MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 119-124

Edge Computing for IoT Networks: Reducing Latency through Intelligent Data Offloading

1 D. Sreevidya, Assistant Professor, Department of CSE -(CyS,DS and AI&DS)
VNRVJIET, Hyderabad,Telangana-500090
sreevidya_d@vnrvjiet.in

2 Mrs. Kolli Veena, Assistant Professor, Department of CSE -AIML&IOT,
Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering &Technology,
Hyderabad,Telangana-500090
veenachowdary93@gmail.com

3 Madhusudhana Reddy Govindu, Lead Engineer,
Department of CTX(consumer technology experience)
AT&T, 2900 W Plano Pkwy, Plano, TX- 75075
madhusudhanreddyg@gmail.com

4 Dr. Reshma M, Assistant Professor, Department of Electronics and communication Engineering,
University BDT college of Engineering, Karnataka - 577004
reshma.m03@gmail.com

5 Nisha Milind Shrirao, Assistant Professor, Department of Electrical
Yeshwantrao Chavan College of Engineering,Nagpur, Maharashtra-441110
sahu.nisha11@yahoo

6 Gaddi Poshamallu, Assistant Professor, Department of ECE,
St. Martin's Engineering College, Kompally, Telangana-500100
gaddi.poshamallu421@gmail.com

Abstract: The fast-growing number of Internet of Things (IoT) devices have contributed to the rapid growth of data generated, which has become a breadth to centralized cloud infrastructures. Conventional cloud-based systems tend to have high latency, bandwidth overload and low real-time response, all of which are severe limitations to latency-sensitive systems like autonomous vehicles, smart healthcare, and industrial automation. Edge computing has been identified as a prospective paradigm that takes computation and storage services nearer to the data sources. In this paper, intelligent data offloading plans in an edge computing setup are discussed to shorten the latency in IoT networks. An adaptive offloading system according to workload description and network characteristics is suggested which allows allocating tasks optimally between the edge node and the cloud. Experimental results reveal that intelligent data offloading will greatly decrease the end-to-end latency and enhance the efficiency of the network in comparison to the traditional cloud-only strategies.Some of the issues though, include practical limitations such as resource constraints at edge nodes, security vulnerability and large-scale deployment. Future research directions consist of incorporating the concept of AI driven predicted offloading, federated learning in making distributed intelligence and adaptive security frameworks to make edge-IoT systems more scalable and resilient.
Keywords: Edge computing, Internet of things (IoT), Data offloading, Latency reduction, Cloud computing, Distributed system, Network optimization.

I. INTRODUCTION

A major change that has occurred due to the rapid expansion of the Internet of Things (IoT) is a complete change of the interactions between devices, systems, and users within digital ecosystems. Millions of smart devices, actuators, and sensors are connected in real-time and produce huge amounts of data in fields like smart cities, medical monitoring, industrial automation, transportation of the smart world and environmental sensors[1]. To be trusted in terms of reliability, safety, and operating efficiency, these applications usually involve real-time or close-to-real-time decision-making. Apparently, the classical IoT architectures are based on centralized cloud computing infrastructures to store, process, and analyze data. Although cloud computing is virtually infinite in terms of computational power and scalable storage, it also creates a great deal of latency through long-range data transfer and network overloads.The constraints of cloud-based models are becoming more apparent as more IoT implementations are rolled out. Round trip communication between devices and remote cloud servers cannot be allowed in such applications because it introduces delays that are not tolerated by latency sensitive applications like autonomous vehicles, remote robotic surgery, predictive industrial maintenance, and emergency response systems. Additionally, the increasing number of connected devices exposes network bandwidth to extreme loads, causing congestion, lost packets and poor Quality of Service (QoS). Moreover, the continuous transmission of data to data centers located in one place, consumes more energy and threatens privacy in cases where sensitive data is being dealt with.The problem has been countered by Edge computing which has become a promising paradigm. Edge computing draws computational and storage resources nearer to the IoT devices, located on the network edge, and minimizes the physical and logical distance that the data has to travel[2]. This is a major way of minimizing communication latency, improving responsiveness and minimizing bandwidth usage. Edge nodes can process, filter, and analyze data locally instead of sending all of the raw data to the cloud, and only post relevant or aggregated information to the centralized servers. This distributed architecture is not only useful in improving performance but also in increasing the system-scaling and resiliency.Nevertheless, it cannot be assumed that the only way to ensure optimal performance of IoT networks is by introducing edge nodes[3]. The placement of tasks to be processed and their timing is one of the most essential issues in edge-enabled IoT systems. IoT devices are usually resource-capped both in computation power, memory, and energy capacity. Edge nodes are relatively strong in terms of computational provisions and are still weak relative to data centers at the clouds. So it is necessary to implement intelligent data offloading to guarantee the efficient use of the resources available whilst reducing the latency and energy usage.The offloading of data is the dynamic transfer of computational loads between the IoT devices and the edge node or cloud server on the basis of the established requirements[4]. Conventional ffloading systems tend to be based on fixed policies or threshold-based decision policies, which cannot

