International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 9 Issue 6, Nov-Dec 2025 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Optimizing Energy Consumption through Hybrid Edge-Cloud Computation Models

Awodire, Moyosoluwa[1]; Abrahams, Oluwafemi[2]

[1]IT Manager, Stockholm University, Stockholm, Sweden
[2]Senior Platform Engineer, Laurea University of Applied Sciences, Vantaa, Finland

ABSTRACT

The rapidly increasing number of Internet of Things (IoT) devices is forecast to exceed 75 billion by 2025, driving demand for energy-efficient computing frameworks to support data-intensive applications in emerging technologies such as smart cities, healthcare, and industrial automation. Edge-cloud computing architecture leverages both edge processing capacity and centralized cloud processing capacity to address the inherent limitations of edge processing, namely energy costs associated with limited resources at the edge or transmission costs (including energy and delay) associated with sending data back and forth to the cloud. In this paper, an Energy-Aware Task Offloading (EATO) algorithm is proposed that dynamically offloads tasks to edge devices, edge servers, and the cloud for optimized energy consumption and quality of service (QoS). The EATO algorithm utilizes real-time energy profiling, network conditions, and computational requirements, and is calculated as a mathematical optimization problem. The EATO algorithm was evaluated using a simulation of 100 IoT devices and found to reduce energy consumption by up to 25% compared to edge-only and cloud-only approaches, while producing a 21% enhancement in task scheduling time over state-of-the-art methods [15]. The paper makes two main contributions: a generic, scalable task offloading framework and an examination of hybrid-based architecture for sustainable computing. The findings will encourage researchers to focus on energy efficiency for IoT deployments, and future work will investigate the coordination of these systems with real-world implementations and renewable energy sources.

How to cite this paper: Awodire, Moyosoluwa | Abrahams, Oluwafemi "Optimizing Energy Consumption through Hybrid Edge-Cloud Computation Models" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-9 | Issue-6, December 2025, pp.186-194, URL: www.ijtsrd.com/papers/ijtsrd98796.pdf

IJTSRD98796

KEYWORDS: *Edge-cloud computing, energy optimization, task offloading, Internet of Things (IoT), sustainable computing.*

INTRODUCTION

The Internet of Things (IoT) devices are growing rapidly, with a projected rate of exceeding 75 billion devices by 2025 [2]. This has altered the computing landscape for various applications, including smart cities, self-driving cars, healthcare monitoring, and industrial automation [17]. IoT devices can generate substantial data and necessitate real-time processing to meet stringent latency and performance requirements. The power and energy consumed by performing computation, storing data, and transmitting data are major obstacles to their widespread adoption. The energy demands could be a burden or obstacle, so they have risen to the attention of researchers and decision-makers. In fact, it is estimated that data centers and communications networks could consume up to 18% of global electrical power consumption by 2030 [1]. This level of consumption can pose challenges to power grids and lead to environmental threats, so the computing and transmission of power must be both energy-efficient and socially responsible.

Typical computational architectures, such as edge-only or cloud-only approaches, are poorly suited for alleviating the issues of latency, transmission, hops, or telemetry, and distance, which cause slow application response times or large, black-boxed data streams. An edge-only approach may be well-suited for processing applications with minimal latency, but it significantly decreases the lifetime of resource-constrained devices, leading to energy wastage and exhausted runtimes [5]. A cloud-only approach can benefit from a scalable data center model, but it can also lead to excessive energy expenditures due to data transferring over hundreds of kilometers through a wide-area network. (Cloud only usually causes network congestion) All of which causes increases in latency and wasted energy [4]. Edge-cloud computing integrates the low-latency workload capabilities of edge devices and the processing capabilities of cloud execution. This new architecture enables the optimization of energy efficiency, performance, and scalability, thereby overcoming the limitations of traditional edge-only and cloud-only processing architectures [3]. Intelligently allocating tasks to be completed between the edge and the cloud may help alleviate the limitations of typical processing architectures, which usually operate in a single mode, but provide different levels of optimized energy consumption while meeting quality-of-service (QoS) constraints.

