Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

OPTIMIZING REAL TIME IOT PROCESSING WITH HYBRID EDGE CLOUD ARCHITECTURE FOR ENHANCED LATENCY AND ENERGY EFFICIENCY

ANIL KUMAR PALLIKONDA[1*], ANIL KUMAR KATRAGADDA[2], JAGADEESWARA RAO ANNAM[3], V.V.RAMA KRISHNA[4], SAI SIRISHA CHITTINENI[5], ESWAR PATNALA[6], VIPPARLA ARUNA[7]

[*1]Department of CSE, PVP Siddhartha Institute of Technology, Andhra Pradesh, India.

[2]Staff Data Engineer, BD (Becton Dickinson), Tampa, FL, USA

[3]Department of CSE, CVR College Of Engineering, Vastunagar,Telangana, India

[4]Department of ECE, Lakireddy Bali Reddy College of Engineering, Andhrapradesh, India.

[5]Department of CSE, DVR&Dr.HS MIC college of technology, Andhrapradesh, India.

[6]Department of CSE, Koneru Lakshmaiah Education Foundation, Andhrapradesh, India

[7]Department of CSE, NRI Institute of Technology, Andhrapradesh, India

E-mail: [1]anilkumar.pallikonda@gmail.com , [2]anilkatragadda.k@gmail.com, [3]ajr@cvr.ac.in, [4]vvrk@lbrce.ac.in , [5]sirisha.chittineni8@gmail.com , [6]peswar@kluniversity.in, [7]aruna.vipparla5@gmail.com

ABSTRACT

The proposed hybrid edge-cloud architecture system aims to maximize efficiency in dynamically processing real-time data for Internet of Things (IoT) powered applications. By strategically placing a fog layer, the system seeks to balance the load, ensuring efficient data processing while minimizing latency, optimizing energy consumption, and enhancing scalability. This is achieved through a Dynamic Task Offloading (DTA) algorithm that intelligently assigns tasks to either the edge or cloud layer. Experiments using a synthetic smart city traffic dataset demonstrate that the hybrid model can reduce latency by 70% and save 55% in energy consumption compared to a cloud-only model, while achieving a task offloading efficiency of 92%. The architecture supports high scalability, resource utilization, and timely decision-making, significantly reducing processing latencies and energy usage for IoT applications. The study highlights the limitations of cloud-only systems, including high latency and scalability issues, and addresses these by proposing a hybrid solution that enhances real-time IoT data processing capabilities in dynamic environments.

Keywords: *Hybrid Architecture, Edge Computing, Cloud Computing, IoT, Task Offloading, Latency Optimization*

1. INTRODUCTION

Over the last few years, the Internet of Things has exploded into various industries, from health care and agriculture to manufacturing and cities. Internet of Things (IoT) connects devices, sensors, and cloud platforms, which allow data to be collected, analyzed, and decisions taken in real-time, thereby increasing operational efficiency and user experience. Nonetheless, for conventional cloud computing systems, it becomes a ticking time bomb moving forward because the seamless number of IoT ecosystems and the data they generate are escalating exponentially. Traditional cloud computing systems will face significant challenges in managing, processing, and delivering everything in real-time. Traditionally, cloud systems cannot address the needs of delay-sensitive IoT applications due to issues like high latency, limited bandwidth, and the massive scale of IoT [1], [2].

For a long time, cloud computing stood as the pillar for IoT data processing, offering centralized computational power and storage. However, this key functionality comes with the hindrance of high latency due to the orchestration of data across remote servers in latency-sensitive applications such as autonomous vehicles, healthcare monitoring, and industrial automation. Such delay in transferring data between IoT devices and cloud servers can lead to ineffectiveness and, in extreme cases, severe failure [3],[4]. To meet these challenges, there has been a shift in the field to explore edge computing in conjunction with cloud systems.

Edge computing executes the data as close to the source as possible on edge devices or servers, reducing long-distance data transmission and latency. These applications involve a significant amount of time-critical data processing, which often requires a response to local events within tens of seconds, making edge computing the best solution [5], [6] in the context of IoT systems by processing locally in real-time at the edge, which can significantly reduce the time taken for IoT systems to respond to these changes. However, edge computing cannot satisfy numerous IoT systems' massive computing and storage requirements. However, due to their limited resources, such as computational power and storage capacity [7], edge devices only do well with processing data in real-time.

