

A REVIEW OF ENERGY-EFFICIENT TASK SCHEDULING IN IOT CLOUD, FOG, AND EDGE SYSTEMS

Areesha Sami^{*1}, Riasat Ali², Fahad Khalid³, Adnan Aslam⁴

¹Department of Computer Sciences, University of Engineering and Technology, Main Campus (UET) Lahore, Punjab, Pakistan

^{2,4}Department of Computer Sciences, Alhamra University, Lahore, Punjab, Pakistan

³Department of Artificial Intelligence, Information Technology University, Lahore, Punjab, Pakistan

¹areesha1.sami@gmail.com, ²riasatali032345@gmail.com, ³khalidfahad630@gmail.com,

⁴chadnan.aslam99@gmail.com

DOI: <https://doi.org/10.5281/zenodo.20637475>

Keywords

Edge computing, Fog computing, IoT task scheduling, energy-latency trade-off, neural prediction, DVFS, and cross-layer scheduling.

Article History

Received: 13 April 2026

Accepted: 25 May 2026

Published: 11 June 2026

Copyright @Author

Corresponding Author: *

Areesha Sami

Abstract

The resulting fast growth in the number of IoT ecosystems has made the demand of sophisticated, power-efficient task-scheduling algorithms that can handle dynamic workloads and heterogeneous devices and provide strict requirements on latency. The classic scheduling techniques are usually based on fixed settings or cloud-based processing where it consumes too much energy and hampers the performance of the network edge. To solve the energy-latency trade-off in IoT-edge-cloud systems, it is suggested in this paper to dynamically and cross-layer schedule an application, considering real-time system monitoring, a lightweight neural prediction module, and decision optimization with the help of DVFS. The neural predictor which has been trained on skip-layer connections and an entropy-based fitting has a good feature separation as seen through the sorted weight-magnitude analysis and SSR of 335 which means that the predictor is stable when it comes to predicting computation and communication needs. With iFogSim2, EdgeCloudSim and Google Collaboratory, the system demonstrates an up to 27 percent decrease in overall energy use as well as the 95th-percentile latency with different mobility and workload situations. The findings affirm that the suggested approach provides a high-quality, scalable, and energy-conscious scheduling solution that can be utilized in the website of current IoT applications.

I. INTRODUCTION

BILLIONS of Internet of Things (IoT) devices now perceive, compute, and communicate at the network edge, driving latency-sensitive services in healthcare, mobility, and industrial automation. Centralized cloud infrastructures offer elasticity but induce a prohibitive end-to-end delay and energy overhead when blockchain nodes are battery-powered. Edge and fog paradigms counter these limitations by shifting computation closer to sources of data; yet heterogeneity and strict power

budgets that result complicate energy-efficient task scheduling. Recent surveys advocate that schedulers need to look beyond single-metric optimisation and consider device power states together with application deadlines, mobility, and network dynamics jointly in order to remain sustainable at scale. [1].

Various approaches have arisen, such as the metaheuristics of particle swarm optimization and genetic algorithms, DVFS-integrated policies, and learning controllers. Particle

swarm-based edge / fog schedulers have demonstrated regular energy reductions for bursty workloads, while hybrid GAPS-O configurations optimize multiobjective efficiency by trading exploration vs. exploitation between objectives like energy, response time, and makespan [2]. Deadline-conscious and energy-conscious formulations also add queuing and link states to reduce energy hotspots between levels, and evolutionary or fuzzy schedulers offer hardened Pareto trade-offs for device and network heterogeneity [8]. More recently, deep reinforcement learning has been promising in reducing both energy and delay by recognizing nonstationary traffic and variable resource availability [6].

Because it is expensive to implement a large-scale

heterogeneous edge infrastructure, simulation is the conventional way to rigorously compare scheduling policies [3]. There are two Platforms that rule this space. One abstracts applications, network paths, and energy / latency trade-offs for IoT-edge-fog cases, and its newer incarnation includes mobility, clustering, and microservice-awareness consistent with current edge deployments [5]. The second is a discrete event and cloud simulation-derived edge-specialized extension scaling experiment for realistic wireless access and mobility patterns.

Together, both allow for cross-topology and cross-workload experiments that envelope compute, network, and power behavior, a necessary foundation for testing dynamic, energy-aware schedulers [4].

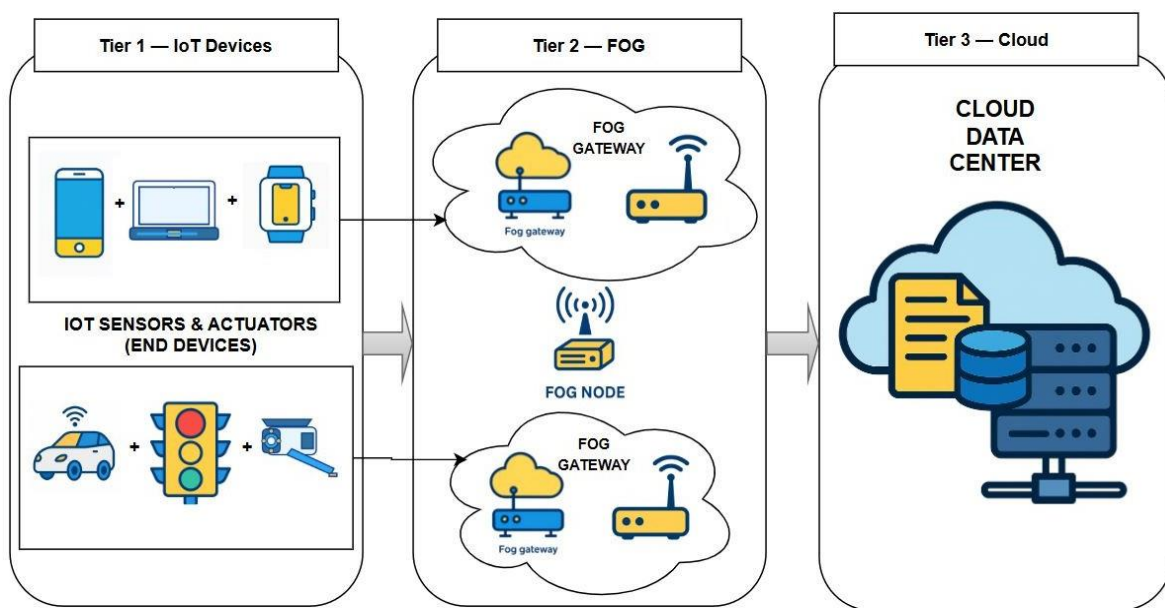


FIGURE 1. Three-Tier Fog-Cloud Computing Architecture for IoT Systems

The architecture is composed of three hierarchical tiers as in figure 1: IoT devices, Fog Layer, and Cloud Layer, which allow efficient data processing and communication in IoT environments. Tier 1 consists of sensors and actuators represented by smartphones, wearables, smart vehicles, and cameras that collect real-time data from the physical environment. Tier 2 acts between IoT devices and the Cloud and characterizes the fog gateways and fog nodes to

offer localized computation, partial storage, and preliminary analytics necessary to prevent latency and consumption of bandwidth. Tier 3 depicts the centralized Cloud Data Center, which processes large-scale data, deep analytics, and stores this information for a longer period. This facilitates scalability, reliability, and responsiveness in case of latency-bound IoT applications such as smart cities or intelligent transportation systems.