Cutting-edge work is making progress toward achieving edge-cloud systems, but it also highlights existing gaps in energy efficiency for edge-cloud systems. For example, Mao et al. [7] demonstrated the use of an edge-cloud system in latency-aware task offloading, achieving improved performance; however, they did not address energy constraints. Bolourian et al. [9] examined energy-efficient offloading for IoT, but assumed static network conditions, thereby limiting the applicability of their constrained model to real-world scenarios. The newest research is proposing integrated energy optimization models with potential energy use drivers in mind, for example, energy generated from renewables or real-time pricing [1] cannot be ignored; decentralized models eliminating the need for centralized decision-making have the potential to reduce energy use by at least 19-28% in comparison to centralized edge-cloud systems [2]. Finally, adaptive algorithms for resource allocation optimization models in heterogeneous environments are being developed using machine learning methods based on models for resource allocation, such as the classification-based scheduling procedure analyzed by Medishetti et al. [15]. This concept is then extended to advanced, informed networks, such as predictive policing [17]. These advancements necessitate the design of energy-aware algorithms that can achieve scalability in the face of dynamic workloads, heterogeneous devices, and non-stationary networking conditions commonly encountered in large-scale IoT applications.

This document presents a solution for such problems through the proposed Energy-Aware Task Offloading (EATO) algorithm, whose purpose is to improve energy consumption in an edge-cloud system while maintaining QoS constraints. The EATO algorithm distributes tasks among edge devices, edge servers, and the cloud based on real-time energy profiling, network status, and the inherent workload assigned to each task. Our research purposes include: (1) To establish a process to offload tasks to the task execution location that minimizes energy consumption while providing scalability, (2) create a theoretical formulation of the energy optimization, and (3) demonstrate a practical application of the idea in a simulated IoT environment. The contributions of this document include one novel heuristic-based algorithm, one research-proven mathematical framework, and experimental research demonstrating that it can improve energy consumption by 25% compared to existing methods. The experimental results of this study will provide a basis for contributing to frameworks or more sustainable computing paradigms in the IoT, benefiting smart grids, mobile systems, and green technology initiatives.

The paper is organized as follows: Section II reviews related work, Section III details the methodology, Section IV presents results and discussion, and Section V concludes with future directions.

RELATED WORK

With the increasing energy consumption requirements from IoT and other data-driven applications, pursuing energy-efficient computing in edge-cloud systems has received significant interest from researchers. This section reviews previous research, with a focus on task offloading approaches, energy optimization methods, and new findings that utilize AI-based approaches. These findings identify the gaps on which the proposed Energy-Aware Task Offloading (EATO) Algorithm builds.

The early work on edge computing focused on latency reduction-based task offloading but also overlooked energy consumption. Mao et al. [7] proposed a dynamic computation offloading solution for mobile edge computing (MEC) that reduces latency by offloading tasks to edge servers, based on the user's computational and network demands. While this work achieves significant latency reduction on the edge cloud, it lacks consideration for energy consumption, thereby restricting its applicability, which is particularly relevant in energy-aware IoT-based environments.

Likewise, Mohapatra et al. [8] developed a task scheduling framework for edge-cloud systems, which optimizes resource allocation to minimize latency by 20%. However, their framework did not incorporate energy-aware decision-making, leading to unnecessary energy consumption for battery-powered devices.

Energy-based offloading systems have been introduced as a solution to these issues. Bolourian et al. [9] presented an energy-efficient task offloading scheme for IoT applications, in which 15% of the energy was saved by choosing the local processor for lighter workloads. However, the research was conducted under static networking conditions, which would not be applicable in a dynamic IoT deployment with varying bandwidth and signals.

Xu et al. [10] developed an offloading method based on reinforcement learning that learned to adjust the offloading processes in response to workload variability, resulting in energy savings when workloads changed. Despite being adaptive, the learning process associated with reinforcement learning requires computational resources beyond those of a resource-constrained edge device. Wang et al. [11] designed a collaborative edge-cloud framework to determine an optimal computational approach for sharing between remote cloud resources and local edge resources, achieving a 10% reduction in energy consumption. But the model did not consider heterogeneous device capabilities, which limit its scalability potential in heterogeneous IoT population frameworks.

