Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

AI-Driven Task and Data Offloading in Replicated Fog Computing Environments for Scalable Smart Infrastructure

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 smart infrastructure, including IoT-based systems, smart cities, and industrial automation, has created a significant demand for real-time data processing with minimal latency. Traditional cloud-centric approaches often struggle to meet these requirements due to network delays and bandwidth limitations. Fog computing has emerged as a promising solution by enabling computation closer to the data source; however, efficient task and data offloading in fog environments remains a challenging problem, especially in scenarios involving replicated data across distributed fog nodes. Existing approaches are often based on static or heuristic strategies and lack the adaptability required to handle dynamic workloads and heterogeneous system conditions. In this study, an AI-driven task and data offloading framework is proposed for replicated fog computing environments, with a focus on improving latency, energy efficiency, and scalability. The proposed approach is based on a Q-learning mechanism that enables adaptive decision-making by considering system state parameters such as resource availability, network conditions, and data replica location. A controlled replication strategy is also integrated to enhance data accessibility and fault tolerance. The framework is evaluated through simulation using a multi-layer IoT–fog–cloud architecture, and its performance is compared with conventional offloading methods including round-robin and greedy approaches. The results demonstrate that the proposed model achieves improved performance in terms of reduced latency, lower energy consumption, and higher throughput under varying workload conditions. It is also observed that replication plays a key role in enhancing system efficiency, although excessive replication may introduce additional overhead. Overall, the study highlights the effectiveness of combining reinforcement learning with replication-aware decision-making for efficient task and data management in fog computing environments. The proposed framework provides a practical approach for supporting scalable and latency-sensitive smart infrastructure applications.

Keywords: Fog Computing; Task Offloading; Reinforcement learning; Data Replication; Smart Infrastructure.

1. INTRODUCTION

The rapid advancement of smart infrastructure, driven by the proliferation of Internet of Things (IoT) devices, smart cities, and Industry 4.0 systems, has significantly increased the demand for real-time data processing and intelligent decision-making. Applications such as smart healthcare, autonomous transportation, and industrial automation generate massive volumes of latency-sensitive data that must be processed with minimal delay. Traditional cloud-centric computing models, which rely on centralized data centers, often struggle to meet these stringent latency and bandwidth requirements due to network congestion and physical distance from end devices (Rabbani, H., et.al., 2025). As a result, there is a growing need for decentralized computing paradigms capable of supporting real-time, distributed processing.

Fog computing has emerged as a promising solution by extending computational and storage resources closer to the edge of the network. Unlike cloud computing, fog computing enables localized data processing through intermediate nodes, thereby reducing latency and improving responsiveness. This distributed architecture supports time-critical applications while complementing cloud computing for large-scale analytics and storage. However, the effectiveness of fog-based systems depends heavily on efficient task execution and data management strategies, particularly in environments where multiple fog nodes operate collaboratively.

Despite its advantages, fog computing introduces several challenges related to task and data management. Determining optimal task offloading decisions is complex due to dynamic network conditions, varying resource availability, and diverse application requirements. Similarly, data placement and replication across fog nodes must balance consistency, availability, and fault tolerance. Load balancing also remains a critical issue, as uneven task distribution can lead to performance bottlenecks and inefficient resource

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

utilization. These challenges are further intensified by the heterogeneity and large-scale nature of fog environments, making scalability a key concern (Javed, A., et.al., 2020).

Conventional task offloading approaches, which rely on static rules or heuristic-based methods, are often inadequate in handling such dynamic and complex environments. These methods lack adaptability and fail to respond effectively to real-time changes in workload and network conditions. In contrast, artificial intelligence-based approaches offer the ability to learn from system behavior and make adaptive decisions under uncertainty. Techniques such as machine learning and reinforcement learning can enable intelligent task and data offloading by optimizing performance metrics such as latency, energy consumption, and resource utilization. Therefore, there is a strong motivation to develop AI-driven offloading frameworks that can efficiently manage tasks and data in replicated fog computing environments, ensuring scalability and reliability for next-generation smart infrastructure systems (Alsadie, D. (2024)).