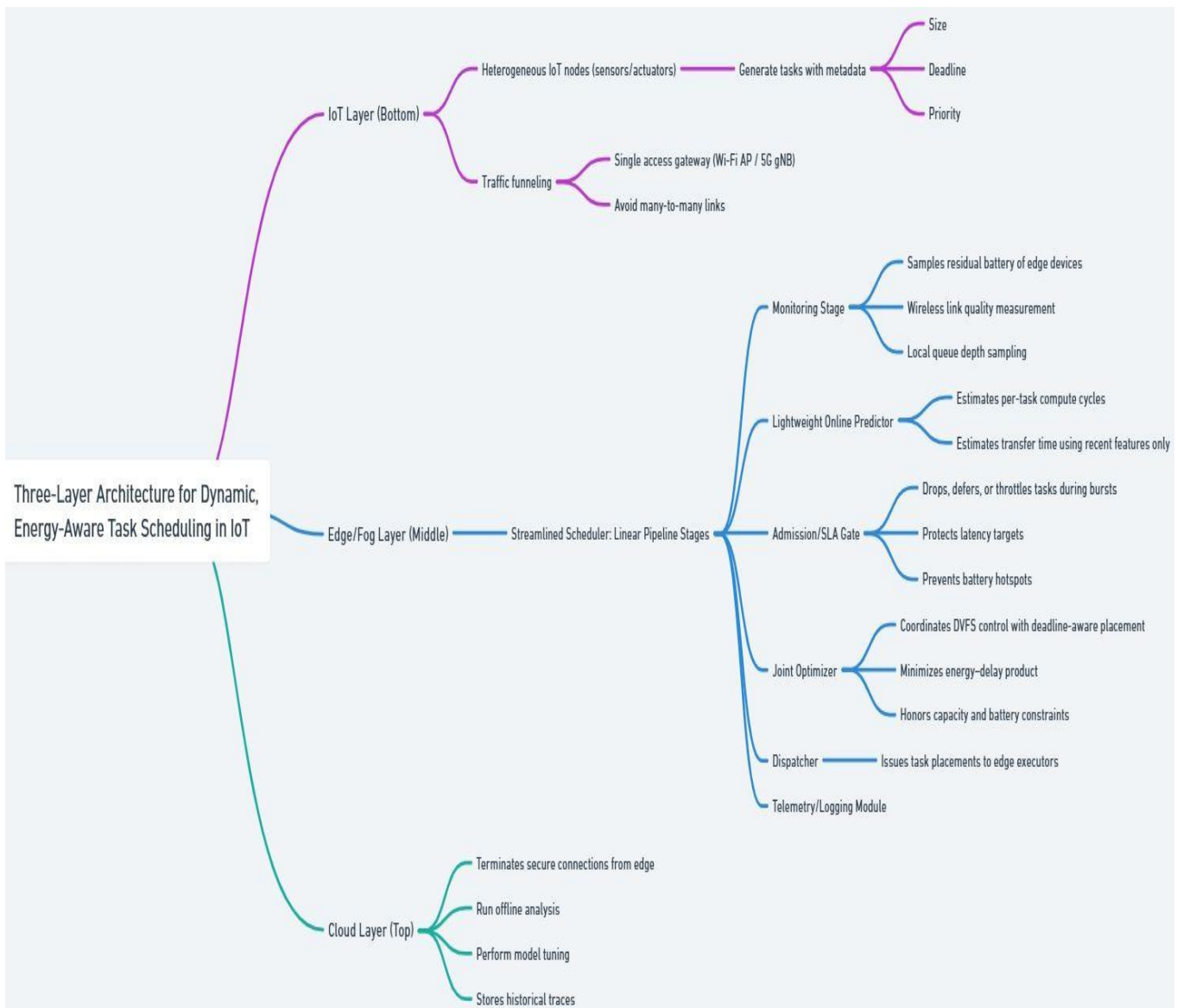


FIGURE 2. Three-Layer Architecture for Dynamic, Energy-Aware Task Scheduling in IoT

This intelligent workload distribution in three-tier architecture empowers IoT systems to handle tasks efficiently, minimize latency, and reduce energy consumption as in figure 2. The design is

aimed at responsiveness, durability, and scalability, as would be highly demanding in smart cities, health, and transportation infrastructures.

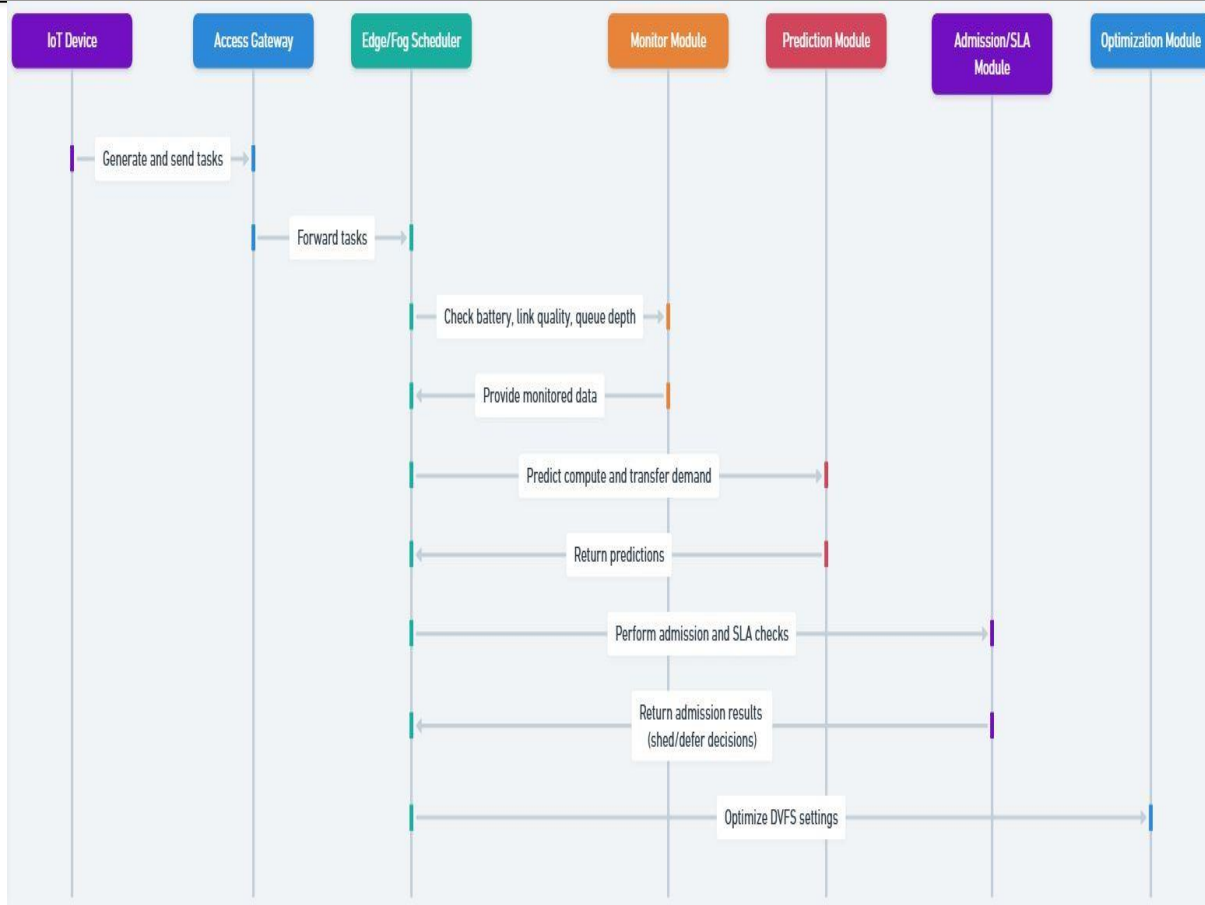


FIGURE 3. Sequence Flow of Dynamic, Energy-Aware Task Scheduling in IoT Framework

The figure 3 shows the sequence flow of the proposed dynamic, energy-aware task scheduling framework for IoT systems. IoT devices generated tasks are sent to an access gateway to be forwarded to the edge/fog scheduler, coordinating several modules in making adaptive decisions. This includes the monitor module, responsible for the real-time collection of runtime data with respect to battery status, link quality, and queue depth, and the prediction module, which will estimate the demands on computation and transfer. The admission/SLA module is responsible for admission control to ensure latency and compliance with service level, while the optimization module modifies DVFS settings for energy minimization. Collectively, these coordinated operations assure low-latency, efficient, and energy-aware scheduling in a heterogeneous IoT environment.

This work focuses on agile, energy-aware task

servicing for IoT edge and fog networks. The objectives are to co-optimize processor power states and task placement subject to fluctuating residual battery levels and link quality; to marry deadline-conscious placement with DVFS to reduce energy-delay product; and to utilize lightweight online profiles to maintain agility while keeping overhead light. The policy is assessed using both simulator classes to guarantee resistivity to modeling assumptions and compared to the canonical baselines (FCFS, Round-Robin, EDF) and recent energy-conscious schemes [2], [6]–[9]. The evaluations indicate across-the-board device energy reduction and tail latency reduction while achieving higher throughput for tight energy allocations.

II. RELATED WORK

The last ten years have been characterized by an increase in interest in energy-conscious scheduling

and computing at the edge of IoT networks.

