future internet

Article

IMBA: IoT-Mist Bat-Inspired Algorithm for Optimising Resource Allocation in IoT Networks

Ziyad Almudayni [1], Ben Soh [1,*] and Alice Li [2]

[1] Department of Computer Science and Information Technology, School of Computing, Engineering and Mathematical Sciences, La Trobe University, Bundoora, VIC 3086, Australia; z.almudayni@latrobe.edu.au
[2] La Trobe Business School, La Trobe University, Bundoora, VIC 3086, Australia; a.li@latrobe.edu.au
* Correspondence: b.soh@latrobe.edu.au

Abstract: The advent of the Internet of Things (IoT) has revolutionised our interaction with the environment, facilitating seamless connections among sensors, actuators, and humans. Efficient task scheduling stands as a cornerstone in maximising resource utilisation and ensuring timely task execution in IoT systems. The implementation of efficient task scheduling methodologies can yield substantial enhancements in productivity and cost-effectiveness for IoT infrastructures. To that end, this paper presents the IoT-mist bat-inspired algorithm (IMBA), designed specifically to optimise resource allocation in IoT environments. IMBA's efficacy lies in its ability to elevate user service quality through enhancements in task completion rates, load distribution, network utilisation, processing time, and power efficiency. Through comparative analysis, IMBA demonstrates superiority over traditional methods, such as fuzzy logic and round-robin algorithms, across all performance metrics.

Keywords: Internet of Things; mist; edge; cloud; load balance; scheduling

1. Introduction

The advent of the Internet of Things (IoT) has reshaped the way we interact with the world around us. By seamlessly connecting sensors, actuators, and humans, IoT networks have transformed many industries, enabling tailored solutions and streamlined processes across various work environments. At the core of IoT networks lies a fundamental objective: to enhance convenience in our daily lives while facilitating real-time data exchange between devices and humans.

As the proliferation of IoT devices continues, the demand for robust and reliable IoT networks is expected to grow exponentially. Projections indicate a staggering 55.7 billion connected devices by the year 2025, underscoring the imperative to ensure the stability, security, and adaptability of IoT technology to meet the evolving needs of modern industries [1].

Efficient task scheduling is essential to strengthening the stability and performance of IoT networks. The allocation of tasks in IoT systems is paramount, as it optimises resource utilisation and ensures the timely execution of tasks. The main method of addressing scheduling challenges lies in aligning with the constraints and objectives of the system, thereby ensuring organisational efficiency and mitigating processing time and power consumption.

Conversely, inefficiencies in task scheduling can impede the efficacy of IoT systems, leading to heightened power consumption and processing delays. By prioritising efficient task scheduling, organisations can have many benefits, including heightened productivity and substantial cost savings [2].

In essence, the optimisation of task scheduling represents a pivotal step towards harnessing the full potential of IoT technology. Accordingly, this paper aims to improve the processing time, energy efficiency, load balancing, and task completion of IoT networks by

check for updates

Citation: Almudayni, Z.; Soh, B.; Li, A. IMBA: IoT-Mist Bat-Inspired Algorithm for Optimising Resource Allocation in IoT Networks. *Future Internet* **2024**, *16*, 93. https://doi.org/10.3390/fi16030093

Academic Editor: Gianluigi Ferrari

Received: 12 February 2024
Revised: 27 February 2024
Accepted: 6 March 2024
Published: 8 March 2024

introducing the IoT-mist bat-inspired algorithm (IMBA), which provides efficient resource optimisation and scheduling for IoT tasks.

The proposed IMBA aims to achieve six objectives. They are to (1) increase the successful rate of task completion; (2) enhance the processing time; (3) reduce the dependency on external computing in cloud and edge, as these require more task waiting time, more offloading steps, a slow response time, and a greater distance for data to travel; (4) allocate more tasks to the mist layer, aiming to provide fast response times and shorter waiting times for performing tasks, as mist notes are the nearest computing resources; (5) reduce the energy consumption of external resources in cloud and edge layers by functioning the internal resources in the mist layer without showing high energy consumption in the mist layer; and (6) reduce the dependency on external networks (WAN networks) to minimise the risk of security vulnerabilities by functioning the tasks in internal networks (LAN network).

The paper consists of five sections. Section 1 introduces the paper. Section 2 analyses recent studies relevant to task scheduling and previous studies on the usage of the bat algorithm to address IoT network challenges. Section 3 gives an overview of our proposed approach. Section 4 presents the results of and validates IMBA. The last section concludes the paper.

2. Related Work

Our research commences with a relatively comprehensive review of the pertinent literature, aimed at pinpointing gaps pertinent to the following challenges:

- How to schedule tasks and resources to optimise IoT network performance;
- How to effectively use the bat algorithm (BA) [3] in IoT environments.

2.1. The Task Scheduling and Resource Optimisation in IoT Networks

Managing task and resource management in IoT networks, especially across cloud, fog, and mist computing environments, poses a significant challenge. However, Refaat [4] proposed a solution by introducing a multi-level IoT task scheduling (MLITS) model. This model adeptly handles resource allocation for IoT tasks, distributing them across relevant environments based on task deadlines and urgency. Notably, the MLITS model has showcased remarkable efficacy, surpassing other algorithms like Min-Min, credit-based scheduling (CBS), and Earliest-Feasible-Deadline-First (EFDF) when evaluated with CloudSim 3.0.2. It boasts accelerated task completion, reduced wait times, and enhanced throughput. Despite its successes, the MLITS model faces hurdles that warrant attention. One such challenge lies in its limited distribution criteria, particularly its oversight of the diverse capabilities inherent in fog, cloud, and mist computing environments. Moreover, the process of offloading tasks from mist to fog and onward to the cloud can incur unnecessary energy expenditures. A more efficient approach would involve tasks autonomously locating suitable resources, with resource capabilities factored into allocation decisions. Nevertheless, the MLITS model represents a pivotal advancement in resource management in IoT networks, offering a promising trajectory for future solutions.

