Krill Herd Algorithm for Live Virtual Machines Migration in Cloud Environments

Hui Cao[1], Zhuo Hou[2]*
HeNan Open University, Zhengzhou, Henan, China, 450046[1]
Henan Vocational College of Agriculture, Zhengzhou, Henan, China, 450046[2]

***Abstract*—Green cloud computing is a modern approach that provides pay-per-use information and communication technologies with a minimal carbon footprint. Cloud computing enables users to access computing resources without the need for local servers or personal devices to operate applications. It allows businesses and developers to access infrastructure and hardware resources conveniently. Consequently, this results in a growing demand for data centers. It becomes crucial in maintaining economic and environmental sustainability as data centers use disproportionate energy. This points to sustainability and energy consumption as being important topics for research in cloud computing. This paper introduces a two-tiered VM placement algorithm. A queuing model is proposed at the first level to handle many VM requests. Models such as cloud simulation are easily implemented and validated using this model. It also provides an alternate method for allocating tasks to servers. Next, a multi-objective VM placement algorithm is proposed based on the Krill Herd (KH) algorithm. Basically, it maintains a balance between energy consumption and resource utilization.**

Keywords—Cloud computing; migration; virtualization; energy consumption

I. Introduction

Virtualization is a major technology that underpins cloud computing. In order to offer access to mainframe computers in an interactive manner, IBM developed this technology in the 1960s, where multiple users (and applications) could utilize expensive hardware simultaneously [1]. Storage technologies and computing power have made computing resources more abundant, cheaper, and powerful than ever before due to rapid technological advancements [2, 3]. This development has led to the development of cloud computing, where computing resources are provided on an on-demand basis over the Internet [4]. Virtualization technology can be utilized in modern cloud computing environments in order to minimize energy costs and optimize resource utilization [5]. Multiple operating systems can be run on one physical machine using virtualization technology. Operating systems are run on separate Virtual Machines (VMs) controlled by hypervisors [6]. The rapid advancement of emerging technologies such as the Internet of Things (IoT) [7, 8], big data [9, 10], artificial intelligence [11, 12], Blockchain [13], machine learning [14-17], and 5G communication technology [18] has resulted in significant challenges associated with traditional cloud computing data centers, including complex operation and maintenance, inefficient configuration, high construction costs, and an absence of effective unified business monitoring tools.

Live VM migration is one of the key advantages of virtualization, as it facilitates resource management in cloud environments [19]. A VM can seamlessly migrate between physical machines (source hosts) and destination hosts, even as it is running [20]. Load balancing is the primary goal of live VM migration, which migrates VMs from heavy hosts to under-loaded ones [21]. It also provides power management since it consolidates light-load VMs onto fewer servers, resulting in reduced IT operations costs and power consumption [22]. Live VM migration also provides hotspot and cold spot mitigation. In order to meet SLA requirements for VMs, active resource consumption needs to be monitored [23]. Cold spots are physical machines with low threshold values, while hot spots are physical machines with high threshold values. By combining proactive and reactive techniques, hot and cold spots can be detected before they occur and mitigated, thereby maintaining system performance [24]. Server consolidation is another advantage of live VM migration. In this technique, VMs are packed onto a small number of physical machines, and the spare or ideal machine can be powered off or placed in sleep mode, thus reducing both server usage and power consumption [25].

VM migration is the process of moving a VM from one physical host (source node) to another host (target node). VMs are transferred from their source hosts to their destination hosts without interrupting network connections. The original VM continues to run during live migration. Live VM migration is relatively quick, making it ideal for fast migrations. The proposed system aims to migrate VMs from overloaded to underloaded physical machines within a short period of time. The term "live migration of VMs" refers to the process of migrating the VM while it is running in its original state. Live VM migration is beneficial for fast migration since it provides a short amount of migration time [26].

This paper introduces a two-tiered VM placement algorithm. A queuing model is proposed at the first level to handle many VM requests. Models such as cloud simulation are easily implemented and validated using this model. It also provides an alternate method for allocating tasks to servers. Next, a multi-objective VM placement algorithm based on the KH algorithm is proposed. Basically, it maintains a balance between energy consumption and resource utilization. The proposed research aims to achieve the following objectives:

- Generating resource utilization profiles for PMs.
- Using the KH algorithm to identify overloaded and underloaded physical servers.

- Migrating VMs with minimal downtime.
- Reducing the amount of energy required for physical resources.