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 119-124

accommodate dynamism in networks, changes in workload, and the capabilities of devices of heterogeneous nature. The ability of the systems to adapt and make decisions based on contexts is needed in highly dynamic IoT environments due to the changing bandwidth, the changing task sizes, user mobility, which is unpredictable, as well as uneven distribution of resources.

The rationale of this work is the necessity to design a smart and scalable data offloading system that can reduce latency and not waste resources in the context of heterogeneous IoT networks. Most of the current methods assume an unchanging network, or do not consider real time flexibility. Moreover, other models are only interested in the reduction of the latency without paying sufficient attention to the bandwidth efficiency, energy consumption, and scalability rates. It is evident that there is need to have an elaborate framework that consolidates various performance measures towards a single model of making a decision.

In 2020 Hamdan [5] proposed et.al.,This paper seeks to fill this gap by suggesting an offloading mechanism that is dynamic and context-sensitive and that constantly tracks the system parameters to optimize placement of tasks between IoT devices, edge nodes, and the cloud servers. The fundamental goal is to lower the end-to-end latency and system reliability and scalability. The suggested framework can assess the characteristics of the tasks, such as the computational complexity and the amount of data, and the real-time network conditions: the latency and the bandwidth resources. Also, the possibility of the availability of resources at edge nodes is included in the decision-making process in order to avoid overload and evenly allocate the workload.

The main aims of this work are the following:

To examine the shortcomings of the conventional cloud-based IoT systems within the latency-intensive systems.

To develop a smart data offloading model to select dynamically the most appropriate processing layer (device, edge or cloud).

To compare the work of the suggested model based on the latency, bandwidth consumption, and energy consumption.

To offer scalable architectural framework, which can be used in heterogeneous IoT environments.

In the present paper, an overview of the proposed edge-based offloading framework will be described with references to its architecture, decision-making mechanisms, and performance analysis. The framework aims at improving the responsiveness of the entire system and its operational efficiency by incorporating both the adaptive workload management and real time monitoring of the system[8]. The results indicate that the intelligent offloading can be very effective in minimizing the latency when compared with the traditional models that are either a static or cloud-only model.

To conclude, the proposal of intelligent data offloading in edge computing environments is a very vital move in supporting next-generation IoT applications, which require low latency, high reliability, and efficient resource utilization. The contribution of this work is in the continuation of the further development of distributed computing paradigm as it provides a practical and scalable solution according to the dynamic needs of IoT networks.

Novelty and Contribution

The originality of this study is that a dynamic, context-sensitive intelligent data offloading framework was developed to be responsible in the alleviation of latency in heterogeneous IoTnetworks[10]. The proposed approach, in contrast to traditional models that may use rigid decision criteria or concentrate on individual indicators of performance only, combines a variety of parameters, such as task complexity, network conditions, availability of edge resources, and energy constraints, to a common adaptive decision operation.

One of the most innovative features of the given work is the ability to monitor in real-time and constantly optimize it. As opposed to one offloading decision, the system responds dynamically to changing network and workload conditions by dynamically changing task assignment. This is to be sure that the performance optimization is preserved even with varying traffic patterns and delicate scenarios with device mobility. Also, the framework focuses on the equal distribution of workloads among edge nodes to avoid workforce overload, which is a typical problem in distributed edge systems. The model improves scalability and reliability of massive IoT deployments by considering the offloading strategy with resource-awareness.

