Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 3, March 2025, pp. 1865~1873
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i3.pp1865-1873

A hybrid approach for hotspot problem using load balancing and advanced ant colony algorithm

Padmini Mysuru Srikantha[1], Sampath Kuzhalvaimozhi[2], Susheela Shimoga Balasubramanya[1], Sklita Buthello[1], Sadana Hulugundi Jagadish[1], Spandana Mallegowdanakoppalu Thammanna[1]
[1]Department of Computer Science and Engineering, The National Institute of Engineering (affiliated to Visvesvaraya Technological University), Karnataka, India
[2]Department of Information Science and Engineering, The National Institute of Engineering (affiliated to Visvesvaraya Technological University), Karnataka, India

Article Info

Article history:

Received May 20, 2024
Revised Oct 2, 2024
Accepted Oct 7, 2024

Keywords:

Advanced ant colony algorithm
Clustering in WSNs
Hotspot problem
Load balancing
Sensor node

ABSTRACT

Wireless sensor networks (WSNs) are crucial in various applications such as environmental surveillance, military operations, transportation monitoring, and healthcare. However, due to a finite set of sensor nodes' resources concerning energy, memory, disk, and CPU processing, nodes in WSNs often face hotspot issues. The sensor nodes that are located near the base station, are responsible for relaying data not only from themselves but also from neighboring nodes. This leads to hotspot issues, where nodes near the base station experience higher traffic loads and faster energy depletion. This paper mainly focuses on mitigating hotspot issues in heterogeneous WSNs using unequal clustering, load balancing, and an advanced ant colony algorithm. This approach involves devising strategies for selecting cluster heads, determining clusters optimal number and formation, and optimizing data transmission processes. Central to the methodology is utilizing load balancing mechanisms and an advanced ant colony algorithm to distribute the workload among sensor nodes more evenly and find the optimum routing path. The proposed algorithm shows promise in alleviating traffic congestion and energy depletion and provides an innovative approach to enhance network performance and prolong the lifespan of sensor nodes.

Corresponding Author:

Padmini Mysuru Srikantha
Department of Computer Science and Engineering, The National Institute of Engineering
(Affiliated to Visvesvaraya Technological University)
Belagavi-590018, Mysuru, Karnataka, India
Email: mspadmini19@nie.ac.in

1. INTRODUCTION

Wireless sensor networks (WSNs) are a major component of the telecommunications sector [1]. WSNs consist of at least one base station (BS) and numerous sensor nodes, which are typically constrained in terms of energy, processing capabilities, and storage capacity [2], [3]. To ensure efficient data transmission, mechanisms that manage the energy consumption of these nodes and enhance the overall network lifetime are essential [4]. Sensor nodes collect and process environmental data, transmitting it to the BS through neighboring nodes. Given their limited resources, implementing efficient data aggregation and energy-saving routing protocols is crucial [5]. To address these challenges, various routing protocols have been developed to optimize energy consumption among sensor nodes [6], [7].

Clustering within WSNs is an effective strategy for enhancing scalability and extending network lifespan. This method involves grouping sensor nodes and selecting a cluster head (CH). The CH aggregates data from its cluster nodes and relays it to a node closer to the BS. However, nodes nearest to the BS often face excessive energy consumption due to uneven traffic loads, leading to premature failure and network disruption. This creates a hole in the network leading to the hotspot problem.

The applications of WSNs have become increasingly widespread, including areas such as target tracking [8], [9], environmental monitoring, security, disaster response, and health monitoring [10], [11]. As such, there is a significant demand to resolve the hotspot issue caused by clustering in WSNs. One approach to mitigate this issue is a hybrid method that incorporates a load-balancing mechanism with an advanced ant colony optimization algorithm. This advanced ant colony optimization is a nature-inspired paradigm where most stochastic algorithms yield different possible solutions at each iteration, thereby increasing the chances of exploring the entire search space [12]. To enhance the initial feasible solution, various mechanisms such as movement, mutation, exchange, and cooperative perception are employed. This iterative improvement process continues until the best possible solution is identified [13]. The approach adopts a metaheuristic method utilizing a fitness function [14], with the stochastic algorithm aiming to find either the global minimum or maximum of this function, delivering high-quality solutions within a reasonable timeframe [15].