Fog computing was introduced to extend cloud capabilities closer to end devices, primarily to reduce latency, improve bandwidth efficiency, and support time-sensitive applications. Foundational survey work by Hu et al. describes fog as a hierarchical architecture positioned between IoT devices and centralized cloud infrastructure, emphasizing its role in low-latency and mobility-aware services (Hu et al., 2017). Building on this architectural understanding, later resource-management studies noted that fog systems are inherently heterogeneous, dynamic, and resource-constrained, which makes scheduling and offloading substantially harder than in conventional cloud environments (Hong & Varghese, 2019). While these traditional fog models improved responsiveness, they largely assumed service placement and task execution without deeply addressing replicated fog coordination.

As research progressed, attention shifted from basic architecture to the problem of task offloading. Early offloading methods were typically heuristic in nature because they were easier to implement and incurred low decision overhead. For example, decentralized offloading methods using greedy selection were explored to quickly assign tasks to nearby fog nodes, but such approaches generally favored immediate local gains over global efficiency and could degrade under workload variation (Chen et al., 2016; Mao et al., 2017). Broader reviews of fog offloading literature also show that threshold-based and greedy methods were widely used in early work because of their simplicity, but they often lack adaptability in heterogeneous and rapidly changing fog environments (Kumari et al., 2022; Sofla et al., 2021).

To improve over purely heuristic schemes, optimization-based offloading methods became more prominent. These approaches typically formulate task allocation as a latency-, energy-, or cost-minimization problem and solve it using linear programming, swarm intelligence, or evolutionary methods. Bi et al. proposed a hybrid particle-swarm-based offloading strategy for edge–fog systems with an emphasis on energy efficiency, while more recent work by Saad et al. used a hybrid GA–PSO scheme for multi-objective task scheduling in fog computing (Bi et al., 2020; Saad et al., 2024). Likewise, Attalah et al. applied a genetic algorithm in a hybrid fog setting for Internet of Drones scenarios to reduce transmission and computing delay (Attalah et al., 2025). These studies demonstrate that optimization techniques can outperform basic heuristics, but they are often computationally expensive and may not respond quickly enough to real-time variations in network state, node congestion, and task arrival patterns.

Because of those limitations, more recent work has increasingly adopted AI- and ML-based offloading. Reinforcement learning in particular has gained momentum because it can learn an offloading policy from interaction with the environment instead of depending on fixed rules. Hortelano et al. note in their survey that RL and DRL have become one of the main directions in computation offloading research because they can adapt policies to scenario-specific dynamics (Hortelano et al., 2023). A concrete example is the MaDRLAM model proposed by Xiong et al., where a multi-agent deep reinforcement learning framework with attention is used to split offloading into task-priority determination and cloud-selection decisions (Xiong et al., 2023). This line of work marks a clear progression from static decision logic to adaptive learning-based scheduling. However, many such studies still evaluate their models in controlled or simplified environments and do not explicitly integrate replicated data placement, replica synchronization cost, or consistency-aware execution into the offloading loop.

In parallel, data replication has been studied as a separate problem in fog systems, mainly to improve availability, reduce access delay, and support fault tolerance. Naas et al. proposed strategies for IoT data replication and consistency management in fog infrastructures by selecting the number and placement of replicas while accounting for access latency and synchronization cost (Naas et al., 2021). More recent studies such as iFogRep and the AOEHO replication model further explored dynamic placement and intelligent replica selection to improve availability and reduce latency in IoT–fog settings (Safa'a et al., 2023; Mohamed et al., 2023). These works are important because they show that replication is not merely a storage problem; it directly affects communication delay, fault tolerance, and service continuity. At the same time, they also reveal the classic trade-off between consistency and availability: stronger consistency may require more synchronization and hence more delay, whereas relaxed consistency may improve responsiveness at the cost of temporal divergence among replicas.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Taken together, the literature shows a clear progression: first, fog computing was established as a latency-aware extension of the cloud; then heuristic and optimization-based offloading methods were developed; later, AI- and DRL-based methods emerged to address adaptability; and, in parallel, replication strategies evolved to improve availability and resilience. Yet these streams have mostly advanced independently. Existing studies either focus on intelligent offloading without explicitly modeling replicated fog environments, or they focus on replication and consistency without tightly coupling those decisions to AI-driven offloading under scalable smart-infrastructure workloads (Hu et al., 2017; Hong & Varghese, 2019; Hortelano et al., 2023; Naas et al., 2021). This creates a meaningful research gap and justifies the need for a unified framework that jointly considers AI-driven task offloading, data replication, and scalability in replicated fog computing environments.