In 2021 Jazaeri [6] suggested et.al.,The other new input is the holistic evaluation approach. The proposed framework also does not concentrate only on the reduction of latency but also takes into consideration bandwidth optimization and energy efficiency. This is a multi-objective optimization approach that offers a broader solution that is in line with the deployment requirements in the real world.

Primary Contribution

The main contributions of this study may be summed up as following:

Wireless Cloud Sensor Network Design:

An IoT-based three layer architecture that incorporates IoT devices, edge nodes, and cloud infrastructure is suggested, which offers a scalable and structured model of deployment.

Intelligent Offloading Model by Adaptation:

An active decision algorithm is created to identify a good position of tasks depending on the current system parameters and workload features. The strategy to be employed in multi-metric optimization is to maximize a multi-metric based on the available data.<|human|>Multi-Metric Optimization Strategy:

The framework is also capable of maximizing the latency, bandwidth usage, and energy use, focusing on several key performance indicators.

Performance Appraisal and Comparative Reports:Experimental results show that there are much lower latencies and better network effectiveness than cloud-only and fixed edge-based strategies.Practical issues like edge resource constraints and workload inequality are also tackled in the study and provide a glimpse of what can be applied in the real world.In general, the article contributes to the development of edge-enabled IoT systems by introducing a performance-driven, adaptive, and intelligent data offloading framework. It offers both theoretical and operational design principles to be used in the development of low-latency, scalable IoT infrastructures that have the capability of supporting next-generation applications.

II. RELATED WORK

The introduction of edge computing to Internet of Things (IoT) networks has gained a lot of research interest because of the constraints associated with conventional cloud-based architectures. Initial IoT applications were very dependent on the use of centralized cloud to store and process their data[11]. Although cloud computing is elastic and offers great power in terms of its computational power, it generates latency due to long distance communication and network overload. With the increasing availability of latency-sensitive IoT applications, scientists started to investigate the idea of distributed computing to alleviate these constraints.The introduction of the concept of fog computing as an extension of cloud services nearer to end devices was also one of the foundational developments in this direction. Fog computing architecture sought to accelerate the delay in communication and enhance the context awareness capability through localized computing. These architectures were found to be far more responsive than all-cloud based systems. Nevertheless, most of the early fog models did not pay much attention to smart task allocation schemes but more on architecture designing. They tended to make assumptions about the condition of the network which was static and failed to properly consider the dynamic changes of workloads in heterogeneous IoT settings.

Later research changed to a more task scheduling and workload management in the edge-enabled IoT systems. Scheduling algorithms of different tasks were introduced to establish the manner in which the computational tasks are allocated among the edge nodes and cloud servers. Such strategies had the common goal of minimizing latency, minimizing energy use or optimizing throughput. Others employed heuristic-based scheduling methods which had rather low computational cost, but were not flexible to network variations. Whereas these methods worked better in controlled conditions, they failed in large scale or high-turnover deployments.In 2020 Jha [7] introduced et. al., order to increase flexibility, optimization-based methods were used more and more in research. Mathematical models were created to formalize task offloading and these are usually optimization problems, which may have the objective of minimizing total execution time or energy consumption. These models often utilized a linear programming, convex optimization or game-theoretics. Whereas optimization-based models had better resource allocation efficiency, most of them made the assumption that there was effective prior knowledge of system parameters. However, network latency, bandwidth availability and device mobility in real-world IoT networks are subject to unpredictable variability, which limits the range of applicability of the static optimization frameworks used in practice. The recent studies have discussed intelligent offloading mechanisms with the development of artificial intelligence and machine learning. Based on machine learning, such approaches can forecast the conditions of the network, workload, and the resource availability to take proactive decisions on offloading. An example of such reinforcement learning is reinforcement learning which has received much attention in dynamic task allocation as it is able to learn optimum policies by interacting with the environment. These techniques showed good results in latency reduction and the fact that the system can perform better. However, when they are applied to real-world IoT systems, there are challenges of complexity in training, and computational load and the requirement of large datasets. There are also models of machine learning that can add decision latency, which can counterbalance performance improvements in applications that need speed. The other research area is energy-efficient offloading strategies. IoT devices are also often limited by the capacity of the battery, which is why the issue of energy consumption becomes the central one. Research in this field studies trade-offs between local execution and remote processing. Local processing consumes less energy in communication but can consume more energy in battery drainage of the devices whereas offloading consumes less energy in the local device and consumes more energy in network transmission. A number of energy-conscious algorithms dynamically decide on optimal points of execution based on remaining battery levels and energy models. Although these strategies are effective in extending device lifetime, not necessarily should it ensure a minimum of latency particularly where the conditions of the network are not consistent. Optimization of bandwidth has also been one of the themes of related studies. With the increasing amount of IoT devices, it is ineffective and expensive to transfer raw data to the cloud. To minimize unnecessary transmissions, data compression, filtering as well as aggregation at edge nodes has been proposed. Other works are focusing on content-aware processing of data, in which only relevant data is sent so further levels. Even though these methods can reduce bandwidth to a large extent, they can also incur extra computational costs on the edge that could create processing bottlenecks. The issue of trade-off between bandwidth reduction and computational load is a continuing problem.