II. Related Work

Farahnakian, et al. [27] presented an architecture based on the Ant Colony Optimization (ACO) algorithm for dynamic VM consolidation that reduces the energy consumption of cloud data centers while improving quality of service. Kansal and Chana [28] proposed an energy-aware VM migration approach for cloud computing based on the Firefly algorithm. By migrating over-loaded VMs to under-loaded nodes, the energy efficiency of data centers is improved. By comparing the proposed technique with other techniques, the efficacy of this technique is demonstrated. By cutting an average of 73 % of migrations and reducing 35 % of hosts, the data center has achieved an average reduction in energy consumption of 45 %.

Fu, et al. [29] presented a layered VM migration algorithm. Cloud data centers are initially divided into several regions based on bandwidth utilization rates. VM migrations balance network load between regions, resulting in load balancing of cloud resources. Experiments indicate that the proposed algorithm is shown to be able to effectively balance network resource load in cloud computing. Chien, et al. [30] have proposed a novel VM migration algorithm based on the reduction of migrations in cloud computing that can enhance efficiency, meet user requirements, and prevent service level agreements (SLA) violations. The proposed algorithm was found to be more effective than existing algorithms based on experimental results. A threshold algorithm was proposed by Kaur and Sachdeva [31] to allocate tasks to the most capable machine and host and to maintain checkpoints on VMs. Overloaded VMs need to migrate tasks to another VM. This study proposes a weight-based approach for migrating cloudlets between VMs.

Xu and Abnoosian [32] presented a hybrid optimization algorithm based on genetic and particle swarm optimization algorithms for improving VM energy consumption and execution time during VM migration. In the hybrid algorithm, GA is utilized to overcome the limitations of the PSO algorithm, which suffers from slow convergence and limited global optimization. According to the results, the proposed method has improved energy consumption by an average of 23.19% compared to the other three methods. Results also revealed a 29.01% improvement in execution time over the other three methods. Zhou, et al. [33] introduce an energy-efficient algorithm for VM migrations. In this algorithm, host location, VM selection, and trigger time are optimized when memory and CPU factors are taken into account. It migrates some VMs from lightly loaded hosts to heavily loaded hosts using virtualization technology. Energy is conserved by switching idle hosts to the low-power mode or shutting them down. This algorithm reduces SLA violations by 13% and saves 7% of energy over the Double Threshold (DT) algorithm.

VM migration provides an effective and efficient approach to managing cloud resources by providing flexibility in terms of security guaranteeing [34-36], network traffic optimization [37, 38], reduction of SLA violations [39-41], migration cost minimization [42, 43], and energy minimization [44-46]. However, these approaches do not take into account the performance reduction that occurs when VMs are migrated. Several performance-aware VM migration methods have been proposed [47-49]. However, their performance optimization focused not on maximizing VM performance but on reducing SLA violations or migration downtime. Specifically, the works [39-41] attempted to reduce SLA violations or migration downtime rather than optimize VM performance, and Zhang and Zhou [50] guaranteed that running tasks would meet the VM processing time constraint. Çağlar and Altılar [51] aimed to minimize power consumption while meeting performance requirements rather than maximizing VM performance for their users. Moreover, none of the above VM migration techniques provided a specific performance model to describe how VM performance declines over time.

III. System Model

A VM request must be arranged for deployment on physical servers. A scheduler determines a server from the available servers to place the specific VM. A queuing model is proposed to schedule the placement of VMs. In Section III A, the queuing structure is explained. Section III B discusses the KH-based VM placement algorithm.

A. Single Queue Single Service Facility

The queuing scheme follows a single queue single service facility - M/M/1 queue. VM requests are processed according to a FIFO discipline before being forwarded to the data center for placement. Assume μ and λ reflect service and arrival metrics in the queue at various intervals. ($N-1$) VM requests are handled in this way. In the stable situation, P_n represents the likelihood of having n VM requests.

$$\lambda_n = \begin{cases} \lambda, & n = 0,1,\ldots,n-1 \\ 0, & n = N, N+1 \end{cases} \quad (1)$$

$$\mu_n = \mu, \quad n = 0,1,\ldots \quad (2)$$

$$\rho = \frac{\lambda}{\mu} \quad (3)$$

$$P_n = \begin{cases} \rho^n P_0, & n \le N \\ 0, & n > N \end{cases} \quad (4)$$

Using P_n, the expected number of VM requests in the system (R_s) is determined as follows.

$$R_s = \sum_{n=1}^{N} nP_n = \frac{\rho[(N+1)\rho^N + N\rho^{N+1}]}{(1-\rho)(1-\rho^{N+1})}; \ \rho \neq 1 \quad (5)$$

$$\mathrm{R_s} = \sum_{\mathrm{n}=1}^{\mathrm{N}} \mathrm{nP_n} = \frac{\mathrm{N}}{2}; \ \rho = 1 \quad (6)$$