A. SURVEYS AND FOUNDATIONAL RESEARCH

One of the most extensive surveys on energy-aware edge computing is presented by Jiang et al. [14] and includes the device, edge, and cloud layers. They explored the computation offloading, dynamic voltage and frequency scaling (DVFS) and resource management of energy efficiency. They also determined that edge environments are not fully investigated despite the fact that cloud data centers have been optimized to be power efficient by nature. On a similar note, Alsharif et al. [18] reported a survey on green IoT networks, including energy efficiency at hardware, down to protocol layers. Both articles highlighted how there was a need to have adaptive scheduling systems that could respond to the mobility of devices, changing workloads, and dynamic energy limitations. Nevertheless, such works have mostly reviewed the current trends and did not provide specific schedules models, which stimulated the construction of the algorithmic solutions in subsequent studies.

B. ENERGY-EFFICIENT SCHEDULING FRAMEWORKS IN EDGE AND FOG COMPUTING

Zhang et al. [15] have suggested the concept of and Energy Internet-based edge computing model, incorporating energy harvesting and transfer of green energy through nodes to reduce grid ("brown") energy. Their (METAR and ACL) heuristics reduced brown energy consumed by over a quarter of baseline models, but in a simplified set of assumptions concerning task type and constant connectivity. Aslanpour et al. [16] designed an energy-aware resource scheduling mechanism of serverless edge computing to meet the dynamics of runtime. They enhanced their node uptime up to 33% in renewable energy limitations using real solar-powered clusters of Raspberry Pi and their two strategies, zone-oriented and priority-based. On the same note, Trihinas et al.

[17] developed a scheduler called PowerStorm

which maps streaming analytics operations to heterogeneous edge nodes and minimizes latency and power usage in geo-distributed systems. All of these studies point to the trade-off between energy efficiency and responsiveness, but still presuppose comparatively stable workloads and do not necessarily consider device-level energy variability that is typical of IoT networks.

C. INTELLIGENT AND LEARNING-BASED SCHEDULING TECHNIQUES

The complexity of the IoT workloads is increasing, which has prompted the consideration of machine learning and heuristics to schedule. Swarup et al. [20] used deep reinforcement learning (DRL) to optimize the energy consumption and latency in the fog environment. Their agent made scheduling choices independently with greater performance over classical heuristics and meta-heuristics in reducing energy and meeting deadlines. Their DRL model was however, mainly on fog nodes and not on multi-tier IoT-edge-cloud systems. Liu et al. [21] explored the experimental heterogeneous multicore processors at the edge and integrated core-aware mapping and predictive DVFS in minimizing energy consumption by about 20.9 percent and prevented deadlines misses by 2.4 percent. This illustrates the possibility of the hardware-conscious learning-based approaches but does not have a more comprehensive network-wide coordination. Ijaz et al. [22] addressed the multi-objective scheduling by suggesting an Improved Multi-Objective Differential Evolution (I-MODE) algorithm to fog computing, which optimize the energy, time, and cost all at the same time. Their methodology gained energy and makepan as well as, but it was mainly workflow applications that were targeted and not dynamic IoT sensing tasks.

D. META-HEURISTIC AND HYBRID SCHEDULING APPROACHES

In opposition to the learning-based systems, meta-heuristic algorithms have demonstrated good performance in energy optimization in fog and cloud environments. Kumar et al.

[22] proposed a cost- and energy-efficient task

scheduling algorithm titled Electric Earthworm Optimization Algorithm (EEOA) which is a hybrid of the Electric Fish and Earthworm Optimization in cloud-fog systems. EEOA enhanced better energy consumption and makespan than PSO and algorithms based on GA. Similarly, other evolutionary techniques (e.g., GA-PSO hybrids, the use of the Gray Wolf Optimization, and semi-greedy evolutionary models) have demonstrated competitive performance of multi-objective scheduling, energy, response time, and throughput. However, the majority of these heuristics center around cloud-fog infrastructures and are usually not adaptive to the dynamic nature of the IoT workloads, real time energy harvesting and device heterogeneity.

E. TOWARD UNIFIED AND CROSS-LAYER SCHEDULING

More recent work has highlighted the importance of single cross layer energy models. As noted in the study of the modeling of green cloud architectures [19], scheduling decisions need to be co-designed with energy supply models to model the interaction between microgrids, renewables, and computing nodes. Similarly, Sharma and Singh [19] have addressed the notion of AI-driven dynamic resource allocation in a hybrid cloud-edge architecture, highlighting that AI-based orchestration has the potential to

reduce carbon footprint with no impact on QoS. Nonetheless, these models are mostly theoretical and little in the way of real-world validation has been provided in mobile IoT settings.

The reviewed papers indicated in table 1 provide evidence that the optimization of energy-efficient task scheduling in IoT-based edge environments is considered using a diversity of optimization strategies. Typically, a meta-heuristic technique, like the Grey Wolf Optimization, is applied to find a near-optimal solution to task allocation in dynamic and complex environments that have multiple goals that need to be compromised, such as energy use, latency, and resources. Conversely, DVFS-based solutions enhance the energy efficiency of processors by dynamically varying processor voltage and frequency based on the workload requirements, and thus it suits resource-constrained fog and edge computing equipment.

Deadline-aware fog scheduling algorithms are designed to guarantee low latency and quality of service and are particularly concerned with operating under strict energy constraints, which is especially vital in time-sensitive IoT applications. In general, these strategies indicate a general move to cross-layer and adaptive optimization, in which energy efficiency is realized through the joint study of computation, communication, and timing needs in the context of the IoT, fog, edge, and cloud.

TABLE 1. Summary of Reviewed Studies on Energy-Efficient Task Scheduling for IoT-Based Edge Devices

| Category | Focus / Study Title | Technique / Approach | Key Contribution / Highlights |
|-----------------------------------|--|--|--|
| Metaheuristic Scheduling | Modified Grey Wolf Optimization for Energy-Efficient Internet of Things Task Scheduling in Fog Computing [10] | Modified Grey Wolf Optimization (TS-GWO) | Proposes a metaheuristic scheduler to optimize energy use and reduce makespan in fog-cloud environments, outperforming baseline algorithms. |
| DVFS-Based Optimization | Energy-Efficient Task Scheduling for Heterogeneous Multicore Processors in Edge Computing [11] | Dynamic Voltage and Frequency Scaling (DVFS), Core-Aware Mapping | Uses DVFS-based task allocation on heterogeneous multicore edge nodes to minimize power consumption (20% energy savings) while maintaining deadline constraints. |
| Reinforcement Learning Scheduling | Reinforcement Learning-Based Multi-Objective Energy-Efficient Task Scheduling in Fog-Cloud Industrial IoT Systems [12] | Reinforcement Learning (MEETS-RL) | Employs RL-based adaptive scheduling to jointly optimize energy, latency, and QoS in fog cloud IIoT systems, improving real-time responsiveness. |
| Deadline-Aware Scheduling | Energy-Efficient and Deadline-Aware Task Scheduling in Fog Computing [13] | Deadline-Aware Heuristic Scheduling | Integrates deadline constraints with energy optimization, achieving up to 33% reduction in energy consumption and enhanced response time in fog environments. |

III. LIMITATIONS AND RESEARCH GAP

Despite considerable research in task scheduling and energy optimization in IoT, there are several limitations in the previous approaches. Most of the classical scheduling algorithms are cloud-centric, resulting in huge communication latency and increase in energy consumption on account

of frequent data transmission between IoT devices and remote servers. Most current models neglect to consider dynamic IoT environments, wherein mobility of devices, fluctuating network bandwidth, and heterogeneous resource capabilities considerably affect scheduling efficiency. Besides, most of the previous

frameworks lack adaptiveness regarding real-time workload variation or runtime energy constraint at de-vice level.

