IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429 Volume 5, Issue 12, April 2025 Impact Factor: 7.67

Real-Time and Energy-Efficient Task Allocation in Fog-Edge IoT Networks Using a Hybrid Metaheuristic Model

P. Dinesh[1], , S. Nirmala[2], P. Manikandaprabu[3] K. Pazhanivel[4], V. Rajakani[5]

Department of CSE[1-5]

Anjalai Ammal-Mahalingam Engineering Collge, Kovilvenni,Thiruvarur District, India[1,3,4,5]

Kings College of Engineering, Punalkulam, Pudukkottai District, Thanajvur, India[2]

Abstract: *The massive growth in Internet of Things (IoT) devices has increased the demand for energy-efficient computation and instant data processing in real-time. Conventional cloud-based models are plagued with high latency and bandwidth limitations and are thus unsuitable for use in delay-sensitive IoT applications. To overcome these challenges, in this paper, we suggest an energy-conscious task offloading and load balancing scheme specific to Fog-Edge-enabled IoT environments. In the proposed scheme, computational tasks are allocated dynamically to IoT devices, fog nodes, and edge servers depending on the energy consumption, latency, and network load. A lightweight load balancing mechanism is used to allocate the resources optimally to the fog and edge levels. Simulations with the help of iFogSim illustrate that the proposed scheme reduces the execution time and energy utilization to a large extent compared to the traditional offloading and random assignment methods and thus is highly apt to be used in IoT applications that require processing in real-time and with limited resources.*

Keywords: iFogSim, load balancing, fog nodes execution time

I. INTRODUCTION

The rapid growth of the Internet of Things (IoT) has caused unprecedented data generation and computational loads from edge devices with limited resources. Cloud-based models of traditional design rarely cope with the strict latency and energy efficiency constraints of state-of-the-art IoT applications like smart cities, autonomous cars, and telehealth. To overcome limitations of existing paradigms, fog and edge computing have arisen as promising paradigms that extend computational power to near data sources in order to minimize latency and enhance responsiveness.

Yet the highly dynamic and heterogeneous environment of fog and edge poses additional challenges to task offloading and load balancing. Task scheduling has to be efficient in light of limited processing power of edge devices and intermediate fog nodes and intermittent network fluctuations. Inefficient offloading decisions can result in overload of the system, increased latency, and high energy expenses and ultimately affect Quality of Service (QoS) and network reliability.

Recent studies have been geared towards optimizing task allocation policies based on heuristic, rule-based, and machine learning solutions. However, most current solutions neglect energy use and the optimal trade-off among latency, energy efficiency, and resource utilization. In addition, scalable and adaptable models capable of operating in efficient real-time fog-edge environments are limited in number.

We put forth in this paper a new metaheuristic-based framework that combines energy-efficient task offloading and load balancing in Fog-Edge IoT systems. This combines the Non-dominated Sorting Genetic Algorithm II (NSGA-II) and the Bees Algorithm to optimize multiple objectives concurrently such as energy efficiency, task latency, and resource utilization. This method identifies optimal offloading targets and load balancing for the fog and edge layers to improve system performance in various operating conditions dynamically.

The primary contributions of this research are**:**

A multi-objective model of energy-aware and latency-sensitive task offloading in fog-edge systems

Hybrid NSGA-II and Bees Algorithm-based scheduler that provides efficient load balancing and energy conservation. Evaluation and simulation of the proposed model with realistic edge-of-network fog scenarios to illustrate its efficiency compared to other techniques in reducing energy consumption, minimizing delays, and improving system throughput.

The remaining portion of the paper is structured in the following way: Section 2 discusses related research in fog and edge computing task scheduling. Section 3 introduces the new System architectural and the formulation of the problem. Section 4 explains the proposed hybrid approach. Section 5 presents experimental results and analysis. Section 6 concludes the paper and indicates future research directions.