This paper is organized as follows. Section 2, outlines the methodology for clustering in WSNs and the algorithm is proposed for load balancing with advanced ant colony optimization. Section 3 discusses the results. Section 4 draws the concluding remarks and future work.

2. METHOD

This section explains the clustering in WSNs. Then provides detailed information about the proposed algorithm.

2.1. Clustering in WSNs

Clustering is a widely used technique in WSN topology management, where nodes are organized into groups called clusters based on factors like energy levels and location [16]. Each cluster has a CH, selected either through distributed methods, where nodes share status and the one with the highest energy is chosen, or centralized methods, where a BS selects CHs based on a global view, though this requires more communication. WSNs may form nested clusters within a supercluster, enabling multi-hop communication where data is relayed through smaller CHs to a super CH and then to the BS. Clustering helps organize sensor nodes hierarchically, optimizing energy use and reducing the need for frequent reconfiguration, which occurs mainly at the CH level. This technique enhances resource utilization, minimizes redundant transmissions, improves scalability, and extends network lifespan by uniformly distributing energy. The figure of clustering in WSN is represented in Figure 1.

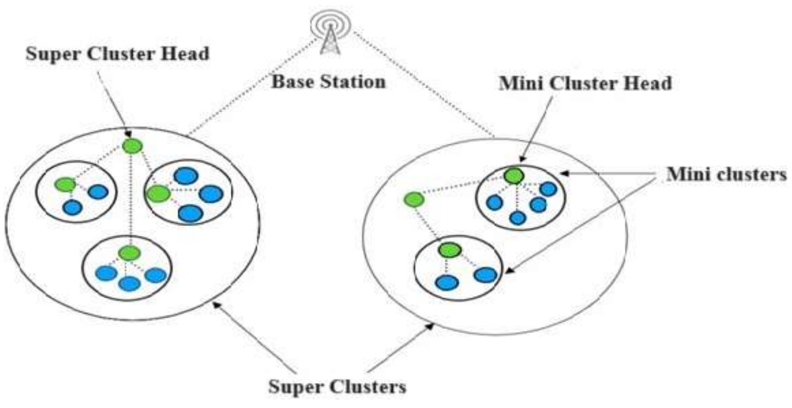

Figure 1. Clustering in wireless sensor networks (WSNs)

2.2. Proposed algorithm

This research introduces a novel load-balancing algorithm with advanced ant colony optimization (LACO). The algorithm is thoroughly detailed in Algorithm 1. LACO consists of two key strategies: the load-balancing mechanism and the advanced ACO operator. In the dynamic environment of WSNs load balancing is the general problem that affects network performance [17]. The LACO mechanism manages workload by using heuristic and fitness functions to ensure efficient scheduling and minimize processing time. LACO mimics ant behavior to effectively balance the load and uses advanced ant colony optimization strategies to reduce both processing time and energy during scheduling.

2.2.1. Load balancing with advanced ant colony optimization

Effective load balancing and advanced ant colony algorithm relies on a solid initialization process, which sets the stage for optimal resource allocation. This involves configuring key parameters such as pheromone levels, evaporation rate and maximum number of iteration, defining system resources which includes CPU, disk and memory resources, and identifying tasks to be scheduled. These parameters directly influence the performance and convergence of the optimization process. The steps involved in balancing the load are experimentally shown in Algorithm 1.