Barik et al. [5] presented the Rao-1 algorithm, devoid of metaphors, aiming to enhance energy efficiency in IoT networks. This algorithm comprises two pivotal phases: firstly, minimising the count of idle microcomputers, and secondly, intelligently assigning tasks to eligible microcomputers. Leveraging Python 3.7, the researchers meticulously implemented the algorithm and conducted a comparative analysis against the MaxMaxUtil algorithm. The findings revealed the superior efficiency of the proposed algorithm over MaxMaxUtil in terms of power consumption and overall performance. However, to further refine their approach, the authors should prioritise workload balance. Ensuring an equitable distribution of tasks across all available microcomputers is essential to alleviate the burden on cloud and fog computing environments while optimising resource utilisation.

Energy conservation is crucial in IoT networks, given the real-time processing demands and energy constraints of IoT devices. In this context, Tian M. et al. [6] introduced a pivotal method known as decisive task scheduling for energy conservation (DTS-EC). This method classifies tasks based on their energy requirements, with the goal of efficiently allocating and queuing data for processing. Validation of the proposed approach was conducted using the Cooja simulator. The findings demonstrated superior performance of DTS-EC compared to existing methods such as scalable energy efficiency scheme (SEES), composition based on concurrent request integration optimisation–particle swarm optimisation (CRIO-PSO), and deep reinforcement learning-based offloading (DRLO) across metrics including data dissemination, energy utilisation, and latency. However, for further refinement, the authors should consider additional factors beyond energy requirements in scheduling, particularly focusing on the accuracy of data arrival and task completion. This broader perspective would enhance the comprehensiveness and effectiveness of the DTS-EC method.

Efficient task scheduling in fog and cloud environments is essential for ensuring the productivity and efficacy of IoT systems. In this vein, Liu Q. et al. [7] devised the hybrid meta-heuristic (MH) algorithm to proficiently allocate and schedule tasks across the fog cloud system, with the primary criterion being the minimisation of the makespan. The evaluation of the optimal virtual machine (VM) for task allocation is based on this metric. Executing simulations using MATLAB R2018b, the authors demonstrated the superior performance of their proposed approach compared to traditional methods such as particle swarm optimisation (PSO), firefly algorithm (FA), and Harris Hawks optimisation (HHO) in terms of makespan and throughput. However, for a more comprehensive evaluation, it is imperative for the authors to broaden their focus beyond makespan and throughput. Targeting additional performance metrics, such as task completion success rates and energy consumption, would provide a more holistic assessment of the MH algorithm's efficacy in enhancing IoT network performance.

Balancing loads and optimising power consumption are critical challenges in IoT environments, given the vast geographical distribution and sheer abundance of IoT sensors and devices. Ensuring minimal energy consumption by these devices and pinpointing optimal computing resources are crucial endeavours. Addressing these concerns, Shuaib M. et al. [8] introduced the dynamic energy-efficient load balancing (DEELB) algorithm. The DEELB algorithm comprises three key phases: identifying nodes' storage capacities, allocating resources, and balancing load. To assess its efficacy, the authors employed CloudSim for validation. The simulation results unequivocally demonstrated the superior performance of DEELB over existing methods such as energy-efficient opportunistic (EEFO), dynamic energy-efficient resource allocation (DEERA), efficient load-balancing security (ELBS), and delay energy balanced task scheduling (DEBTS). Notably, DEELB exhibited advantages in terms of delay, response time, packet loss, reliability, and throughput. However, while having these achievements, it is imperative for the authors to consider energy consumption. Ensuring that the DEELB algorithm operates within sustainable energy usage bounds is paramount, as energy efficiency stands as a cornerstone in IoT network management. By integrating energy considerations into their algorithmic framework, the authors can further fortify the viability and sustainability of DEELB in real-world IoT deployments.

Ensuring the optimal activation of IoT sensor nodes is pivotal for maximising the performance of IoT networks. Recognising this, Liu Y. et al. [9] introduced the spatio-temporal scope information model (SSIM), designed to facilitate informed decisions regarding sensor activation by leveraging the spatio-temporal characteristics of valuable sensor data. Through the analysis of space and time parameters, the SSIM enables the system to schedule sensor activation effectively. The experimental findings underscored the efficacy of the proposed SSIM, showcasing improvements in sensor activation scheduling that translated into enhanced overall system sensing accuracy and energy efficiency. However, to further refine their approach, the authors should integrate considerations of task performance alongside sensor activations. This comprehensive approach would offer a more holistic perspective, ensuring that system optimisation efforts address both sensor activation strategies and task execution performance, thereby advancing the effectiveness and efficiency of IoT networks.

The influx of data received by fog devices in IoT edge networks can significantly impact system performance, leading to delays and increased power consumption. Addressing these challenges, efficient task scheduling plays a pivotal role. Liu W. et al. [10] devised a two-step approach, leveraging advanced algorithms to enhance energy efficiency and reduce time delays in IoT networks. Initially, the authors employed particle swarm optimisation (PSO) to achieve optimal load balancing among fog nodes, considering factors such as energy cost and processing time. Subsequently, they designed a particle swarm genetic joint optimisation artificial bee colony algorithm (PGABC) to schedule tasks among fog clusters, building upon the outputs of the PSO algorithm. The experimental results validated the effectiveness of the proposed approach, demonstrating improvements in energy efficiency and reductions in delay times. Notably, the approach outperformed alternative methods such as moth–flame optimisation (MFO), inter-cluster task scheduling based on random resource allocation (PGABC-R), and shortest job first (SJF) inter-cluster task Scheduling based on PSO for load-balancing resource allocation (SJF–PSO). However, the authors should remain aware of the task completion accuracy. Integrating considerations of task completion rates alongside energy efficiency and delay reduction efforts would offer a more comprehensive evaluation of system performance and further refine the proposed approach.