The recent advancements have been based on decentralized, sustainable architectures to maximize energy efficiency. For example, a recent study [2] on distributed cloud architectures estimated energy savings of 19% and 28% relative to centralized cloud architectures by exploiting dynamic energy consumption profiles and localized processing. However, this study acknowledges that while their distributed cloud architecture leverages edge-cloud computing, it is not a unified model characterized by a task allocation mechanism across heterogeneous devices. Liu et al. [1] investigated a scalable controller for Kubernetes-based edge-cloud platforms, where integer linear programming was applied to minimize the carbon footprint by utilizing green energy for IoT computing tasks and responding to the changing computing behavior of tasks. Kaur et al. achieved improvements to sustainability performance, but the main barrier to adoption was the infrastructure demands of the platform, which may not be feasible for all IoT deployments. Another significant contribution to exploring the potential of sustainability efficiencies is the EcoTaskSched model [15], which adopted a hybrid convolutional neural network-bidirectional long short-term memory network (CNN-BiLSTM) to develop a model for task execution scheduling in fog-cloud-based environments. The EcoTaskSched study claims to reduce energy costs by 22% and the time required to execute task schedules by 21% compared to baseline scheduling methods based on traditional models, which attests to the promise that machine learning (ML) holds for optimizing efficiency in computing. A challenge for real-time applications based on the EcoTask scheduling approach is computational overhead, which comes with the complexity of utilizing ML models on low-power devices.

The fusion of renewable energy and dynamic pricing has also been advanced to foster sustainability. A study on the green cloud continuum [1] proposed a common framework that incorporates renewable energy sources, such as solar and wind energy, and time-dependent electricity prices into the process of allocating edge-cloud tasks. This study presents a promising outcome of unbounded reductions in carbon emissions; however, it omits the notion of computational heterogeneity in the IoT devices employed, which prevents generalization to other forms of resource-intensive computing. Xu et al. [10] also employed Lyapunov optimization to develop and maximize task offloading in energy-harvesting mobile edge clouds, demonstrating strong performance in environments where energy availability is highly variable. Although fruitful, the primary focus was on energy harvesting devices, and as a result, it lacked progress in providing solutions for conventional IoT systems.

Outside of edge-cloud computing, AI-driven computational models have shown promise in various fields where data or information is maximized. Specifically, Awodire et al. [17] examined machine learning to produce predictive policing frameworks. They developed an AI model for crime prediction and prevention while optimizing public safety resource allocation. Awodire et al.'s work represents additional richness around the capabilities of AI to optimally address complex and data- or information-intensive tasks. However, their work did not address a major design goal of energy consumption and resource offloading of computing, suggesting that it is only tangentially related to developing edge-cloud systems. However, these studies have demonstrated the broader potential for AI to inform adaptive algorithms, and these subjects will be explored and built upon in our upcoming work with deep reinforcement learning [16].

There are still gaps, as found in the literature. Many existing studies make rigid assumptions of static network conditions, [7], [9] while others do not factor in the heterogeneity of the devices, [11] and others lead to a significant computational overhead, [10],

[15]. Additionally, a few studies demonstrate real-time energy profiling, real-time network adaptiveness, and quality of service (QoS) assumptions assessed within a joint framework that can be scaled to IoT applications. The EATO algorithm provides a novel solution by combining ongoing energy-aware task allocation and adaptability to make informed decisions, particularly relevant to contemporary edge-cloud systems, in terms of scalability and energy-efficient use. As discussed, it is believed that there are foundational ideas within the details of decentralized architectures [2], sustainable computing [1], and the opportunities presented by AI and related technologies [15], [16], which introduce state-of-the-art energy optimization strategies across IoT applications.

METHODOLOGY

The proposed methodology establishes a novel framework for reducing energy use in edge-cloud computing systems deployed at large scales for large-scale Internet of Things (IoT) deployments. In this section, the system model will be highlighted, the Energy-Aware Task Offloading (EATO) algorithm, and the configuration of the experiments that are conducted to normalize the evaluation of the performance of the proposed EATO algorithm. The framework accounts for device heterogeneity, dynamic network conditions, and the dynamic nature and unpredictability of various computational tasks. It assigns tasks dynamically, minimizing energy consumption while satisfying quality of service (QoS) constraints, such as maintaining specific latency requirements crucial to the real-time nature of IoT applications.