Algorithm 1. Pseudocode of load balancing with advanced ant colony optimization

```
Step 1. Initialization of the parameters;
Step 2. Initialization of system resource set R and a definite set of tasks J
Step 3. Load balancing tasks;
Step 4. While the task is left to be done do
        Nearest Neighbour Operator: For each task, find the nearest available resource (like
        finding the closest worker to do the job);
        Optimal path storage: keep track of the current best path found for scheduling
tasks;
        Updating routing table: Information about the best paths found is updated in the
        routing table;
        Ant Sorting, Ants (representing potential task schedulers) are sorted based on the
        updated routing table to guide their search;
        Construction of path: Ants then construct their paths based on the sorted
        information;
        Updating pheromone table: As ants move along paths, the pheromone table (which
        represents the desirability of paths) is updated;

        if every ant has finished exploring its path options, then
                Pheromone table update: Evaluating the updated pheromone table to find the
                most promising paths;
                Best path selection: Selecting the best path based on the evaluated
                pheromone table;
If Best_solution_obtained then
                        Best Path Selection: Selecting the best path based on the evaluated
                        pheromone table;
                         Task Assignment to Optimal Node: Assigning the task to this optimal
                        node;
                else
                        Next Optimal Solution: If the best solution is not yet found,
                        continue the search for the next best solution.
                end if
        end if
        end while
```

Given a finite set of jobs and a set of resources, the task is to minimize the time required to compute the jobs. Effective job scheduling is achieved through the LACO mechanism. The proposed algorithm schedules tasks efficiently by first selecting the task order and then using a local search phase to create a problem graph, similar to ant foraging. Pheromone levels are updated based on hardware performance and load, to minimize execution time. Tasks are assigned to nodes with high pheromone concentrations and ample memory. LACO integrates advanced ant colony optimization with load balancing, offering a hybrid approach that enhances task scheduling and system efficiency through ant-inspired behavior and optimization strategies. The formula to calculate the execution time for a job i is defined in (1):

$$E_i = T_{out} - T_{in} \tag{1}$$

where T_{in} and T_{out} is the time of arrival of the task i and time of completion of processing time of task i. The total execution time required for the computation of n tasks [18] is depicted in (2):

$$E_n = \sum_{i=1}^{n} E_i \tag{2}$$

the initial energy E_o is set to $0.5J$ .The formulas used for energy are as follows:

$E_{TX} = 50 \times 10^{-9}$
$E_{RX} = 50 \times 10^{-9}$
$E_{fs} = 10 \times 10^{-12}$

E_{mp}= 0.0013×10^{-12}
$E_{DA} = 5 \times 10^{-9}$
$E_{TX}(k,d) = E_{elec} \times k + Emp \times k \times d^2$
$E_{RX}(k) = Eelec \times k$

$E_{TX}(E_{elec} * TX)$: represents the energy consumed per bit for transmitting one bit, E_{RX} $(E_{elec} * RX)$: represents the energy consumed per bit for receiving one bit, E_{mp}: represents the energy amplifier's energy consumption per bit per square meter, E_{fs}: represents the energy consumption per bit per square meter when the free space model applies (for short distances), d_o: represents the threshold distance beyond which the multipath fading model is considered, E_{DA}: represents the additional energy consumed per packet at the cluster head for data aggregation. This includes operations like combining and compressing data from member nodes, E_o : represents the initial energy of a sensor node [19].

2.2.2. Load balancing mechanism with advanced ant colony optimization colony (LACO)

In highly trafficked networks, the load balancing mechanism is crucial for ensuring the quality of service [20]. The LACO algorithm checks the remaining memory on each node to manage new task requests efficiently. For example, with two nodes, $S1$ and $S2$, having 30 MB and 50 MB of free memory respectively, and two incoming jobs $J1$ and $J2$, requiring 40 MB and 60 MB, the function assesses the maximum available memory (50 MB). $J1$ which needs 40 MB is approved as it fits within the available memory of $S2$. However, $J2$ which requires 60 MB exceeds any single node's capacity and is thus rejected and queued. Tasks are only scheduled if sufficient memory is available on any node, ensuring optimal resource utilization. If a task exceeds the total available memory, it waits in a queue until a node can accommodate it. This process prioritizes queued tasks over new ones, optimizing task allocation and reducing computation time. In summary, the LACO ensures that jobs are only scheduled if there is enough memory available on the computing nodes, optimizing task allocation, reducing computation time, and ensuring efficient task management within the system.

The nearest neighbor strategy is a method used in pattern recognition for distributed systems to create a routing table for all "ants" (representing tasks or resources). It is known for its simplicity and effectiveness in generating short tours using Euclidean distance calculations.

