithm has numerous points of interest and one of them is that it can merge at faster rates during introductory stages by changing from investigation

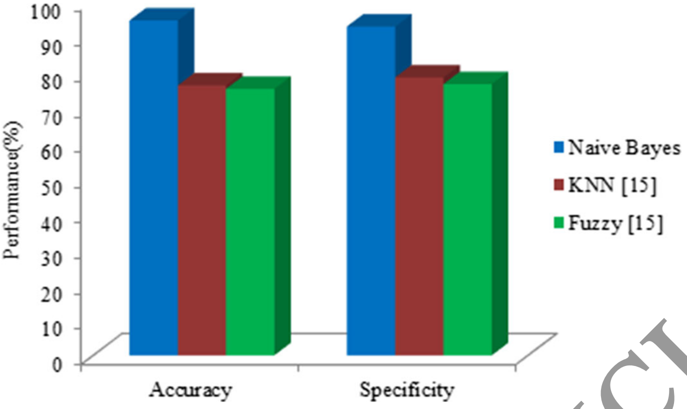

Fig. 4 Prediction results

to abuse. This makes it a productive algorithm where a fast arrangement is required. To empower VM energy consumption with the end goal to make it as an ideal arrangement, the previously-mentioned conditions need to be utilized to manage the molecule position refresh activity.

5 Performance evaluation results

To assess the execution of the proposed method, a demonstration was conducted which was actualized in java programming with JDK 1.7.0 out of a windows machine containing configurations such as Intel (R) Core i5 processor with cloud test system. This reenactment display considered 50 machines in excess considering arranged sources. This evaluation procedure was measured in few metrics in view of various migration process.

5.1 Success rate

It was characterized by the probability of successful migration after a migration being inquired for, was received by Open Nebula orchestrator. The fundamental purpose behind the migration failure is that a migration is not a nuclear activity, i.e., but it is made up of various individual tasks. Every operation, in any case, can come up short which brings about entire migration failure.

5.2 Failure rate

The failure rate of a framework ordinarily relies upon a time with the rate differing over the existence cycle of the framework. Many failure occasions in VM migration might be delivered by virtue of the related physical hosts' non-accessibility.

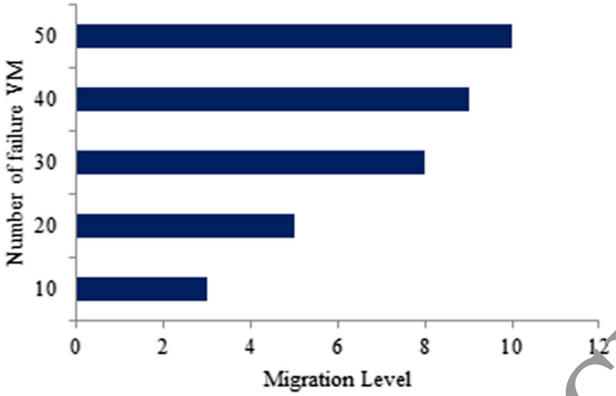

Fig. 5 Migration results

Prediction and migration process execution are illustrated in Figs. 4, 5. The proposed naive Bayes classifier foresee the greatest accuracy and specificity such as 92.3% and 94.56% for VM arranging model when compared with other classifiers. Ordering VMs, according to the CPU utilization proportion and memory utilization proportion is discussed as a nonstop time arrangement. Likewise, utilizing characterization for VMs as rates is definitely not an unpredictable strategy, it also diminishes the cost and time and there is no loss of process time for looking to locate an accessible VM. On the off chance which contributed 20 implies, according to the condition i.e., threshold and tolerance esteems-based group, the contribution is considered as a failure or non-failure machine. Then Fig. 5 demonstrates the migration process with different VMs. In the event that the rate of refreshing pages is high, migration time ascends to a high esteem. Yet, the benefit of this approach is that all refreshing can be accessed at the goal. It can be actuated whenever and here the number of migrations increase as the quantity of Virtual Machines increase for the purpose of expansion of designation of VMs among neighborhood machines rather than remote machines.