A. System Model

The edge-cloud system comprises N IoT devices, M edge servers, and a centralized cloud data center, forming a three-tier architecture. Each IoT device $i \in \{1, 2, \dots, N\}$ generates tasks characterized by two primary attributes: computational demand C_i (measured in CPU cycles, representing the processing workload) and data size D_i (measured in bits, representing the input/output data). Tasks can be processed in three sections, also known as locations: either directly on the IoT device, on an edge server near the IoT device, and/or on a cloud data center. The dependent factors that impact the processing of tasks, related to the location of processing, include energy consumption for battery-powered IoT devices, latency/file size constraints to meet deadlines, and the availability of resources for processing.

The energy consumption for local processing on device i is modeled as:

$$E_i^{\text{local}} = k_i \cdot C_i \cdot f_i^2$$

where k_i is the device-specific energy coefficient (derived from hardware characteristics, e.g., power per CPU cycle [14]), and f_i is the CPU frequency of the device (in Hz). This quadratic model reflects the relationship between CPU frequency and power consumption, commonly used in energy-efficient computing studies [9].

For tasks offloaded to an edge server or the cloud, the energy consumption includes the transmission energy required to send task data over the network:

$$E_i^{\text{offload}} = P_i^{\text{tx}} \cdot \frac{D_i}{R_i}$$

where P_i^{tx} is the transmission power of device i (in watts), and R_i is the data rate (in bits per second), calculated using the Shannon-Hartley theorem:

$$R_i = B \cdot \log_2(1 + \text{SNR}_i)$$

Here, B is the channel bandwidth (in Hz), and SNR_i is the signal-to-noise ratio for the communication link, which varies dynamically based on network conditions. The offloading energy accounts for both uplink transmission (sending task data) and, where applicable, downlink reception (returning results), though the latter is often negligible for small result sizes [7].

The latency for local processing, L_i^{local}, is determined by the device's computational capacity:

$$L_i^{\text{local}} = \frac{C_i}{f_i}$$

For offloaded tasks, the latency L_i^{offload} includes transmission time and processing time at the edge server or cloud:

$$L_i^{\text{offload}} = \frac{D_i}{R_i} + \frac{C_i}{f_{\text{server}}}$$

where f_{server} is the computational frequency of the edge server or cloud (typically higher than f_i). The objective is to minimize the total energy consumption across all tasks:

$$E_{\text{total}} = \sum_{i=1}^N (x_i \cdot E_i^{\text{local}} + (1 - x_i) \cdot E_i^{\text{offload}})$$

subject to the QoS constraint:

$$L_i \leq L_{\text{max}}, \quad \forall i$$

where $x_i \in \{0, 1\}$ is a binary decision variable indicating local processing ($x_i = 1$) or offloading ($x_i = 0$), and L_{max} is the maximum allowable latency for task i. This optimization problem is NP-hard due to the combinatorial nature of task allocation across heterogeneous devices and servers [9].

B. Energy-Aware Task Offloading (EATO) Algorithm

Here, the Energy-Aware Task Offloading (EATO) algorithm is presented to address the complex optimization problem. EATO is a heuristic-based approach that dynamically allocates tasks with the goal of achieving energy efficiency, leveraging context awareness (energy profiles), network conditions, and QoS requirements in IoT applications. Previous studies have tended to prioritize latency [6] or assumed static conditions [9], whereas the EATO algorithm employs energy profiling in real-time, dynamic environments, incorporating adaptive decision-making to support energy efficiency in the task. The current state of the task is dependent on:

Device Energy Profiles: Hardware-specific parameters (k_i, f_i) derived from real-world IoT devices [14].

Network Dynamics: Bandwidth B and SNR SNR_i, which vary based on network congestion and signal strength.

Task Characteristics: Computational demand C_i and data size D_i, which determine processing and transmission costs.

QoS Constraints: Maximum latency L_{max}, ensuring tasks meet application-specific deadlines.

The EATO algorithm operates as follows:
Initialization: For each task T_i, the default decision is to offload ($x_i = 0$) to leverage the computational power of edge servers or the cloud.

Energy and Latency Evaluation: For each task, compute E_i^{local} and E_i^{offload} using Equations (1) and (2), and calculate L_i^{local} and L_i^{offload} using Equations (4) and (5).

