Network load balancing and data categorization in cloud computing

Arunachalam Komathi[1], Somala Rama Kishore[2], Athiyoor Kannan Velmurugan[3], Maddipetlolu Rajendran Pavithra[4], Yoganand Selvaraj[5], Akbar Sumaiya Begum[6], Dhakshnamoorthy Muthukumaran[7]

[1]Department of Computer Science, Nadar Saraswathi College of Arts and Science College, Theni, India
[2]Department of Electronics and Communication Engineering, CMR Engineering College, Hyderabad, India
[3]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vijayawada, India
[4]Department of Management Studies, Rajalakshmi School of Business, Chennai, India
[5]School of Computer Science and Engineering, Vellore Institute of Technology, VIT University, Vellore, India
[6]Department of Electronics and Communication Engineering, R.M.D. Engineering College, Chennai, India
[7]Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

Article Info

Article history:

Received Jan 31, 2024
Revised May 7, 2024
Accepted May 12, 2024

Keywords:

Ant colony optimization
Cloud computing
Data categorization
Load balancing
Support vector machine

ABSTRACT

Cloud computing (CC) is rising quickly as a successful model presenting an on-demand structure. In the CC, the present investigation shows that load-balancing methods established on meta-heuristics offer better solutions for appropriate scheduling and allotment of resources. Conversely, several traditional approaches believe in only some quality of service (QoS) metrics and reject several significant components. Network load balancing and data categorization (NBDC) is proposed. This approach aims to enhance load balancing in the cloud field. This approach consists of two phases: the support vector machine (SVM) algorithm-based data categorization and the ant colony optimization (ACO) algorithm for distributing the network load on the virtual machine (VM). The SVM algorithm performs several data formats, such as text, image, audio, and video, resultant data class that offers high categorization accuracy in the cloud. The ACO algorithm reaches an efficient load balancing based on the time of execution (TE), time of throughput (TT), time of overhead (TO), time of optimization, and migration count (MC). Simulation results related to the baseline approach demonstrate an enhanced system function in terms of service level agreement violation, throughput, execution time, energy utilization, and execution time.

Corresponding Author:

Arunachalam Komathi
Department of Computer Science, Nadar Saraswathi College of Arts and Science College
Theni, Tamil Nadu, India
Email: komathiramakrishnan@gmail.com

1. INTRODUCTION

The internet of things (IoT) involves a novel computing paradigm, which submits to the scalability of the cloud while reducing the network [1]. An efficient load-balancing method contains several running virtual machines (VMs) and virtual machine hosts (VMHs) [2]. It desires to observe, notice, and make a decision how to discharge the inequity condition of VMHs. An efficient load balancing method can take actions that discharge an overloaded VMH and do not outcome in overloads on another. The heuristic-based rule approach is efficient but unsuccessful because it gains overloading in other VMHs [3]. Advanced balancing approaches do mathematical work efficiently, though they require computing several sophisticated functional arguments and significant calculation power to choose motion targets. A fuzzy logic model is used to minimizing packet loss, selecting the relay by pause time, maximum speed, and maximum connection [4]. As a result, it improved efficiency. A hybrid method that evaluates the allocation algorithm to measure the feasible solutions for VM loads next utilizes a genetic algorithm to adjust these solutions [5]. A heuristic task deployment approach using glowworm swarm optimization is utilized for long-term load balancing. A hybrid met heuristic hybridizes an artificial bee colony and an ant colony optimization (ACO) for VM load balancing in the cloud.

The ACO algorithm is a possible technique that searches for an optimal solution [6]. The behavior of ants is to seek the shortest route between their colony and the food origin. Pheromone deposits reveal the route, while, ants move arbitrarily. The route is preferred by the maximum pheromone deposit from the beginning node and is examined for optimality [7]. In the ACO algorithm, the real ant system is updated in three features, such as the election of the route being biased towards the route with a large volume of pheromones [8].

Problem statement: a dynamic load-balanced scheduling (DLBS) mechanism enhances the throughput when equating the workload [9]. An efficient heuristic scheduling method for the two typical open flow models correlates data flow time slots. The DLBS approach uses round robin scheduling algorithms that imbalance the degree of data flows. However, this approach needs to balance the network load more efficiently. It also raises the routing load. A support vector machine (SVM) algorithm for data categorization with equating cloud computing (CC) load is proposed to solve these issues. Structure of this paper arrangement is specified

Task scheduling significantly gives scheduling tasks dramatically adheres to the necessities of the service level agreement (SLA), a document accessible through cloud developers to users [10]. For example, deadline and significant SLA parameters are addressed in the dynamic load balancer (LB) method. It aims to tackle the VM violation problem, optimize resources, and enhance the quality of service (QoS) task parameters and resource allocation. A meta-heuristic algorithm using the dominant firefly algorithm optimizes load balancing of tasks amongst the several VMs [11]. The predictive priority-based modified heterogeneous earliest finish time approach can evaluate the upcoming resource burden [12]. The prediction-based model is used for efficient and dynamic resource to accomplish the end user's requirements. This approach reduced the workflow makespan by enhancing the load balancing across all the VMs [13]. The hybrid artificial bee colony and ACO proved their efficiency. This approach solves the bio-inspired algorithms' issues and balances the load between physical machines. Additionally, it improves the QoS that evaluates SLA violations and performance degradation.