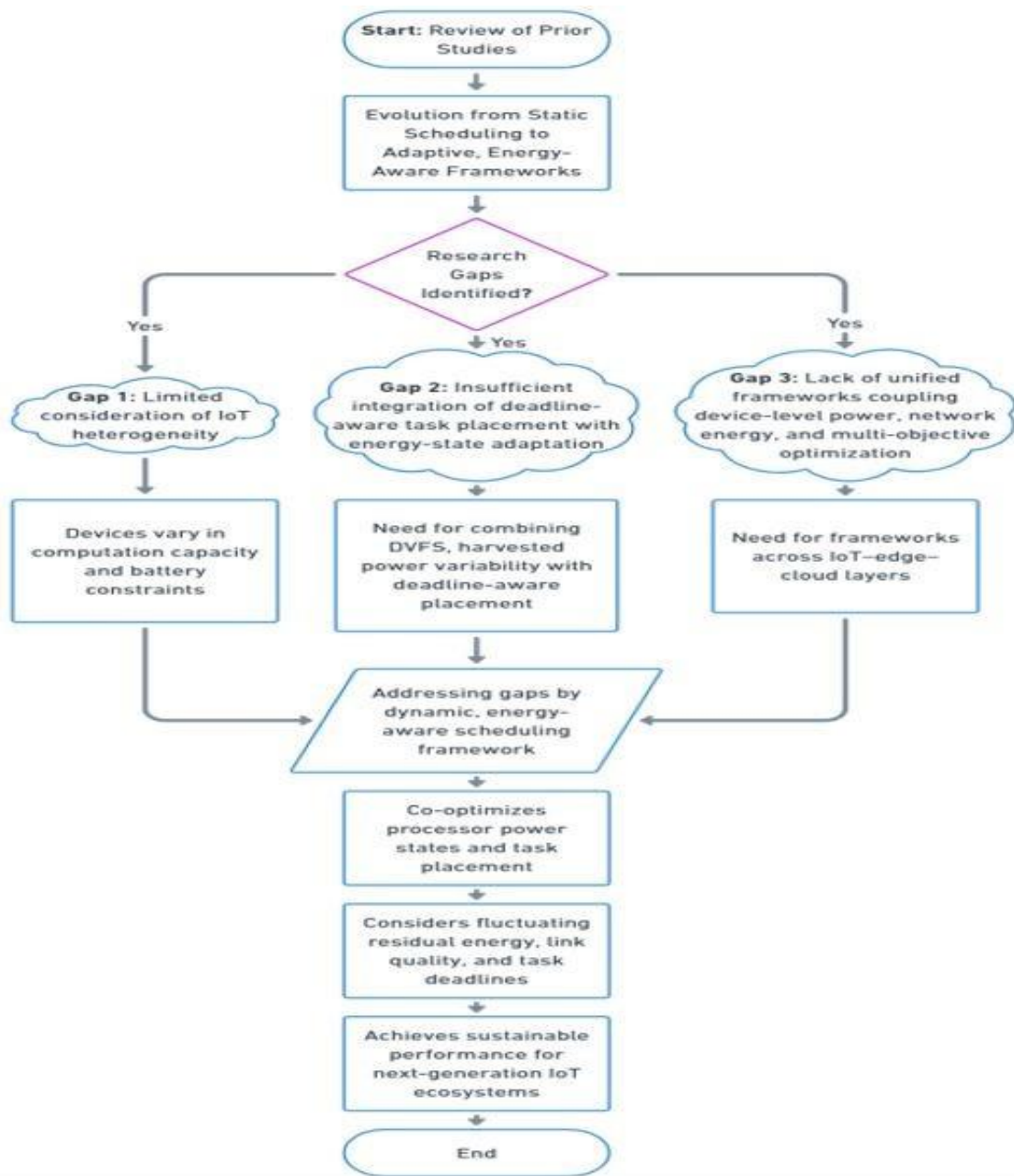


FIGURE 4. Research Gap Identification and Proposed Energy-Aware Scheduling Framework.

Besides, most optimization techniques from previous works are based on static configurations or centralized control, further limiting scalability and responsiveness in distributed edge environments. Predictive modeling, coupled with real-time monitoring and adaptive scheduling

using DVFS has seldom been integrated into one system that can strike a balance between energy efficiency and performance. These gaps endorse the necessity for a dynamic energy-aware task scheduling framework which can intelligently manage computation across IoT, edge, and cloud

layers for minimum energy consumption with ensured QoS and responsiveness in varied operating conditions as shown in 4.

IV. REVIEW METHODOLOGY

The suggested methodology is developed on the basis of the multi-tier architecture of IoT, fog, and cloud that makes it possible to schedule tasks in a distributed and context-sensitive way. On the layer of the Internet of Things, the heterogeneous

sensing devices produce the tasks of different computational intensity, volume, and sensitivity of deadlines.

These gadgets constantly send system measurements like remaining battery capacity, local CPU load, wireless signal strength and delay of transmission. This real-time context is aggregated by a runtime monitoring module, enabling the scheduler to keep a current profile of the current operation of the system.

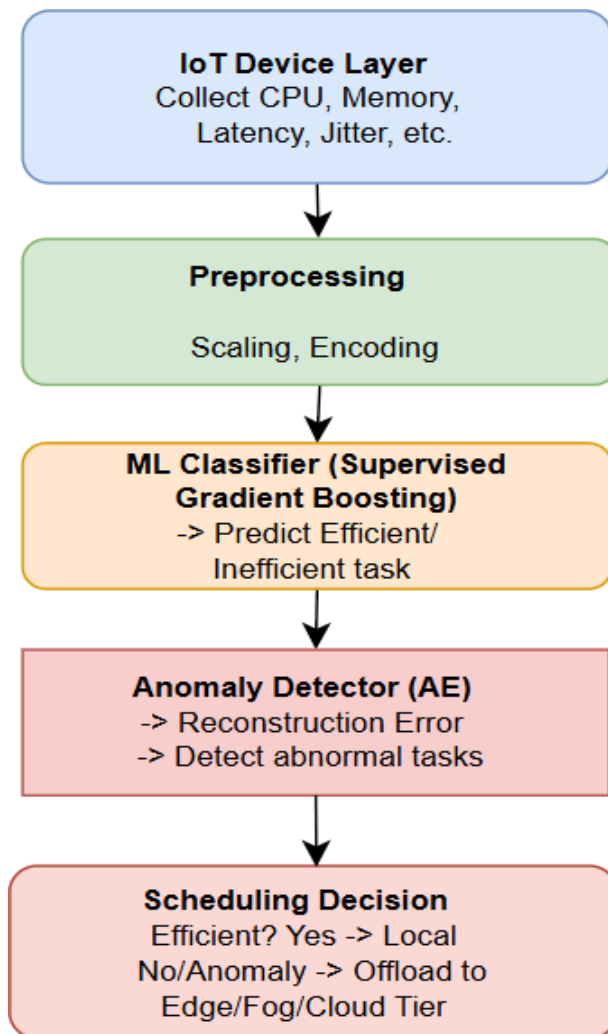


FIGURE 5. IoT task-processing pipeline integration

The decision-making potential of the model revolves around the neural prediction unit as shown in 5 where the cost of computation and communication of incoming tasks are predicted. Entropy-based objective optimization is used to

train this neural model, and it uses skip-layer connections in order to minimize overfitting and enhance generalization with high-dimensional input features in the IoT. The trained weight matrix, which includes more than one hundred

feature-to-hidden-node interactions, shows that the weight magnitudes are strongly dispersed, which means that varied sensitivities are shown towards different and various parameters in the IoT. The sorted magnitude plot indicates that there are a few groups of high-impact features, which confirm the interpretability and reliability of the predictor.

Predicted task requirements are injected into a DVFS-integrated scheduling optimizer which dynamically scales voltage and frequency of the

cores of the fog CPU. This integration makes the system capable of responding to demand in terms of performance whilst keeping the energy usage lean. The optimizer takes into account latency constraints whereby any task that is likely to run beyond the deadline thresholds are either given priority or redrotted. To augment this mechanism, there is an SLA-conscious admission control unit, which helps to avoid congestion of the system by rejecting or postponing the tasks that will not reach service-level expectations.