Energy consumption of the proposed model is illustrated in Fig. 6. It can be inferred that, when compared with Particle Swarm optimization (PSO), dynamic adaptive PSO(DAPSO) and ABC, the proposed model demonstrate good results. Here, in this model, the quantity of virtual machines is less due to which the energy consumption is also less, yet as the quantity of Virtual Machines increases, the energy consumption too increments. Therefore, the host with a more-grounded figuring power which demonstrates that it has a superior 'execution per watt' for the situation must be considered and the appraised control is similar which leads to less energy consumption. In the event that fluctuating the quantity of Virtual Machines to at least the energy of 1200 for hybrid ABC-BA approach, when compared all VMs with the proposed method, the energy at least was only in half breed. VM attributes thereby diminished the aggregate system resource consumption and aggregate energy consumption in the server farm.

Figure 7a, b demonstrates the success and failure rate of VM migration process. In the event that VMs are to be moved far from failure-inclined hosts, the failure

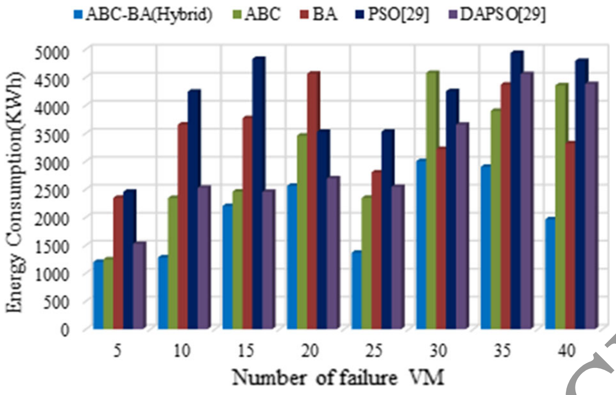

Fig. 6 EC versus number of VM

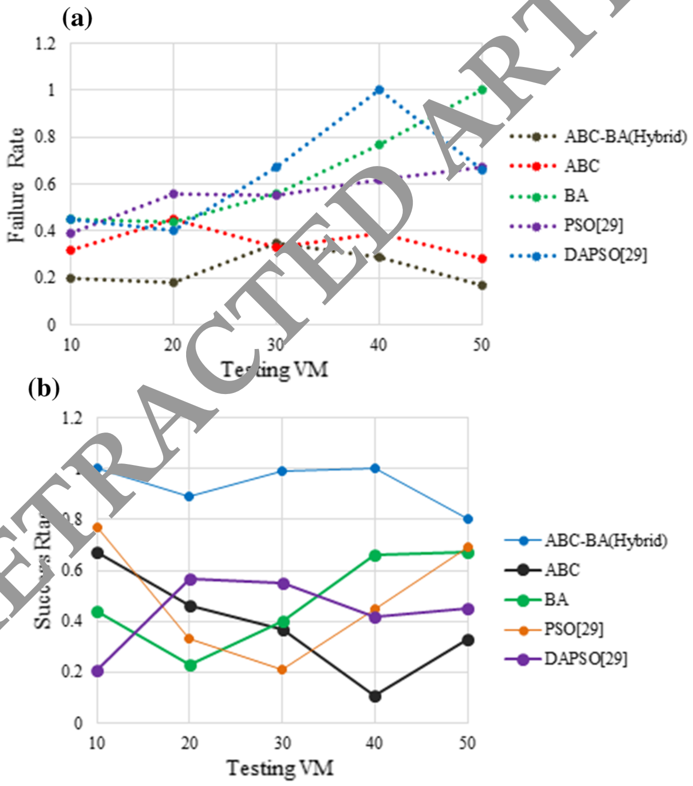

Fig. 7 Compassion for Performance. **a** failure rate, **b** success rate

prediction algorithm needs to anticipate failures in the future than the aggregate term of migration. The capacity with which crossbreed ABC-BA adjusts to a dynamic domain is additionally inside a specific range whereas the quantity of failure in VM migration occasions increases as per the ascending rate of failure of hosts. It not only limits the incremental energy consumption of a cloud server farm, but also limits the quantity of failure in VM migration occasions moderately. From this analysis, the least failure rate and the maximum success rate were achieved for the proposed optimization process where these two measurements were assessed for various optimization systems. The migration strategy is not a nuclear task and it comprises of numerous consecutive advances which are commonly needy. This implies that each time a migration is performed, then the migration procedure must be checked with a specific end goal to affirm the effective migration achievement.