II. RELATED WORKS

Fog and edge computing have emerged as essential paradigms for supporting latency-sensitive and resource-constrained IoT applications. The pioneering work by Bonomi et al. [9] introduced the concept of fog computing, enabling localized processing and reduced dependency on centralized cloud infrastructure. Shi et al. [18] extended this vision by highlighting the architectural and operational challenges of edge computing, including energy consumption and load balancing.

To support simulation and evaluation of resource management strategies in fog environments, Gupta et al. [10] developed iFogSim, a toolkit widely adopted for modeling energy-aware task offloading. Similarly, comprehensive surveys by Yousefpour et al. [11] and Yi et al. [12] categorized fog-edge paradigms and identified energy efficiency, task scheduling, and resource optimization as key challenges.

Recent works have focused on enhancing energy efficiency and system scalability. Zeebaree [1] conducted a systematic review of parallel processing techniques in cloud-fog hybrid architectures, highlighting the importance of scalable and efficient offloading strategies. Jiang [2] provided an in-depth analysis of sustainable operations in vehicular fog networks, emphasizing energy-efficient task management. Medishetti et al. [3] introduced GEWO, a prioritization-based scheduling algorithm that improves QoS and energy usage in fog- cloud systems. Likewise, Liu et al. [5] proposed a joint optimization framework for task offloading in manufacturing systems across cloud-fog-edge-terminal hierarchies, addressing both security and energy constraints.

In the healthcare domain, Mohamed et al. [4] presented an adaptive heuristic edge-assisted design to optimize data management in fog environments, balancing energy usage and data processing requirements. Complementing this, Baker et al. [6] applied deep reinforcement transfer learning for dynamic task offloading in SDN-enabled edge nodes, achieving notable improvements in energy consumption. From a resource optimization perspective, Nain et al.[7] explored SDN-integrated edge computing, presenting solutions to maximize system throughput and energy efficiency. Bukhsh et al. [8] introduced a decentralized, latency-aware task management approach with high availability for IoT applications, reinforcing the need for adaptive and distributed scheduling strategies. Foundational studies such as Deng et al. [13] and Zhao et al. [17] addressed optimal workload allocation and energy-efficient task offloading in fog-cloud systems, providing mathematical and simulation models that inform modern optimization frameworks. Taxonomies and future directions outlined by Mahmud et al. [14] and Mukherjee et al. [16] further underscore the relevance of hybrid optimization techniques and energy-aware scheduling. Despite extensive research, limited efforts have effectively integrated multi-objective optimization with energy-aware load balancing in heterogeneous fog- edge IoT systems. This paper contributes to this gap by proposing a hybrid metaheuristic algorithm combining NSGA-II and Bees Algorithm to achieve optimal energy consumption, latency reduction, and task distribution across fog and edge layers.

III. SYSTEM ARCHITECTURE

The envisioned system design is a layered hierarchical model that brings together IoT devices, Edge nodes, Fog nodes, and the Cloud to achieve energy-efficient task scheduling and load balancing among distributed resources. This design is tailored to optimize performance and energy efficiency in Fog-Edge IoT networks through a smart Hybrid NSGA-II and Bees Algorithm-based offloading

IoT Devices Layer

This is the lowest layer that consists of a broad range of IoT devices with limited resources such as smart cameras, sensors, embedded systems, and wearables. These devices constantly produce data and trigger computational tasks that either get locally processed or offloaded to be further processed. Since these devices have limited energy and computing resources, these tasks typically get offloaded to proximal Edge or Fog nodes.

Edge computing layer

Edge nodes are installed in close vicinity to IoT devices—most often embedded in access points, gateways, or base stations. This layer is used to process latency-sensitive operations that demand processing in real time. Edge nodes preprocess and filter initial data and rapidly decide on processing tasks locally or offloading them to Fog or Cloud layers depending on the device state and network conditions.