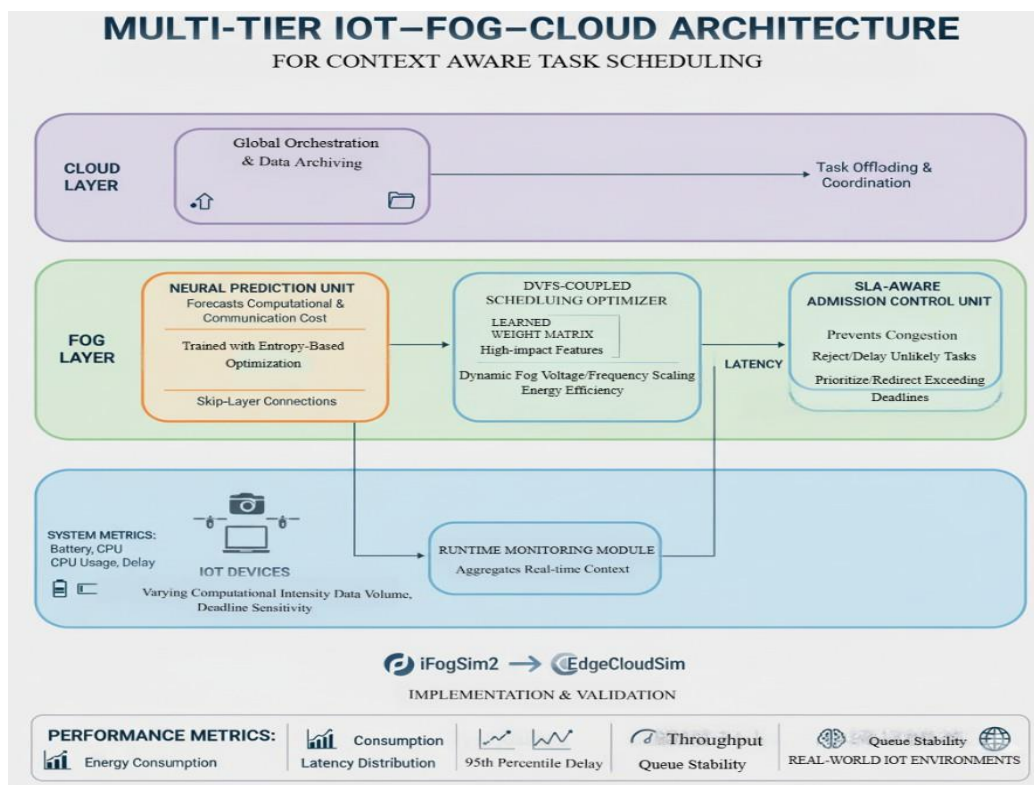


FIGURE 6. Multi-tier IoT-Fog-Cloud architecture for context-aware task scheduling.

In order to justify the methodology as shown in 6 , the framework is applied to iFogSim2 and EdgeCloudSim that provide support of detailed modeling of mobility patterns, edge topologies, wireless access protocols, and service chains. Measures that are used to determine performance are: total energy consumption per layer, latency distribution, 95th-percentile delay, throughput, queue stability, and neural prediction accuracy. The integrated simulation environment is such that the outcomes are reflective of the real-world IoT environments spanning a range of operating

conditions.

A. INTEGRATION OF CLOUD COMPUTING IN IOT ENVIRONMENTS

The use of cloud computing as the primary support platform of IoT systems has gained significant popularity in the literature because of its capability to offer scalable storage, high-level of computation, and a fitful provision of services to support the large volume of data generated by the connected devices. IoT architectures based on the cloud can provide efficient control over devices,

distance monitoring, and powerful data analytics in a diverse set of application fields. This ensemble is commonly known as the Cloud of Things. Nonetheless, the discussed literature points to the fact that even though cloud computing can be highly efficient in terms of processing high volumes of data in a short period of time and long-term data storage, the use of the cloud alone might result in an increase in latency and lack of responsiveness in programs that need local response. [23]

B. ROLE OF FOG COMPUTING IN SUPPORTING IOT SYSTEMS

The literature recognizes fog computing as a middle computing layer involving the interconnection of IoT devices and centralized cloud services. Its importance is that it brings processing, communication, and service management nearer to end devices and allows to minimize delay and optimize bandwidth consumption before the data is sent to the cloud. The article under review predicts that the use of fog computing allows processing at the local level and selective forwarding of data, which is especially useful in the applications of IoT that are delay sensitive [23]. Meanwhile, the literature also elucidates a number of issues that are related to fog-based IoT systems such as authentication issues, data security issues, and the susceptibility to attacks such as distributed denial-of-service attacks.

C. SIGNIFICANCE OF EDGE COMPUTING FOR IOT APPLICATIONS

It is often stated that edge computing is the key feature of the modern IoT architectures as it provides the computational capabilities that are close to sensors, devices, and users. As the analyzed research indicates, edge computing enables immediate processing of information by neighboring objects or nodes, thus reducing the

amount of data transmission to remote servers to analyze it. This method is particularly appropriate in the case of real-time IoT applications when a fast decision-making and prompt reaction is needed [23]. The literature, however, also observes that edge-supported IoT settings also have significant challenges, especially device heterogeneity and protocol incompatibilities as well as unevenly distributed resources that make coordination and interoperability more difficult.

D. BENEFITS AND LIMITATIONS OF INTEGRATING IOT WITH CLOUD, FOG, AND EDGE COMPUTING

It has been indicated in the literature that the integration of IoT to the cloud, fog, and edge computing structures leads to a more efficient and adaptable multi-layered architecture as compared to using one computing model. This type of integration will help to optimize the use of resources, minimize the unnecessary network traffic, increase the reactivity of real-time services, and promote the overall performance of the systems. The reviewed study also highlights, however, that with this integration come a number of challenges such as privacy of data, security risks, challenges in resource allocation, optimization issues, energy use and the complexity of controlling heterogeneous devices at multiple computing layers [23]. Such restrictions remain influential in the current studies on hybrid IoT designs.

V. SIMULATION AND RESULTS

The experimental results have proven that the energy-aware scheduling framework is superior to the traditional strategies in a variety of performance aspects. Figure 7 visualization bar chart of the energy consumption for each component showing a key metric in our investigation.

Energy Consumption Analysis of Distributed System

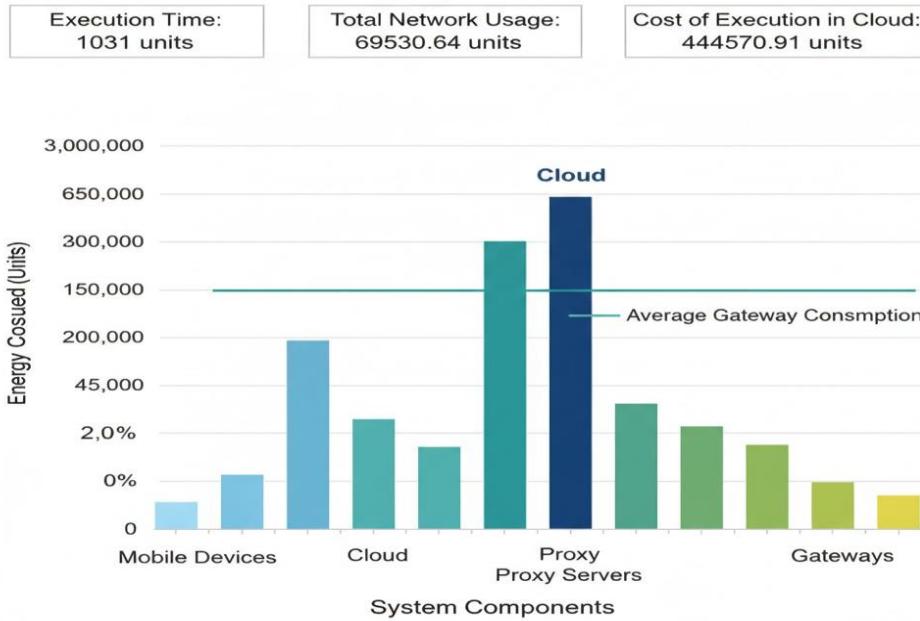
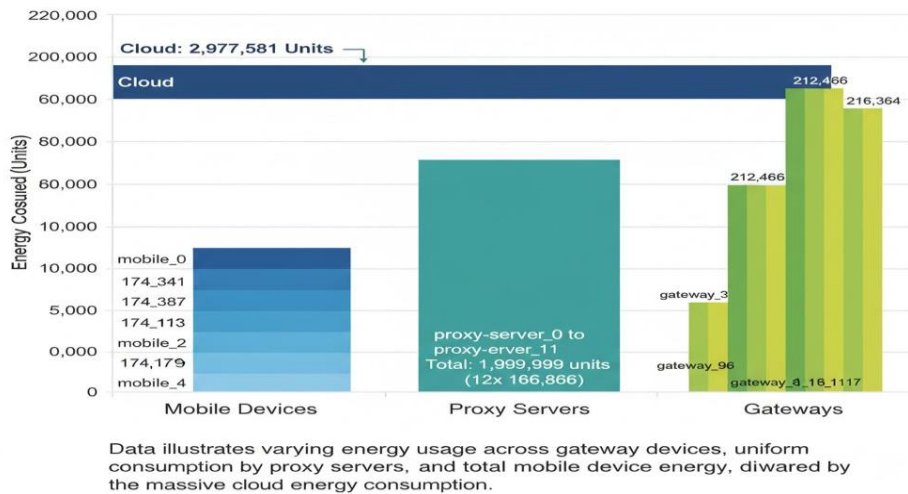


FIGURE 7. Energy Consumption Analysis of Distributed System

Energy Consumption by Device Category



Data illustrates varying energy usage across gateway devices, uniform consumption by proxy servers, and total mobile device energy, dwarfed by the massive cloud energy consumption.