The issue of security and privacy has also affected the studies of edge based IoT systems. Distributed data processing brings new attack points and weaknesses. Various works have suggested safe offloading systems, such as the encryption-based transmission protocols and blockchain-based trust management systems. Privacy-friendly models, including distributed learning models that do not require aggregation of data in a central point have been explored. Although these techniques can effectively improve system security, they can add extra computational and latency overhead, which should be properly controlled in systems that require low latency.

Another research area is mobility-aware offloading strategies. In other applications like connected cars and mobile medical devices, the position of IoT nodes in a network is often varying. Mobility-aware networks seek to provide continuity of the services by dynamically reallocating tasks to the edge nodes within the proximity. Such strategies usually make use of predictive models to anticipate the movement pattern of the user and allocate resources ahead of time. Despite the fact that mobility-aware mechanisms enhance the reliability of the services, they add complexity to the system and demand precise prediction mechanisms that will eliminate unnecessary resources allocation.

The literature has also given much attention to scalability. With the IoT ecosystems reaching millions of devices, centralized ways of control become impractical. Models of distributed orchestration have been put forward to coordinate edge resources in an efficient manner. Containerization and lightweight virtualization technology is commonly added to the edge infrastructures to allow flexible provisioning of resources. Nevertheless, the coordination of the high numbers of distributed edge nodes poses the problem of orchestration, such as synchronization overheads and fault tolerances.

Although the contributions in the existing literature are rather extensive, there are still some gaps. A significant number of previous works are working on the optimization of one of the performance metrics, like latency, energy, or bandwidth, but do not fully cover the multi-objective trade-offs. Some make the homogeneous edge resources or constant network conditions assumptions which do not necessarily correspond to real world deployments. Moreover, there are intelligent offloading solutions with a high computation cost and, hence, are not always feasible within resource-constrained settings.

Altogether, the presented related literature indicates the potential of edge computing to promote the performance of IoT by distributing tasks better and processing them locally. Nevertheless, it is necessary for flexible, contextualized frameworks that would combine various measures of performance, being scalable and having minimal computational footprints. It is necessary to address these challenges in order to achieve robust and low-latency IoT networks that can accommodate the new real-time and mission-critical applications.

.III. PROPOSED METHODOLOGY

The current methodology is an adaptive and smart data offloading framework that is used to minimize the latency and maximize the resource utilization in IoT networks. The system is based on the three-layer infrastructure comprising of IoT devices, edge nodes and cloud servers. When a task is created by an IoT device, a monitoring module constantly assesses important system parameters (task characteristics, network conditions, available bandwidth, edge resource capacity and device energy levels). It works on this real-time data to have a decision engine decide on whether the task is to be completed on the edge or sent to the cloud. Latency sensitive tasks are given priority in the framework as edge processing tasks whereas the computationally intensive, yet less time sensitive tasks are sent to the cloud. Furthermore, workload balancing systems are used to eliminate the overloading of edge nodes and to guarantee effective redistribution of work throughout the network[12]. This is a dynamic and context-sensitive method of operation that is more responsive, and better bandwidth use and scalable deployment in heterogeneous IoTenvironments.Flowchart Title