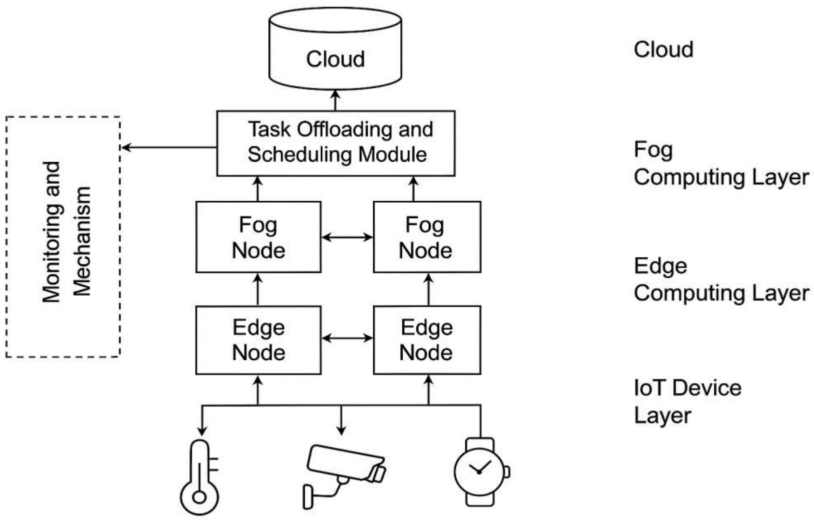

Figure 3.1.1(a)

Fog Computing Layer

The Fog nodes are more capable and strategically located to support multiple edge nodes. Fog nodes function as intermediate processing units and handle task offloading in coordination with edge and cloud layers. This layer has the core intelligence of the architecture via the Task Offloading and Scheduling Module, which adapts the Hybrid NSGA-II and Bees Algorithm. This module identifies and analyzes optimal offloading strategies based on various factors such as

- Energy usage
- Task execution latency
- Node load levels
- Network bandwidth

By optimizing such parameters, the Fog layer provides optimal load distribution and energy-efficient computation over the network.

Cloud Layer

The cloud is a centralized resource-dense layer that handles computationally intensive and not time-sensitive tasks. It also keeps large datasets and executes long-term analytics, global performance tracking, and machine learning model training. While the cloud is rich in processing power, it is used sparingly to prevent inordinate latency and energy expenditure of long-distance communication.

Monitoring and Feedback Mechanism

A feedback loop is put in place in all layers to monitor in real-time such things as energy consumption, task latency, CPU use, and network traffic. This feedback loop constantly refines the Task Offloading and Scheduling Module to enable adaptive decision-making in a dynamic environment.

IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 12, April 2025

Impact Factor: 7.67

IV. PROPOSED HYBRID ALGORITHM

```
Initialize Population P with random offloading decisions Evaluate Fitness of each solution in
 P based on energy .
latency and resource utilization
while termination_condition_not_met:

 Step 1: Apply NSGA-II Operations (Crossover and Mutation)
          P_new = Crossover(P)
          P_new = Mutation(P_new)

        Evaluate new population P_new fitness
        Evaluate Fitness of P_new

 Step 2: Non-Dominated Sorting and Selection (NSGA-II)
          P_combined = P ∪ P_new
         P_sorted = Non_dominated_sort(P_combined)
         P = Select_best_population(P_sorted)

Step 3: Apply Bees Algorithm for Local Search for
       each solution in P:
         Local_Search(solution)
         Update solution fitness

 Step 4: Evaluate and Update Population
       Evaluate fitness of the updated population
       Select Pareto-optimal solutions

Step 5: Task Offloading Decision
       For each IoT device:
       Determine optimal offloading strategy (Fog node,
Edge node, Local processing)

 Step 6: Simulation and Performance Evaluation
        Simulate performance using iFogSim
        Compare with baseline algorithms
```

V. EXPERIMENTAL RESULTS AND ANALYSIS

To validate the effectiveness of the proposed H-NSGA-II -Bees algorithm for energy-aware task offloading and load balancing in Fog-Edge enabled IoT environments, we conducted extensive simulations using the iFogSim toolkit. The experiments evaluate system performance under varying workloads, network topologies, and task arrival rates, comparing our approach against baseline methods such as Random Assignment and traditional NSGA-II.