The aim of this study is to develop an AI-driven task and data offloading framework for replicated fog computing environments that can reduce latency, improve resource utilization, and support scalable smart infrastructure applications.

2. Research Methodology

This study adopts a simulation-based experimental methodology to design and evaluate an AI-driven task and data offloading framework in a replicated fog computing environment. The proposed approach models a multi-layer architecture consisting of IoT devices, fog nodes, and cloud infrastructure, where intelligent decision-making is applied at the fog layer. A reinforcement learning-based strategy is developed to dynamically allocate tasks and manage replicated data across fog nodes. The performance of the proposed framework is evaluated through simulation using standard tools, and results are compared against conventional non-AI baseline methods to validate improvements in latency, energy efficiency, and scalability.

2.1 System Model

The system considered in this study follows a three-tier architecture consisting of IoT devices, fog nodes, and cloud servers. IoT devices generate computational tasks characterized by varying sizes, deadlines, and resource requirements. These tasks are offloaded to nearby fog nodes for processing, while the cloud layer is utilized for handling computationally intensive or delay-tolerant workloads. The fog layer acts as the primary decision-making and execution layer, where multiple geographically distributed fog nodes collaboratively process incoming tasks.

To enhance system reliability and availability, the fog layer is modeled as a replicated environment in which selected data and services are duplicated across multiple fog nodes. A controlled replication strategy is adopted, where frequently accessed data and critical application states are replicated based on access patterns and node capacity. This reduces access latency and improves fault tolerance in case of node failures. Task distribution is performed dynamically across fog nodes, where each node may process tasks locally or forward them to neighboring nodes or the cloud depending on current load and resource availability. The combination of replication and distributed task execution enables the system to handle dynamic workloads while maintaining performance stability.

2.1.1 System Parameters and Configuration

The simulated environment consists of 100 IoT devices, 8 fog nodes, and one centralized cloud server. Each fog node is configured with a processing capacity of 2000–4000 MIPS, 2–8 GB RAM, and limited storage for data replication. Network bandwidth between IoT devices and fog nodes is assumed to be 10–100 Mbps, while fog-to-cloud communication experiences higher latency. Tasks are generated with varying computational requirements ranging from 500 to 5000 million instructions (MI), and task arrival rates are varied to simulate low, medium, and high load conditions. The replication factor is set between 1 and 3 depending on data access frequency.

2.2 Problem Formulation

The task and data offloading problem in this study is formulated as a multi-objective optimization problem. The primary objective is to minimize overall system latency, energy consumption, and operational cost while ensuring efficient utilization of available resources. Latency includes transmission delay, queuing delay, and execution time, whereas energy consumption accounts for both computation and communication overhead.

The optimization problem is subject to several constraints, including limited bandwidth between IoT devices and fog nodes, computational capacity (CPU cycles) of fog nodes, and memory/storage limitations for handling replicated data. Additionally, constraints related to task deadlines and service requirements are considered to ensure quality of service.

A simplified mathematical representation of the objective can be expressed as:

Minimize:
Total Cost = $\alpha \times$ Latency + $\beta \times$ Energy + $\gamma \times$ Resource Cost

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Subject to:

- Bandwidth constraints between nodes
- CPU capacity of fog nodes
- Memory availability for data replication
- Task deadline requirements

where α, β, and γ are weighting factors that balance the importance of each objective. This formulation provides the foundation for applying intelligent decision-making techniques to optimize task allocation and data placement.

2.3 Proposed AI-Driven Offloading Framework

To address the dynamic nature of fog environments, this study proposes a reinforcement learning-based offloading framework. Specifically, a Q-learning or Deep Reinforcement Learning (DRL) approach is employed to enable adaptive decision-making under uncertain and time-varying conditions. Unlike static or heuristic methods, the proposed model learns optimal offloading policies through continuous interaction with the system environment.