Proposed Intelligent Data Offloading Framework for IoT Networks

This flowchart illustrates the step-by-step process of intelligent task evaluation and dynamic offloading between IoT devices, edge nodes, and cloud servers. It highlights how real-time parameter monitoring and cost-based decision-making are used to minimize latency and optimize resource utilization.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 119-124

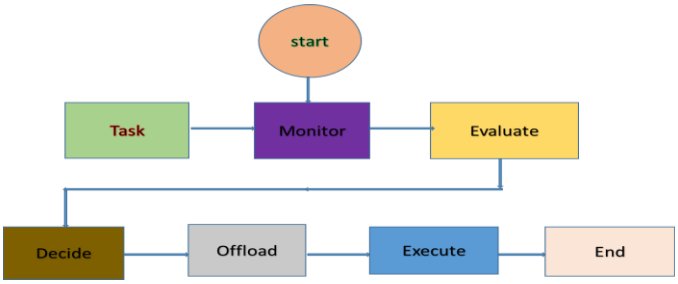

Fig 1Proposed Intelligent Data Offloading Framework for IoT Networks

The flowchart reflects the working process of the suggested intelligent data offloading scheme of the IoT networks. Start node is the starting point of the process, which implies the initiation of the system. After being activated, an IoT device produces a computational Task that can contain sensor data processing, analytics requests or real-time control instructions. This is the main input to the offloading system that initiates the monitoring and evaluation cycle.During the second stage, the Monitor module is constantly monitoring the critical parameters in the system like the size of the tasks at hand, workload intensity, bandwidth availability, network latency, and availability of resources at the edges. The purpose of this monitoring stage is that the system is in real-time aware of the network conditions as well as the computational capacity. Proper monitoring is highly required since dynamic IoT environments tend to have changing loads of traffic and varying resource conditions.

Monitoring will be followed by the Evaluate stage of the system. In this case, gathered parameters will be evaluated to determine the viability and effectiveness of options of various processes. The analysis identifies the level of performance of processing the task on the local server, edge processor nodes, or cloud[13]. The step is a performance assessment layer that prepares information, which is also needed to make intelligent decisions.Subsequently, it goes into the Decide stage where the system decides the most ideal location of execution based on the results of the evaluation. When there is enough edge resources and the latency requirement is high, then the task is scheduled to edge processing. Otherwise, cloud processing is chosen to process the task. After this decision, it passes to the Offload stage, whereby it is sent to the selected computational layer.Lastly, each node (edge or cloud) of the selected node (edge or cloud) performs the task in the Execute stage and sends the result back to the requesting device or application. The End node is the final node in the workflow that implies the task cycle completion. It is a sequence that is structured to guarantee low latency, effective resource usage, and flexible resource allocation in the IoT network.

IV. RESULT& DISCUSSIONS

The experimental evaluation demonstrates the effectiveness of the proposed intelligent data offloading framework in reducing latency, optimizing bandwidth usage, and minimizing energy consumption compared to traditional cloud-only and static edge architectures.