- Euclidean distance calculation: the distance between two nodes, $S1$ and $S2$, in a system with n dimensions [21], is calculated using (3):

$$\mathrm{E}(\mathrm{S1}, \mathrm{S2}) = \sqrt{\sum_{i=1}^{n} \left(S_{1_\mathrm{i}} - \mathrm{S}_{2_\mathrm{i}}\right)^2} \quad (3)$$

- Nearest neighbour steps: the process starts by randomly selecting a node as the starting point. It then finds the nearest unvisited node by calculating distances from the current node. This step is repeated, selecting the closest unvisited node each time, until all nodes have been visited. This approach ensures that the path taken is relatively short, making it an efficient method for routing in distributed systems. In the LACO algorithm, after constructing the routing table using the nearest neighbour strategy, the ants (representing tasks or resources) need to opt for the next node to move to. This is where the probability matrix T_{xy} comes into play, as represented in (4). The (probability of transition) fitness function, T_{xy}, determines the likelihood of an ant moving from node x to node y. In the LACO algorithm, each "ant" explores resource nodes to build potential solutions, choosing nodes based on a probability matrix T_{xy} which considers pheromone levels, distance, and heuristics presented in (4). This matrix helps ants decide their next move, balancing exploring new paths with exploiting known efficient ones. By using this probabilistic approach, ants effectively search for high-quality solutions, mimicking natural ant behaviour.
- Optimizing task allocation: ants select nodes likely to lead to efficient task allocation using transition probabilities. Nodes with higher probabilities might indicate lower execution times, more memory, or better pheromone levels. The probability matrix T_{xy} guides ants' decisions in the LACO algorithm, optimizing task allocation and ensuring tasks are assigned to the most suitable nodes. This adaptive approach continuously refines path choices based on changing conditions. Given a finite set of resources R.

$$T_{xy} = \frac{A_{xy}^{a} \times B_{xy}^{\beta}}{\sum\limits_{z \epsilon R} (A_{xz}^{a} \times B_{xz}^{\beta})}, y \epsilon R \quad (4)$$

Where A_{xy} is the value of the pheromone for the resource transition from node x to node y, B_{xy} is a stochastic function that represents the willingness to move from node x to node y. α is a parameter utilized to manage the effect of concentrations of pheromone and stochastic data which is relative to scheduling order value; β is the stochastic component, which represents the relative stochastic data used to select the scheduling sequence for an ant. R represents the unassigned requests set that remain. Among the set of remaining tasks T, one of the tasks will be chosen by the ant in the next move. Resource nodes are chosen based on two important factors as per (4) which are A_{xy} and B_{xy}[22]. B_{xy} the stochastic function is defined by (5).

$$B_{xy} = f_1 \times Ce(x) + f_2 \times Me(x) + f_3 \ x \ De(x) \tag{5}$$

Where f_1 is the effective weight of the utilization of CPU, f_2 denotes the effective weights of the utilization of memory and f_3 represents disk utilization; $Ce(x)$,$Me(x)$, and $De(x)$ denote resource node x's efficiency in (6), (7), and (8):

$$Ce(x) = \frac{Max_{cu} - CPU_X}{Max_{cu}} \tag{6}$$

$$Me(x) = \frac{RemM_x}{Max_m} \tag{7}$$

$$De(x) = \frac{Max_{du} - DU_x}{Max_{du}} \tag{8}$$

Where CPU_x refers to the utilization of the CPU for the sensor node x, DU_x and $RemM_x$ denote the utilization of disk for the node x and the amount of memory remaining, respectively. Max_{cu}, Max_m and Max_{du} are the maximum CPU, memory, and disk capacity respectively. In the LACO algorithm, each ant has its way of making decisions when it comes to selecting a path. This means that they consider the available resources on each server before making their choice. The heuristic function in LACO is based on the local point of each ant. This local point reflects the status of resources, specifically CPU and disk usage, and the amount of remaining memory on a server. In (6) relates lower CPU usage to higher resource availability, while (8) links lower disk usage to the same. In (7) indicates that more remaining memory on a server signifies better efficiency. So, when an ant is deciding which path to take, it considers these factors: CPU usage, disk usage, and memory availability.

