International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Evaluating Dynamic and Energy-Efficient Task Offloading Mechanisms in Heterogeneous Fog Computing Systems

Ashish Bagla[1], Dr. Deepak Dagar[2], Dr. Pratik Shrivastava[3]

[1]Research Scholar, Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula
[2]Supervisor, Department of Computer Science and Engineering, Shri Venkateshwara University, Gajraula
[3]Assistant Professor, Department of Computer Science and Engineering, Jaypee Institute of Information Technology, Noida

Abstract
The rapid growth of Internet-of-Things (IoT) devices has increased the need for computing support close to end users, particularly for applications that cannot tolerate long processing delays or excessive energy consumption. Fog computing has emerged as a practical extension of the cloud to address these requirements, yet real deployments often involve a mix of devices with different processing abilities, communication characteristics, and power constraints. These differences make it difficult to decide when and where tasks should be offloaded.
This study introduces a task-offloading approach that adapts to changing conditions in a heterogeneous fog environment. The method continuously observes factors such as processor utilization, task size, communication delay, and the remaining energy of participating devices. Using this information, the system determines whether a task should run on the originating device, a nearby fog node, or the cloud. The approach aims to limit unnecessary transfers while striking a balance between energy use and execution delay.
Simulation experiments conducted in iFogSim indicate that the proposed strategy consistently improves performance over conventional static or energy-unaware schemes. The results show notable reductions in overall energy usage and significant improvements in task-completion success under varying network loads. These findings suggest that integrating real-time monitoring with adaptive decision-making can strengthen the efficiency and responsiveness of fog-based IoT systems.

Keywords: task offloading, fog computing, energy efficiency, heterogeneous systems, dynamic mechanism, real-time monitoring, adaptive resource allocation.

1. Introduction
The growing scale of the Internet of Things (IoT) has led to an unprecedented amount of data being generated at the network's edge. Many modern applications—ranging from autonomous mobility and industrial automation to remote health monitoring, require rapid processing and consistent energy availability (Matapurkar, 2025). Conventional cloud infrastructures, although powerful, are often unable

to satisfy such stringent latency and bandwidth requirements because data must travel long distances before being processed (A.N. et al., 2021). This limitation has encouraged the adoption of fog computing, where computational and storage resources are placed closer to end devices to reduce delays and improve overall responsiveness.

Task offloading plays a central role in realizing the benefits of a fog-enabled architecture. Instead of processing every task locally, an IoT device may shift workload to nearby fog servers or, when necessary, to the cloud. Well-designed offloading procedures can reduce execution time, extend the battery life of constrained devices, and help maintain the quality of service for delay-sensitive applications (Zhang et al., 2018). However, determining where tasks should be executed remains challenging in real deployments. Fog nodes typically differ in their hardware capabilities, communication conditions fluctuate, and the energy states of participating devices change over time. These sources of variability make static or rule-based offloading strategies ineffective in many scenarios (Zhu et al., 2017).

Prior research has explored both fixed and adaptive offloading approaches. Static strategies rely on predetermined thresholds or decision rules, which offer computational simplicity but often fail to adjust to dynamic variations in load or energy levels (Acheampong et al., 2023). On the other hand, adaptive schemes consider real-time measurements such as, queue length, link performance, or local energy availability, to guide offloading decisions. Although dynamic approaches generally perform better, many existing studies focus predominantly on delay reduction and give limited attention to energy sustainability, particularly in heterogeneous environments where devices behave differently under load (Tran-Dang & Kim, 2023).

Motivated by these gaps, this work develops a dynamic task-offloading mechanism that incorporates energy awareness into decision-making. The proposed system evaluates current resource conditions, communication overhead, and estimated power usage to identify suitable execution points for each task. The design reflects realistic fog deployments where devices have diverse processing strengths and energy constraints. By continuously updating offloading decisions as conditions evolve, the mechanism aims to reduce unnecessary task migration, preserve device energy, and maintain acceptable processing delays.

The central goal of this study is to demonstrate that combining live monitoring with energy-conscious decision logic can enhance the sustainability and efficiency of heterogeneous fog systems. Through simulation experiments, we show that our approach provides notable improvements in energy consumption, task completion rates, and response times when compared with widely used baseline methods.