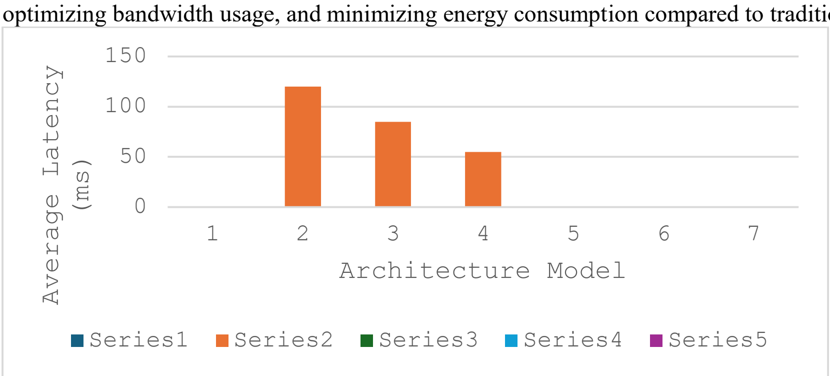

Figure 2: Latency Comparison

Figure 2 shows the mean end to end latency in the three architectures Cloud Only, Static Edge, and the Proposed Model. The most latency is registered by the cloud-only solution because of the central processing and because of requiring the data transfer over a long distance between the IoT devices and the remote data center. These delays are magnified by network traffic, or the devices themselves may be too far away to be connected to the cloud infrastructure. The model of the edge statistic is improved as the computation of part is done nearer to the source of data and the time is lessened.The intelligent offloading model proposed has the least latency when compared to any other architecture. This is the benefit of dynamic monitoring and active task assignment, which makes sure that the tasks that are latency sensitive are handled at the most appropriate layer. The framework reduces the time wastage in waiting time during communication by making wise decision on whether to perform tasks on edge or on cloud. The drastic drop of latency proves the appropriateness of the suggested model to be used in real-time IoT systems like healthcare monitoring, autonomous systems, and industrial automation.

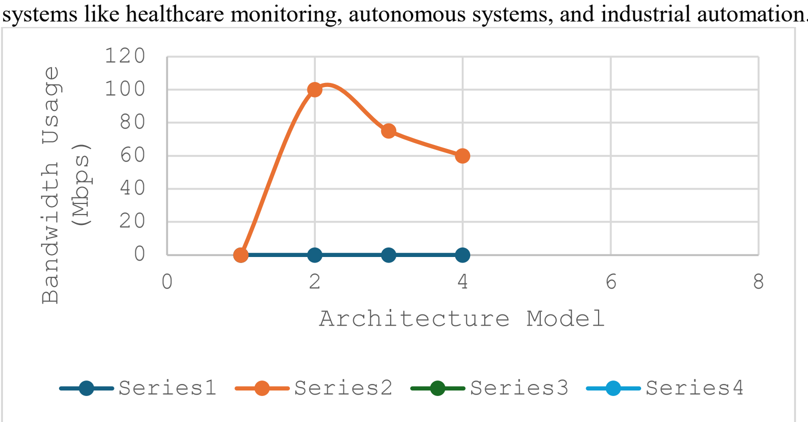

Figure 3: Bandwidth Utilization Comparison

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 119-124

The bandwidth consumption of all three architectures is shown in figure 3. The cloud-only model has the largest bandwidth since it sends all the raw IoT data to the centralization servers without preprocessing and local filtering. This method will cause network congestion and degraded Quality of Service as more and more connected devices are added to the network. The model that minimizes bandwidth was the static edge model that eliminates the need to preprocess much at the edge node and, hence, reduces data transmission to the cloud.
The intelligent data filtering and selective transmission mechanisms make the proposed model the lowest bandwidth consuming model[14]. Through task demand analysis and network environment, the system only sends out critical or processed data to the higher layers. It is a highly efficient data manipulation to minimize network loads and scalability. The utilization of bandwidth is considered especially effective in the case of defining the large-scale IoT deployments involving thousands of devices simultaneously that produce data streams that are continuous.

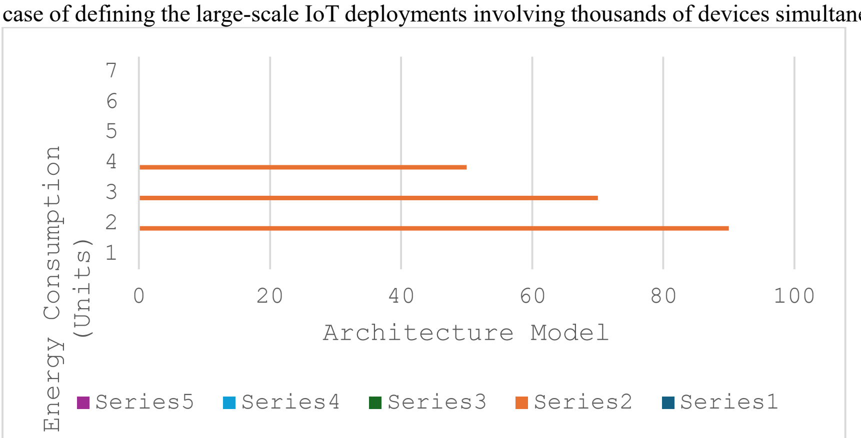

Figure 4: Energy Consumption Comparison
Figure 4 comparatively evaluates the energy consumption of the IoT devices in the three architectures. The cloud-only solution leads to maximum energy consumption since gadgets are required to run large amounts of data over long distances. Constant communication consumes more power particularly in battery operated IoT systems. The static edge model is moderate in terms of energy saving as certain processing is done locally, and long-range communication does not always have to occur.
The intelligent offloading proposal has the lowest energy consumption when compared to the other two models. The system limits the amount of communication overhead and manages the consumption of device energy by optimizing task allocation dynamically and by minimizing unnecessary data transmission. Reduced energy use increases the life of the devices and improves sustainability of IoT ecosystems. It is specifically useful to remote sensors, wearable and smart infrastructure systems which use sparse power resources