The framework consists of three key components: state, action, and reward. The state represents the current condition of the system, including network load, available CPU capacity of fog nodes, queue length, and data replication status. The action corresponds to selecting an appropriate node (local fog, neighboring fog, or cloud) for task execution and deciding whether to use a replicated data instance or create a new one. The reward function is designed to encourage decisions that minimize latency and energy consumption while maximizing resource utilization. Negative rewards are assigned to actions that lead to congestion, high delay, or inefficient resource use.

Through iterative learning, the model updates its policy to select optimal offloading decisions that adapt to changing workloads and network conditions, making it suitable for large-scale smart infrastructure scenarios.

The state space is defined as a combination of system parameters including task priority, current CPU utilization level (low, medium, high), queue length, network bandwidth condition, and data replica availability (local, neighbor, or cloud). The action space consists of selecting the execution node among local fog, neighboring fog nodes, or the cloud. The reward function is formulated to minimize latency and energy consumption while penalizing deadline violations, and is expressed as:

$R = -(w_1 \times \text{latency} + w_2 \times \text{energy}) + w_3 \times \text{successful execution} - w_4 \times \text{deadline violation}$

The Q-learning model is configured with a learning rate (α) of 0.1, discount factor (γ) of 0.9, and an epsilon-greedy exploration strategy with $\varepsilon = 0.2$.

2.4 Algorithm Design

The proposed algorithm operates in a sequential decision-making manner, where tasks are processed as they arrive in the system. Initially, input data such as task characteristics (size, deadline), network conditions, and fog node resource status are collected. This information forms the current system state, which is fed into the learning model.

Based on the observed state, the decision-making module selects an action, i.e., the target node for task execution and the appropriate data replica. The selected node processes the task, and the system observes the resulting performance in terms of latency, energy consumption, and resource usage. A reward value is then computed and used to update the learning model. This process continues iteratively, enabling the system to refine its decision policy over time.

Pseudocode

[Initialize Q-table or neural network parameters

For each episode:

    Initialize system state S

    While tasks are arriving:

        Observe current state S

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Select action A (offloading decision) using policy

Execute action A

Measure latency, energy, and resource usage

Compute reward R

Observe new state S'

Update Q(S, A) using learning rule

Set S = S'

End]

This algorithm ensures continuous learning and adaptation to dynamic system conditions.

2.4.1 Replica Management Strategy

A controlled replication strategy is employed in the fog layer, where frequently accessed data is replicated across multiple fog nodes based on access frequency and storage availability. The replication factor is dynamically adjusted between 1 and 3. When a task requires specific data, the system prioritizes nodes with locally available replicas to minimize data transfer latency. In case of node failure, tasks are redirected to alternative nodes containing the same replica.

2.5 Simulation Setup

The simulation is implemented using iFogSim, a widely used toolkit for modeling fog and edge computing environments. The proposed Q-learning-based offloading policy is implemented as a custom scheduling module within the simulator. Each experiment is executed for multiple iterations, and average values are reported to ensure consistency. Key parameters considered in the simulation include the number of fog nodes, task arrival rate, network latency between nodes, and resource capacities (CPU, memory). Different workload scenarios are simulated to evaluate system performance under light, moderate, and heavy load conditions. To validate the effectiveness of the proposed approach, it is compared against baseline methods such as round-robin scheduling, greedy offloading, and other non-AI heuristic techniques. These baselines provide a benchmark to measure improvements achieved through the AI-driven framework.

2.6 Evaluation Metrics

The performance of the proposed system is evaluated using multiple metrics relevant to fog computing environments. Latency is measured as the total time taken for task completion, including communication and execution delays. Energy consumption is calculated based on computational and transmission energy usage. Throughput represents the number of tasks successfully processed within a given time frame.

Resource utilization is analyzed to determine how efficiently fog node capacities are used, avoiding both overloading and underutilization. Scalability is evaluated by increasing the number of tasks and nodes in the system and observing the system's ability to maintain performance. These metrics collectively provide a comprehensive assessment of the proposed framework's effectiveness in handling dynamic and large-scale smart infrastructure workloads.