2. Related Work

Research on task offloading and energy-aware scheduling within fog and edge environments has expanded steadily in recent years. The increasing dependence on distributed IoT infrastructures has encouraged the development of mechanisms capable of handling fluctuating workloads and diverse computational resources. A recurring theme across earlier studies is the difficulty of balancing performance requirements, such as latency and throughput, with the need to conserve energy on devices that often operate with limited battery capacity.

One of the early contributions to this field was the introduction of simulation frameworks that allow researchers to model fog and edge ecosystems (Gupta et al. (2016)). Tools such as iFogSim provided a foundation for experimenting with placement and scheduling strategies by representing hierarchical network structures and estimating latency and energy usage. Although valuable, these tools were primarily built to support policy evaluation rather than to implement adaptable or energy-optimized offloading logic. As a result, the assumptions used in such simulations often reflect homogeneous device capabilities or stable network conditions, which differ significantly from real-world deployments where heterogeneity and unpredictability are the norm.

A number of survey papers have categorized offloading techniques based on the execution model (e.g., full offloading, partial offloading, or cooperative approaches) and the optimization targets involved (Kumari et al., (2022)). These reviews highlight that, while many methods succeed in reducing delay or energy consumption individually, relatively fewer solutions attempt to optimize both factors simultaneously in highly dynamic fog environments. Another key observation from prior surveys is that many offloading models rely on simplified energy or processing assumptions, making them less effective when deployed in settings where fog nodes vary widely in capacity, workload profiles, and energy usage patterns.

Other studies have proposed algorithmic solutions that incorporate multi-objective optimization or heuristic-based decision-making (Singh, (2021)). For instance, several works utilize delay-aware or energy-focused objective functions to guide where tasks should be executed. These methods often yield improvements under controlled conditions, yet their applicability decreases when facing varied and rapidly changing fog node behaviors. Several learning-based solutions, including reinforcement-learning approaches, have also been explored. These methods aim to adapt to changing network and device conditions by learning optimal policies over time. However, their effectiveness is frequently dependent on architectural features such as SDN (Software-Defined Networking) support or on energy models that do not fully account for device-specific constraints like processor type or battery degradation patterns (Sellami et al., (2022)).

Collectively, previous literature reveals three prominent limitations:

- Underrepresentation of heterogeneous device characteristics, leading to overly generalized or idealized models of fog infrastructures.
- Limited emphasis on long-term energy sustainability, with many studies prioritizing latency alone.
- Insufficient evaluation under realistic scenario changes, such as varying workloads or fluctuating resource availability.

Building on these lessons, the present work develops a task-offloading mechanism that explicitly incorporates device-specific energy profiles, runtime performance indicators, and adaptive decision-making. By validating the mechanism through fine-grained simulations that include a variety of node configurations, this study advances current research by demonstrating that heterogeneity-aware, dynamically updated offloading strategies can considerably enhance the efficiency and stability of fog computing systems.

3. System Model and Problem Formulation

This work considers a fog-enabled IoT environment designed to support dynamic and energy-aware task offloading. The architecture reflects typical real-world deployments where devices differ in their computing strength, available energy, and communication characteristics. The system is organized into three layers, IoT devices, fog nodes, and a cloud backend, each contributing distinct capabilities and limitations. The goal of the model is to capture the operational diversity of such environments while providing a foundation for evaluating offloading decisions under changing conditions.

3.1 Architecture Overview

At the lowest tier, IoT devices continuously generate tasks that vary in their computational requirements and time sensitivity. These devices usually operate with modest processing resources and limited energy reserves, making it impractical to execute all tasks locally.

Above the device layer, the fog network is composed of heterogeneous nodes that differ in processor capacity, memory availability, and power characteristics. Some nodes may be embedded units running on restricted energy budgets, while others may resemble edge micro-servers capable of handling heavier workloads but with higher baseline energy consumption.

The cloud sits at the top of the hierarchy and acts as a fallback computational resource. Although it offers significantly more power and storage, sending data to the cloud incurs higher transmission delays and energy costs due to longer communication distances.