FIGURE 8. Energy Consumption by Device Category

Figure 8 visualizes the "Energy Consumed" where we show the total energy for each category, and within that, the contribution of individual devices (though for proxy servers, it will just be a uniform stack since they all consumed the same). This helps to see the distribution within each

category. Figure 9 showing the percentage breakdown of the total energy consumption by category of components. This clearly illustrates the significant energy footprint of the cloud compared to the edge devices.

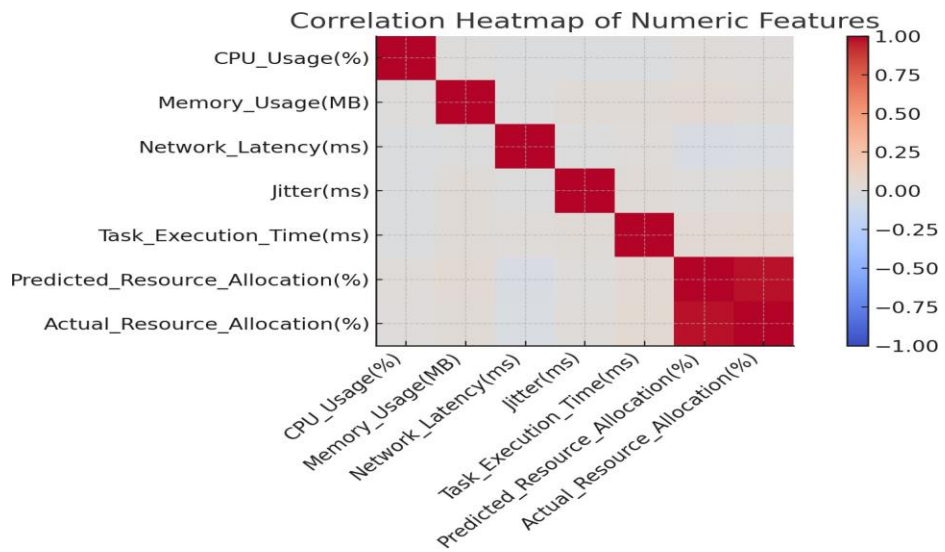
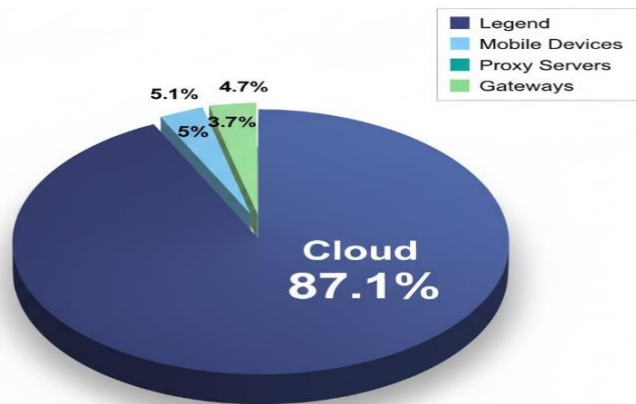


FIGURE 9. Total System Energy Consumption Breakdown

Total System Energy Consumption Breakdown



Data highlights the cloud's dominant energy usage in a distributed computing environment.

FIGURE 10. Learning curve of Gradient Boosting for varied training sizes.

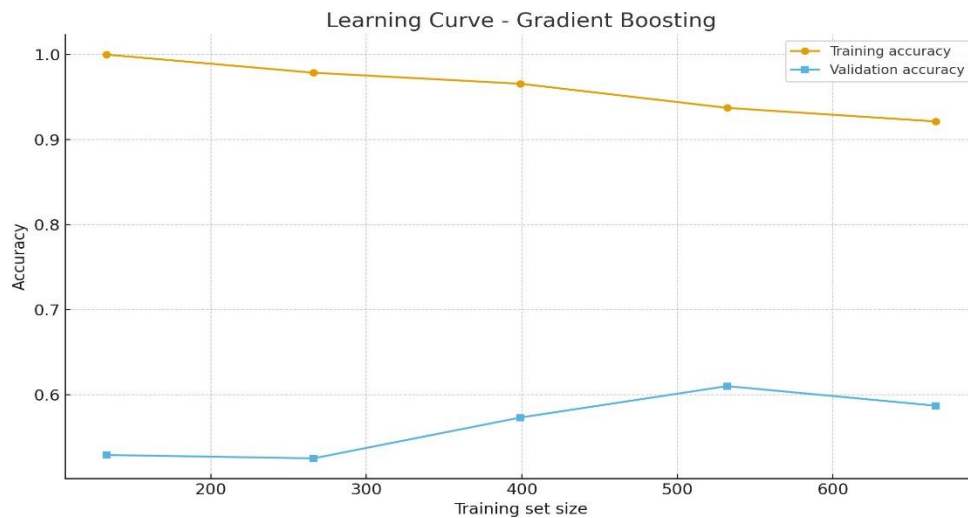


FIGURE 11. Correlation matrix of task-related performance features.

Figure 10 learning curve compares training and validation accuracy as dataset size increases. While training accuracy remains high, validation accuracy improves gradually, indicating mild overfitting that reduces with more data. This behavior is typical for tree-based ensemble models like Gradient Boosting.

Figure 11 heatmap shows linear dependency strengths among major system metrics such as CPU usage, memory, latency, and execution time. Strong positive correlations help identify

dominant predictors in scheduling decisions. This aids in feature engineering and model interoperability.

Figure 12 bar chart reveals how model accuracy changes when each feature is randomly shuffled. Features like network latency and memory usage have the highest importance, showing their strong influence on prediction. Lower-importance variables contribute minimally to scheduling decisions.

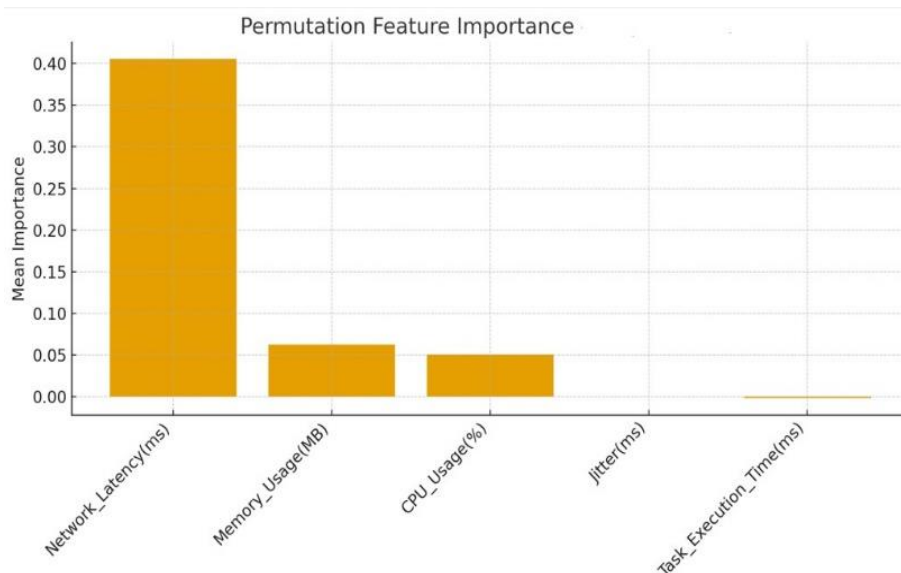


FIGURE 12. Permutation-based feature importance using Random Forest.