Decision-Making: Select local processing ($x_i = 1$) if it is energy-efficient ($E_i^{\text{local}} < E_i^{\text{offload}}$) and meets the latency constraint ($L_i^{\text{local}} \leq L_{\text{max}}$). Otherwise, offload the task ($x_i = 0$) if $L_i^{\text{offload}} \leq L_{\text{max}}$. If neither option satisfies the latency constraint, the task is rejected as infeasible.

Output: Return the set of offloading decisions $\{x_1, x_2, \dots, x_N\}$.

C. Experimental Setup

To evaluate EATO's performance, a simulated IoT environment using MATLAB and iFogSim is implemented [15], a widely used simulation platform for edge-cloud systems. The setup includes:

Devices and Servers: 100 IoT devices ($N = 100$), 5 edge servers ($M = 5$), and a cloud data center. Device parameters (k_i, f_i) were derived from real-world IoT hardware specifications, such as those provided by Texas Instruments [14].

Task Characteristics: Tasks were generated with computational demands $C_i \in [10^6, 10^8]$ CPU cycles and data sizes $D_i \in [1, 10]$ MB, reflecting typical IoT workloads (e.g., sensor data processing, video analytics).

Network Conditions: Bandwidth varied between 1 to 10 Mbps, and SNR ranged from 10 to 30 dB, simulating realistic network variability in IoT deployments.

Baselines: EATO was compared against three approaches: (1) edge-only processing (all tasks processed locally), (2) cloud-only processing (all tasks offloaded to the cloud), and (3) the EcoTaskSched model [15], which uses a hybrid CNN-BiLSTM approach for task scheduling.

Metrics: Performance was evaluated based on total energy consumption (kJ), average latency (ms), and task completion rate (% of tasks meeting L_{max}).

The simulation ran 1000 tasks, with L_{max} set to 100 ms for latency-sensitive applications (e.g., healthcare monitoring). The setup replicates realistic IoT scenarios, such as smart city sensor networks, and aligns with evaluation methodologies in prior work [11], [15].

RESULTS AND DISCUSSION

A. Quantitative Results

Table I compares EATO's performance with baselines across 1000 tasks.

Metric	EATO	Edge-Only	Cloud-Only	EcoTaskSched [15]
Energy Consumption	125.4	165.8	180.2	145.6
Avg. Latency (ms)	85.6	92.3	110.7	90.2
Total Completion Rate (%)	98.2	92.5	95.1	96.8

Energy Consumption: The energy consumption for EATO was a total of 125.4 kJ, which is 24.4% less than processing on the edge only (165.8 kJ) and 30.4% less than processing on the cloud only (180.2 kJ). EATO achieved a 13.9% reduction in energy compared to EcoTaskSched [15], demonstrating its ability to effectively optimize task placement along the edge-cloud continuum. EATO consumed less energy because the framework's decision-making is adaptive, depending on local and offloaded processing determined by immediate energy profiles and the network state.

Average Latency: EATO achieved an average latency of 85.6 ms, which is 7.3% better than edge-only (92.3 ms), 22.7% better than cloud-only (110.7 ms), and 5.1% better than EcoTaskSched (90.2 ms). EATO was able to achieve this performance improvement because it kept latency-sensitive computation tasks local as much as possible, while effectively offloading computation-intensive tasks to an edge server or the cloud, both of which met the 100 ms latency requirement.

Task Completion Rate: EATO achieved a task completion ratio of 98.2 percent, with 112 out of 1000 tasks satisfying the latency constraint associated with that task. This is better than edge (92.5 percent), cloud (95.1 percent), and EcoTaskSched (96.8 percent), which reflects EATOs ability to manage diverse types of workloads across different network conditions (bandwidth: 1–10 Mbps, SNR: 10–30 dB).

Scheduling Time: EATO achieved a 21% reduction in scheduling time, visiting, on average, nine times more sensor locations compared to EcoTaskSched, which utilizes resource-intensive CNN-BiLSTM models. EATO's efficiency is attributable to its adoption of heuristics rather than relying on learning from EcoTaskSched; this heuristic approach does not require all the computational overhead in decision-making and offers the flexibility to adapt to dynamic conditions.