The number of VM requests in the queue (Rq) is determined by Rq = λTq, in which Tq indicates the estimated time required to place the VM. In addition, Ts represents the estimated time for placing the VM in the system. These parameters are determined by Eq. (7) and Eq. (8). Hence, Eq. (9) can be used to calculate the total time required to place a VM request.

$$\mathrm{T_s} = \frac{\mathrm{R_s}}{\lambda} \quad (7)$$

$$\mathrm{T_q} = \mathrm{T_s} - 1 \quad (8)$$

$$\mathrm{T} = \mathrm{T_q} + \mathrm{T_s} \quad (9)$$

B. VM Allocation using the KH Algorithm

Appropriate mapping of VMs to hosts is essential for optimizing key performance indicators, including resource wastage and power consumption. The mapping of VMs to proper PMs is known as VM allocation. The VM array {vm_1, vm_2, vm_3, …, vm_n} comprises *n* VMs, each requesting resources in the memory and CPU dimensions. A host array {H_1, H_2, …, H_p}t signifies the total number of PMs. A krill matrix is used to model the VM request set. Power consumption and resource waste are optimized simultaneously in the proposed method. The following subsections provide mathematical definitions of the parameters mentioned above in order to optimize them.

1) Power consumption calculation: Total power consumption is calculated using Eq. (10), where $\mathrm{ut_i}$ stands for host utilization, $\mathrm{power_i^{min}}$ denotes the minimum power consumption at minimum utilization, and $\mathrm{power_i^{max}}$ refers to the maximum average power consumption at maximum utilization. As determined by Eq. (11), efficiency is the percentage of total power consumed to total workload.

$$\mathrm{pow_i} = \left(\mathrm{pow_i^{max}} - \mathrm{pow_i^{min}}\right) \times \mathrm{ut_i} + \mathrm{pow_i^{min}} \quad (10)$$

$$\mathrm{E_{pwr}} = \frac{\text{Total workload}}{\text{Power consumed}} = \frac{\mathrm{ut_{cpu}}}{(\mathrm{pow^{max}} - \mathrm{pow^{min}}) \times \mathrm{ut_{cpu}} + \mathrm{pwr^{min}}} . \left((\mathrm{pow^{max}} - \mathrm{pow^{min}}) + \mathrm{pow^{min}}\right), \ \ 0 \le \mathrm{E_{pow}} \le 1 \quad (11)$$

In this case, efficiency ranges between 0 and 1, with higher efficiency indicating better server utilization. Eq. (12) calculates the aggregate efficiency.

$$\mathrm{E_{tot}} = \mathrm{E_{res}} + \mathrm{E_{pwr}}$$
$$0 \le \mathrm{E_{tot}} \le 2 \quad (12)$$

2) Resource wastage calculation: Due to the VM's unpredictable resource usage pattern, server utilization is stochastic in nature. An important criterion for determining the appropriate utilization of a server is the proper use of its resources across all dimensions. Eq. (13) calculates the total amount of resources wasted by a host, where r^{min} signifies the normalized wastage. Eq. (14) provides a formula for determining the efficiency of a given assignment.

$$waste_i = \sum_{d \neq min} (r^d - r^{min}) \quad (13)$$

$$E_{res} = ut_{cpu} . ut_{mem}; \ \ 0 \le E_{res} \le 1 \quad (14)$$

E_{res} measures the efficiency of resource utilization. The goal is to maximize the utilization of resources in a variety of dimensions. Physical machines are measured in terms of their CPU and memory. A higher efficiency indicates better packing of VMs. This efficiency ranges from 0 to 1. Eq. (15) can be used to calculate the utilization of the i^{th} host along the d^{th} dimension, where cap_i^d represents the physical host's capacity and $aloc\ (H_i)$ represents the set of VMs allocated to the i^{th} host.

$$ut_i^d = \frac{\sum_{\forall vm_i \in aloc(H_i)} vm_i^d}{cap_i^d}; \ \ d \in \{CPU, mem\} \quad (15)$$

3) Formulation of the problem: Numerous engineering problems have been solved using the KH algorithm. VMP can also be viewed as an optimization problem. The inspiration comes from the KH algorithm, where the krills continually move around the environment in search of food sources. Each VM corresponds to a krill, and the optimum food source corresponds to an optimal host for placement. According to Eq. (16), the global solution is a configuration that fulfills the requirements. The KH algorithm is used to obtain a suboptimal solution to the VM placement problem.

$$min \sum_{i=1}^{p} waste_i \ \ and \ \ \sum_{i=1}^{p} pow_i \quad (16)$$

Virtual machines are assigned to physical machines based on the following criteria.