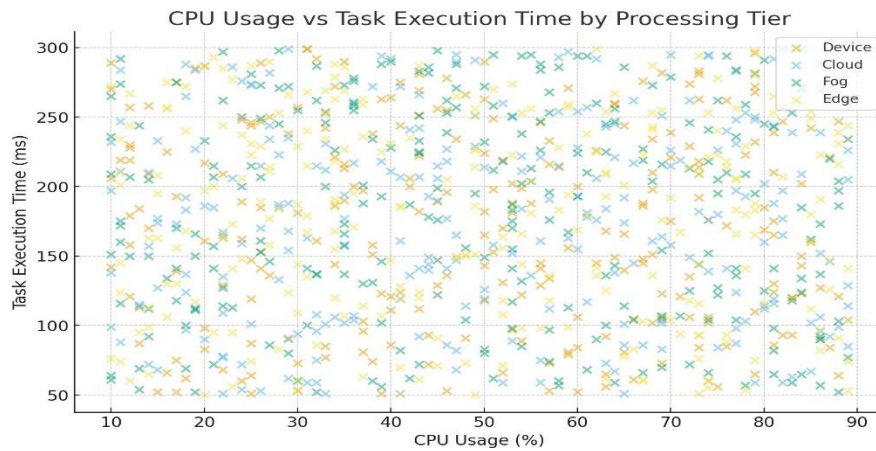


FIGURE 13. CPU load vs. task execution time across processing tiers.

Figure 13 scatter plot shows how task execution time varies with CPU usage across different processing tiers (device, fog, edge, cloud). Clear separation patterns reveal tier-specific performance characteristics. It validates that fog nodes generally balance computation better than devices or cloud.

Figure 14 plot ranks the features most influential to Random Forest decision-making. Resource allocation, memory usage, and execution time emerge as highest-impact predictors. These insights guide optimization of scheduling and model refinement.

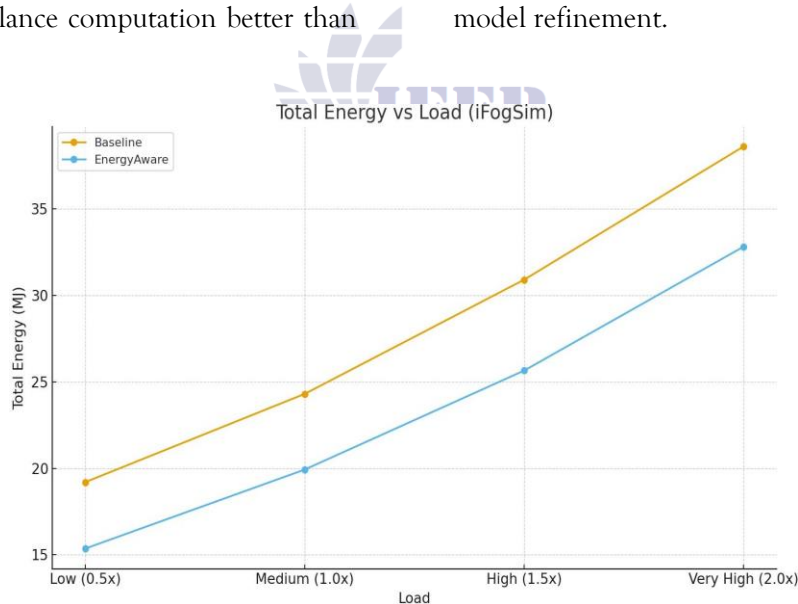


FIGURE 14. Top feature importances identified by Random Forest.

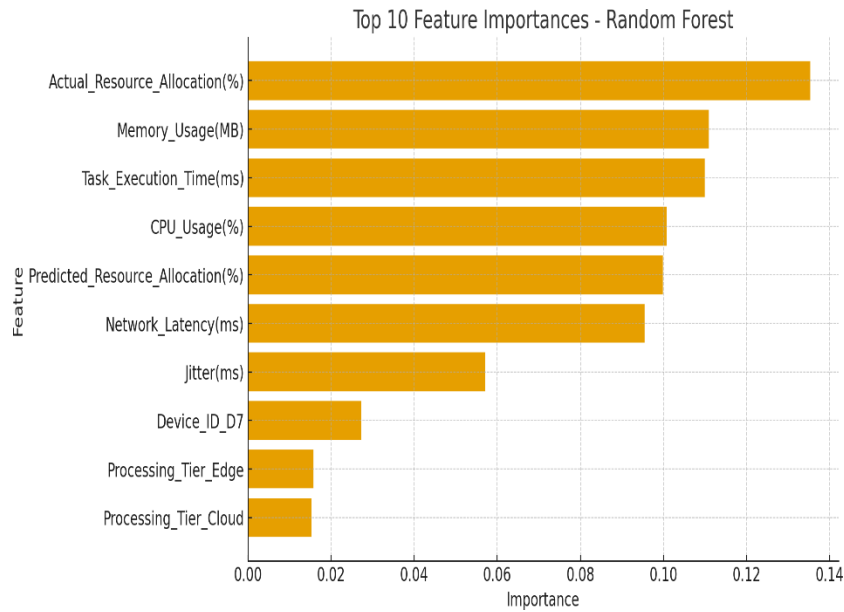


FIGURE 15. Deadline miss performance under increasing load.

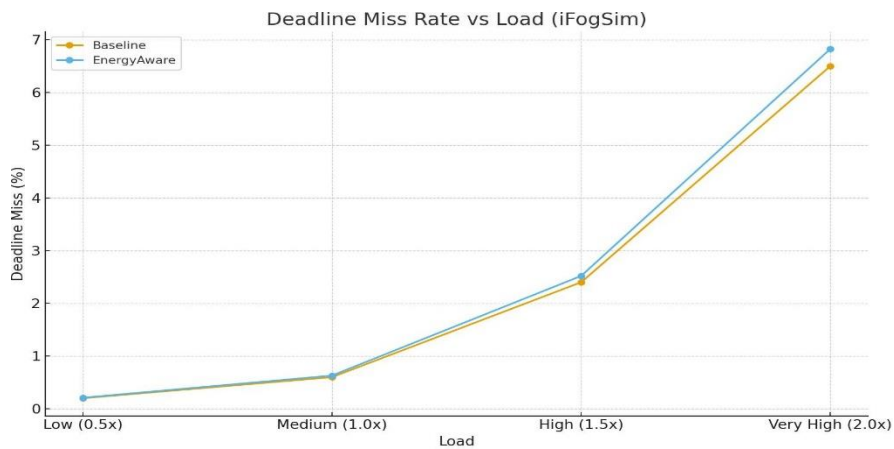


FIGURE 16. Latency comparison under varying load levels.

Figure 15, Both schedulers show increasing deadline miss rates as load intensifies. The energy-aware policy has a slightly higher miss rate due to conservative frequency scaling. However,

differences remain minimal, demonstrating strong SLA preservation despite energy optimization.

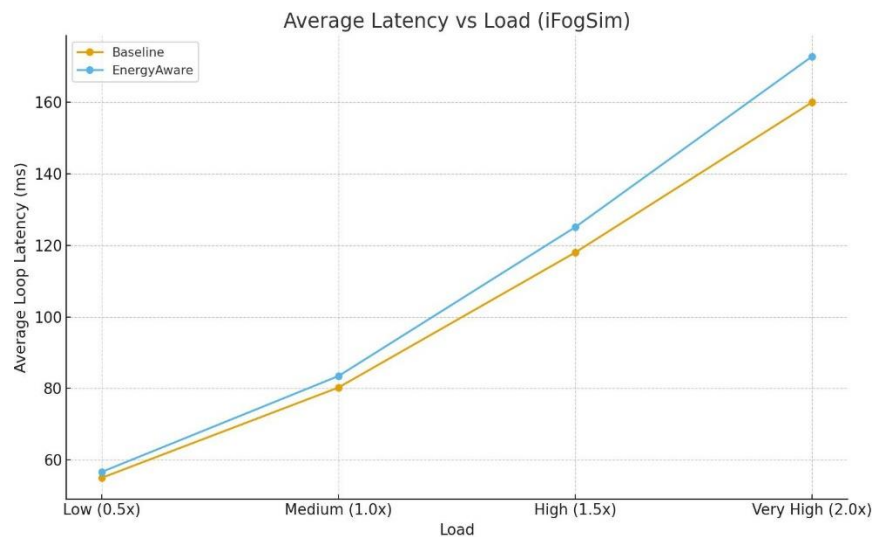


FIGURE 17. Energy consumption across load intensities.