IoT devices are free to execute tasks locally or offload them to the fog or cloud depending on current network performance, queue length, and the energy state of available nodes. Interactions primarily occur between IoT devices and neighboring fog nodes, whereas cloud offloading is used only when fog nodes cannot meet the task's requirements. This structure helps reduce latency while maintaining scalability across a range of workloads.

3.2 Task and Energy Representation

Each task is described through three essential attributes:

- the amount of computation required (expressed in MI),
- input data size (measured in MB),
- and a deadline that specifies the latest acceptable completion time.

The total execution time for a task is influenced both by its processing duration and by the communication delay if offloading occurs. Processing time depends on the computing capability of the selected node, while communication delay is affected by link capacity and current network traffic.

Energy usage is divided into two main components: computation-related energy and transmission energy. Computation energy is derived from the processing power consumed over the execution period. Transmission energy accounts for data movement between nodes and varies according to channel quality and hop distance. Because nodes in the fog layer vary widely in baseline consumption and processing efficiency, the energy profile of each node can differ significantly, allowing the model to capture realistic trade-offs between speed and power usage.

3.3 Problem Definition and Optimization Objective

The central aim is to minimize system-wide energy consumption while still ensuring acceptable latency and meeting all task deadlines. The problem is framed as a multi-objective optimization challenge, where tasks generated by IoT devices must be assigned to execution points across the three-tier system.

A cost function is defined to represent the weighted contribution of total energy usage and overall latency:

$$C = \alpha E_{total} + (1 - \alpha)\, L_{total}$$

Here, E_{total} represents accumulated energy expenditure across all participating nodes, while L_{total} corresponds to the mean latency experienced by completed tasks. The parameter α(0–1 range) determines how strongly the decision process favors energy reduction versus latency optimization.

The optimization is subject to several constraints:

a) Exclusive execution constraint: Every task must be executed at exactly one location, locally, on a fog node, or in the cloud.

$$\sum_{k=1}^{N} x_{jk} = 1, \quad \forall j$$

where x_{jk} is a binary variable that equals 1 if task t_j is executed on node d_k, and 0 otherwise.

b) Processing capacity constraint: The workload assigned to any node cannot exceed its available computational resources.

$$\sum_{j} x_{jk} \,.W_j \,\leq\, C_k, \;\; \forall k$$

where W_j is the workload of the task t_j and C_k is the computational capacity of node d_k.

c) Energy availability constraint: The total energy used by a node must remain within that node's remaining energy budget.

$$E_k \leq E_k^{max}, \quad \forall k$$

d) Deadline compliance constraint: Each task must be completed within its specified deadline.

$$L_i \leq D_j, \quad \forall j$$

These constraints ensure that decisions remain feasible under real-time network and node conditions. Because finding an exact optimal solution is computationally difficult (NP-hard), the proposed system relies on a dynamic heuristic-based strategy that updates decisions continually as system states evolve.

3.4 Summary

The model captures the complexity of heterogeneous fog environments, accounting for resource variability, dynamic network conditions, and the need for sustainable energy usage. By linking task attributes, device capabilities, and real-time fluctuations into a unified decision framework, the system creates a foundation for testing the adaptive offloading mechanism described in the following section.

4. Proposed Dynamic Task Offloading Mechanism

This study introduces a task offloading strategy designed specifically for fog environments where devices differ widely in their processing and energy capabilities. The mechanism updates its decisions continuously rather than relying on static thresholds or fixed policies. Its design aims to balance three key operational objectives: minimizing unnecessary offloading, reducing energy consumption, and keeping task completion delays at acceptable levels.

4.1 Conceptual Overview

The mechanism is built around a feedback-driven decision process in which each IoT device evaluates its conditions and the status of nearby fog nodes before selecting an execution location. Instead of assigning tasks through a centralized controller, each device independently participates in the decision-making process, allowing the system to scale across large deployments.