- Placement constraints: The constraint ensures that a VM will be distributed to only a single host if all required resources are available.
- Capacity constraints: This condition ensures that VM resource requirements do not exceed the total resources all the hosts can share in the federation.
- Assignment constraints: VMs will be placed on servers that meet all their requirements under this constraint. These constraints can be expressed in mathematical terms as follows:

4) Krill herd algorithm: The KH algorithm employs swarm intelligence to solve continuous optimization problems. In comparison to existing algorithmic techniques, it appears to perform better or provide comparable results. Compared to other swarm-intelligence algorithms, this algorithm requires few control parameters and is easy to implement. The KH algorithm models the krill population searching for food within a multidimensional search space with the locations of individual krills serving as decision variables, whereas the distance between the krills and the rich food represents an objective cost. Based on Fig. 1, the KH optimization process comprises three stages, including the movement of other krill individuals, foraging motion, and physical diffusion. These actions can be expressed as a mathematical expression by Eq. (17), where D_i stands for physical diffusion, F_i denotes foraging motion, and N_i signifies other krill movements [52].

$$\frac{dX_i}{dt} = N_i + F_i + D_i \quad (17)$$

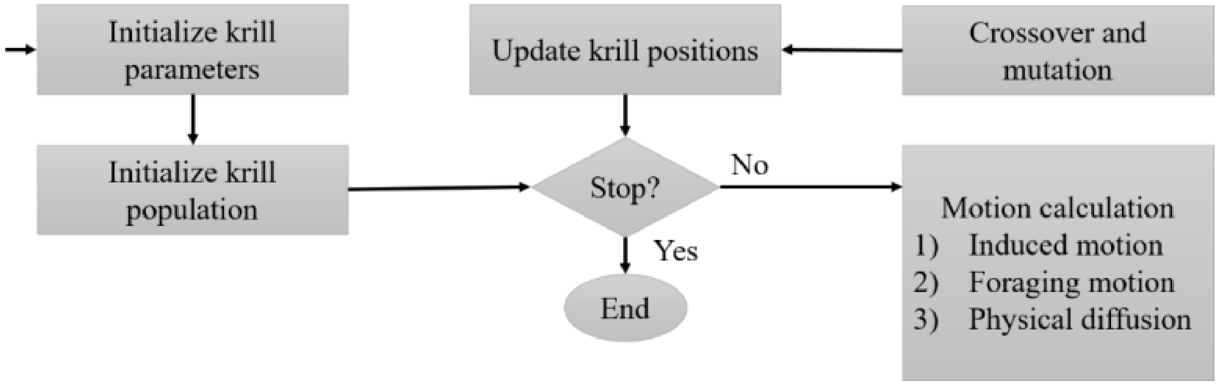

Fig. 1. Flowchart of KH algorithm.

There are three components to the first motion, namely the target effect, the local effect, and the repellent effect. A mathematical representation of the krill i can be derived as follows:

$$N_i^{new} = N^{max} a_i + \omega_n N_i^{old} \tag{18}$$

$$a_i = a_i^{local} + a_i^{target} \tag{19}$$

In the above equations, a_i^{target} indicates the effect of target direction, a_i^{local} represents the local effect, N_i^{old} refers to the last motion induced, ω_n corresponds to the inertia weight ranging from 0 to 1, and N^{max} refers to the maximum speed induced. Food location and previous experience can be described as two aspects of the foraging process. This can be expressed as follows for the i^{th} krill.

$$F_i = V_f \beta_i + \omega_f F_i^{old} \tag{20}$$

$$\beta_i = \beta_i^{food} + \beta_i^{best} \tag{21}$$

In Eq. (20) and Eq. (21), V_f refers to the foraging speed, ω_f represents the inertia weight between 0 and 1, F_i^{old} indicates the last foraging motion, β_i^{food} indicates the food's attractiveness, and β_i^{best} reflects the best fitness of the i^{th} krill. Each krill moves between high- and low-density levels based on Eq. (22), in which d represents a random array of values between 0 and 1 while D^{max} determines the diffusion speed.