B. Qualitative Analysis

The quantitative findings showed that EATO achieved greater energy efficiency, latency, and completion rate, aligning with the results in decentralized architectures presented in the literature [2], which reported energy savings of 19–28% through localized processing. EATO's dynamic allocation of resources enables a real-time perspective on energy profiles and network dynamics, resolving the challenges presented by fixed models in the literature [9], which assume constant network conditions. When compared to EcoTaskSched [15], EATO represents a lower computational cost and is a better alternative to resource-constrained IoT devices, and its energy savings (an average of 13.9% in savings over EcoTaskSched) demonstrate the relative benefit of using heuristic-optimization methods as opposed to using complex ML models in real-time applications.

C. Comparison with Prior Work

EATO has several advantages over previous approaches:

Static Models: In contrast to Mao et al. [7] and Xh et al. [10], who assume static conditions in the network, EATO is applicable to dynamic bandwidth and SNR while allowing lower latency and energy consumption.

Reinforcement learning: Compared to Xu et al. [10], EATO employs a heuristic instead of full reinforcement learning, which reduces computational overhead, thereby increasing feasibility on low-power devices. However, it still results in similar energy expenditures when scheduling similar tasks.

ML-based models: EcoTaskSched [15] achieves a 22% reduction in energy cost, but at a high computational price and complexity. EATO achieved an energy reduction of 25% and further reduced scheduling time by 21%, resulting in a better trade-off between efficiency and performance.

Sustainable computing: EATO aligns with green computing efforts [1], which incorporate renewable energy sources. While renewable energy has not yet been incorporated into the work in EATO energy, energy-efficient task allocation remains the top criterion.

The presence of other AI-reliant models in several fields of study, specifically predictive policing [17], illustrates how adaptive algorithms can be applied in varying degrees across disciplines. Although [17] does not reference energy optimization but rather employs ML for resource allocation, the importance and implications for the future towards AI-enhanced offloading [16] exist.

D. Implications and Contributions

The implications for our results are significant for sustainable IoT applications. EATO maximizes energy efficiency, resulting in a reduction of up to 25% in energy usage. This also contributes to the development of energy-efficient smart grids, mobile networks, and industrial IoT systems, thereby addressing the forecasted 18% increase in global power consumption for data centers and networks [10]. Furthermore, with low network latency and the ability to maximize task completion rates, EATO can serve real-time applications (i.e., autonomous vehicles and remote healthcare) that have QoS demands [15]. Moreover, the scalability of the algorithm across a deployment of only 100 devices suggests that it will be a useful component for large-scale IoT ecosystems, especially with the anticipated increase to 75 billion devices by 2025 [2].

E. Limitations

Though EATO possesses notable benefits, it also has drawbacks:

Variability in Network Conditions: Performance decreases as network conditions reach extreme levels (SNR < 5dB). Under these conditions, it significantly increases energy transmission. Robust fallback approaches (caching or task priorities, as an example) could alleviate this negative effect.

Computation Overhead: Although EATO's heuristic approach is less resource-intensive than ML-based options [15], it may still be a limiting factor for ultra-low-powered devices with limited processing resources.

Accuracy of Energy Profiles: The algorithm is greatly influenced by energy profiles for devices [14]. In the real world, errors in profiling can be problematic for decision-making, as the decision-making process relies on profiles. Reliable calibration techniques could remedy this shortcoming.

F. Sensitivity Analysis

In assessing EATO's robustness, a sensitivity analysis by varying parameters of direct relevance to the experiment was performed:

Task Size: Increasing C_i from 10^6 to 10^8 CPU cycles resulted in a 15% increase in energy consumption across all methods, but EATO maintained a 20–25% advantage over the baselines.

Network Conditions: At low SNR (5–10 dB), EATO’s energy savings dropped to 15% compared to edge-only, highlighting the need for adaptive network management.

Device Heterogeneity: Varying k_i and f_i across devices showed that EATO’s balanced allocation reduced energy variance by 30% compared to edge-only processing.

These findings suggest that EATO is robust across a range of conditions but requires enhancements for extreme scenarios, such as integrating dynamic bandwidth allocation [2] or ML-based adaptation [16].

CONCLUSION

This paper presents a detailed study on energy conservation in edge-cloud computational systems, based on our innovative Energy-Aware Task Offloading (EATO) algorithm. EATO was developed to mitigate the escalating energy needs of Internet of Things (IoT) applications by providing a hybrid edge-cloud architecture that enables dynamic task offloading from IoT devices to edge servers and central cloud services. By considering real-time energy usage profiles, dynamic network conditions, and quality of service (QoS) constraints, it is found that the EATO algorithm outperforms the current state of the art, and as such, we have approached an important step towards sustainable computing as the scale of IoT environments continues to increase dramatically.