TABLE 1: Simple Architecture Comparison

Cloud Only	Static Edge	Proposed Model
High Latency	Medium Latency	Low Latency
High Bandwidth Use	Medium Bandwidth Use	Low Bandwidth Use
High Energy Consumption	Medium Energy Consumption	Low Energy Consumption
Low Adaptability	Moderate Adaptability	High Adaptability
Limited Real-Time Support	Partial Real-Time Support	Full Real-Time Support

A comparison of the three architectures could be simplified as it is presented in Table 2 Cloud Only Static Edge and Proposed Model. Cloud-only architecture has latency that is high, bandwidth consumption is high and the consumption of energy is also high since all processing occurs at a centralized servers. It is not as flexible and has less support to real-time applications, so it is not as applicable to delay-sensitive IoT systems. Despite cloud infrastructure having good computational power, its reliance on long distance communication introduces performance bottlenecks[15].
The use of the static edge architecture enhances performance as it can perform part processing around the IoT devices leading to moderate latency, bandwidth consumption, and energy consumption. Nevertheless, it is not quite dynamic in network conditions. Conversely, the intelligent offloading model proposed has low latency, low bandwidth and low energy as a result of its adaptive and context-sensitive decision-making mechanism. It is scalable and more efficient and reliable in supporting modern IoT deployments with full real-time application needs.

Table 2:Numeric Performance Comparison Table

Cloud Only	Static Edge	Proposed Model
120 ms	85 ms	55 ms
100 Mbps	75 Mbps	60 Mbps
90 Units	70 Units	50 Units

Table 2 provides a comparison of he quantitative comparison in terms of the three architectures Cloud Only, Static Edge, and the Proposed Model is made directly in the numeric table. The first row is the average values of latency, where under the cloud-only architecture the latency is 120 ms, under the static edge architecture the latency is lower to 85 ms, and under the proposed model the average latency is lower to 55 ms. The bandwidth utilisation in the second row has cloud-only, 100 Mbps; static edge, 75 Mbps; and some proposed system with 60 Mbps. The third row shows the consumption of energy in which cloud only is 90 units, the statical edge is 70 units and the proposed framework is 50 units. These values are a clear indication of steady performance increase in all the measures.
The findings show that the intelligent offloading model proposed is more efficient and optimized than the traditional methods. The latency reduction indicates enhanced responsiveness in real-time IoT applications, whereas the reduction in bandwidth consumption indicates the decrease in congestion in the network and enhanced scalability. Moreover, reduced power usage improves the longevity of the device and increases its life cycle. On the whole, the quantitative analysis of the data proves that an adaptive and context-aware distribution of tasks has a significant beneficial impact on the work of the system in the environment of edge-enabled IoT.

V. CONCLUSION

Edge computing proposal provides a feasible method to address the latency issues within an IoT network since it enables intelligent data offloading among devices, edge nodes, and cloud servers. This paper will prove that adaptive offloading strategies are very effective as they help minimize latency, better bandwidth use, and better system efficiency. The suggested framework is dynamically adapted to the changes in workloads and networks, which is superior to the classical cloud-based and fixed edge models.

Although these may have the advantages, there are real life inefficiencies. The edge nodes have few computing capabilities than cloud data centers, which may cause bottlenecks during heavy workloads. The issue of security and privacy is based on the distributed data processing among several nodes. Moreover, massive implementation needs standard interoperability frameworks and effective management systems.
Research on AI-based predictive offloading models which predict workload variations and network variations are in need of investigation in the future. Federated learning may be realized through the integration to facilitate collaborative intelligence between distributed edge nodes with privacy of data. In addition, it will be essential to come up with lightweight security procedures and dynamic orchestration systems to make IoT ecosystems on the edge resilient, safe, and efficient.