$$D_i = D^{max}(1 - \frac{I}{I_{max}})^{\delta} \tag{22}$$

IV. Experimental Results

Cloudsim is the most popular toolkit for simulating cloud environments and conducting simulation-based evaluations. Cloud computing can be continuously exhibited, simulated, and investigated on this completely adaptable platform. In this way, the research community and industry-based designers have the ability to focus on detailed system design. A description of the simulation process is provided in Table I. The simulation was repeated 40 times with up to 200 virtual machines and 200 hosts. This section examines the efficiency of the proposed algorithm in two scenarios and compares it with previous algorithms. In the first scenario, the proposed algorithm's energy consumption is compared to previous algorithms. The results demonstrate that the proposed algorithm is more efficient than previous methods that require fewer hosts and migrations. Fig. 2 illustrates the energy consumption of First Fit Decreasing (FFD), ACO, and Firefly Optimization (FFO) algorithm. Fig. 3 and 4 depict convergence and stability graphs, respectively. As depicted in Fig. 5, the proposed method produces fewer migrations than the FFO, ACO, and FFD. Fig. 6 illustrates the method's stability for the second scenario.

TABLE I. Simulation Parameters

Parameter	Value
Number of physical machines	10-200
Number of virtual machines	10-200
VM size	2 GB
VM RAM	1-4 GB
VM MIPS	1000-2500
PM ram	4 GB
Bandwidth	2 Gbps

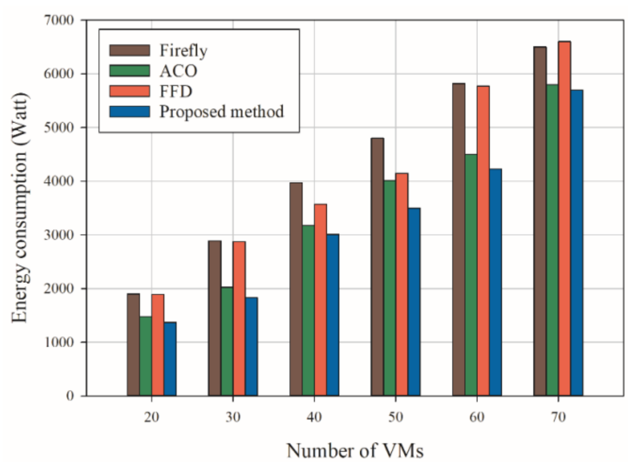

Fig. 2. Energy consumption comparison.

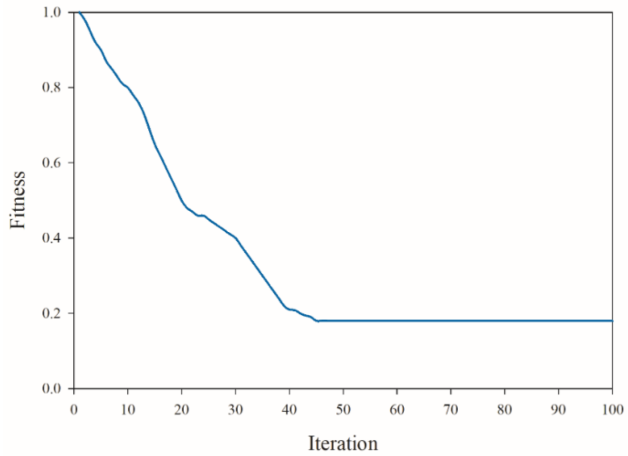

Fig. 3. Coverage graph for the first scenario.

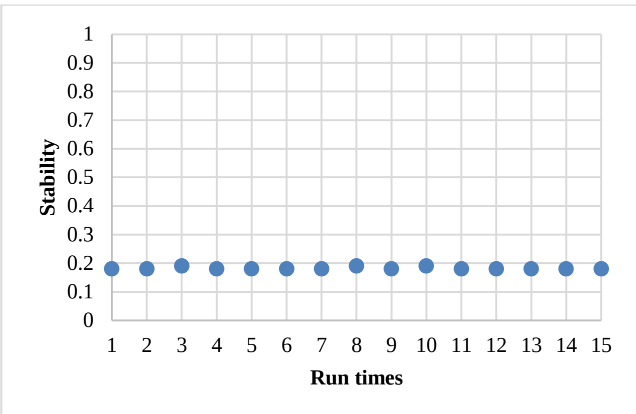

Fig. 4. Stability for the first scenario.