The sum of these three values represents the average remaining resource on a server, as represented in (5). In simple terms, each ant in the LACO algorithm looks at the CPU and disk usage, as well as the remaining memory, of each server before choosing its path. Lower CPU and disk usage, along with more remaining memory, indicate better resource availability. This aids the ants make decisions about the next path that has to be taken for efficient task allocation. Moreover, the heuristic function not only helps ants select paths based on immediate resource conditions but also contributes to the collective intelligence of the ant colony. This adaptive decision-making process enhances the algorithm's ability to navigate complex environments and find optimal solutions effectively. In the LACO method, the pheromone update process is crucial for load balancing and takes into account the performance of resource nodes' hardware. By adjusting pheromone levels based on path efficiency and resource availability, LACO optimizes task allocation across servers, ensuring balanced workload distribution and efficient utilization of network resources. The algorithm adjusts pheromone levels according to load balancing and node performance. Ants seek the shortest paths each iteration, with pheromones increasing on the best routes and decreasing on others through evaporation. Pheromone levels are updated after all ants complete their tours. The global pheromone update, a critical step, is calculated using (9). This process optimizes task allocation and path selection. LACO updates pheromone levels based on load balancing and node performance, with ants searching for the shortest paths for each iteration. Pheromones increase on the best paths and decrease on others through evaporation, ensuring efficient load balancing and path optimization. This pheromone update occurs in an ant cycle, where all ants complete their tours before updating the pheromone levels globally using (9). This approach helps ants make decisions about the allocation of resources, ultimately optimizing task scheduling and improving system efficiency.

$$A_{xy} \ (t+1) = \ (1 - \ p) \ A_{xy} \ (\ t) + \sum_{k=1}^{m} \Delta \mathrm{A}_{\mathrm{xy}}^{\mathrm{k}} \tag{9}$$

Where the global residual pheromones coefficient decay rate is denoted by $1 - p, 0 < p < 1$. The evaporation of the Pheromone trail acts as a mechanism to erase or diminish the influence of suboptimal decisions made in the past. ΔA_{xy}^{k} denotes the amount of pheromones drizzled by an ant k on its path from x to y and ΔA_{xy}^{k} is calculated using (10):

$$\Delta A_{xy}^{k} = \begin{cases} (QP)_{k/Len_k} & , if\ k\ travels\ the\ path\ from\ x\ to\ y \\ else\ 0 \end{cases} \tag{10}$$

where Q_{pk} is the amount of pheromone that has to be dropped by an ant k as it travels along the path. Len_k is the length of the path [23].

3. RESULTS AND DISCUSSION

The proposed approach was created and put into action using the MATLAB environment R2023a, known for its numerical computing capabilities. MATLAB is a high-performance language for technical computing [24]. Table 1 shows the initial values of various parameters that were set in MATLAB during the simulation of WSNs.

The comparative analysis of the genetic algorithm (GA), swarm optimization algorithm (SA) with our proposed algorithm, load balancing with advanced ant colony optimization (LACO) is done. This research enhances WSN applications in environmental surveillance, military operations, transportation monitoring, healthcare, smart agriculture, and industrial automation [25].

Table 1. Parameter initialization

Parameters	Values
α	0.8
β	10
P	0.5
Total number of nodes	50
Number of rounds (or iterations)	2000
Number of tasks	100
Initial Energy (E_o)	0.5 J

3.1. Remaining energy analysis

Figure 2 illustrates the relationship between clustering rounds and the remaining energy levels of nodes in the wireless sensor network. The graph tracks energy consumption over time during the clustering process. Higher remaining energy indicates better conservation and efficiency. The analysis reveals that LACO maintains a consistently higher remaining energy curve compared to the other two algorithms.

3.2. Number of access to the fitness function

In Figure 3 the LACO curve always gives better results when repeated. The model of the LACO curve shows that the LACO curve converges faster or reaches a better solution than the GA and the SA which implies that LACO exhibits good performance.