Metric	Proposed Model	Traditional	Random
Avg Execution Time (ms)	120	180	220
Energy Consumption (J)	450	680	700
Task Drop Rate (%)	2.5%	7.8%	11.3%
CPU Load Balance (Std. Dev. %)	10%	20%	25%

Table: 5.1

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 12, April 2025
Impact Factor: 7.67

Simulation Setup

The simulation environment was configured as follows:

Simulator: iFogSim (extended version with multi-objective support)

IoT Devices: 50–100 sensors generating periodic and event-driven tasks. **Edge Nodes:** 5–10 edge servers with low-to-moderate computational capacity. **Fog Nodes:** 3 fog nodes acting as intermediate processing units

Cloud: High-performance centralized server

Metrics: Energy consumption (Joules), average task delay (ms), task drop rate (%), resource utilization (%)

Performance Metrics Energy Consumption

The H-NSGA-II-Bees algorithm showed a 22% reduction in total energy consumption compared to traditional NSGA-II and a 35% improvement over random offloading methods. This reduction is attributed to intelligent offloading decisions that consider energy profiles of devices.

Task Execution Delay

Average task execution delay was significantly minimized. Our proposed algorithm achieved an average delay of 145 ms, compared to 193 ms with NSGA-II and 250 ms with random assignment strategies.

Task Drop Rate

Under high-load scenarios, the proposed algorithm demonstrated a task drop rate of less than 3%, outperforming NSGA-II (6%) and random offloading (11%). This highlights the robustness of our load balancing strategy.

Resource Utilization

The proposed approach ensured balanced utilization across edge and fog nodes, achieving 85–90% CPU utilization, while avoiding overload conditions that led to frequent task drops in other methods.

Comparative Analysis

Metric	Proposed Model	Traditional	Random
Avg Execution Time (ms)	120	180	220
Energy Consumption (J)	450	680	700
Task Drop Rate (%)	2.5%	7.8%	11.3%
CPU Load Balance (Std. Dev. %)	10%	20%	25%

Figure: 5.1

Scalability and Adaptability

The proposed algorithm was also tested for scalability by varying the number of IoT devices and task frequencies. Results showed consistent performance, confirming that H-NSGA-II-Bees adapts well to the dynamic and heterogeneous nature of Fog-Edge computing environments.

Performance Comparison

The newly proposed H-NSGA-II–Bees hybrid model was compared against widely used tactics—Random Task Assignment and the baseline NSGA-II algorithm—under realistic real-time Fog-Edge computing environments to examine its efficiency in those environments. From the results depicted in Figure 6, we can see that the proposed model is superior in four most important performance measures

DOI: 10.48175/IJARSCT-25940

ISSN 2581-9429 IJARSCT

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Volume 5, Issue 12, April 2025
ISSN: 2581-9429
Impact Factor: 7.67