Saurabh et al. [11] proposed sigmoid neural network clustering (SNNC) and entropy-based scheduling (EBS) techniques to elevate the quality of service (QoS) in IoT networks. The methodology revolves around data collection from IoT sensors and subsequent transmission to the fog layer for in-depth analysis. In the fog computing domain, SNNC and EBS facilitate the calculation of score values for each fog node, aiding in the selection of optimal clustering and scheduling strategies. The experimental findings underlined the superiority of the proposed approach over traditional methods such as first come first serve (FCFS) and deep q-learning mode (DQN) across various performance metrics, including latency, response time, energy consumption, network traffic, bandwidth, transmission rate, and throughput. However, expanding the algorithm's scope to encompass internal resources such as mist nodes would further enhance its applicability and effectiveness in IoT network management. Integrating mist nodes into the algorithmic framework could lead to more comprehensive resource utilisation and optimisation, thereby bolstering the overall performance and scalability of the system.

2.2. The Effective Use of the Bat Algorithm in IoT Environments

Suresh G. et al. [12] proposed a novel, self-adaptive, energy-efficient bat algorithm to fortify software-defined IoT networks (SDN) by bolstering the intrusion detection system. The method unfolds in two stages: initially, real-time network traffic packets from SDN undergo analysis, and feature selection is executed. Subsequently, packets are classified as normal or malicious, triggering appropriate preventive measures upon the detection of an attack. The evaluation results underscored the outperformance of the proposed approach over swarm-based bat algorithms in terms of accuracy, precision, and time overhead, indicating its superior efficacy for the specified problem.

Alharbi A. et al. [13] proposed a robust solution to safeguard commercial IoT devices against two types of botnet attacks. Their local–global best bat algorithm for neural networks (LGBA-NN) exhibited high efficiency in detecting and securing nine different IoT devices. An evaluation using N-BaIoT showcased superior results compared to weight optimisation using particle swarm optimisation (PSO-NN).

In the area of smart homes, Razwan M. et al. [14] tackled the challenge of balancing energy efficiency and smart home services. Their bat-algorithm-based solution aimed to optimise temperature, illumination, and indoor air quality settings to enhance home convenience while minimising energy costs. The experimental findings demonstrated the superiority of the proposed approach over alternative algorithms such as firefly, genetic, ant colony optimisation, and artificial bee colony algorithms.

Addressing the challenges of energy efficiency and rerouting in mobile ad hoc Networks (MANET), Oluesesi A. et al. [15] devised an adaptive information weight bat algorithm (AIW-BA) model. MATLAB simulations revealed that AIW-BA surpassed existing routing protocols, including genetic algorithm ad hoc on-demand multipath distance vector (GA-AOMDV), bat-optimised link state routing (BOLSR), ant colony harmony search algorithm (ACO–HSA), particle swarm optimisation routing protocol (PSO-RP), and enhanced ant colony ad hoc on-demand vector (EA-AODV) in terms of enhancing the network's lifetime and reducing link failures.

In cloud computing, Sundas A. et al. [16] proposed a modified bat algorithm (MBA) to optimise task allocation to cloud resources, ensuring optimal load balancing. The evaluation results demonstrated the superior performance of MBA compared to the original bat algorithm, highlighting its efficacy in achieving efficient resource allocation and load balancing in cloud environments.

3. Proposed IMBA

The bat algorithm (BA) harnesses the ingenuity of nature to tackle intricate problems, drawing inspiration from how bats navigate and capture prey. Central to this algorithm are five pivotal parameters that mirror the sensory and locomotive capabilities of bats:

1. Frequency (fi): Corresponds to the sound waves emitted by the bat during its quest for food, akin to a sonar system. It interacts with the received frequency to guide the bat's navigation.
2. Position (xi): Signifies the spatial coordinates of the bat in its environment, determining its current location.
3. Pulse value (ri): Represents the vibration or wave rate, mirroring the echo feedback received by the bat during its exploration.
4. Loudness (Ai): Reflects the intensity and amplitude of the bat's vocalisations, influencing its ability to detect and communicate with prey.
5. Velocity (Vi): Denotes the speed and direction of the bat's movement, crucial for the efficient pursuit and capture of prey targets.

These interconnected parameters synergise to enable the bat to locate and sense its prey effectively. By leveraging the inherent intelligence of this algorithm, we can tap into the immense problem-solving potential offered by nature [17]. Our approach involves deploying the original bat algorithm while carefully configuring the fitness function and bat parameters to align with the specific objectives outlined earlier. This strategic adaptation empowers us to efficiently address the challenges at hand, leveraging the natural wisdom encapsulated within the bat's evolutionary adaptations.

3.1. IMBA Phases

Our approach revolves around harnessing the innate power of the original bat algorithm while tailoring it to suit our specific objectives. We define the fitness function as per Equations (1) and (2) and tune the parameters outlined in Table 1 to align with our goals:

$$Fitness(i) = \sum_{j=1}^{n} Exec\_time(j, VM) \tag{1}$$

$$Exec\_time(j, VM) = \frac{Task\_lengthj(VM)}{MIPS(VM) \times VM(count)} \tag{2}$$

Table 1. IMBA parameters.

Parameter	Value
Population size	100
Iterations	10
loudness_min: A_min—loudness	0
loudness_max: A_max—loudness	0.8
pr_min: pulse rate/emission rate min	0
pr_max: pulse rate/emission rate max	0.95
pf_min: pulse frequency min	0.0
pf_max: pulse frequency max	10
Lower bound of values in the population	0
Upper bound of values in the population	The count of VMs

This approach has been empirically proven to be highly effective in achieving the objectives outlined in our paper.