A cooperative fog-CC function reduces the bandwidth cost and balances the network load [14]. This approach reduces the bandwidth cost and equates the load efficiently. This approach balances the load based on the priority of minimizing the delay. This approach determines that link-level load balancing can minimize the link queuing delay. A joint cloud model approach using the heuristic task clustering method and the glowworm swarm optimization (GSO) algorithm reach better search capability and local convergence abilities [15]. Gene expression programming (GEP) is used for forecasting loads [16]. Next, the VMH load is measured by GEP. The task load balancing strategy enhances the average response time and the system's makespan [17]. The balanced state possibly obtains the predictable utilizations for the VM that cooperate an essential job in task allocation. In this approach, the LB is a central server that applies a fair task allocation method to the incoming tasks. In this approach, the particle swarm optimization and the honey bee foraging method reach load balancing. Aware of resource intensity the load balancing approach dynamically allots various weights to different resources and their usage intensity, appreciably minimizes the time and cost to reach the load balance, and evades future load inequity [18]. It reduces the bandwidth cost and transfers VMs to physical machines (PMs) with less VM performance deprivation. Load balancing with server consolidation improves resource utilization and QoS metrics [19]. This method presents taxonomy with categorization for balancing the load, such as hardware threshold, migration overhead, network traffic, and reliability. Resource-aware load balancing algorithm approach is an optimal choice regarding the tradeoff between complexity and the operation in terms of resource consumption and machine-level load balancing [20].

Load-balancing is a cost-effective system management system that builds decentralized load-balancing issues to reduce the cost of violation [21]. This approach locally creates probable assignments of requests to resources and then helpfully chooses an assignment such that their combination enhances edge utilization and reduces the service execution cost [22]. Power quality monitoring offers forecasted insights into power quality fluctuations via examining past data and patterns that enhances grid stability as well as allocation of resources [23]. The energy-efficient light-path establishment is achieved by introducing an ACO-split bypass heuristic to enhance energy efficiency [24]. A heuristic method is established on the ACO method that minimizes the network energy path via swarm intelligence to discover the most energy-efficient routes from sender to receiver to reduce the computational complexity. The ACO algorithm is used to discover cost-optimal related covers of a grid-based region of interest. The ACO method makes efforts to improve the energy-hole issue [25]. Swarm intelligence, such as the ACO method, is an intelligence technique that involves cooperative behavior that interacts with one other locally in a distributed environment. The ACO algorithm discovers optimal path routing and fast route discovery. ACO method for reducing node’s energy utilization and enhancing the lifetime is described in [26].

2. PROPOSED METHOD

This approach utilizes the SVM and ACO algorithms to construct a hybrid technique known as the network load balancing and data categorization (NBDC) mechanism to enhance load balancing in the cloud. SVM algorithm-based data categorization, and the ACO algorithm module for balancing the network load. Different data are obtained arbitrarily in the data categorization phase, and then execute the categorization on these data. The portioned data category is an output of the SVM algorithm. The ACO algorithm reaches an efficient load balancing by the time of execution (TE), time of throughput (TT), time of overhead (TO), time of optimization, and migration count (MC). The feature extraction, specifically image, text, video and audio features, is extracted by the SVM classifier algorithm. The SVM algorithm, compared to k-nearest neighbor, and Naive Bayes classifier algorithms, provide 82% accuracy than the other algorithms. Hence, this approach selects the SVM algorithm for data classification.

2.1. SVM algorithm-based data classification

Next, the selection of optimal feature subsets is made by applying SVM [27]. This approach gathered the data from various cloud sources, including images, text, audio, and video. The SVM algorithm categorizes the data based on features and allows it into a specific category. In a cloud field, each VM allows the task to be established based on task requirements, and every VM has storage resources and different processing. The VM is divided into four types: text, audio, image, and video, by size, requirements, and task features. The SVM aims to categorize and equivalently match data to the appropriate VM and data types. Four audio features are measured for recognizing types of audio classes such as trendy music, traditional music, speech, crowd noise, and simple noise for audio data categorization. In this approach, two kinds of classic kernel operations are utilized in SVMs: radial origin function kernel and polynomial kernel. Here, ai represent the support vector, indicates the lagrange multiplier and bj denotes the label of membership class (+1, -1); here n=1, 2, 3...N. This expression demonstrates the polynomial operation.

$$poly(a, b) = (a^k b + 1)^c \tag{1}$$

where, *c* represents the polynomial degree. The polynomial kernel operation is utilized with SVMs and other kernel models, indicating the similarity among features over the polynomials of the original variables. The polynomial kernel is explained as:

$$K(x, x_i) = (x, x_i)^d \tag{2}$$

where, *d*=1 verifies the linear kernel. The output of the categorization is categorized tasks; as a result, minimizing the computational cost, for instance, evading preprocessing of learning features, data adaptation, features removal, data revolution, and data categorization at the scheduling phase.