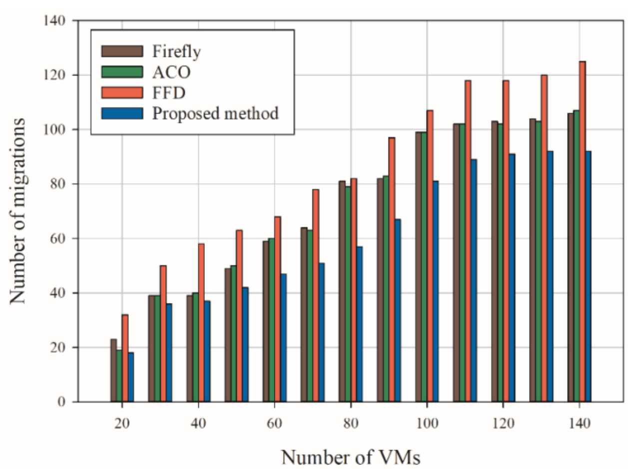

Fig. 5. Number of migrations vs. number of VMs.

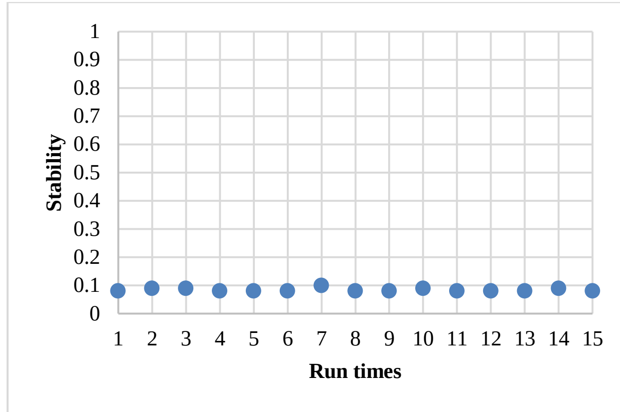

Fig. 6. Stability for the second scenario.

V. Conclusion

Cloud computing provides unlimited computing resources that can be accessed from anywhere, anytime, on demand. Recent research in cloud computing emphasizes the importance of energy efficiency in data centers. Cloud architecture is characterized by high power consumption and inadequate utilization of physical resources. An idle VM tends to consume 50% to 70% of the total server energy, resulting in an imbalance and insufficient power for the active VMs. This paper proposed a new VM migration method based on the KH algorithm that minimized energy consumption and maximized the utilization of computational resources. The algorithm reduces power consumption by putting idle machines into sleep mode. It also minimizes the number of migrations compared to previous works.

References

[1] B. Pourghebleh, A. A. Anvigh, A. R. Ramtin, and B. Mohammadi, "The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments," Cluster Computing, pp. 1-24, 2021.

[2] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," Concurrency and Computation: Practice and Experience, vol. 34, no. 5, p. e6698, 2022.

[3] P. S. Priya, P. Malik, A. Mehbodniya, V. Chaudhary, A. Sharma, and S. Ray, "The Relationship between Cloud Computing and Deep Learning towards Organizational Commitment," in 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), 2022, vol. 2: IEEE, pp. 21-26.

[4] S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation," The Journal of Supercomputing, vol. 78, no. 1, pp. 18-42, 2022.

[5] S. Bharany et al., "Energy efficient fault tolerance techniques in green cloud computing: A systematic survey and taxonomy," Sustainable Energy Technologies and Assessments, vol. 53, p. 102613, 2022.

[6] S. Vinoth, H. L. Vemula, B. Haralayya, P. Mamgain, M. F. Hasan, and M. Naved, "Application of cloud computing in banking and e-commerce and related security threats," Materials Today: Proceedings, vol. 51, pp. 2172-2175, 2022.

[7] A. Mehbodniya et al., "Energy-Aware Routing Protocol with Fuzzy Logic in Industrial Internet of Things with Blockchain Technology," Wireless Communications and Mobile Computing, vol. 2022, 2022.

[8] M. Mohseni, F. Amirghafouri, and B. Pourghebleh, "CEDAR: A cluster-based energy-aware data aggregation routing protocol in the internet of things using capuchin search algorithm and fuzzy logic," Peer-to-Peer Networking and Applications, pp. 1-21, 2022.

[9] H. Kosarirad, M. Ghasempour Nejati, A. Saffari, M. Khishe, and M. Mohammadi, "Feature Selection and Training Multilayer Perceptron Neural Networks Using Grasshopper Optimization Algorithm for Design Optimal Classifier of Big Data Sonar," Journal of Sensors, vol. 2022, 2022.

[10] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017.

[11] M. Sarbaz, M. Manthouri, and I. Zamani, "Rough neural network and adaptive feedback linearization control based on Lyapunov function," in 2021 7th International Conference on Control, Instrumentation and Automation (ICCIA), 2021: IEEE, pp. 1-5.

[12] C. Han and X. Fu, "Challenge and Opportunity: Deep Learning-Based Stock Price Prediction by Using Bi-Directional LSTM Model," Frontiers in Business, Economics and Management, vol. 8, no. 2, pp. 51-54, 2023.