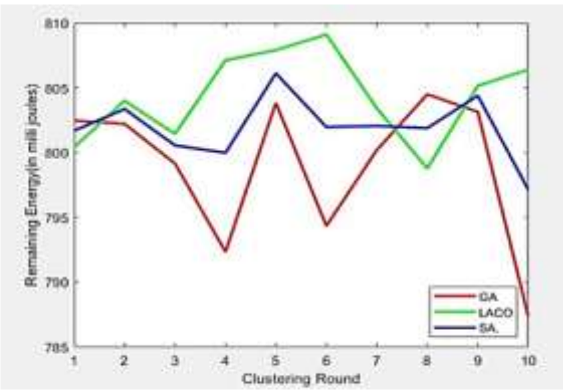

Figure 2. Remaining energy analysis

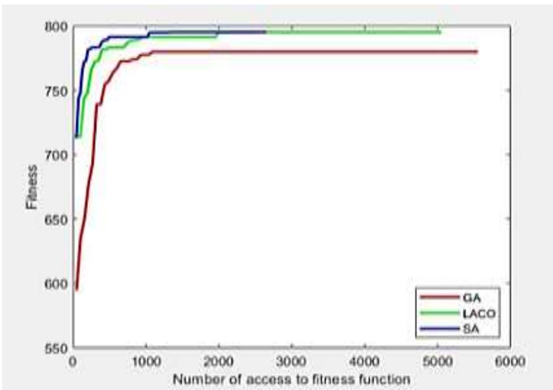

Figure 3. Number of access to fitness function

3.3. Best fit analysis

Figure 4 refers to how well a solution performs relative to some objective function. A higher fitness value indicates a better solution. LACO consistently demonstrates the highest performance compared to genetic algorithms and swarm optimization algorithm in this graph. This indicates that LACO consistently achieves superior results.

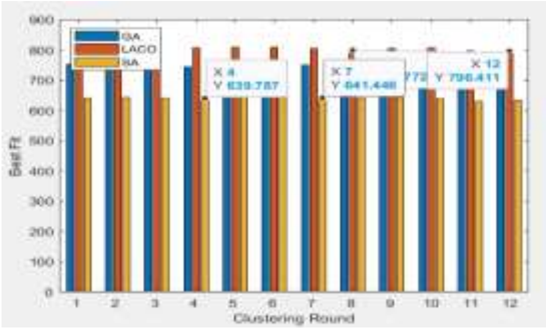

Figure 4. Best fit analysis

4. CONCLUSION

The proposed algorithm significantly improves energy efficiency, reliability, and lifespan in sensor networks by using real-time load distribution based on node energy levels. This dynamic approach surpasses traditional static methods, effectively addressing the hotspot problem near the base station (BS). The study demonstrates that load-balancing and advanced ant colony optimization prevents hotspots, enhancing WSN applications across various fields. It advances clustering techniques, boosts energy efficiency, and lays the groundwork for future research. Future implementations will focus on optimizing performance, ensuring practical deployment, and enhancing adaptability, security, scalability, and fault tolerance.

ACKNOWLEDGMENT

The authors would like to thank The National Institute of Engineering, Mysuru for supporting this project.

REFERENCES

[1] F. M. Mahmud, M. N. Salleh, and M. F. M. Alnoor, “Wireless sensor network security: A recent review based on state-of-the-art works,” *International Journal of Engineering Business Management*, vol. 15, 2023, doi: 10.1177/18479790231157220.

[2] W. Abidi and T. Ezzedine, “Effective clustering protocol based on network division for heterogeneous wireless sensor networks,” *Computing*, vol. 102, no. 2, pp. 413–425, 2020, doi: 10.1007/s00607-019-00757-w.

[3] G. M. Abdulsahib and O. I. Khalaf, “Accurate and effective data collection with minimum energy path selection in wireless sensor networks using mobile sinks,” *Journal of Information Technology Management*, vol. 13, pp. 139–153, 2021, doi: 10.22059/JITM.2021.80359.

[4] I. K. Osamah, M. A. Ghaida, I. K. Osamah, and S. Muayed, “A modified algorithm for improving lifetime WSN,” *Journal of Engineering and Applied Sciences*, vol. 13, pp. 9277–9282, 2022, doi: 10.1155/2022/7909472.