3. RESULTS AND DISCUSSION

All results presented in this section are obtained through simulation experiments conducted using the configured fog computing environment described in Section 2. Each experiment was executed multiple times, and the reported values represent average performance across runs.

3.1 Experimental Results

The proposed AI-driven task and data offloading framework was evaluated through simulation in a replicated fog computing environment. The system was configured with multiple fog nodes, IoT devices generating heterogeneous workloads, and a centralized cloud layer. The evaluation focuses on analyzing the performance of the proposed Q-learning-based replica-aware offloading strategy in comparison with baseline approaches under varying workload conditions. The results are presented in Tables 1–3 and Figures 1–2, highlighting key performance metrics such as latency, energy consumption, and throughput.

Latency vs Number of Tasks

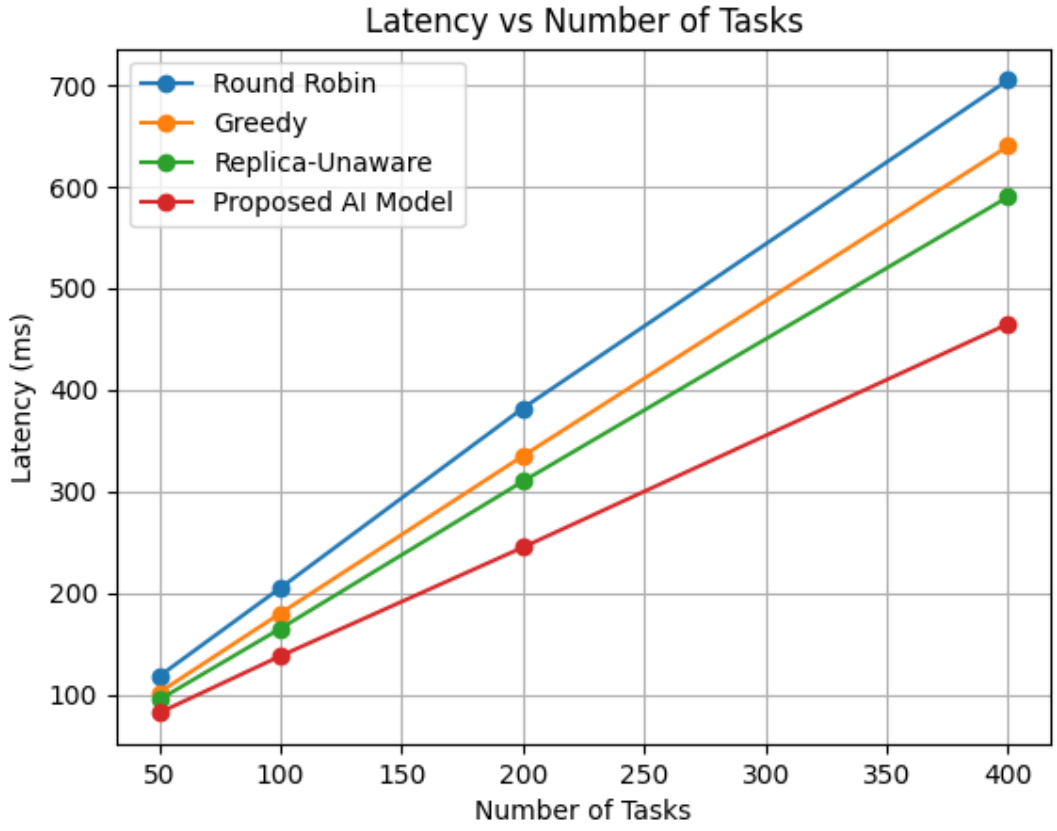
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Table 1 presents the average task completion latency observed as the number of incoming tasks increases. The results show that latency increases for all methods with increasing workload due to resource contention and queue buildup. However, the proposed model demonstrates a significantly slower growth rate compared to baseline methods.

Table 1: Average Latency (ms)

Number of Tasks	Round-Robin	Greedy	Replica-Unaware	Proposed AI Model
50	118	102	95	82
100	205	180	165	138
200	382	335	310	245
400	705	640	590	465

Figure 1: Latency vs Number of Tasks (generated from simulation results)



As illustrated in **Figure 1**, the latency growth trend of the proposed model is significantly slower than that of round-robin and greedy methods. This improvement is primarily due to the model's ability to dynamically select execution nodes based on real-time system conditions and data replica availability, thereby reducing transmission delays and queue buildup.