[13] S. Meisami, M. Beheshti-Atashgah, and M. R. Aref, "Using Blockchain to Achieve Decentralized Privacy In IoT Healthcare," arXiv preprint arXiv:2109.14812, 2021.

[14] R. N. Jacob, "Non-performing Asset Analysis Using Machine Learning," in ICT Systems and Sustainability: Proceedings of ICT4SD 2020, Volume 1, 2021: Springer, pp. 11-18.

[15] M. Sadi et al., "Special Session: On the Reliability of Conventional and Quantum Neural Network Hardware," in 2022 IEEE 40th VLSI Test Symposium (VTS), 2022: IEEE, pp. 1-12.

[16] J. Akhavan, J. Lyu, and S. Manoochehri, "A deep learning solution for real-time quality assessment and control in additive manufacturing using point cloud data," Journal of Intelligent Manufacturing, pp. 1-18, 2023.

[17] P. Alipour and S. E. Charandabi, "Analyzing the Interaction between Tweet Sentiments and Price Volatility of Cryptocurrencies," European Journal of Business and Management Research, vol. 8, no. 2, pp. 211-215, 2023.

[18] R. Singh et al., "Analysis of Network Slicing for Management of 5G Networks Using Machine Learning Techniques," Wireless Communications and Mobile Computing, vol. 2022, 2022.

[19] K. Ramana, R. Aluvalu, V. K. Gunjan, N. Singh, and M. N. Prasadhu, "Multipath Transmission Control Protocol for Live Virtual Machine Migration in the Cloud Environment," Wireless Communications and Mobile Computing, vol. 2022, 2022.

[20] Y. Kumar, S. Kaul, and Y.-C. Hu, "Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey," Sustainable Computing: Informatics and Systems, vol. 36, p. 100780, 2022.

[21] S. E. Motaki, A. Yahyaouy, and H. Gualous, "A prediction-based model for virtual machine live migration monitoring in a cloud datacenter," Computing, vol. 103, no. 11, pp. 2711-2735, 2021.

[22] K. J. Naik, "An Adaptive Push-Pull for Disseminating Dynamic Workload and Virtual Machine Live Migration in Cloud Computing," International Journal of Grid and High Performance Computing (IJGHPC), vol. 14, no. 1, pp. 1-25, 2022.

[23] A. Gupta, P. Dimri, and R. Bhatt, "An Optimized Approach for Virtual Machine Live Migration in Cloud Computing Environment," in Evolutionary Computing and Mobile Sustainable Networks: Springer, 2021, pp. 559-568.

[24] M. Noshy, A. Ibrahim, and H. A. Ali, "Optimization of live virtual machine migration in cloud computing: A survey and future directions," Journal of Network and Computer Applications, vol. 110, pp. 1-10, 2018.

[25] M. H. Shirvani, A. M. Rahmani, and A. Sahafi, "A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: taxonomy and challenges," Journal of King Saud University-Computer and Information Sciences, vol. 32, no. 3, pp. 267-286, 2020.

[26] T. He, A. N. Toosi, and R. Buyya, "Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers," Journal of Parallel and Distributed Computing, vol. 131, pp. 55-68, 2019.

[27] F. Farahnakian et al., "Using ant colony system to consolidate VMs for green cloud computing," IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 187-198, 2014.

[28] N. J. Kansal and I. Chana, "Energy-aware virtual machine migration for cloud computing-a firefly optimization approach," Journal of Grid Computing, vol. 14, no. 2, pp. 327-345, 2016.

[29] X. Fu, J. Chen, S. Deng, J. Wang, and L. Zhang, "Layered virtual machine migration algorithm for network resource balancing in cloud computing," Frontiers of Computer Science, vol. 12, no. 1, pp. 75-85, 2018.

[30] N. K. Chien, V. S. G. Dong, N. H. Son, and H. D. Loc, "An efficient virtual machine migration algorithm based on minimization of migration in cloud computing," in International Conference on Nature of Computation and Communication, 2016: Springer, pp. 62-71.

[31] G. Kaur and R. Sachdeva, "Virtual machine migration approach in cloud computing using genetic algorithm," in Advances in Information Communication Technology and Computing: Springer, 2021, pp. 195-204.

[32] Y. Xu and K. Abnoosian, "A new metaheuristic-based method for solving the virtual machines migration problem in the green cloud computing," Concurrency and Computation: Practice and Experience, vol. 34, no. 3, p. e6579, 2022.