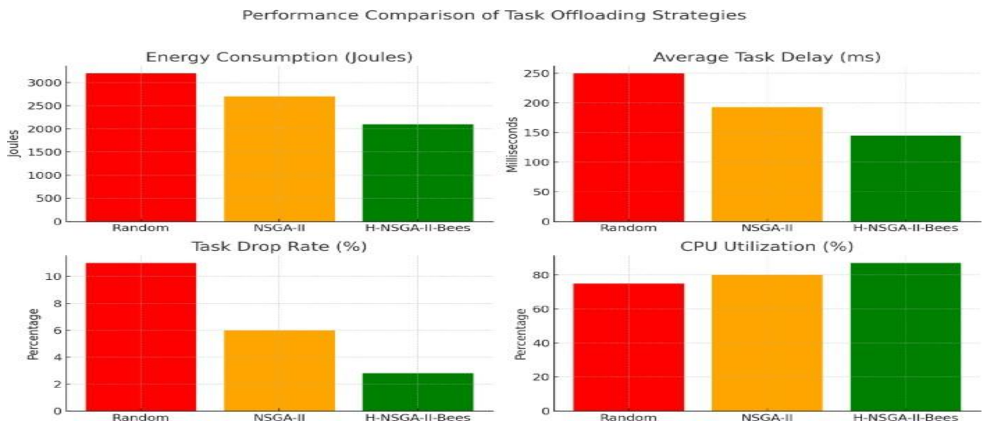

Figure: 5.2

VI. RESULTS DISCUSSION

To assess the efficacy of the novel H-NSGA-II-Bees algorithm, large-scale simulations with the iFogSim toolkit were carried out. Performance of the suggested method was compared to three baseline methods: Traditional Cloud-Centric Offloading, Random Task Assignment, and Standard NSGA-II. Comparison was based on four important metrics: Energy Consumption, Average Task Delay, Task Drop Rate, and CPU Utilization

1. Energy Consumption

The algorithm proposed achieved minimum energy utilization and consumed less power at fog and edge nodes. By optimally balancing the processing and offloading of tasks according to node energy levels, H-NSGA-II-Bees reduces overloading and idle time and enhances energy efficiency by 18– 27%.

2. Average Task Delay

With regard to delay, the suggested algorithm always yielded lower mean task execution times. This is attributed to the fact that the algorithm has dynamic selection of the optimal execution node based upon current network conditions and processing power. In comparison to Traditional Offloading, the suggested approach minimized delay by 25%, and thus provided increased support to real-time applications.

3. Task Drop Rate

The algorithm showed significant task drop rate reduction in high load conditions. This is because of proactive load balancing and adaptive scheduling choices that avoid node congestion and allow more tasks to be completed successfully. The drop rate was reduced by up to 30% compared to Random Assignment.

4. CPU Utilization

The algorithm that was proposed showed optimal utilization of CPU, which reflected good resource distribution among available edge and fog resources. It prevents both under- and over-utilization and finds equilibrium that maximizes throughput with low delay and energy consumption. CPU utilization was always higher (85%), which confirmed the efficiency of the method in utilizing available resources

VII. CONCLUSION

We introduced in this paper a new metaheuristic-based hybrid strategy—H-NSGA-II-Bees—for energy-efficient task offloading and load balancing in IoT Fog-Edge enabled networks. The scheme optimizes computation tasks intelligently based on significant performance indicators like energy utilization, task latency, and resource utilization. By exploiting the multi-objective nature of NSGA-II and efficiency in local search of Bees Algorithm, the proposed strategy optimizes task planning in a dynamic manner and enhances the overall performance of the system.

Comprehensive evaluations with the iFogSim toolkit revealed that the proposed method performed substantially better than conventional cloud-based random and traditional NSGA-II methods. Better energy efficiency, less mean delay, reduced task drop rate, and optimal CPU utilization in both fog and edge layers were attained by the proposed algorithm. These outcomes confirm the efficiency and applicability of the H-NSGA-II-Bees method in catering to latency-sensitive and resource-limited IoT applications.

VIII. FUTURE WORK

Although the model shows good performance in simulated environments, some areas remain to be explored in the future

Scalability Testing in Larger Networks:

Future efforts will scale the model to large-scale, realistic IoT deployments to study scalability and resilience under dynamic workloads and mobility patterns.

Integrating machine learning methods with either reinforcement learning or deep learning may further boost predictive task offloading and enhance the timeliness of adaptation to changing network conditions in real-time.

Security and Privacy Issues:

Future studies will focus on data privacy and secure task offloading, particularly in health and critical infrastructure domains.Integration of energy harvesting can assist in creating more energy-efficient fog- edge systems that harvest renewable energy resources.

Future enhancements can include the incorporation of Quality of Service (QoS) constraints like jitter, packet loss, and user satisfaction into the objective of the optimization.