REFERENCES

[1] Ai, Y., Peng, M., & Zhang, K. (2017). Edge computing technologies for Internet of Things: a primer. Digital Communications and Networks, 4(2), 77–86. https://doi.org/10.1016/j.dcan.2017.07.001

[2] Andriulo, F. C., Fiore, M., Mongiello, M., Traversa, E., &Zizzo, V. (2024). Edge Computing and Cloud Computing for Internet of Things: a review. Informatics, 11(4), 71. https://doi.org/10.3390/informatics11040071

[3] Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., &Fortino, G. (2023). At the Confluence of Artificial Intelligence and Edge Computing in IoT-Based Applications: A Review and New Perspectives. Sensors, 23(3), 1639. https://doi.org/10.3390/s23031639

[4] Fazeldehkordi, E., &Grønli, T. (2022). A survey of Security Architectures for Edge Computing-Based IoT. IoT, 3(3), 332–365. https://doi.org/10.3390/iot3030019

[5] Hamdan, S., Ayyash, M., &Almajali, S. (2020). Edge-Computing Architectures for Internet of Things Applications: A survey. Sensors, 20(22), 6441. https://doi.org/10.3390/s20226441

[6] Jazaeri, S. S., Jabbehdari, S., Asghari, P., &Javadi, H. H. S. (2021). Edge computing in SDN-IoT networks: a systematic review of issues, challenges and solutions. Cluster Computing, 24(4), 3187–3228. https://doi.org/10.1007/s10586-021-03311-6

[7] Jha, D. N., Alwasel, K., Alshoshan, A., Huang, X., Naha, R. K., Battula, S. K., Garg, S., Puthal, D., James, P., Zomaya, A., Dustdar, S., &Ranjan, R. (2020). IoTSim-Edge: A simulation framework for modeling the behavior of Internet of Things and edge computing environments. Software Practice and Experience, 50(6), 844–867. https://doi.org/10.1002/spe.2787

[8] Khan, W. Z., Ahmed, E., Hakak, S., Yaqoob, I., & Ahmed, A. (2019). Edge computing: A survey. Future Generation Computer Systems, 97, 219–235. https://doi.org/10.1016/j.future.2019.02.050

[9] Mehmood, M. Y., Oad, A., Abrar, M., Munir, H. M., Hasan, S. F., Muqeet, H. a. U., &Golilarz, N. A. (2021). Edge computing for IoT-Enabled smart grid. Security and Communication Networks, 2021, 1–16. https://doi.org/10.1155/2021/5524025

[10] Minh, Q. N., Nguyen, V., Quy, V. K., Ngoc, L. A., Chehri, A., &Jeon, G. (2022). Edge computing for IoT-Enabled smart Grid: The future of energy. Energies, 15(17), 6140. https://doi.org/10.3390/en15176140

[11] Mocnej, J., Miškuf, M., Papcun, P., &Zolotová, I. (2018). Impact of edge computing paradigm on energy consumption in IoT. IFAC-PapersOnLine, 51(6), 162–167. https://doi.org/10.1016/j.ifacol.2018.07.147

[12] Rupanetti, D., &Kaabouch, N. (2024). Combining Edge Computing-Assisted Internet of Things Security with Artificial Intelligence: Applications, Challenges, and Opportunities. Applied Sciences, 14(16), 7104. https://doi.org/10.3390/app14167104

[13] Sha, K., Yang, T. A., Wei, W., &Davari, S. (2019). A survey of edge computing-based designs for IoT security. Digital Communications and Networks, 6(2), 195–202. https://doi.org/10.1016/j.dcan.2019.08.006

[14] Singh, A., Satapathy, S. C., Roy, A., &Gutub, A. (2022). AI-Based Mobile Edge Computing for IoT: applications, challenges, and future scope. Arabian Journal for Science and Engineering, 47(8), 9801–9831. https://doi.org/10.1007/s13369-021-06348-2

[15] Sittón-Candanedo, I., Alonso, R. S., García, Ó., Muñoz, L., & Rodríguez-González, S. (2019). Edge computing, IoT and social computing in smart energy scenarios. Sensors, 19(15), 3353. https://doi.org/10.3390/s19153353