2.2. Balancing the network load by ACO algorithm

The ACO algorithm solves node distribution issues and minimizes the routing overhead. Figure 1 represents the undirected weighted graph G (V, E) formed by VMs. Here, V indicates the VM, and E denotes that the edge has a weighted possibility. This weight possibility determines the overload and under load intensities between two nodes.

The SVM algorithm completes the data categorization, and then balances the network load using the ACO algorithm. Let’s assume the set of VMs for text, image, audio, and video corresponds. Every set of VMs is dependable for one task. The mapping of tasks on VMs is calculated using the SVM; here, every VM is allocated a task by the task's size, requirements, and features.

$$Image\,VM = \{VM_1\ , VM_2\, , \dots VM_n\ \} \tag{3}$$

$$Text\,VM = \{VM_1\ , VM_2\, , \dots VM_n\ \} \tag{4}$$

$$Audio\,VM = \{VM_1\ , VM_2\, , \dots VM_n\ \} \tag{5}$$

$$Video\,VM = \{VM_1\ , VM_2\, , \dots VM_n\ \} \tag{6}$$

Each ant moves from the present VM_i to the next node VM_j by calculating the possibility of the route preference rate (RP (k)). In this approach, the ACO algorithm is an agent-based technique that provides an optimized route. This method resulted in the procedure for active pheromone update, and the detection quality of the agent established a computational system that offers an optimal solution. The journey creation procedure for ants comprises the ants' adjacency and heuristic facts. The cost function is applied to measure the quality of resolutions obtained from ants. The ACO algorithm is precise through three factors; for example, a definition of neighborhood, ant, and pheromone trial value. At every step, an ant applies a possibility transition policy to select the operation point it will see next time. Figure 2 demonstrates the flowchart of the NBDC approach.

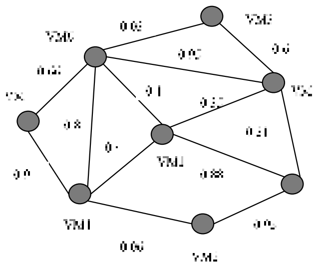

Figure 1. VM's with weight possibility

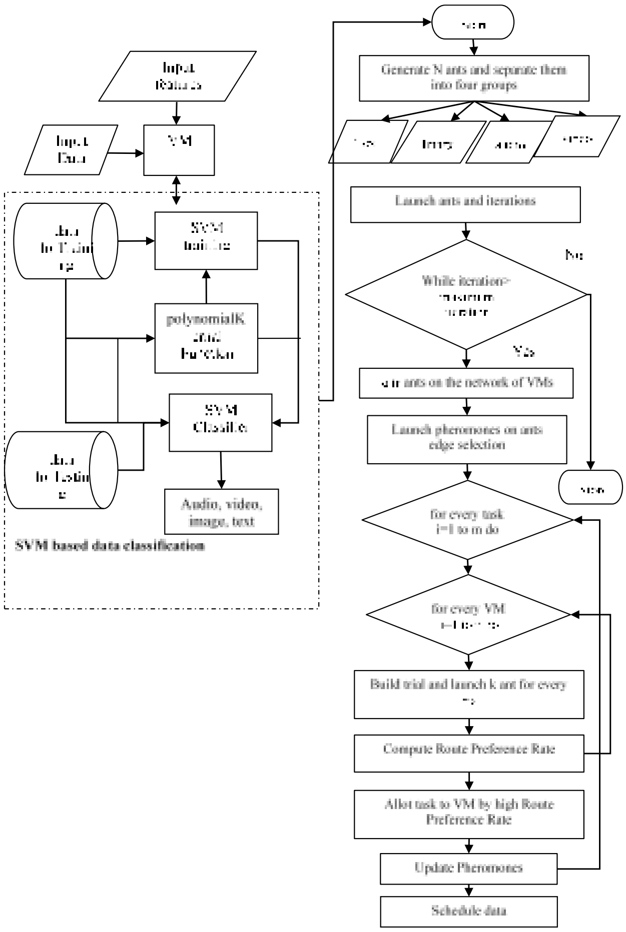

Figure 2. Flowchart of NBDC approach

Throughout the route detection, the forward ants (Fants) determine the route to R by pheromone value (PV), and the backward ants (Bants) return with the subsequent traces. The sender (S) distributes fant to its neighborhood VM to discover which has the greatest good rate necessitated threshold. Then confirm which neighborhood is higher than the threshold, and the S then distributes the Fant message to those nodes. This fant message contains {SID, DID, Hop count, and TTL}. Additionally, this Fant message traces the TE, TT, TO, and MC. While the fant message reaches the D, it calculates RP (k) by hop count, capacity, and delay parameter. The RP (k) calculation among nodes i and j in (7).