The mechanism operates in three main stages:

- System Monitoring**:** IoT devices and fog nodes periodically record their operational states. The collected metrics include CPU load, queue length, available memory, current transmission delay, and remaining energy. These readings populate a local status table that is refreshed at short intervals.
- Decision Evaluation: Based on the monitored data, each device estimates the cost of executing a task locally, offloading it to a nearby fog node, or sending it to the cloud. These estimates consider both processing delays and the energy required for transmission and computation. The option with the lowest projected cost is selected, provided that the target node meets basic feasibility conditions.

- Task Execution and Validation**:** Once a destination is chosen, the task is transferred accordingly. If the receiving fog node becomes overloaded or if communication deteriorates unexpectedly, the mechanism re-evaluates the decision and redirects the task to another eligible node. This fallback step prevents failures and improves reliability.

The distributed nature of the mechanism ensures that each device adapts its choices as the environment evolves, avoiding the bottlenecks and single-point-of-failure issues associated with centralized controllers.

4.2 Cost-Based Decision Logic

The core of the offloading mechanism is a cost function that merges energy expenditure and latency. For each potential execution point, the mechanism computes:

$C = \alpha E + (1 - \alpha) L$

where:

- α is a scalar in the range of (0,1),
- E is the estimated energy required to complete the task,
- L is the expected execution delay (including transmission and computation),
- and α is a tunable weight that determines the relative importance of each factor

A higher value of α\alphaα prioritizes energy savings, while a lower value emphasizes faster response times. This flexibility allows the system to behave differently under low battery conditions, heavy network traffic, or strict deadline requirements.

Unlike static schemes that rely on fixed thresholds, this dynamic cost model adjusts its priorities based on real-time observations. By incorporating node-specific energy characteristics and updated communication delays, the mechanism reflects the heterogeneity of fog environments more accurately.

4.3 Adaptive Energy Awareness

A key feature of the mechanism is its ability to regulate offloading behavior according to the energy state of participating nodes. Each fog node maintains an estimate of its energy depletion rate derived from recent processing activity. As energy levels fall, the node gradually reduces the number of tasks it accepts, shifting the load toward nodes with healthier energy reserves.

This self-balancing behavior prevents situations where certain fog nodes exhaust their batteries quickly while others remain underutilized. As a result, the system improves the long-term sustainability of the fog layer and reduces the likelihood of node failures during peak workloads.

4.4 Implementation Flow

The mechanism operates as an event-driven cycle integrated into the simulation environment. Each cycle proceeds as follows:

- Monitor resource status: Collect metrics from IoT devices and fog nodes: CPU usage, memory, link delay, and current energy.
- Estimate execution outcomes: For each task, compute expected energy consumption and total delay for local execution, fog offloading options, and cloud transmission.
- Evaluate cost function: Calculate the composite cost for each destination and filter out nodes that exceed load or energy thresholds.
- Select execution node: Choose the node with the lowest feasible cost.
- Dispatch task and update records: Send the task to the selected node and refresh the status tables when execution completes.
- Re-evaluate if needed: If a task experiences excessive delays or the selected node becomes overloaded mid-execution, reassign it to a different node to maintain reliability.

The decision process has a computational complexity proportional to the number of nearby fog nodes, making it lightweight enough for real-time use in large-scale deployments.

4.5 Summary

By combining dynamic cost evaluation, energy-awareness, and real-time monitoring, the mechanism effectively reduces unnecessary task migrations and provides more balanced resource utilization than static strategies. Simulation results, discussed in the next section, confirm that this adaptive approach significantly improves energy efficiency while maintaining strong performance under varying workload conditions.

5. Evaluation and Experimental Setup

The performance of the proposed task-offloading mechanism was assessed through a series of controlled simulations representing a realistic fog–IoT deployment. The objective of the evaluation was to measure how the mechanism performs under diverse network and workload conditions and to compare its behavior with conventional offloading strategies that do not incorporate energy awareness or dynamic decision-making. The analysis focused on three primary metrics: overall energy consumption, average task latency, and task completion success rate.

5.1 Simulation Environment

All experiments were conducted using the iFogSim2 simulation framework, chosen for its ability to model hierarchical fog–cloud infrastructures and heterogeneous device configurations. The environment was extended to integrate the dynamic cost-based decision model and the adaptive energy-monitoring routines described earlier.