3.2. The Proposed IMBA Steps Are Listed Below

1. Initialisation
    a. Initialise the population of bats with random solutions. In this simulation, a random solution is generated in the form of a list of tasks. Each task is then allocated to be executed on either cloud, edge, or mist nodes in a random manner.
    b. Set the pulse rate (A) and loudness (r) for each bat.
    c. Set the iteration counter (t) to 0.
2. Evaluate the fitness function of each bat's solution using the objective function (Equation (1)). This function is based on finding the solution with the least execution time (Equation (2)). The fitness function is the sum of time taken by a virtual machine (mist, edge, or cloud) to complete a task, where n is the number of tasks in a solution/population, j is the task id, and VM is the id of the machine assigned to it, which can be cloud, edge, or mist.
3. While stopping criteria are not met:
    a. Increase the iteration counter (t);
    b. Generate new solutions;
    c. Generate new solutions for each bat by adjusting their positions based on frequency and loudness;
    d. Evaluate fitness;
    e. Evaluate the fitness of the new solutions;
    f. Update the best solution;
    g. Update the loudness and pulse rate;
    h. Update the loudness (r) and pulse rate (A) for each bat;
    i. Check stopping criteria;
    j. Check whether the stopping criteria, such as maximum iterations or convergence, are met.

4. Evaluation and Results

PureEdgeSim simulator 1.0 stands as the pinnacle solution for simulating large-scale Internet of Things (IoT) environments. With its distributed and dynamic infrastructure, researchers can examine the complexities of highly heterogeneous applications in IoT networks. The simulator boasts realistic infrastructure models that facilitate the in-depth exploration of mist-to-cloud interactions. Moreover, PureEdgeSim covers all layers of computing modelling and simulation, as depicted in Figure 1 [18], providing researchers with a holistic platform to investigate various aspects of IoT systems.

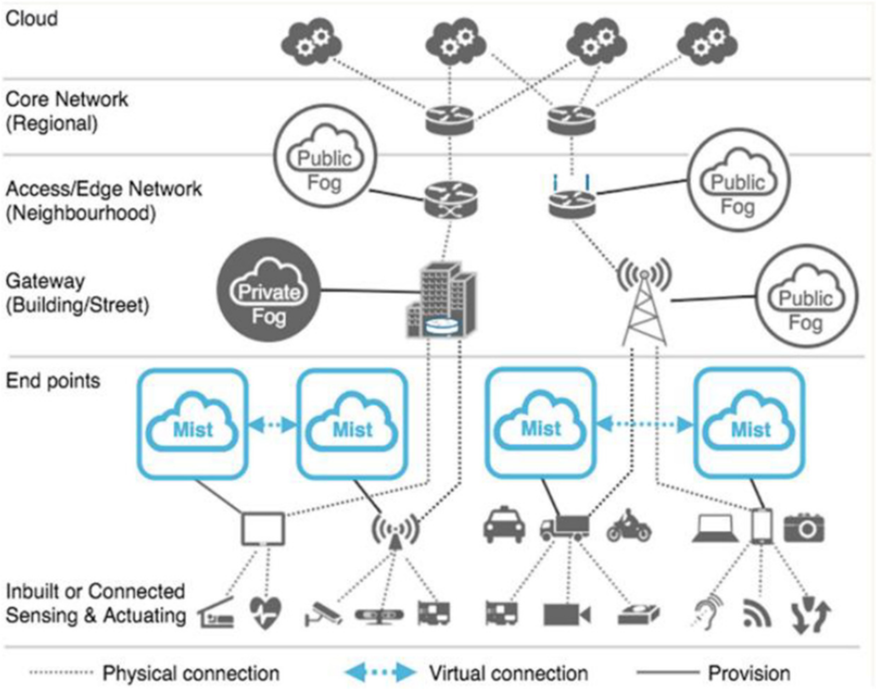

Figure 1. End point to cloud infrastructure.

In this study, we conducted a simulation employing mist nodes ranging from 50 to 300, serving as internal resources, while external resources comprised one edge data centre and one cloud data centre. The performance evaluation of our proposed bat algorithm—IMBA —was carried out under escalating task loads, starting from 5000 tasks up to 30,000 tasks. Several key parameters were scrutinised to assess performance, encompassing the number of completed tasks, CPU utilisation, processing time, LAN and WAN usage, and energy consumption.

The performance study analysed various metrics and compared them against fuzzy logic-based algorithms and round robin scheduling. Initially, the task completion count was scrutinised as the primary metric, revealing that the proposed IMBA achieved the highest count of completed tasks, as illustrated in Figure 2. Furthermore, the IMBA exhibited a notably shorter processing time compared to the fuzzy logic-based algorithm, while its processing time closely aligned with that of the round robin algorithm, as depicted in Figure 3. Notably, Figure 2 shows that the IMBA completed approximately 700 more tasks than the round robin algorithm. This observation unequivocally highlights the superior performance of the IMBA in terms of processing time, thereby outperforming both the fuzzy logic-based and round robin algorithms.

Figures 4 and 5 provide a comprehensive analysis of the load reduction achieved over the cloud and edge, thereby enhancing the system's performance. Notably, the proposed IMBA demonstrated remarkable effectiveness in alleviating the load burden on both the central cloud server and edge when compared to the fuzzy logic and round robin algorithms.

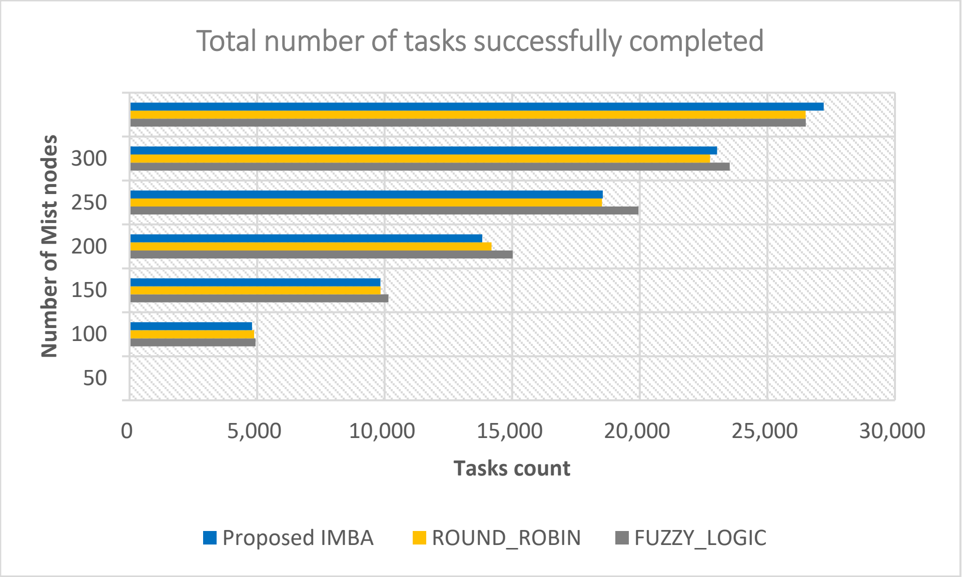

Figure 2. Total number of tasks successfully completed.