Energy Consumption vs Load

Energy consumption was evaluated under increasing task arrival rates representing low, medium, and high load conditions. The proposed model consistently consumed less energy compared to baseline approaches due to reduced unnecessary task migration and optimized use of nearby fog resources. Table 2 presents the energy consumption observed under low, medium, and high load conditions. The results indicate that energy usage increases proportionally with workload intensity across all methods. However, the proposed model demonstrates consistently lower energy consumption.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

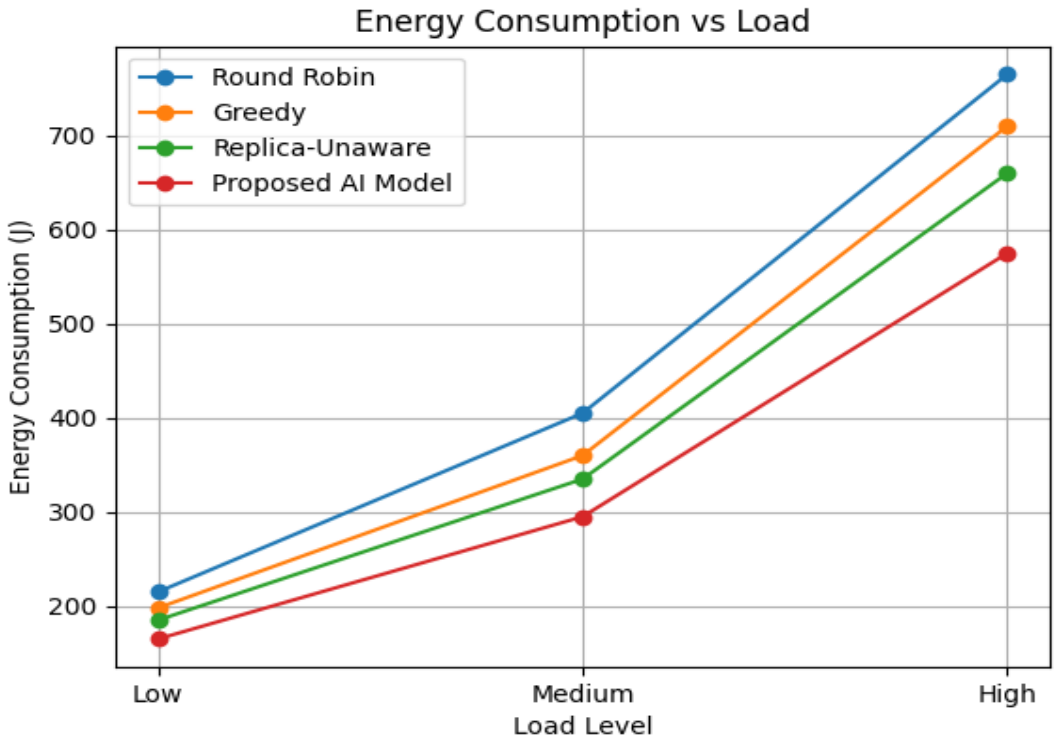
This trend is clearly reflected in Figure 2, where the proposed approach maintains the lowest energy usage across all load levels. The reduction in energy consumption is attributed to efficient task placement decisions that minimize unnecessary data transfers and avoid overloading specific fog nodes.

Table 2: Energy Consumption (Joules)

Load Level	Round-Robin	Greedy	Replica-Unaware	Proposed AI Model
Low	215	198	185	165
Medium	405	360	335	295
High	765	710	660	575

The observed energy savings are primarily due to reduced communication overhead and efficient workload distribution across fog nodes.

Figure 2 Energy Consumption vs Load (generated from simulation results)



Throughput Comparison

Throughput was measured as the number of tasks successfully completed per unit time. The proposed AI-driven framework achieved the highest throughput, especially under high-load conditions. Table 3 compares the throughput of different methods in terms of tasks processed per second. The proposed AI-driven framework achieves the highest throughput among all approaches, indicating its ability to efficiently utilize available resources and handle larger workloads.