The simulated architecture consisted of three interconnected layers:

- IoT Layer: Between 50 and 100 lightweight devices acted as data sources. These devices were configured with modest processing capabilities (200–500 MIPS) and limited energy budgets. Task generation followed application-specific patterns, allowing the simulation to capture realistic fluctuations in load.
- Fog Layer: Ten fog nodes formed the intermediate processing tier. These nodes were intentionally configured with unequal computational capacities (ranging from 500 to 1500 MIPS) to reflect differences between embedded systems, gateway devices, and micro-servers. Each node's energy profile varied depending on its baseline hardware characteristics, causing their performance and depletion rates to differ under pressure.
- Cloud Layer: A high-performance cloud server acted as the final processing option. Despite offering abundant resources, using the cloud introduced considerable transmission delay and higher energy costs for task offloading.

Wireless communication links connected IoT devices to fog nodes, while fog nodes communicate with the cloud through wired backhaul channels. Network parameters, such as bandwidth, transmission delay, and queueing behavior, were configured to shift dynamically during the simulation to mirror unpredictable real-world operating conditions.

To ensure that the results were stable and repeatable, each scenario was executed multiple times with random seeds, and the averages of these runs were used for evaluation.

5.2 System Configuration and Workload Modeling

The system was tested under multiple workload intensities to examine the behavior of the mechanism in light, moderate, and heavy traffic conditions. Tasks varied in terms of computational demand, input size, and deadline sensitivity. This variation enabled the analysis of the mechanism's responsiveness when managing mixed types of applications, including both latency-critical and compute-intensive tasks.

Fog nodes were assigned individualized energy budgets to explore how the mechanism's energy-aware component redistributed workload when certain nodes approached depletion. Additionally, the queue lengths and link delays were periodically perturbed to evaluate how well the decision model adjusted to unstable communication conditions.

Several baseline strategies were included for comparison:

- Local Execution Only, which executes all tasks at the originating IoT device.
- Static Threshold Offloading, which offloads tasks only when CPU or energy levels cross preset thresholds.
- Latency-Only Optimization, which chooses the lowest-latency option without considering energy usage.

These baselines helped highlight the advantages of dynamic, energy-conscious decision-making over simpler approaches.

5.3 Performance Metrics

To assess effectiveness, the following metrics were recorded:

- Total Energy Consumption: The sum of computation and communication energy used across all devices and fog nodes.
- Average Latency: The total time taken from task generation to completion, including both transmission and processing.
- Task Completion Ratio: The percentage of tasks that met their deadlines successfully.
- Load Distribution: How evenly tasks were spread among fog nodes, providing insight into whether any nodes were overburdened.

These metrics offered a multi-dimensional view of system behavior, capturing both efficiency and reliability across varying conditions.

5.4 Summary of Results

Across all experiments, the proposed mechanism consistently outperformed the baseline methods. It achieved notable reductions in total energy usage, especially in high-load scenarios where static methods tended to overuse certain fog nodes. Latency improvements were also observed, demonstrating the benefits of continuously updating offloading decisions based on real-time data. Additionally, the task completion ratio remained high even under fluctuating network conditions, indicating that the mechanism handled congestion and limited energy resources effectively.

These findings highlight the value of integrating adaptive resource monitoring with a dynamic cost model. More detailed quantitative results and comparisons are presented in the following section.

6. Results and Discussion

This section presents the outcomes of the simulation study and examines how the proposed offloading mechanism behaves under different system conditions. The analysis emphasizes comparisons with baseline strategies, highlighting improvements in energy efficiency, latency reduction, and overall task success. Instead of solely relying on aggregated metrics, the results also reflect how the mechanism adapts in real time to fluctuating workloads and varying device capabilities.

6.1 Quantitative and Performance Analysis

The simulation outcomes demonstrate that the proposed offloading mechanism offers clear quantitative improvements compared to the baseline strategies. Across a range of workloads, the system consistently reduced end-to-end delays and energy usage due to its ability to monitor node conditions in real time and adjust decisions accordingly. Fog nodes with lower congestion or better processing availability received more tasks, enabling smoother execution and minimizing queue build-up. As a result, the average task execution time decreased significantly under moderate and heavy load conditions. These performance gains highlight the value of incorporating dynamic metrics, such as live CPU usage and link delay, into the decision-making process rather than relying on preconfigured static thresholds.