$$RP(k) = \frac{(TE_G TT_G TO_G MC_G)}{\sum_{j \in R_m}(TE_G TT_G TO_G MC_G)} \quad (7)$$

Where, TE_G→ time of execution good value, TTG→ time of throughput good value, TO_G→ time of overhead good value, MC_G→ migration count good value, and R_m→ Set of the route from S to D. Next, the Bant message is returned to the S, and such a Bant message is created through RP (k). Each relay node accepts the Bant message and modifies the PV. The PV is given in (8).

$$V_{i,j}\left(1 + PV_{i,j}\right)RP(k) \quad (8)$$

Where, RP (k) indicates the route preference rate of the k^{th} route that satisfied the QoS requirement. Primarily, the PV among the node i and j is PVi,j=0. Then, the $\Delta PV_{i,j}$ denotes the updated PV between the node i and j and obtains the F_{ant} message specified in equation 9. Lastly, the S achieves many B_{ant} messages; next, the S chooses the greatest PV node for scheduling the data to VMs. Hence, the data reaches the VM effectively and accurately.

$$PV_{i,j} = \Delta PV_{i,j} + PV_{i,j} \quad (9)$$

3. SIMULATION ANALYSIS

In this approach, CloudSim 4.0 simulators are executing cloud-related exploration work [28]. Table 1 explains simulation parameters of DLBS and NBDC approaches. NBDC mechanism use 1,000 VMs, correspondingly, and the entirety of 50,000 tasks are executed. Four metrics are compared: SLA violations, energy utilization, execution time and throughput [29]. The SLA violation happens if VM acquires additional time than the allowed CPU time. SLA violations against count of VMs for NBDC and DLBS approaches is shown in Figure 3.

Table 1. Simulation parameters of DLBS and NBDC approaches

Parameter	Value
VM	1,000
Tasks	50,000
Approaches	NBDC and SVMC
Data type	audio, video, image, and text
Algorithms	ACO, SVM
Evaluation tool	CloudSim 4.0 simulator
Iteration	1,000
VM memory	1,024 bytes
Operating system (OS)	Windows-10

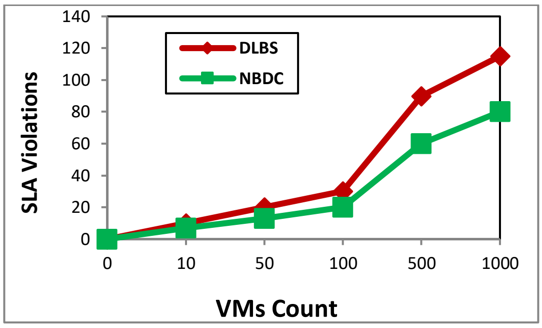

Figure 3. SLA violation of DLBS and NBDC approaches based on VMs count

From Figure 3, till 100 VMs, little alters in SLA violations after 500 and above VMs highly increases the SLA violations. The NBDC approach has the least violations equated to the DLBS approach. The least violations show that the NBDC approach has a better performance and routing efficiency. The optimization time is called convergence time, shown in Figure 4. Here, the DLBS approach is viewing an exponential rise in optimization time; as a result, greater unstable behavior. From this figure, the NBDC approach takes less optimization time since it optimizes relatively rapidly and generates better results than the DLBS approach. In general, DLBS has taken a lot of time to optimize. In the NBDC approach, the PSO algorithm minimizes the error and receives the global best features.

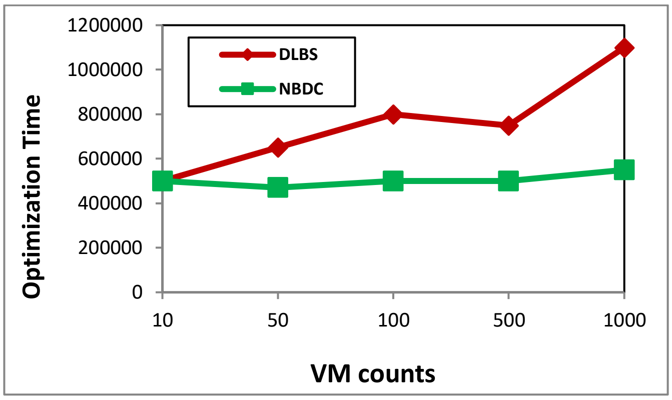

Figure 4. Optimization time of DLBS and NBDC approaches based on VMs count

Figure 5 demonstrates that increased VMs from 10 to 1,000 and raise an energy utilization that keeps on receiving raised with each added VMs. With the regular raise in the tasks count from 100 to 1,000, the DLBS approach utilizes more energy than the NBDC approach. Compared to the DLBS approach, the NBDC approach reduces energy utilization since it balances the network load efficiently. Figure 6 demonstrates the execution time of DLBS and NBDC approaches based on VMs count.