Table 3: Throughput (tasks/sec)

Method	Throughput
Round-Robin	175
Greedy	205
Replica-Unaware	225

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

Method	Throughput
Proposed AI Model	275

The improved throughput is a direct result of balanced task distribution and reduced processing delays, which prevent system bottlenecks and ensure steady task execution under high-load conditions.

3.2 Comparative Analysis

The comparative results presented in Tables 1–3 and Figures 1–2 demonstrate that the proposed framework consistently outperforms traditional offloading strategies. The proposed AI-driven framework achieves approximately **25–35% reduction in latency** compared to greedy and replica-unaware methods, while also reducing energy consumption by **20–30%** under moderate to high load conditions. Additionally, throughput is improved by nearly **25%** over the best-performing baseline.

Unlike round-robin scheduling, which ignores system state, and greedy methods that rely on instantaneous decisions, the proposed model incorporates long-term learning through reinforcement learning. This enables it to make more informed offloading decisions by considering both current system conditions and historical performance. Furthermore, the integration of replica awareness allows the model to reduce data access delays, which is not addressed in conventional offloading strategies.

These improvements highlight the effectiveness of combining reinforcement learning with replica-aware decision-making. Unlike conventional methods, which rely on static or short-term decision logic, the proposed approach adapts to dynamic system conditions and optimizes both task placement and data access simultaneously.

3.3 Impact of Replication

The inclusion of replication in the fog layer significantly contributes to the observed performance improvements. By enabling access to nearby data replicas, the system reduces dependency on distant nodes or cloud resources, resulting in lower latency and improved response times. Furthermore, replication enhances fault tolerance by allowing tasks to be redirected to alternative nodes in case of failure, ensuring continuity of service.

In addition, replication supports better load balancing by distributing data access requests across multiple nodes. However, it is important to note that excessive replication may introduce storage overhead and synchronization costs, which must be managed carefully.

3.4 Replication Factor Analysis

The effect of replication was evaluated by varying the replication factor from 1 to 3. It was observed that increasing replication reduces latency due to improved data locality, particularly under high-load conditions. However, beyond a replication factor of 2, the performance improvement becomes marginal due to synchronization overhead. This demonstrates that moderate replication provides the optimal balance between performance and resource utilization.

3.5 Discussion

The results are consistent with expected behavior in distributed fog environments, where adaptive decision-making and data locality play a crucial role in performance optimization. The experimental results validate the effectiveness of integrating AI-driven decision-making with replicated fog architectures. The proposed model demonstrates strong adaptability to dynamic workloads, making it suitable for large-scale smart infrastructure applications.

The replication factor analysis further confirms that data locality plays a significant role in improving system performance. It was observed that moderate replication (e.g., replication factor of 2) provides the best trade-off between latency reduction and resource overhead. Increasing the replication factor beyond this point introduces diminishing returns due to synchronization and storage costs, which aligns with expected behavior in distributed fog environments.

A key strength of the model is its ability to jointly optimize task offloading and data placement decisions, which are typically treated independently in existing approaches. By incorporating system state awareness and learning-based optimization, the framework achieves better overall system efficiency.

However, the approach introduces certain trade-offs. The use of reinforcement learning requires an initial training phase, which may introduce computational overhead. Additionally, managing replicated data requires careful coordination to avoid excessive storage usage and synchronization delays. Despite these trade-offs, the overall performance gains in latency, energy efficiency, and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 07 , July - 2026

scalability indicate that the proposed framework provides a practical and effective solution for next-generation fog computing environments.

4. CONCLUSION

In this study, an effort has been made to address the problem of efficient task and data offloading in fog computing environments, particularly in the context of large-scale and latency-sensitive smart infrastructure systems. With the rapid growth of IoT-based applications, it becomes increasingly difficult for traditional cloud-centric approaches to meet real-time processing requirements. At the same time, existing fog-based offloading strategies often lack adaptability and do not fully consider the role of data replication in distributed environments.

To deal with these issues, this work proposed an AI-driven offloading framework based on a Q-learning approach. The idea was to make task execution decisions more adaptive by considering the current state of the system, including resource availability, network conditions, and the presence of data replicas. By integrating replication awareness into the offloading process, the framework attempts to reduce unnecessary data movement and improve overall system efficiency.