6 Conclusion

In this paper, Naive Bayes with hybrid optimization demonstrated to limit the energy consumption for VM migrations in distributed computing. From this exploration, possibility of prediction of VM failure is exhibited along with the outcomes that utilize cloudsim. The outcomes reveal that the proposed approach yielded better outcomes in prediction of VM failure in cloud server farms. In the analysis of the quantity of failures in VM migration occasions, the ABC-BA approach has less number of failures in VM migration occasions than that of random migration and ideal migration. It has been moved from servers that neglect to fulfill the heap adjust condition to the goal servers and the load mindful migration algorithm was utilized. The execution of the proposed framework is compared with previous studies and contracted with the current strategies using achievement and failure rate, and energy consumption as measures. In future, innovative classifier and optimization with different performance measures should be considered for VM migration.

References

1. Refaat TK, Kantarci B, Mouftah HT (2016) Virtual machine migration and management for vehicular clouds. Veh Commun 4:47–56
2. Liu Y, Gong B, Xing C, Jian Y (2014) A virtual machine migration strategy based on time series workload prediction using cloud model. Math Probl Eng 2014. https://doi.org/10.1155/2014/973069
3. Papadopoulos AV, Maggio M (2015) Virtual machine migration in cloud infrastructures: problem formalization and policies proposal. In: 2015 IEEE 54th Annual Conference on Decision and Control (CDC), IEEE, pp 6698–6705
4. Kaur Ramandeep (2017) A hybrid approach for virtual machine migration in cloud computing environment. J Adv Res Comput Sci Softw Eng 7(4):30–35
5. Alonso-Monsalve S, García-Carballeira F, Calderón A (2018) A heterogeneous mobile cloud computing model for hybrid clouds. Future Generat Comput Syst. https://doi.org/10.1016/j.future.2018.04.005
6. Andonovski G, Mušič G, Škrjanc I (2018) Fault detection through evolving fuzzy cloud-based model. IFAC-PapersOnLine 51(2):795–800
7. Yadav RK, Kushwaha V (2014) An energy preserving and fault tolerant task scheduler in Cloud computing. In: 2014 International Conference on Advances in Engineering and Technology Research (ICAETR), IEEE, pp 1–5
8. Cerroni W, Esposito F (2016) Optimizing live migration of multiple virtual machines. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2016.2567381
9. Akram SA, Ghaleb S, Hamaid SB, Vasanthi V (2017) Survey study of virtual machine migration techniques in cloud computing. Migration 177(2):19–22