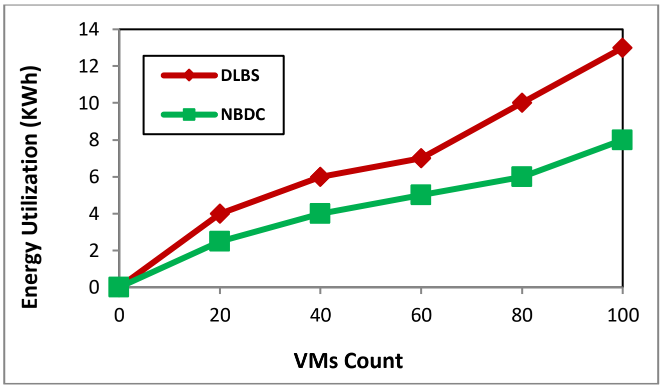

Figure 5. Energy utilization of DLBS and NBDC approaches based on VMs count

From Figure 6, the NBDC firstly executed fine and took the least time and then initiated to increase while VMs obtains 50 in size, NBDC approach execution time gradually then little increase the execution time. Though, overall the method like the DLBS approach increases the highest execution time than the NBDC approach since the DLBS approach causes additional delay. Figure 7 illustrates the overhead time of DLBS and NBDC approaches based on VMs count.

This is mostly due to their computational complexity. The DLBS and NBDC approaches initially obtain the highest overhead time at 50 VMs. However, again unsteadiness is noticed after 500 VMS that obtains raised after each run. Table 2 explains SLA violations, optimization time, energy utilization, and execution time and overhead time of DLBS and NBDC based on VMs count.

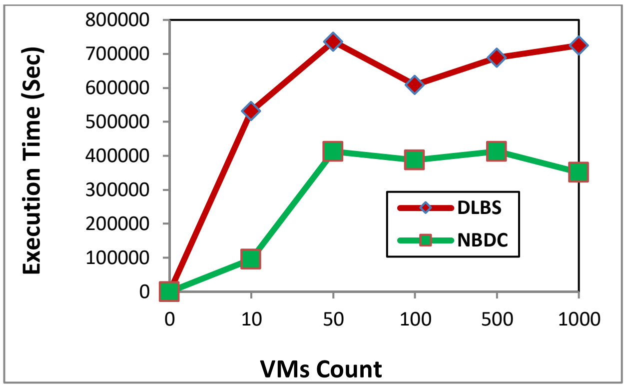

Figure 6. Execution time of DLBS and NBDC approaches based on VMs count

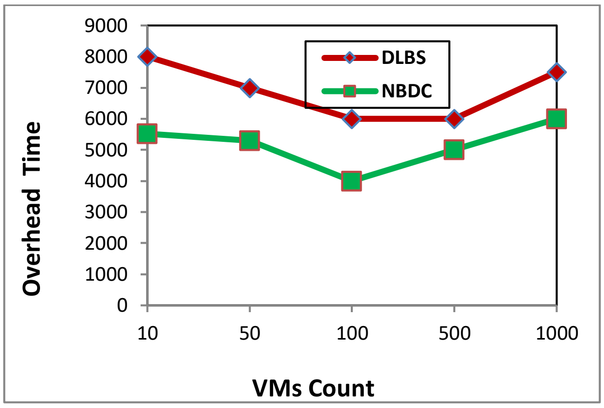

Figure 7. Overhead time of DLBS and NBDC approaches based on VMs count

Table 2. SLA violations, optimization time, energy utilization, and execution time and overhead time of DLBS and NBDC based on VMs count

VM count	SLA violation		Optimization time		Energy utilization		Execution time		Overhead time	
	DLBS	NBDC	DLBS	NBDC	DLBS	NBDC	DLBS	NBDC	DLBS	NBDC
0	0	0	500,100	50,010	0	0	0	0	0	0
10	10	7	650,000	65,000	4	2.5	53,247	95,000	8,000	5,520
50	20	13	800,000	80,000	6	4	73,645	41,200	7,000	5,300
100	30	20	750,000	75,000	7	5	60,893	38,800	6,000	4,000
500	90	60	500,100	11,000	10	6	68,851	41,305	6,000	5,000
1,000	115	80	1,100,000	50,010	13	8	72,461	35,124	7,500	6,000

From Table 2, the NBDC approach compared to DLBS approach, the proposed NBDC mimizes the routing overhead and execution time. Furthermore, the proposed approach NBDC has a lesser and a lesser energy utilization. Moreover, the traditional DLBS mechanism to taken more execution time for processing than a NBDC approach in the network. The lesser computational complexity, smallest iterations in discovering global optima, small transmission cost, lesser overhead have prepared the NBDC approach better choice over the DLBS approach.