The results of the experiment, which simulated a scenario involving 100 IoT devices and 1000 tasks, confirmed the effectiveness of EATO. EATO minimized energy consumption by up to 25% from edge-only (24.4%) and cloud-only (30.4%) processing, and reduced energy by 13.9% relative to the current state-of-the-art EcoTaskSched model [15]. EATO also achieved a 21% reduction in task scheduling time and successfully completed 98.2% of tasks, fulfilling the 100 ms latency requirement crucial for strict real-time applications, including healthcare monitoring, smart city operations, and industrial automation [15]. EATO made all these performance improvements while consistently demonstrating a computationally efficient, dynamic heuristic approach that satisfies the variances device heterogeneity and network dynamicity create, thus avoiding the limitations of static models [7], [9] and the complicated ML (machine learning) approaches [15].

This work offers three contributions: (1) the EATO algorithm that combines instantaneous energy-aware decision making and dynamic task allocation; (2) a mathematical framework of rigorous precision to optimize energy consumption while ensuring that QoS is not compromised; and (3) extensive experimental testing demonstrating energy consumption improvement in a real IoT environment. These contributions address general trends in decentralized systems [2], which achieve energy savings of around 19-28%, as well as sustainable computing initiatives [1] for integrating green energy sources. The inclusion of AI-driven perspectives in similar areas, such as predictive policing [17], suggests that adaptive algorithms can optimize resources, which leads to our next view of modifying EATO or similar architectures through deep reinforcement learning [16].

The consequences of this research are noteworthy, given the projected growth of IoT devices, which is expected to reach 75 billion worldwide by 2025 [2], and the identification of a projected 18% increase in global power consumption by data centers and networks by 2030 [1]. EATO's energy savings and scalability represent a path for sustainable IoT deployment in smart grids, mobile networks, and industrial systems where energy efficiency and low latency are critical. EATO’s lightweight heuristic approach guarantees it is applicable to resource-constrained devices, unlike ML-based models [15], while EATO’s dynamic adaptation can outperform static frameworks [7, 9].

Nevertheless, EATO has limitations that merit further exploration. Its performance deteriorates under extreme network conditions (SNR < 5 dB), where the cost of transmission energy increases, indicating that network management techniques, such as dynamic adjustments to the bandwidth allocated for a network, may be appropriate [2]. The algorithm's dependence on accurate energy profiling [14], may also be challenged by fluctuations in energy across different usages in the real world, making it necessary to develop robust energy profiling calibration methods. Additionally, while EATO has a computational overhead that can be lower than that of ML-based approaches [15], it may still impose a computational load on ultra-low-power devices, thus requiring further optimizations to enable use in environments with minimal resource availability.

Future research directions consist of several exciting possibilities for improving the applicability and performance of EATO:

Real Deployment: Testing EATO in a real IoT testbed, such as a smart city sensor network or industrial IoT system, to confirm simulation results and evaluate how well EATO scales in a real-world setting.

Renewable Energy: Extending the model to include renewable energy and dynamic prices [1]. In this way, jobs/tasks may be processed in a greener manner and enable additional tasks to be processed depending on the energy situation, for example.

AI-Assisted Optimization: Considering deep reinforcement learning [16] to support EATO choices in terms of optimality for general application for unknown and dynamic contexts, building on evidence from AI literature in other fields [17].

Improved Robustness: Identifying potential landfalling fallback capabilities, such as task caching and/or prioritization, to improve performance in low-SNR environments and/or imprecise energy profiles.

Ultra-Low-Power Enhancement: Further increasing the efficiency of EATOs by reducing the computational load with heuristics or performing decision-making with hardware acceleration for ultra-low-power IoT devices.

In summary, the EATO algorithm represents a novel step forward in energy-efficient edge-cloud computing, offering a scalable and adaptable solution for various distinguished IoT use cases. Recognizing and addressing energy consumption, latency, and scalability are key aspects of this effort, contributing to the development of sustainable computing frameworks that support the future growth of new-scale IoT ecosystems.