[33] Z. Zhou, J. Yu, F. Li, and F. Yang, "Virtual machine migration algorithm for energy efficiency optimization in cloud computing," Concurrency and Computation: Practice and Experience, vol. 30, no. 24, p. e4942, 2018.

[34] D. Sun, J. Zhang, W. Fan, T. Wang, C. Liu, and W. Huang, "SPLM: security protection of live virtual machine migration in cloud computing," in Proceedings of the 4th ACM international workshop on security in cloud computing, 2016, pp. 2-9.

[35] J. Doyle, M. Golec, and S. S. Gill, "Blockchainbus: A lightweight framework for secure virtual machine migration in cloud federations using blockchain," Security and Privacy, vol. 5, no. 2, p. e197, 2022.

[36] A. Celesti, A. Salici, M. Villari, and A. Puliafito, "A remote attestation approach for a secure virtual machine migration in federated cloud environments," in 2011 First International Symposium on Network Cloud Computing and Applications, 2011: IEEE, pp. 99-106.

[37] W. Zhang, S. Han, H. He, and H. Chen, "Network-aware virtual machine migration in an overcommitted cloud," Future Generation Computer Systems, vol. 76, pp. 428-442, 2017.

[38] X. Fu, C. Zhang, J. Chen, L. Zhang, and L. Qiao, "Network traffic based virtual machine migration in cloud computing environment," in 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019: IEEE, pp. 818-821.

[39] M. Jalali Moghaddam, A. Esmaeilzadeh, M. Ghavipour, and A. K. Zadeh, "Minimizing virtual machine migration probability in cloud computing environments," Cluster Computing, vol. 23, pp. 3029-3038, 2020.

[40] L. Liu, S. Zheng, H. Yu, V. Anand, and D. Xu, "Correlation-based virtual machine migration in dynamic cloud environments," Photonic Network Communications, vol. 31, pp. 206-216, 2016.

[41] A. R. Hummaida, N. W. Paton, and R. Sakellariou, "Scalable virtual machine migration using reinforcement learning," Journal of Grid Computing, vol. 20, no. 2, p. 15, 2022.

[42] X. Wang, X. Chen, C. Yuen, W. Wu, M. Zhang, and C. Zhan, "Delay-cost tradeoff for virtual machine migration in cloud data centers," Journal of Network and Computer Applications, vol. 78, pp. 62-72, 2017.

[43] A. Zhou, S. Wang, X. Ma, and S. S. Yau, "Towards service composition aware virtual machine migration approach in the cloud," IEEE Transactions on Services Computing, vol. 13, no. 4, pp. 735-744, 2019.

[44] J. A. Jeba, S. Roy, M. O. Rashid, S. T. Atik, and M. Whaiduzzaman, "Towards green cloud computing an algorithmic approach for energy minimization in cloud data centers," in Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing: IGI Global, 2021, pp. 846-872.

[45] S. K. Kosuru, D. Midhunchakkaravathy, and M. A. Hussain, "An intelligent energy minimization algorithm with virtual machine consolidation for sensor-based decision support system," Measurement: Sensors, p. 100778, 2023.

[46] S. Sohrabi, A. Tang, I. Moser, and A. Aleti, "Adaptive virtual machine migration mechanism for energy efficiency," in Proceedings of the 5th International Workshop on Green and Sustainable Software, 2016, pp. 8-14.

[47] A. A. Khan, M. Zakarya, R. Buyya, R. Khan, M. Khan, and O. Rana, "An energy and performance aware consolidation technique for containerized datacenters," IEEE Transactions on Cloud Computing, vol. 9, no. 4, pp. 1305-1322, 2019.

[48] X. Zhang, Y. Zhao, S. Guo, and Y. Li, "Performance-aware energy-efficient virtual machine placement in cloud data center," in 2017 IEEE International Conference on Communications (ICC), 2017: IEEE, pp. 1-7.

[49] V. Mongia and A. Sharma, "Energy Efficient and Performance Aware Multi-Objective Allocation Strategy in Cloud Environment," in 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), 2020: IEEE, pp. 368-373.

[50] P. Zhang and M. Zhou, "Dynamic cloud task scheduling based on a two-stage strategy," IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772-783, 2017.

[51] İ. Çağlar and D. T. Altılar, "Look-ahead energy efficient VM allocation approach for data centers," Journal of Cloud Computing, vol. 11, no. 1, pp. 1-16, 2022.

[52] A. L. a. Bolaji, M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and L. M. Abualigah, "A comprehensive review: Krill Herd algorithm (KH) and its applications," Applied Soft Computing, vol. 49, pp. 437-446, 2016.