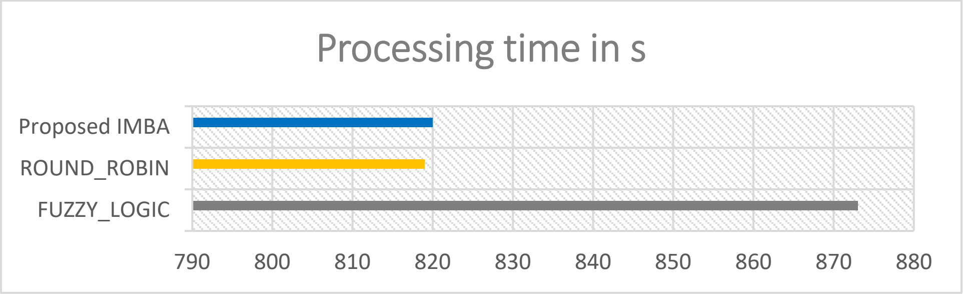

Figure 3. Processing time in s.

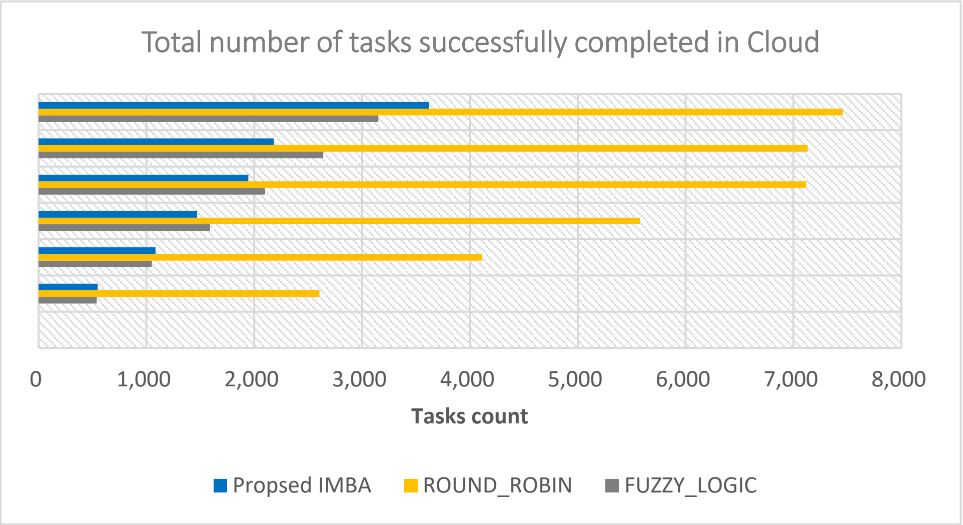

Figure 4. Total number of tasks successfully completed in cloud.

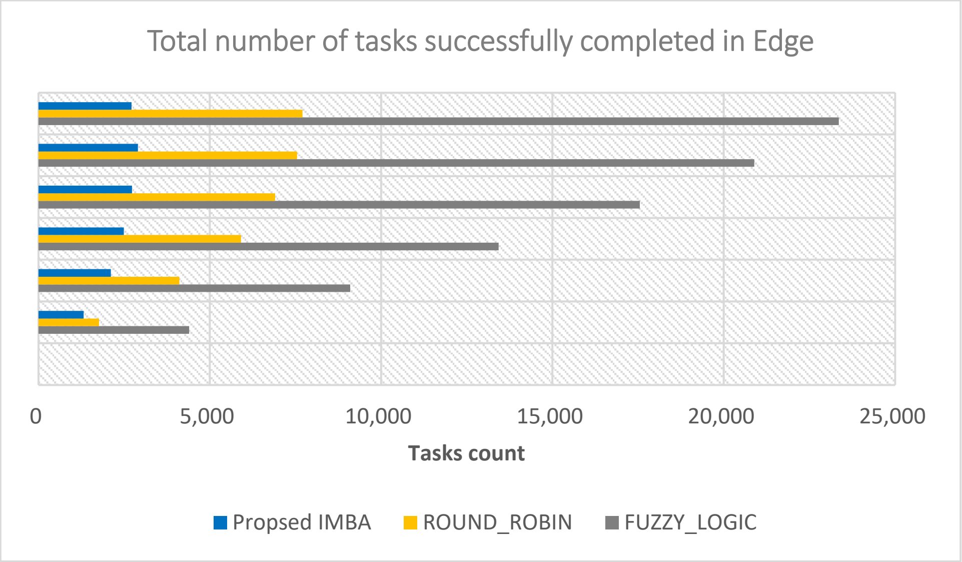

Figure 5. Total number of tasks successfully completed in edge.

However, it is worth noting that the fuzzy logic approach exhibited a lower count of completed cloud tasks, primarily attributable to task failures amounting to approximately 700 tasks. This failure resulted in an elevated CPU usage within the cloud infrastructure for the fuzzy logic approach.

In contrast, the IMBA successfully minimised CPU usage in both the cloud and edge, aligning closely with one of the primary objectives of the IMBA approach, as evidenced by Figures 6 and 7. This outcome underscores the efficacy of the IMBA in optimising resource utilisation across diverse computing environments in IoT networks.