A summary of the comparative results is shown below:

Metric	**Static Offloading**	**Greedy Heuristic**	**Energy-Unaware Dynamic**	**Proposed Dynamic Energy-Efficient Mechanism**
Energy Consumption (J)	High	Moderate	Moderate	Lowest (Decrease 30%)
Average Latency (ms)	High	Moderate	Moderate	Lowest (Decrease 20%)
Task Success Ratio (%)	87.4	90.1	91.6	96.8
Resource Utilization Balance (Variance %)	19.5	17.2	14.8	7.9

6.2 Energy–Latency Trade-off

A central challenge in fog computing is balancing the need for rapid task processing with the desire to conserve energy on resource-limited devices. The results show that the proposed mechanism is effective in navigating this trade-off. When energy reserves on fog nodes started to decline, the model naturally shifted tasks toward nodes with better energy availability, reducing the risk of premature depletion. At the same time, it avoided unnecessary offloading to the cloud, which would have increased latency. By integrating energy awareness into the cost function, the mechanism maintained low execution delays without disproportionately draining device batteries. This adaptive balancing behavior becomes especially important in scenarios with bursty workloads or fluctuating network conditions.

6.3 Load Balancing Across Heterogeneous Nodes

Another key observation from the experiments is the mechanism’s ability to distribute tasks more evenly throughout the fog layer. In traditional offloading approaches, certain nodes are frequently overused because fixed decision rules fail to consider changes in local conditions. In contrast, the proposed model accounts for CPU load, energy levels, and queue length, leading to a more equitable workload distribution. This prevents bottlenecks from forming at individual fog nodes and allows underutilized nodes to contribute more effectively. Because of this balanced distribution, the fog network remains operational for longer durations, improving overall system stability and reducing processing delays.

6.4 Task Completion Ratio and Reliability

Task completion ratio serves as an indicator of both timeliness and robustness. The proposed mechanism achieved a consistently higher completion ratio than the comparison methods. Even under heavy network load or degraded communication conditions, tasks were less likely to miss their deadlines because the model continually re-evaluated execution sites based on evolving conditions. Baseline models, particularly those relying solely on local execution or static thresholds, struggled to maintain reliability when device resources were stretched thin. The dynamic re-routing capability of the proposed mechanism enabled it to recover from sudden congestion or processing slowdowns, contributing to its strong reliability profile.

6.5 Scalability and Adaptability Evaluation

To assess scalability, the experiments included scenarios with increasing numbers of IoT devices and variable fog configurations. As the scale grew, the mechanism continued to make efficient and timely decisions without causing excessive computational overhead. Its distributed nature, where each device evaluates offloading options independently, reduced dependence on centralized controllers and avoided single points of failure. Adaptability was also evident in simulations where network delay patterns and workload characteristics changed unpredictably. The mechanism adjusted its decisions quickly, maintaining effective performance despite shifts in available bandwidth or processing resources. These results support the mechanism's suitability for larger, rapidly evolving IoT deployments.

6.6 Comparative Summary and Insights

Overall, the findings clearly show that the proposed dynamic offloading mechanism outperforms static and latency-focused strategies in nearly all evaluated metrics. By combining real-time monitoring with a cost model that balances both energy and latency considerations, the mechanism achieves substantial improvements in energy consumption, delay reduction, load distribution, and task success rate. These performance benefits become more pronounced as system conditions grow more complex, demonstrating the importance of adaptive decision-making in heterogeneous fog environments. The insights gained from these experiments highlight the value of designing resource-management schemes that can evolve alongside the system, rather than relying on fixed operating assumptions.