[5] T. M. Behera, S. K. Mohapatra, U. C. Samal, M. S. Khan, M. Daneshmand, and A. H. Gandomi, “I-SEP: an improved routing protocol for heterogeneous WSN for IoT-based environmental monitoring,” *IEEE Internet of Things Journa*l, vol. 7, no. 1, pp. 710–717, 2020, doi: 10.1109/JIOT.2019.2940988.

[6] A. Sarkar and T. M. Senthil, “Cluster head selection for energy efficient and delay-less routing in wireless sensor network,” *Journal of Wireless Networks*, vol. 25, no. 1, pp. 303–320, 2019, doi: 10.26438/ijcse/v7i6.618622.

[7] O. I. Khalaf and B. Sabbar, “An overview on wireless sensor networks and finding optimal location of nodes,” *Periodicals of Engineering and Natural Sciences*, vol. 7, no. 3, pp. 1096–1101, 2019, doi: 10.1166/jctn.2019.8134.

[8] O. I. Khalaf, G. M. Abdulsahib, and B. M. Sabbar, “Optimization of wireless sensor network coverage using the bee algorithm,” *Journal of Information Science and Engineering*, vol. 36, pp. 377–386, 2020, doi: 10.6688/JISE.202003_36(2).0015.

[9] O. I. Khalaf, F. Ajesh, A. A. Hamad, G. N. Nguyen, and D. N. Le, “Efficient dual-cooperative bait detection scheme for collaborative attackers on mobile ad-hoc networks,” *IEEE Access*, vol. 8, pp. 227962–227969, 2020, doi: 10.1109/ACCESS.2020.3045004.

[10] S. Kavianpour, B. Shanmugam, S. Azam, M. Zamani, G. N. Samy, and F. De Boer, “A systematic literature review of authentication in internet of things for heterogeneous devices,” *Journal of Computer Networks and Communications*, vol. 2019, 14 pages, 2019, doi: 10.1155/2019/5747136.

[11] A. D. Salman, O. I. Khalaf, and G. M. Abdulsahib, “An adaptive intelligent alarm system for wireless sensor network,” *Science*, vol. 15, no. 1, pp. 142–147, 2019, doi: 10.1155/2022/6186275.

[12] A. Tzanetos and G. Dounias, “Nature inspired optimization algorithms or simply variations of metaheuristics,” *Artificial Intelligence Review*, vol. 54, pp. 1841–1862, 2021, doi: 10.1007/s10462-020-09893-8.

[13] F. Mbarek and V. Mosorov, "Hybrid nearest-neighbor ant colony optimization algorithm for enhancing load balancing task management," *Applied Sciences*, vol. 11, p. 10807, 2021, doi: 10.3390/app112210807.

[14] N. Nedjah, L. D. M. Mourelle, and R. G. Morais, "Inspiration-wise swarm intelligence meta-heuristics for continuous optimization: A survey-part I," *International Journal of Bio-Inspired Computing*, vol. 15, pp. 207–223, 2020, doi: 10.1504/IJBIC.2020.10030553.

[15] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: a survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 766–26791, 2021, doi: 10.1201/9780429504044-8.

[16] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, "Clustering objectives in wireless sensor networks: A survey and research direction analysis," *Computer Networks*, vol. 180, 2020, p. 107376, doi: 10.1016/j.comnet.2020.107376.

[17] K. Manjula and S. Kalpana, "Load Balancing in cloud computing with enhanced genetic algorithm," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 2S6, 2019, doi: 10.35940/ijrte.B1176.0782S619.

[18] A. M. Chirkin *et al.*, "Execution time estimation for workflow scheduling," *Future Generation Computer Systems*, vol. 75, pp. 142–147, 2017, doi: 10.1016/j.future.2017.01.011.

[19] B. Saoud, I. Shayea, M. H. Azmi, A. A. El-Saleh, "New scheme of WSN routing to ensure data communication between sensor nodes based on energy warning," *Alexandria Engineering Journal*, vol. 80, pp. 397–407, 2023, doi: 10.1016/j.aej.2023.08.058.