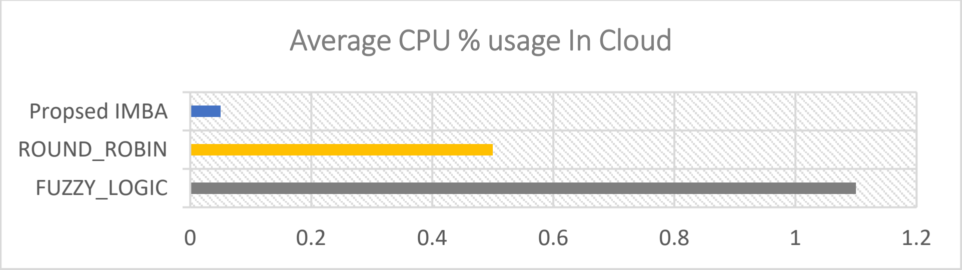

Figure 6. Average CPU % usage in cloud.

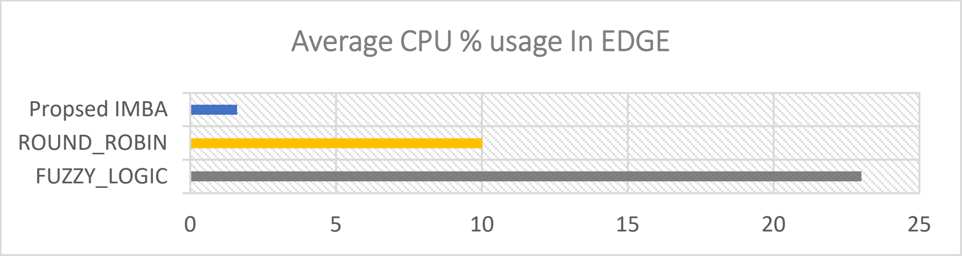

Figure 7. Average CPU % usage in edge.

The research findings underscore the effectiveness of our proposed IMBA in minimising energy consumption across external resources in both the cloud and edge layers. This achievement is illustrated in Figures 8 and 9. The algorithm accomplished this by significantly reducing CPU usage in both the edge and cloud layers, thereby optimising resource utilisation and enhancing energy efficiency in IoT networks.

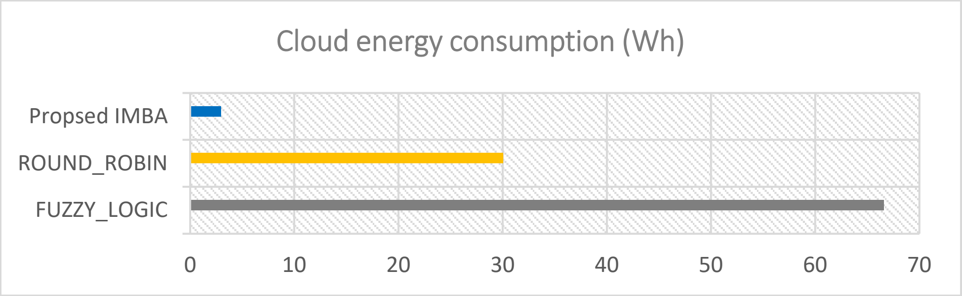

Figure 8. Cloud energy consumption (Wh).

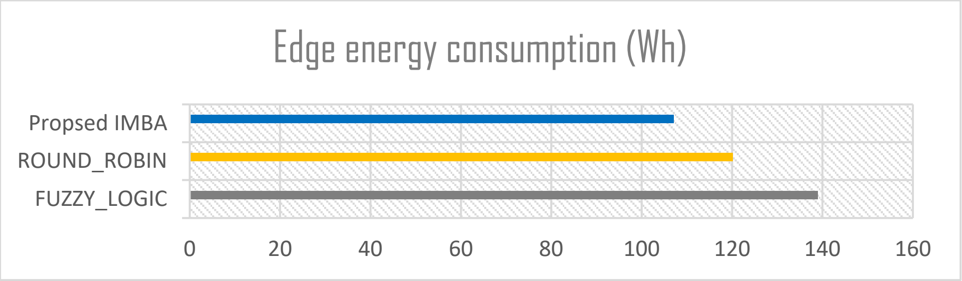

Figure 9. Edge energy consumption (Wh).

The proposed IMBA has demonstrated its capability to allocate a significantly higher number of tasks to the mist layer, along with increased CPU mist usage, as depicted in Figures 10 and 11. This strategic allocation aims to ensure fast response times and minimise waiting times for task execution, leveraging the proximity of mist nodes to IoT devices.

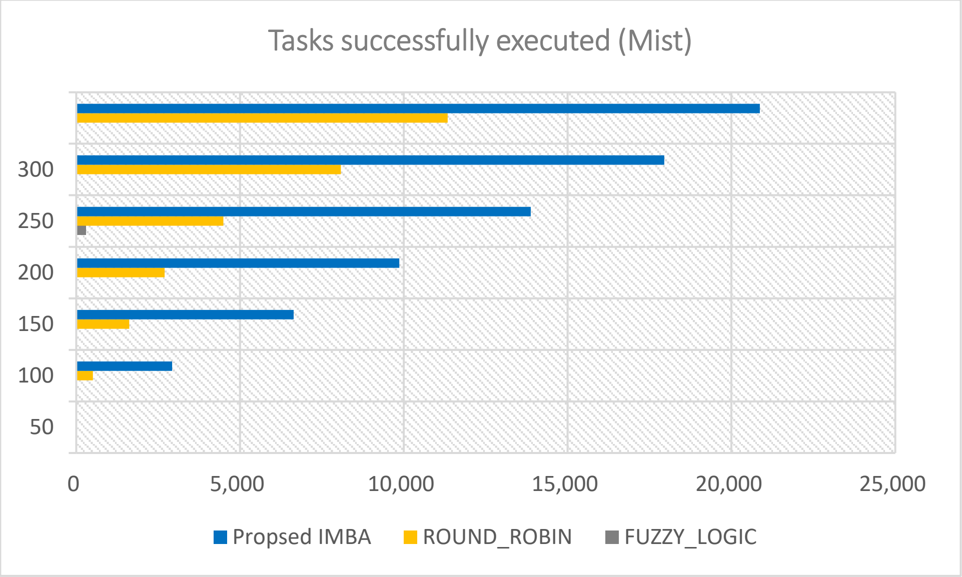

Figure 10. Tasks successfully executed (mist).