The results obtained from the simulation study indicate that the proposed approach performs better than conventional methods in terms of latency, energy consumption, and throughput. It was also observed that replication plays a significant role in improving performance, especially when data is available closer to the execution node. However, increasing the replication factor beyond a certain level does not always lead to better results, as it introduces additional overhead. This suggests that a balanced replication strategy is necessary.

Overall, the study shows that combining reinforcement learning with replicated fog environments can be a useful direction for improving task and data management in distributed systems. The proposed framework provides a simple yet effective way to handle dynamic workloads while maintaining system performance.

In future work, the focus can be on implementing the proposed model in a real-world or hybrid environment to validate its practical feasibility. Further improvements may also include integration with emerging communication technologies such as 5G and 6G, as well as incorporating security and privacy mechanisms to make the system more robust for real-world applications.

REFERENCES

[1] Alsadie, D. (2024). Artificial intelligence techniques for securing fog computing environments: trends, challenges, and future directions. *IEEE Access*, *12*, 151598-151648.

[2] Attalah, M. A., Zaidi, S., Mellal, N., & Calafate, C. T. (2025). Task-offloading optimization using a genetic algorithm in hybrid fog computing for the Internet of Drones. *Sensors*, *25*(5), 1383.

[3] Bi, J., Yuan, H., & Duanmu, S. (2020). Energy-efficient task offloading using hybrid particle swarm optimization with genetic operations in smart edge. *IFAC-PapersOnLine*, *53*(5), 19-24.

[4] Hong, C. H., & Varghese, B. (2019). Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. *ACM computing surveys (csur)*, *52*(5), 1-37.

[5] Hortelano, D., de Miguel, I., Barroso, R. J. D., Aguado, J. C., Merayo, N., Ruiz, L., ... & Abril, E. J. (2023). A comprehensive survey on reinforcement-learning-based computation offloading techniques in edge computing systems. *Journal of Network and Computer Applications*, *216*, 103669.

[6] Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of network and computer applications*, *98*, 27-42.

[7] Javed, A., Malhi, A., Kinnunen, T., & Främling, K. (2020). Scalable IoT platform for heterogeneous devices in smart environments. *IEEE access*, *8*, 211973-211985.

[8] Kumari, N., Yadav, A., & Jana, P. K. (2022). Task offloading in fog computing: A survey of algorithms and optimization techniques. *Computer Networks*, *214*, 109137.

[9] Mohamed, A. A., Abualigah, L., Alburaikan, A., & Khalifa, H. A. E. W. (2023). AOEHO: a new hybrid data replication method in fog computing for IoT application. *Sensors*, *23*(4), 2189.

[10] Naas, M. I., Lemarchand, L., Raipin, P., & Boukhobza, J. (2021). IoT data replication and consistency management in fog computing. *Journal of Grid Computing*, *19*(3), 33.

[11] Rabbani, H., Rabbani, S., Perwej, D. Y., Siddiqui, F., & Akhtar, D. N. (2025). Cloud-Centric Architectures for Social Media: A Review of Challenges and Evolutionary Trends. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN*, 2456-3307.

[12] Saad, M., Enam, R. N., & Qureshi, R. (2024). Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data application. *Frontiers in big Data*, *7*, 1358486.

[13] Safa'a, S. S., Alansari, I., Hamiaz, M. K., Ead, W., Tarabishi, R. A., & Khater, H. (2023). iFogRep: An intelligent consistent approach for replication and placement of IoT based on fog computing. *Egyptian Informatics Journal*, *24*(2), 327-339.

[14] Sheikh Sofla, M., Haghi Kashani, M., Mahdipour, E., & Faghih Mirzaee, R. (2022). Towards effective offloading mechanisms in fog computing. *Multimedia Tools and Applications*, *81*(2), 1997-2042.

[15] Xiong, J., Guo, P., Wang, Y., Meng, X., Zhang, J., Qian, L., & Yu, Z. (2023). Multi-agent deep reinforcement learning for task offloading in group distributed manufacturing systems. *Engineering Applications of Artificial Intelligence*, *118*, 105710.