[20] V. K. A. and K. N. RamaMohanBabu, "An Intelligent Server load balancing based on Multi-criteria decision-making in SDN," *International Journal of Electrical and Computer Engineering Systems (IJECES)*, vol. 14, no. 4, pp. 433–442, Apr. 2023, doi: 10.32985/ijeces.14.4.7.

[21] S. Dhimal and K. Sharma, "Energy conservation in wireless sensor networks by exploiting inter-node data similarity metrics," *International Journal of Energy, Information and Communications*, vol. 6, no. 2, pp. 57–63, 2015, doi: 10.14257/ijeic.2015.6.2.03.

[22] S. Li, Y. Wei, X. Liu, H. Zhu, and Z. Yu, "A new fast ant colony optimization algorithm: the saltatory evolution ant colony optimization algorithm," *Mathematics*, vol. 10, no. 6, p. 925, 2022, doi: 10.3390/math10060925.

[23] P. Duan and Y. AI, "Research on an improved ant colony optimization algorithm and its application," *International Journal of Hybrid Information Technology*, vol. 9, no. 4, pp. 123–134, 2016, doi: 10.14257/ijhit.2016.9.4.2.

[24] Q. Ali, A. Abdulmaojod, and H. M. Mohammed, "Simulation & performance study of wireless sensor network (WSN) using Matlab," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 7, no. 2, pp. 57–63, 2011, doi: 10.33762/eeej.2011.49662.

[25] S. K. Jaiswal and A. K. Dwivedi, "A security and application of wireless sensor network: a comprehensive study," *International Journal of Communication Systems*, vol. 31, no. 12, p. e3551, 2018, doi: 10.1002/dac.3551.

BIOGRAPHIES OF AUTHORS

Padmini Mysuru Srikantha is currently working as Assistant Professor, in the Department of Computer Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India. She has more than 14 years of teaching experience. Her areas of interest include Wireless sensor network and internet of things. She has presented papers in various Conferences and has also published papers in International Journals. She can be contacted at email: mspadmini19@nie.ac.in.

Dr. Sampath Kuzhalvaimozhi is currently working as Professor in the Department of Information Science and Engineering, The National Institute of Engineering, Mysore, Karnataka, India. She has obtained BE degree from Bharathidasan University, ME from PSG College of Technology and PhD degree in the field of Cryptography from Visvesvaraya Technological University, Belagavi. She has more than 24 years of teaching experience. Her areas of interest include Cryptography and Compilers. She has published and presented more than 20 national and international papers in reputed journals and conferences. She can be contacted at email: kuzhali_mozhi@nie.ac.in.

Susheela Shimoga Balasubramanya is a diligent Computer Science and Engineering student at The National Institute of Engineering, located in Mysore, India. Scheduled to graduate in 2024, she has demonstrated a keen interest in full-stack web development and machine learning. A passion has marked her academic journey for exploring the intricacies of these technologies. She can be contacted at email: 2020cs_susheelasb_b@nie.ac.in.

Sklita Buthello is an industrious Computer Science and Engineering student at The National Institute of Engineering, situated in Mysore, India. With an anticipated graduation in 2024, she showcases a robust enthusiasm for exploring the multifaceted domains of full-stack web development, DevOps practices, data science methodologies, and cloud computing. Throughout her academic journey, Sklita has consistently demonstrated an insatiable curiosity in unraveling the intricacies of these dynamic fields. She can be contacted at email: 2020cs_sklitabuthello_b@nie.ac.in.

Sadana Hulugundi Jagadish is a Computer Science and Engineering student at The National Institute of Engineering, in Mysore, India. Will be graduating in the year 2024, and she has a keen interest in the fields of Web development and data analytics. She can be contacted at email: 2020cs_sadanahj_b@nie.ac.in.

Spandana Mallegowdanakoppalu Thammanna has completed her diploma in Computer Science and Engineering at CPC Polytechnic, Mysore and completed her Engineering in Computer Science and Engineering at The National Institute of Engineering, Mysore in the year 2024. Her interest includes programming, internet of things and web development. She can be contacted at email: 2021lcs_spandanamt_b@nie.ac.in.