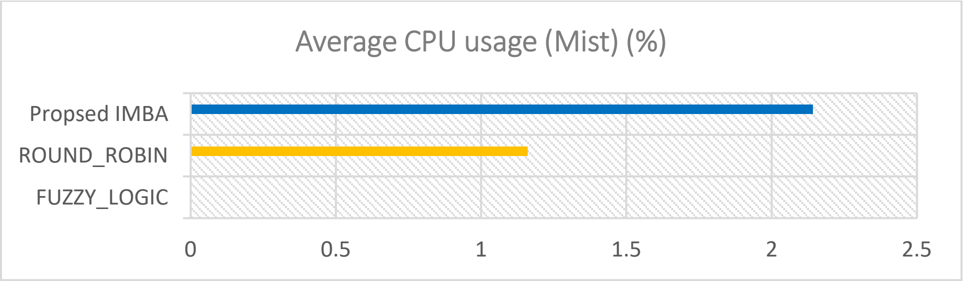

Figure 11. Average CPU usage (mist) (%).

Despite the expectation of increased power consumption due to the intensified task allocation to mist nodes, the IMBA remarkably outperformed expectations. Even with the allocation of 20,870 more tasks than the fuzzy logic algorithm and 9529 more tasks than the round robin algorithm in the mist layer, the algorithm managed to achieve this objective without exhibiting high energy consumption in the mist layer, as illustrated in Figure 12. This remarkable efficiency underscores the effectiveness of the IMBA in optimising resource utilisation while maintaining energy efficiency, thus enhancing the overall performance of IoT networks.

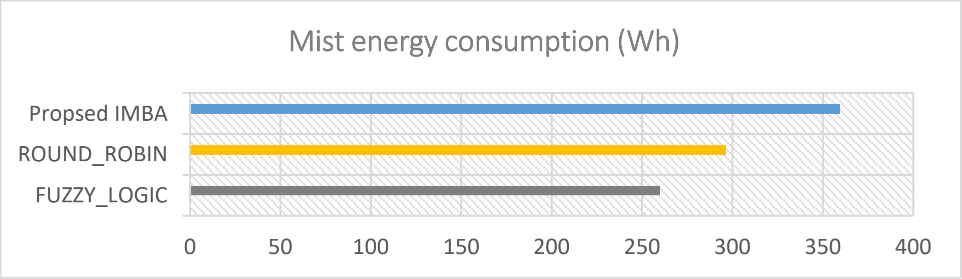

Figure 12. Mist energy consumption (Wh).

The IMBA exhibited superior efficiency in utilising the LAN network by allocating a higher number of tasks compared to other algorithms, as is evident in Figure 13. This strategic allocation aimed to mitigate the dependence on external networks (WAN networks), as depicted in Figure 14, thereby minimising the risk of security vulnerabilities by confining tasks to the internal networks (LAN network), as is also shown in Figure 14.

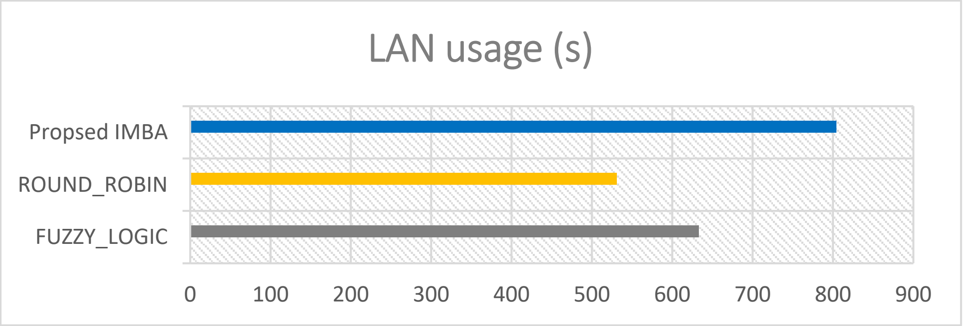

Figure 13. LAN usage (s).

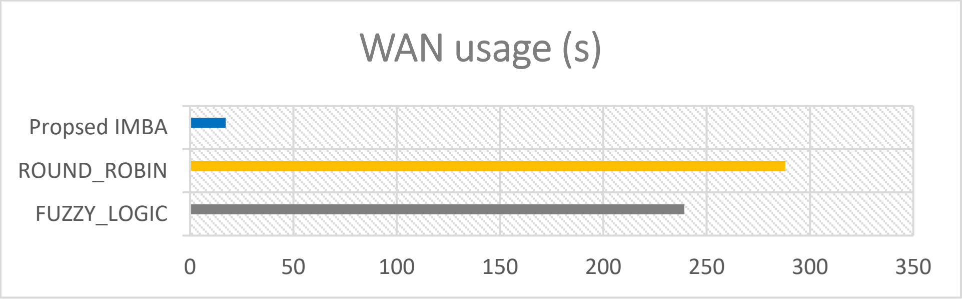

Figure 14. WAN usage (s).

Admittedly, the IMBA demonstrated increased LAN usage compared to other algorithms, aligning with one of the study's primary objectives to leverage LAN resources by allocating more tasks to it. However, remarkably, it achieved this objective without demonstrating excessive LAN usage compared to other algorithms. This balanced utilisation underscores the efficacy of the IMBA in optimising network resources while maintaining security and efficiency in IoT environments.