4. CONCLUSION

This work aims to reach an efficient task distribution and balance the load. This approach presented a SVM algorithm for data categorization by equating CC load. In the data categorization phase different data obtain the input arbitrarily and then execute the categorization on these data. The SVM algorithm categorizes the data. The portioned data category is an output of the SVM algorithm. The ACO algorithm reach an efficient load balancing through the TE, TT, TO, time of optimization, and MC. Initially categorized as a data class, this approach has illustrated the best categorization and an ACO method that reaches the best outcomes in load balancing. Simulation results indicate that the NBDC approach minimized energy utilization and routing overhead and enhanced the throughput. In the future, we will provide data security for allocating resources by employing cryptography algorithm. Migration count and violations for better performance test these algorithms.

REFERENCES

[1] R. Raman, V. Sujatha, C. B. Thacker, K. Bikram, M. B Sahaai, and S. Murugan, "Intelligent parking management systems using IoT and machine learning techniques for real-time space availability estimation," in *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Nov. 2023, pp. 286–291, doi: 10.1109/ICSCNA58489.2023.10370636.

[2] U. K. Jena, P. K. Das, and M. R. Kabat, "Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2332–2342, Jun. 2022, doi: 10.1016/j.jksuci.2020.01.012.

[3] M. Pesaran H.A, P. D. Huy, and V. K. Ramachandaramurthy, "A review of the optimal allocation of distributed generation: Objectives, constraints, methods, and algorithms," *Renewable and Sustainable Energy Reviews*, vol. 75, pp. 293–312, Aug. 2017, doi: 10.1016/j.rser.2016.10.071.

[4] A. Yahya and A. G. E. Abdalla, "Enhancement of voip perfromance in manet using fuzzy logic," *International Journal of Advances in Signal and Image Sciences*, vol. 6, no. 2, pp. 40–49, 2020, doi: 10.29284/ijasis.6.2.2020.40-49.

[5] Y. Dong, G. Xu, Y. Ding, X. Meng, and J. Zhao, "A 'Joint-Me' task deployment strategy for load balancing in edge computing," *IEEE Access*, vol. 7, pp. 99658–99669, 2019, doi: 10.1109/ACCESS.2019.2928582.

[6] V. K. Arora, V. Sharma, and M. Sachdeva, "ACO optimized self-organized tree-based energy balance algorithm for wireless sensor network," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 12, pp. 4963–4975, Dec. 2019, doi: 10.1007/s12652-019-01186-5.

[7] A. Nayyar and R. Singh, "A comprehensive review of ant colony optimization (ACO) based energy-efficient routing protocols for wireless sensor networks," *International Journal of Wireless Networks and Broadband Technologies*, vol. 3, no. 3, pp. 33–55, Jul. 2014, doi: 10.4018/ijwnbt.2014070103.

[8] C. C. Sekhar, V. V, K. Vijayalakshmi, M. B. Sahaai, A. S. Rao and S. Murugan, "Cloud-based water tank management and control system," *Second International Conference On Smart Technologies For Smart Nation*, 2023, pp. 641-646. 10.1109/SmartTechCon57526.2023.10391730.

[9] F. Tang, L. T. Yang, C. Tang, J. Li, and M. Guo, "A dynamical and load-balanced flow scheduling approach for big data centers in clouds," *IEEE Transactions on Cloud Computing*, vol. 6, no. 4, pp. 915–928, Oct. 2018, doi: 10.1109/TCC.2016.2543722.

[10] M. Gamal, R. Rizk, H. Mahdi, and B. E. Elnaghi, "Osmotic bio-inspired load balancing algorithm in cloud computing," *IEEE Access*, vol. 7, pp. 42735–42744, 2019, doi: 10.1109/ACCESS.2019.2907615.

[11] L.-H. Hung, C.-H. Wu, C.-H. Tsai, and H.-C. Huang, "Migration-based load balance of virtual machine servers in cloud computing by load prediction using genetic-based methods," *IEEE Access*, vol. 9, pp. 49760–49773, 2021, doi: 10.1109/ACCESS.2021.3065170.

[12] M. Sohani and S. C. Jain, "A predictive priority-based dynamic resource provisioning scheme with load balancing in heterogeneous cloud computing," *IEEE Access*, vol. 9, pp. 62653–62664, 2021, doi: 10.1109/ACCESS.2021.3074833.