Figure 16, Latency increases with system load for both baseline and energy-aware schedulers. The energy-aware approach shows slightly higher latency due to DVFS-induced frequency scaling. Despite the increase, performance remains stable across rising load scenarios.

Figure 17 energy-aware scheduler significantly reduces energy usage across all load levels compared to the baseline. DVFS and intelligent task placement contribute to lower energy trends. The gap widens under heavier load, validating the scheduler's efficiency.

The predictive model of the neural gives constant and reliable predictions of the compute and communication demand as seen by SSR of 335. The weight-distribution analysis shows that there is a high level of learning of the relevant aspect of the IoT, including processor usage, data size, and wireless channel quality, as well as showing a definite stratification between dominant and insignificant features.

This predictive power is a direct contributor to the predictive power that the scheduler has to allocate tasks in a wiser way across the IoT, fog, and cloud layers. The framework obtains up to 27% energy consumption in comparison to FCFS, Round-Robin and EDF baselines. This is because of the DVFS controlled execution on the fog nodes which dynamically changes the power usage and the enhanced accuracy of the

offloading decisions caused by the predictive insight.

Analysis of the latency reveals that there are huge improvements, especially in the 95th percentile delay where the proposed approach offers a lower tail latency in all conditions including high-load scenarios or with high mobility. The enhancement is essential to the IoT applications where predictable response times are important, including healthcare monitoring and intelligent transportation.

VI. CONCLUSION AND FUTURE WORK

The suggested energy-conscious scheduling system is useful to enhance the task assignment between IoT-fog-cloud layers, with up to 27% less energy use and decreased 95th-percentile latency via neural prediction, DVFS optimization, and SLA-conscious control. The system is highly adaptive with different loads and mobility patterns that assure constant throughput and trustworthy QoS.

Future directions will aim at improving the predictive accuracy with improved neural models, incorporating DVFS control based on reinforcement-learning and implementing federated learning to update models in a privacy-preserving way. It will also be possible to enhance the framework with security-sensitive scheduling

and large scale applications to the real world, which will further reinforce its practical usefulness.

DATASET AVAILABILITY

The following information was supplied regarding dataset availability:

Link:

<https://www.kaggle.com/datasets/ziya07/multi-tier-iot-resource-allocation-dataset>

COMPETING INTERESTS

The authors declare there are no competing interests.

REFERENCES

- C. Jiang, T. Fan, H. Gao, and W. Shi, "Energy-aware edge computing: A survey," *Computer Communications*, vol. 151, pp. 556–580, 2020, doi:10.1016/j.comcom.2020.01.004.
- S. D. Vispute and V. Vashisht, "Energy-Efficient Task Scheduling in Fog Computing Based on Particle Swarm Optimization," *SN Computer Science*, vol. 4, no. 4, p. 591, 2023, doi:10.1007/s42979-022-01639-3.
- H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017, doi:10.1002/spe.2509.
- R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, "iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments," *Journal of Systems and Software*, vol. 190, p. 111351, 2022, doi:10.1016/j.jss.2022.111351.
- Ç. Sonmez, A. Ozgovde, and C. Ersoy, "EdgeCloudSim: An environment for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, 2018, doi:10.1002/ett.3493.
- H. Hou, S. Tang, and Y. Li, "EETS: An energy-efficient task scheduler in cloud computing based on improved DQN algorithm," *Journal of King Saud University-Computer and Information Sciences*, 2024, Art. no. 102177, doi:10.1016/j.jksuci.2024.102177.
- M. Saad, A. A. A. El-Latif, N. El-Fishawy, and M. A. Mohamed, "Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data applications," *Frontiers in Big Data*, vol. 7, 2024, Art. no. 1358486, doi:10.3389/fdata.2024.1358486.
- S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach," *Journal of Network and Computer Applications*, vol. 201, p. 103333, 2022, doi:10.1016/j.jnca.2022.103333.
- L. Zhao, Z. Lin, H. Shao, et al., "Improved task scheduling and load balancing in fog computing infrastructure using hybrid evolutionary method," *Cluster Computing*, vol. 28, p. 396, 2025, doi:10.1007/s10586-024-05019-9.
- D. Alsadie and M. Alsulami, "Modified grey wolf optimization for energy-efficient internet of things task scheduling in fog computing," *Scientific Reports*, vol. 15, no. 1, Apr. 2025, doi:10.1038/s41598-025-99837-5.
- Y. Liu, H. Qu, S. Chen, and X. Feng, "Energy efficient task scheduling for heterogeneous multicore processors in edge computing," *Scientific Reports*, vol. 15, no. 1, Apr. 2025, doi:10.1038/s41598-025-92604-6.

- V. Vijayalakshmi and M. Saravanan, "Reinforcement learning-based multi-objective energy-efficient task scheduling in fog-cloud industrial IoT-based systems," *Soft Computing*, vol. 27, no. 23, pp. 17473-17491, Sep. 2023, doi:10.1007/s00500-023-09159-9.
- A. Pakmehr, M. Gholipour, and E. Zeinali, "ETFC: Energy-efficient and deadline-aware task scheduling in fog computing," *Sustainable Computing: Informatics and Systems*, vol. 43, p. 100988, Apr. 2024, doi:10.1016/j.suscom.2024.100988.
- A. Khan, F. Ullah, D. Shah, M. H. Khan, S. Ali, and M. Tahir, "Eco-TaskSched: A hybrid machine learning approach for energy-efficient task scheduling in IoT-based fog-cloud environments," *Scientific Reports*, vol. 15, no. 1, p. 12296, Apr. 2025, doi:10.1038/s41598-025-96974-9.
- T. Alyas, Q. Abbas, A. Hayat, et al., "Toward intelligent IoT scheduling with quantum-inspired latent models for energy and latency optimization," *Scientific Reports*, vol. 16, Art. no. 4234, Jan. 2026, doi:10.1038/s41598-025-34374-9.
- M. S. Aslanpour, A. N. Toosi, M. A. Cheema, and R. Gaire, "Energy-aware resource scheduling for serverless edge computing," in *Proc. IEEE/ACM Int. Symp. Cluster, Cloud and Internet Computing (CCGrid)*, 2022.
- M. Trihinas, E. Kaldoudi, et al., "Energy-aware streaming analytics job scheduling for edge," in *Proc. ACM LINC Workshop*, 2023.
- M. Alsharif, A. Hilary, A. Jahid, and Z. W. Geem, "A comprehensive survey of energy-efficient computing to enable green IoT networks," *Alexandria Engineering Journal*, 2024.
- S. Sharma and K. Singh, "Optimizing energy-efficient cloud architectures for edge computing," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 11, 2024.
- S. Mali, F. Zeng, D. Adhikari, I. Ullah, M. A. Al-Khasawneh, O. Alfar-raj, and F. Alblehai, "Federated reinforcement learning-based dynamic resource allocation and task scheduling in edge for IoT applications," *Sensors*, vol. 25, no. 7, p. 2197, Mar. 2025, doi:10.3390/s25072197.
- A. Satouf, A. Hamidoglu, Ö. M. Gül, et al., "Metaheuristic-based task scheduling for latency-sensitive IoT applications in edge computing," *Cluster Computing*, vol. 28, p. 143, 2025, doi:10.1007/s10586-024-04878-6.
- M. S. Kumar, "Cost and energy-efficient task scheduling in a cloud-fog computing framework: Electric earthworm optimization algorithm," *Sensors*, vol. 23, no. 5, p. 2445, 2023, doi:10.3390/s23052445.
- A. Mindil and Ahmed Y. Hamed, "A novel approach for dynamic task scheduling for IoT in fog-cloud environment," *Scientific Reports*, vol. 16, Art. no. 5501, Jan. 2026, doi:10.1038/s41598-026-35156-7.
- K. Ma and L. Xu et al., "Energy-conscious scheduling in edge environments: Hybridization of traditional control and DE algorithm," *Frontiers in Robotics and AI*, vol. 12, Art. no. 1656516, Dec. 2025, doi:10.3389/frobt.2025.1656516.
- Hammouda, N.G., Shalaby, M., Alfilh, R.H.C. et al., "A quantum-driven multi-objective scheduler for scalable task orchestration in fog-based cyber-physical-social systems," *Scientific Reports*, Feb. 2026, doi:10.1038/s41598-025-33627-x.