5. Conclusions and Future Work

This paper presented the IMBA, specifically tailored to optimise resource allocation for IoT devices across the cloud, edge, and mist layers. The algorithm's primary objective is to elevate user service quality in IoT environments. This is achieved through enhancements in successful task completion, load distribution, CPU utilisation, WAN and LAN network usage, and power consumption, serving as key performance metrics for validation. Comparative analysis against fuzzy logic and round robin algorithms reveals IMBA's superiority across all measured metrics.

Overall, IMBA demonstrated its potential to enhance the performance of IoT devices across diverse environments. Our future research work will focus on validating security enhancements by subjecting IoT networks to security attacks, alongside efforts to further reduce LAN usage. Additionally, future efforts will focus on enhancing the scalability of IMBA to efficiently handle increased workloads, thereby mitigating traffic congestion on both the cloud and the edge.

Author Contributions: Conceptualization, Z.A. and B.S.; methodology, Z.A. and B.S.; software, Z.A.; validation, Z.A., B.S. and A.L.; formal analysis, Z.A.; investigation, Z.A. and B.S; resources, Z.A. and B.S.; data curation, Z.A.; writing—original draft preparation, Z.A.; writing—review and editing, Z.A., B.S. and A.L.; visualization, Z.A.; supervision, B.S. and A.L.; project administration, Z.A., B.S. and A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Almudayni, Z.; Soh, B.; Li, A. Enhancing Energy Efficiency and Fast Decision Making for Medical Sensors in Healthcare Systems: An Overview and Novel Proposal. *Sensors* **2023**, *23*, 7286. [CrossRef] [PubMed]
2. Almudayni, Z.; Soh, B.; Li, A. Energy Inefficacy in IoT Networks: Causes, Solutions and Enabling Techniques. In *Energy Inefficacy in IoT Networks: Causes, Solutions and Enabling Techniques*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 110–120.
3. Yang, X.-S. Bat algorithm for multi-objective optimisation. *Int. J. Bio-Inspired Comput.* **2011**, *3*, 267–274. [CrossRef]
4. Eldin Refaat, H. MLITS: Multi-Level tasks scheduling model for IoT Service Provisioning. *FCAI-Inform. Bull.* **2020**, *2*, 1–9.
5. Barik, R.K.; Patra, S.S.; Kumari, P.; Mohanty, S.N.; Hamad, A.A. A new energy aware task consolidation scheme for geospatial big data application in mist computing environment. In Proceedings of the 2021 8th international Conference on Computing for Sustainable Global Eevelopment (INDIACom), New Delhi, India, 17–19 March 2021; IEEE: New York, NY, USA, 2021; pp. 48–52.

6. Tian, M.-W.; Yan, S.-R.; Guo, W.; Mohammadzadeh, A.; Ghaderpour, E. A New Task Scheduling Approach for Energy Conservation in Internet of Things. *Energies* **2023**, *16*, 2394. [CrossRef]
7. Liu, Q.; Kosarirad, H.; Meisami, S.; Alnowibet, K.A.; Hoshyar, A.N. An Optimal Scheduling Method in IoT-Fog-Cloud Network Using Combination of Aquila Optimizer and African Vultures Optimization. *Processes* **2023**, *11*, 1162. [CrossRef]
8. Shuaib, M.; Bhatia, S.; Alam, S.; Masih, R.K.; Alqahtani, N.; Basheer, S.; Alam, M.S. An Optimized, Dynamic, and Efficient Load-Balancing Framework for Resource Management in the Internet of Things (IoT) Environment. *Electronics* **2023**, *12*, 1104. [CrossRef]
9. Liu, Y.; Dong, C.; Qin, X.; Xu, X. Efficient Sensor Scheduling Strategy Based on Spatio-Temporal Scope Information Model. *Sensors* **2023**, *23*, 5437. [CrossRef] [PubMed]
10. Liu, W.; Li, C.; Zheng, A.; Zheng, Z.; Zhang, Z.; Xiao, Y. Fog Computing Resource-Scheduling Strategy in IoT Based on Artificial Bee Colony Algorithm. *Electronics* **2023**, *12*, 1511. [CrossRef]
11. Saurabh; Dhanaraj, R.K. Enhance QoS with fog computing based on sigmoid NN clustering and entropy-based scheduling. *Multimed. Tools Appl.* **2024**, *83*, 305–326. [CrossRef]
12. Suresh, G.M.; Madhavu, M.L. AI based intrusion detection system using self-adaptive energy efficient BAT algorithm for software defined IoT networks. In Proceedings of the 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 1–3 July 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
13. Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.T.; Damaševičius, R. Botnet attack detection using local global best Bat algorithm for industrial internet of things. *Electronics* **2021**, *10*, 1341. [CrossRef]
14. Malek, M.R.A.; Aziz, N.A.A.; Alelyani, S.; Mohana, M.; Baharudin, F.N.A.; Ibrahim, Z. Comfort and energy consumption optimization in smart homes using Bat algorithm with inertia weight. *J. Build. Eng.* **2022**, *47*, 103848. [CrossRef]
15. Olusesi, A.T.; Olaniyan, O.M.; Omodunbi, B.A.; Wahab, W.B.; Adetunji, O.J.; Olukoya, B.M. Energy Management Model for Mobile Ad hoc Network using Adaptive Information Weight Bat Algorithm. e-Prime-Advances in Electrical Engineering. *Electron. Energy* **2023**, *5*, 100255.
16. Sundas, A.; Badotra, S.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. Modified Bat Algorithm for Optimal VM's in Cloud Computing. *Comput. Mater. Contin.* **2022**, *72*, 2877–2894. [CrossRef]
17. Nakamura, R.Y.M.; Pereira, L.A.M.; Rodrigues, D.; Costa, K.A.P.; Papa, J.P.; Yang, X.-S. Binary Bat algorithm for feature selection. In *Swarm Intelligence and Bio-Inspired Computation*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 225–237.
18. Mechalikh, C.; Taktak, H.; Moussa, F. PureEdgeSim: A simulation framework for performance evaluation of cloud, edge and mist computing environments. *Comput. Sci. Inf. Syst.* **2021**, *18*, 43–66. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.