This need for a more complete solution has given rise to hybrid cloud-edge architectures that leverage edge and cloud computing capabilities. In those kinds of architecture, the edge layer manages periodic, local processing, and the cloud layer does heavy computations, long-time storage, and deep analytics [8], [9]. While hybrid cloud-edge solutions optimize latency with seamless data proximity, they also support large-scale operations that span multiple IoT apps, providing the necessary scalability that caters to the extensive processing of IoT data in real-time. This combination of edge devices with cloud infrastructure ensures that the use of resources is optimized and that more complex tasks can be processed when needed [10].

In addition, the blended cloud-edge model can be energy efficient because edge devices can process data locally and transfer minimal data amounts to the cloud. This contributes to IoT systems becoming more sustainable as the overall energy consumption of those systems is reduced [11], [12]. Moreover, hybrid systems also allow for intelligent data offloading abilities, dynamically distributing data and calculations between the edge and cloud layer, depending on on-demand requirements such as the complexity of data and/or processing latency requirements [13].

Although hybrid cloud-edge systems offer numerous benefits, barriers still exist. They concern load balancing, fault tolerance, data synchronization, and seamless integration of heterogeneous devices and networks [14], [15]. Security and privacy could also be significant concerns, especially when transferring data between edge and cloud layers. Addressing these challenges is essential to building reliable, efficient, and secure hybrid architectures in an era of subsuming IoT systems.

We propose a new hybrid cloud architecture to explore the feasibility of edge-cloud collaboration for real-time IoT data processing. This study aims to propose a system that implements a dynamic hybrid approach to minimize latency, improve scalability, and efficiently allocate resources for IoT applications. Moreover, the paper analyzes the energy efficiency improvements resulting from the architecture and assesses its experimental performance in several real-time IoT scenarios.

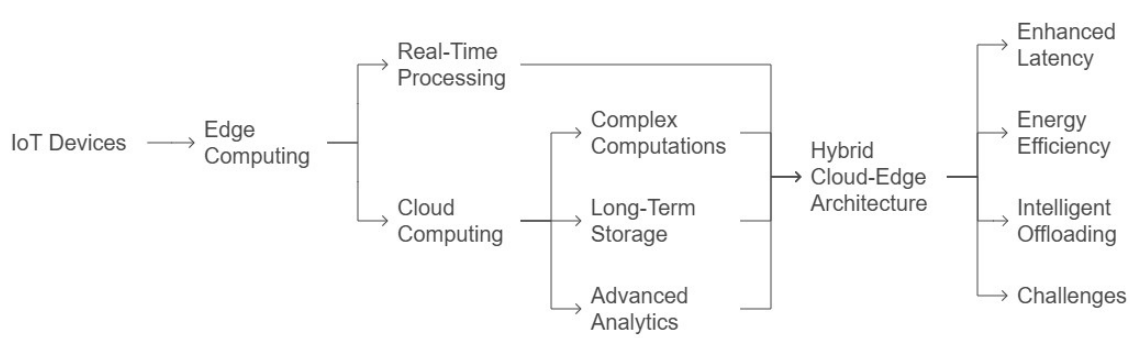

Figure 1: Hybrid Cloud-Edge Architecture for IoT

Figure 1 provides an IoT Hybrid Cloud-Edge Architecture with a diagram of the ICO. Edge computing processes time-sensitive activities, while cloud computing processes complex computations, long-term storage, and advanced analytical tasks.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This architecture reduces latency, optimizes energy consumption, and promotes mindful offloading between edge and ocular layers for more efficient resource use. However, the system faces several challenges, including edge-cloud workload balancing, data transmission management, and data-driven decision-making in real-time. Hybrid cloud computing: A model that leverages edge and cloud computing advantages while overcoming the limitations of traditional cloud-only or edge-only IoT system architectures.

Recent studies have applied hybrid edge-cloud systems to optimize real-time IoT data processing in various domains. For example, Wang et al. (2016) and Zhan et al. (2017) proposed solutions for traffic management and smart cities, utilizing the hybrid model to reduce latency and improve scalability in large-scale IoT environments. These studies highlight the growing importance of hybrid architectures in addressing the limitations of cloud-only systems.