[13] M. M. S. Maswood, M. R. Rahman, A. G. Alharbi, and D. Medhi, "A novel strategy to achieve bandwidth cost reduction and load balancing in a cooperative three-layer fog-cloud computing environment," *IEEE Access*, vol. 8, pp. 113737–113750, 2020, doi: 10.1109/ACCESS.2020.3003263.

[14] D. A. Shafiq, N. Z. Jhanjhi, A. Abdullah, and M. A. Alzain, "A load balancing algorithm for the data centres to optimize cloud computing applications," *IEEE Access*, vol. 9, pp. 41731–41744, 2021, doi: 10.1109/ACCESS.2021.3065308.

[15] K. Sekaran, M. S. Khan, R. Patan, A. H. Gandomi, P. V. Krishna, and S. Kallam, "Improving the response time of M-learning and cloud computing environments using a dominant firefly approach," *IEEE Access*, vol. 7, pp. 30203–30212, 2019, doi: 10.1109/ACCESS.2019.2896253.

[16] J. Hu, J. Gu, G. Sun, and T. Zhao, "A scheduling strategy on load balancing of virtual machine resources in cloud computing environment," in *2010 3rd International Symposium on Parallel Architectures, Algorithms and Programming*, Dec. 2010, pp. 89–96, doi: 10.1109/PAAP.2010.65.

[17] S. Souravlas, S. D. Anastasiadou, N. Tantalaki, and S. Katsavounis, "A fair, dynamic load balanced task distribution strategy for heterogeneous cloud platforms based on markov process modeling," *IEEE Access*, vol. 10, pp. 26149–26162, 2022, doi: 10.1109/ACCESS.2022.3157435.

[18] H. Shen and L. Chen, "A resource usage intensity aware load balancing method for virtual machine migration in cloud datacenters," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 17–31, Jan. 2020, doi: 10.1109/TCC.2017.2737628.

[19] M. Ala'anzy and M. Othman, "Load balancing and server consolidation in cloud computing environments: a meta-study," *IEEE Access*, vol. 7, pp. 141868–141887, 2019, doi: 10.1109/ACCESS.2019.2944420.

[20] A. Hussain, M. Aleem, M. A. Islam, and M. A. Iqbal, "A rigorous evaluation of state-of-the-art scheduling algorithms for cloud computing," *IEEE Access*, vol. 6, pp. 75033–75047, 2018, doi: 10.1109/ACCESS.2018.2884480.

[21] Z. Nezami, K. Zamanifar, K. Djemame, and E. Pournaras, "Decentralized edge-to-cloud load balancing: service placement for the internet of things," *IEEE Access*, vol. 9, pp. 64983–65000, 2021, doi: 10.1109/ACCESS.2021.3074962.

[22] J. Kumar, A. K. Singh, and A. Mohan, "Resource-efficient load-balancing framework for cloud data center networks," *ETRI Journal*, vol. 43, no. 1, pp. 53–63, Feb. 2021, doi: 10.4218/etrij.2019-0294.

[23] T. Meenakshi, R. Ramani, A. Karthikeyan, N. S. Vanitha, and S. Murugan, "Power quality monitoring of a photovoltaic system through IoT," in *2023 International Conference on Sustainable Communication Networks and Application (ICSCNA)*, Nov. 2023, pp. 413–418, doi: 10.1109/ICSCNA58489.2023.10370494.

[24] C. A. Kyriakopoulos, G. I. Papadimitriou, P. Nicopolitidis, and E. Varvarigos, "Energy-efficient lightpath establishment in backbone optical networks based on ant colony optimization," *Journal of Lightwave Technology*, vol. 34, no. 23, pp. 5534–5541, Dec. 2016, doi: 10.1109/JLT.2016.2623678.

[25] D. S. Deif and Y. Gadallah, "An ant colony optimization approach for the deployment of reliable wireless sensor networks," *IEEE Access*, vol. 5, pp. 10744–10756, 2017, doi: 10.1109/ACCESS.2017.2711484.

[26] D. Karthikeyan and M. Dharmalingam, "Ant based intelligent routing protocol for MANET," in *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering*, Feb. 2013, pp. 11–16, doi: 10.1109/ICPRIME.2013.6496440.

[27] B. Meenakshi, A. Vanathi, B. Gopi, S. Sangeetha, L. Ramalingam, and S. Murugan, "Wireless sensor networks for disaster management and emergency response using SVM classifier," in *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, Aug. 2023, pp. 647–651, doi: 10.1109/SmartTechCon57526.2023.10391435.

[28] M. J. Kumar, S. Mishra, E. G. Reddy, M. Rajmohan, S. Murugan, and N. A. Vignesh, "Bayesian decision model based reliable route formation in internet of things," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 34, no. 3, pp. 1665–1673, Jun. 2024, doi: 10.11591/ijeecs.v34.i3.pp1665-1673.