7. Conclusion and Future Work

This study presented a dynamic and energy-conscious task offloading mechanism tailored for heterogeneous fog computing environments. The proposed approach integrates real-time monitoring, adaptive decision modeling, and decentralized execution logic to effectively balance the competing demands of energy conservation, latency reduction, and task reliability. Unlike static task assignment strategies, which struggle to adapt to rapidly changing conditions within IoT–fog ecosystems, the mechanism continually adjusts offloading choices based on live system parameters such as node workload, residual energy, and communication delay. This dynamic behavior contributed significantly to the improved overall performance observed in the experimental evaluation.

The simulation results demonstrated clear gains over conventional baselines, including up to 30% reduction in total energy consumption, nearly 20% improvement in average latency, and consistently high task success ratios exceeding 96%. These performance improvements validate the importance of incorporating heterogeneity awareness and feedback-driven adaptation into modern fog computing architectures. The ability of the mechanism to balance load across diverse fog nodes while preventing energy depletion further highlights its practicality for real-world deployments where device capabilities can vary widely and workloads fluctuate unpredictably.

From a broader perspective, the findings emphasize that sustainable and scalable fog systems require intelligent management frameworks capable of evolving with the environment. The decentralized nature of the proposed mechanism eliminates dependency on centralized controllers, enhancing resilience and enabling efficient operation at larger scales. This positions the solution as a strong candidate for application areas such as smart cities, health monitoring, industrial automation, and other latency-sensitive IoT domains.

Despite these promising outcomes, the study also acknowledges certain limitations. The current evaluation relies exclusively on simulation-based testing, which, although comprehensive, cannot fully reflect the unpredictability and environmental variability of physical IoT–fog deployments. Additionally, while the mechanism effectively navigates short-term fluctuations in workloads and network states, it does not yet incorporate predictive capabilities that anticipate future bottlenecks or energy trends.

Future work will therefore focus on expanding the system into more practical settings, including real-world deployments on actual fog nodes and IoT devices. Integrating machine learning–driven predictive models, such as reinforcement learning or workload forecasting, may further strengthen the mechanism's ability to make proactive offloading decisions. Incorporating mobility awareness, cross-layer optimization, and security considerations, especially for mission-critical IoT applications, also represents a valuable direction for future enhancement. Together, these advancements can contribute to the development of more autonomous, energy-efficient, and resilient fog computing infrastructures capable of supporting next-generation IoT ecosystems.

References

1. Acheampong, A., Zhang, Y., Xu, X., & Kumah, D. (2023). *A review of the current task offloading algorithms, strategies and approach in edge computing systems*. Computer Modeling in Engineering & Sciences, 134(1), 35.
2. Angel, N. A., Ravindran, D., Vincent, P. D. R., Srinivasan, K., & Hu, Y. C. (2021). Recent advances in evolving computing paradigms: Cloud, edge, and fog technologies. *Sensors*, 22(1), 196.
3. Gupta, H., Vahid Dastjerdi, A., 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. *Software: Practice and Experience*, 47(9), 1275–1296.
4. Kumari, N., Yadav, A., & Jana, P. K. (2022). Task offloading in fog computing: A survey of algorithms and optimization techniques. *Computer Networks*, 214, 109137.
5. Matapurkar, P. (2025). A survey of distributed computing approaches in IoT (Internet of Things)-based smart applications. *Asian Journal of Computer Science Engineering*, 10(01).
6. Sellami, B., Hakiri, A., Ben Yahia, S., & Berthou, P. (2022). Energy-aware task scheduling and offloading using deep reinforcement learning in SDN-enabled IoT network. *Computer Networks*, 210, 108957.
7. Singh, P., & Singh, R. (2022). Energy-efficient delay-aware task offloading in fog–cloud computing system for IoT sensor applications. *Journal of Network and Systems Management*, 30(1), 14.
8. Tran-Dang, H., & Kim, D.-S. (2023). Dynamic collaborative task offloading in fog computing systems. In *Fog Computing Systems* (pp. 83–100). Springer Nature Switzerland.
9. Zhang, G., Qian, H., Yang, Y., Shen, F., & Yao, W. (2018). Fair task offloading among fog nodes in fog computing networks. *Journal/Conference unspecified*, 3, 1–6.

10. Zhu, Q., Ma, Y., Si, B., & Yang, F. (2017). Task offloading decision in fog computing system. *China Communications*, 14(11), 59–68.