10. Smara M, Aliouat M, Pathan ASK, Aliouat Z (2017) Acceptance test for fault detection in component-based cloud computing and systems. Future Gener Comput Syst 70:74–93
11. Han L, Weili C (2015) Research on fault diagnosis of rolling bearing based on wavelet packet energy feature and planar cloud model. In: 2015 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), vol 1. IEEE, pp 36–40
12. Satpathy A, Addya SK, Turuk AK, Majhi B, Sahoo G (2017) Crow search based virtual machine placement strategy in cloud data centers with live migration. Comput Electr Eng. https://doi.org/10.1016/j.compeleceng.2017.12.032
13. Qiu X, Dai Y, Xiang Y, Xing L (2017) Correlation modeling and resource optimization for cloud service with fault recovery. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2017.2691323
14. Wahid F, Kim DH (2016) An efficient approach for energy consumption optimization and management in the residential building using artificial Bee colony and fuzzy logic. Math Probl Eng 2016:1–14
15. Qasem GM, Madhu BK (2017) Proactive fault tolerance in cloud data centers for performance efficiency. Int J Pure Appl Math 117(22):325–329
16. Liu J, Wang S, Zhou A, Kumar S, Yang F, Buyya R (2016) Using proactive fault-tolerance approach to enhance cloud service reliability. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2016.2567392
17. Yang X, Guo S, Yang H (2008) The establishment of energy consumption optimization model based on genetic algorithm. In: IEEE International Conference on Automation and Logistics, 2008. ICAL 2008, IEEE, pp 1426–1431
18. Egwutuoha IP, Chen S, Levy D, Selic B, Calvo R (2013) Energy efficient fault tolerance for high-performance computing (HPC) in the cloud. In: 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD), IEEE pp 762–769
19. Paulraj GJL, Francis SAJ, Peter JD, Jebadurai IJ (2018) A combined forecast-based virtual machine migration in cloud data centers. Comput Electr Eng 69:287–300. https://doi.org/10.1016/j.compeleceng.2018.01.012
20. Noshy M, Ibrahim A, Ali HA (2018) Optimization of live virtual machine migration in cloud computing: a survey and future directions. J Netw Comput Appl 110:1–10. https://doi.org/10.1016/j.jnca.2018.03.002
21. Choudhary A, Govil MC, Singh G, Awasthi LK, Pilli ES, Kapil D (2017) A critical survey of live virtual machine migration techniques. J Cloud Comput 6(1):23
22. Dhanoa IS, Khurmi SS (2015) Analyzing energy consumption during VM live migration. In: 2015 International Conference on Computing, Communication & Automation (ICCCA), IEEE, pp 584–588
23. Ansari S, Hans K, Khatri SK (2017) A Naive Bayes classifier approach for detecting hypervisor attacks in virtual machines. In: 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), IEEE, pp 1–6
24. Chengli FAN, Qiang FU, Guangzheng LONG, Qinghua XING (2018) Hybrid artificial bee colony algorithm with variable neighborhood search and memory mechanism. J Syst Eng Electron 29(2):405–414
25. Abdelaziz A, Elhoseny M, Salama AS, Riad AM, Hassanien AE (2017) Intelligent algorithms for optimal selection of virtual machine in cloud environment, towards enhance healthcare service. In: The 3rd International Conference on Advanced Intelligent Systems and Informatics (AISI2017), Sept. 9–11, 2017, Cairo-Egypt, Springer
26. Kaur P, Rani A (2015) Virtual machine migration in cloud computing. Int J Grid Distrib Comput 8(5):337–342
27. Ganesh A, Sandhya M, Shankar S (2014) A study on fault tolerance methods in cloud computing. In: 2014 IEEE International Advance Computing Conference (IACC), IEEE, pp 844–849
28. Hassan MK, El Desouky AI, Badawy MM, Sarhan AM, Elhoseny M, Gunasekaran M EoT-driven hybrid ambient assisted living framework with naïve Bayes–firefly algorithm. Neural Comput Appl https://doi.org/10.1007/s00521-018-3533-y (**in press**)
29. Shankar K, Eswaran P (2017) RGB based multiple share creation in visual cryptography with aid of elliptic curve cryptography. China Commun 14(2):118–130
30. Shankar K, Lakshmanaprabu SK, Gupta D, Maseleno A, de Albuquerque VHC (2018) Optimal feature-based multi-kernel SVM approach for thyroid disease classification. J Supercomput. https://doi.org/10.1007/s11227-018-2469-4
31. Lakshmanaprabu SK, Shankar K, Khanna A, Gupta D, Rodrigues JJ, Pinheiro PR, De Albuquerque VHC (2018) Effective features to classify big data using social internet of things. IEEE Access 6:24196–24204

32. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, Berlin, pp 65–74
33. Zhao J, Hu L, Ding Y, Xu G, Hu M (2014) A heuristic placement selection of live virtual machine migration for energy-saving in cloud computing environment. PLoS ONE 9(9):e108275

Affiliations

K. Karthikeyan[1] · R. Sunder[2] · K. Shankar[3] · S. K. Lakshmanaprabu[4] · V. Vijayakumar[5] · Mohamed Elhoseny[6] · Gunasekaran Manogaran[7]

K. Karthikeyan
adithyakarthi@gmail.com

R. Sunder
sundar.mtech@yahoo.co.in

K. Shankar
shankarcrypto@gmail.com

S. K. Lakshmanaprabu
prabusk.leo@gmail.com

Mohamed Elhoseny
mohamed.elhoseny@unt.edu

Gunasekaran Manogaran
gunavit@gmail.com

[1] Department of Computer Applications, Anna University, Regional Campus, Madurai, Madurai, India

[2] Department of Computer Science and Engineering, Sahrdaya College of Engineering and Technology, Kodakara, Kerala, India

[3] School of Computing, Kalasalingam Academy of Research and Education, Krishnankoil, India

[4] Department of Electronics and Instrumentation Engineering, B. S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India

[5] School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India

[6] CoViS Lab, University of North Texas, Denton, USA

[7] University of California, Davis, USA