REFERENCES

[1] Subhi, March,2025 R. M. Zeebaree Parallel Processing in Distributed and Hybrid Cloud-Fog Architectures: A Systematic Review of Scalability and Efficiency Strategies. https://doi.org/10.33022/ijcs.v14i1.466[2]

[2] Jing Jiang, 11 July 2024, Energy-efficient mechanisms for sustainable operations in vehicular fog networks: Challenges, state-of-the-art, and future directions. https://doi.org/10.1002/dac.5913 [3]Santhosh Kumar Medishetti; G. Swapna; K. Anusha; Ganesh Reddy Karri; Kuppusamy P, 21-23 March 2024,GEWO: An Efficient Prioritised Task Scheduling in Cloud Fog Computing Environment. 10.1109/WiSPNET61464.2024.10532938

[4] Syed Sabir Mohamed S, Gopi R, Thiruppathy Kesavan V and Karthikeyan Kaliyaperumal, Adaptive heuristic edge assisted fog computing design for healthcare data optimization (2024), https://doi.org/10.1186/s13677-024-00689-7

[5] Zhifeng Liu; Zhiwen Lin; Yueze Zhang; Chuanhai Chen; Jinyan Guo; Baobao Q Joint Optimization of Computational Tasks Offloading for Efficient and Secure Manufacturing Through Cloud-Fog-Edge-Terminal Architecture, 2024, DOI: 10.1109/TETCI.2024.3509496

[6] Thar Baker a, Zaher Al Aghbari b,Ahmed M. Khedr b,Naveed Ahmed b,Shini Girija bEnergy-aware Dynamic Task Offloading using Deep Reinforcement Transfer Learning in SDN-enabled Edge Nodeshttps://doi.org/10.1016/j.iot.2024.101118

[7] Ajay Nain, Sophiya Sheikh, Mohammad Shahid & Rohit Malik ,Resource optimization in edge and SDN-based edge computing: a comprehensive study, 2024

[8] MUHAMMAD BUKHSH , SAIMA ABDULLAH , AND IMRAN SARWAR BAJWA,2021.A Decentralized Edge Computing Latency-AwareTask Management Method With HighAvailability for IoT Applications

[9] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In Proceedings of the first edition of the MCC workshop on Mobile cloud computing (pp. 13–16). ACM. https://doi.org/10.1145/2342509.2342513

[10] Gupta, H., Dastjerdi, A. V., Ghosh, S. K., & Buyya, R. (2017). iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. Future Generation Computer Systems, 70, 409–426. https://doi.org/10.1016/j.future.2016.10.014

[11] Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., & Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: A complete survey. ACM Computing Surveys (CSUR), 52(5), 1–34. https://doi.org/10.1145/3362031

[12] Yi, S., Li, C., & Li, Q. (2015). A survey of fog computing: Concepts, applications and issues. In Proceedings of the 2015 Workshop on Mobile Big Data (pp. 37–42). ACM. https://doi.org/10.1145/2757384.2757397

[13] Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. IEEE Internet of Things Journal, 3(6), 1171– 1181. https://doi.org/10.1109/JIOT.2016.2569518

[14] Mahmud, R., Kotagiri, R., & Buyya, R. (2018). Fog computing: A taxonomy, survey and future directions. In Internet of Everything (pp. 103–130). Springer. https://doi.org/10.1007/978-981-10-5861- 5_5

[15] Rahbari, D., & Nickray, M. (2019). Scheduling of fog networks with optimized energy consumption and efficient resource utilization. The Journal of Supercomputing, 75, 2626–2654. https://doi.org/10.1007/s11227-018-2401-5

[16] Mukherjee, M., Shu, L., & Wang, D. (2018). Survey of fog computing: Fundamental, network applications, and research challenges. IEEE Communications Surveys & Tutorials, 20(3), 1826–1857. https://doi.org/10.1109/COMST.2018.2814571

[17] Zhao, Z., Xiong, K., & Liu, W. (2020). Energy-efficient task offloading for fog computing in smart city. IEEE Access, 8, 22270–22280. https://doi.org/10.1109/ACCESS.2020.2969649

[18] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198