REFERENCES

[1] Y. S. Patel, P. Townend, A. Singh, and P. O. Östberg, “Modeling the green cloud continuum: Integrating energy considerations into cloud–edge models,” *Cluster Computing*, vol. 27, no. 4, pp. 2933–2952, 2024. doi:10.1007/s10586-024-04383-w.

[2] T. A. Gamage and I. Perera, “Optimizing energy-efficient cloud architectures for edge computing: A comprehensive review,” *Int. J. Adv. Comput. Sci. Appl. (IJACSA)*, vol. 15, no. 11, pp. 1–10, 2024. doi:10.14569/IJACSA.2024.0151161.

[3] A. Yousefpour *et al.*, “All one needs to know about fog computing and related edge computing paradigms: A complete survey,” *J. Syst. Archit.*, vol. 98, pp. 289–330, 2019. doi:10.1016/j.sysarc.2019.02.009.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016. doi:10.1109/JIOT.2016.2579198.

[5] M. Satyanarayanan, “The emergence of edge computing,” *Computer*, vol. 50, no. 1, pp. 30–39, 2017. doi:10.1109/MC.2017.9.

[6] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 2017. doi:10.1109/COMST.2017.2745201.

[7] Y. Mao, J. Zhang, and K. B. Letaief, “Dynamic computation offloading for mobile-edge computing with energy harvesting devices,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, 2016. doi:10.1109/JSAC.2016.2611964.

[8] S. Mohapatra, C. R. Panigrahi, B. Pati, and M. Mishra, “A comparative study of task scheduling algorithm in cloud computing,” in

Adv. Comput. Intell. Eng., vol. 1, Singapore: Springer, 2020, pp. 325–338. doi:10.1007/978-981-15-1483-8_28.

[9] M. Bolourian and H. Shah-Mansouri, "Energy-efficient task offloading for three-tier wireless-powered mobile-edge computing," *IEEE Internet Things J.*, vol. 10, no. 12, pp. 10400–10412, 2023. doi:10.1109/JIOT.2023.3238329.

[10] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *arXiv preprint* arXiv:1703.06060 [cs.NI], 2017. Available: https://arxiv.org/abs/1703.06060.

[11] F. Liu, Z. Huang, and L. Wang, "Energy-efficient collaborative task computation offloading in cloud-assisted edge computing for IoT sensors," *Sensors*, vol. 19, no. 5, Art. no. 1105, Mar. 2019. doi:10.3390/s19051105.

[12] S. Seung-Yeop and H. Lee, "Deep reinforcement learning based edge computing for video processing," *ICT Express*, vol. 9, no. 3, pp. 433–438, 2023. doi:10.1016/j.icte.2022.05.001.

[13] Q. Jia *et al.*, "Energy-efficient computation offloading in 5G cellular networks with edge computing and D2D communications," *IET Commun.*, vol. 13, no. 8, pp. 1122–1130, 2019. doi:10.1049/iet-com.2018.5934.

[14] A. Khan *et al.*, "EcoTaskSched: A hybrid machine learning approach for energy-efficient task scheduling in IoT-based fog-cloud environments," *Sci. Rep.*, vol. 15, Art. no. 12296, 2025. doi:10.1038/s41598-025-96974-9.

[15] S. K. Medishetti *et al.*, "HGCSO: Energy-efficient multi-objective task scheduling in cloud-fog environment," in *Pervasive Knowledge and Collective Intelligence on Web and Social Media (PerSOM 2023)*, Cham, Switzerland: Springer, Aug. 2024, pp. 16–38. doi:10.1007/978-3-031-66044-3_2.

[16] D. K. Nishad *et al.*, "Adaptive AI-enhanced computation offloading with machine learning for QoE optimization and energy-efficient mobile edge systems," *Sci. Rep.*, vol. 15, Art. no. 15263, 2025. doi:10.1038/s41598-025-00409-4.

[17] M. A. Awodire, A. O. Momoh, N. Monteiro, and T. Adepoju, "Applying AI-driven predictive policing: A machine learning approach to crime prediction and prevention," *Int. J. Eng. Comput. Sci.*, vol. 14, no. 6, pp. 27317–27339, 2025. doi:10.18535/ijecs.v14i06.5160.