[29] M. Amru *et al.*, "Network intrusion detection system by applying ensemble model for smart home," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, pp. 3485–3494, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3485-3494.

BIOGRAPHIES OF AUTHORS

Dr. Arunachalam Komathi received her Ph.D. in computer science from Bharathiar University (2021) and Master Degree in MCA (1998) from Madurai Kamaraj University. Currently working as Associate Professor at Nadar Saraswathi College of Arts and Science Theni. She has been in the academic field for more than 22 years. Her research interests are on the wireless sensor network. She has been guiding the student community in developing interest in research and development. She can be contacted at email: komathiramakrishnan@gmail.com.

Dr. Somala Rama Kishore is working as Associate Professor in the Department of Electronics and Communication Engineering at CMR Engineering College, Hyderabad, Telangana, India. He has completed his Ph.D. in the area of image processing at Sri Satya Sai University of Technology and Medical Sciences, Sehore, Bhopal, Madhya Pradesh. He has published around 20 research articles in international, national journals. His main area of interest includes image processing, embedded systems, and communications. He is a member in ISTE. He has published 2 patents and received best teacher award. He can be contacted at email: ramki430@gmail.com.

Dr. Athiyoor Kannan Velmurugan is currently as a Professor in the School of Computing at Koneru Lakshmaiah Education Foundation (Deemed to be University), Vijayawada, Andhra Pradesh, India. He graduated in Computer Science and Engineering at Arunai Engineering College, Affiliated to University of Madras Chennai, Tamil Nadu, India. He did his Master of Technology in Computer Science and Engineering at Dr MGR Educational and Research Institute (Deemed University) Chennai, Tamil Nadu, India. He obtained his Ph.D. in the research area of wireless sensor networks in Computer Science and Engineering at St. Peter's Institute of Higher Education and Research (Deemed University) Chennai, Tamil Nadu, India. He published number of research articles in preferred journals, chapters in books and conferences. His area of interests includes wireless sensor networks, internet of things, artificial intelligence, and machine learning. He can be contacted at email: akvel47@gmail.com.

Dr. Maddipetlolu Rajendran Pavithra having 19 years of experience, mesmerized her audience with her profound insights reaped from both academia and industry. Proficient in multiple programming languages, she has meticulously designed and delivered many courses upholding stringent quality benchmarks within the field of statistics to diverse student demographics at both graduate and postgraduate levels. Her research pursuits presently revolve around topics such as quality control, lean six sigma, machine learning, pattern recognition, deep learning, and image processing. She can be contacted at email: pavistats@gmail.com.

Dr. Yoganand Selvaraj is currently working as a senior assistant professor in the Department of Analytics, School of Computer Science and Engineering (SCOPE) at Vellore Institute of Technology, Vellore. His area of interest is wireless sensor networks, IoT, cloud computing, and machine learning. He received his Bachelor's in Computer Science and Engineering from ANNA University in the year 2009. After completing his Master's in Software Engineering from Anna University 2011, He has received his doctorate degree in 2022 from Anna University and has an overall teaching experience of 11+ years in various academic institutions. He has published 22 international journals including the 5 international conferences. He is a lifetime member of ISTE and published a book titled "Distributed workload manager for web services" in the year 2014 under Lambert Publishing, Germany and E-Commerce Fundamentals, and applications by R.K publications in the year 2023. He can be contacted at email: s.yoganand@vit.ac.in.

Dr. Akbar Sumaiya Begum is a Professor in the Department of Electronics and Communication Engineering at R.M.D Engineering College. She has been a faculty member since June 2002. She has completed her B.E. Degree in Electronics and Communication Engineering at R. M. K. Engineering College in the year 2001 under Madras University and M.E Degree in Optical communication at Anna University in the year 2003. She has completed Ph.D. under Anna University in the year 2015. She has 22 years of teaching experience to UG classes and has guided many B.E. projects. She has published several papers in international and national journals and conferences. She has acted as resource person and delivered several guest lectures in various topics like block chain, artificial intelligence, deep neural networks, machine learning, entrepreneurship development skills, and understanding patent rights, at various colleges. She has three patents published to her credit. She has attended several workshops, seminars and faculty development programs. She is a life time member of ACM, ISTE, SDIWC, IAENG and CSTA. Her areas of interest include signal processing, image processing, artificial intelligence, deep learning, block chain and its applications. She can be contacted at email: sumizahoor@gmail.com.

Dr. Dhakshnamoorthy Muthukumaran is currently working as an Assistant Professor, Department of Electronics and Communication Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Avadi, Chennai. He has 13 years of teaching experience and published more than 25 papers in various journals and conferences. His area of research includes cognitive radio networks, wireless networks, and wireless communications. He can be contacted at email: sarvamkumaran@gmail.com.