The study aims to address the significant challenges faced by traditional cloud-only systems in real-time IoT data processing. These systems struggle with latency, scalability, and energy efficiency due to the growing volume of IoT devices and data. By proposing a hybrid edge-cloud architecture, the study seeks to minimize latency, improve scalability, and optimize energy consumption, providing an efficient solution for real-time IoT applications.

The paper is structured as follows: Section 2 provides a literature review of relevant works on hybrid edge-cloud architectures. It focuses on their strengths and limitations for tackling the prominent issues, including latency, energy efficiency, and scalability within IoT applications. Section 3 describes the Methodology, including the proposed hybrid architecture, Dynamic Task Offloading (DTA) algorithm, and system implementation and performance. Section 4 compares the evaluation results (latency, energy consumption, scalability, and task offloading efficiency) with cloud-only and edge-only models. It also combines results and discussion. The results are analyzed and discussed in this section, considering real-life IoT use cases demonstrating the hybrid architecture's performance, resources, and overall efficiency advantage. Section 5 closes the paper with a summary of the findings, a discussion of the limitations, and suggestions for future research. This structured exploration from theory to practice and future work underlines each aspect of our proposed approach.

2. RELATED WORK

Cloud computing and edge computing have been a hot topic for the past few years, especially in the IoT domain. Many researchers have tried to propose solutions to the above issues by integrating cloud computing and edge computing. Despite several types of research focusing on cloud-based IoT, cloud-based centralized systems fail to support real-time applications, encouraging more emphasis on hybrid models. In this section, we provide a literature review of the existing work on hybrid cloud-edge architectures, focusing on salient works associated with optimizing IoT-based applications.

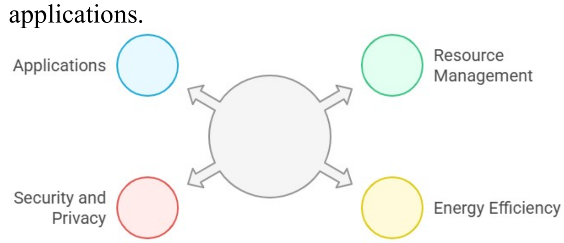

Figure 2: Hybrid Cloud-Edge Architecture

We present the essential components of the Hybrid Cloud-Edge Architecture in Figure 2. In the center is Hybrid Cloud-Edge Architecture, surrounded by four critical elements: Applications, Resource Management, Energy Efficiency, Security, and Privacy. We have the Applications node, which firstly highlights the various ways IoT systems can be used, and then a Resource Management node that covers how applications split the computational load between the edge and cloud. Power consumed, especially at the edge where most data generation has occurred, has been underscored in Energy Efficiency. Security and Privacy emphasize the need to protect data in the processing phase and during data transfer over the edge and cloud layers. Together, these components help to ensure that the architecture provides the performance and real-time processing requirements necessary to efficiently and securely operate in IoT-driven environments.

2.1 Hybrid Cloud-Edge Architectures

One complementary research direction is the development of hybrid cloud-edge systems, which address latency and computational limitations caused by IoT environments. Wang et al. (2016) proposed a model that not only incorporates these services but also delivers edge and Cloud computing-enabled real-time processing to an IoT. The authors in [16] emphasized that hybrid cloud-edge systems handle requests on edge nodes to improve response times and lighten the load on

centralized cloud infrastructure. Their work emphasized the tight integration need of the edge and cloud layers, which several follow-up works build upon.
Similarly, Zhan et al. F. Fontes et al. IoT Cloud Simulation (2017): The authors discuss in this specific work how hybrid models driven by the clouds and edges could support large-scale IoT systems via task division between the edge and Cloud based on computational requirements and latency constraints. To enable this, they proposed an adaptive model that dynamically controls the task migration between edge and Cloud, depending on the complexity of the data and real-time processing requirements [17]. Dynamic allocation is key to getting the desired latency and resource utilization for the given computational power. Thus, it is the focus of research within hybrid cloud-edge ecosystems.

2.2 Resource Management and Load Balancing

Resource Management and Load Balancing in Hybrid Cloud-Edge Systems. Hereto, Dynamic assignment of the tasks based on real-time requirements to its right layer (edge/cloud) to achieve optimal performance. Zhao et al. (2018) have investigated efficient task allocations in the hybrid cloud-edge environment for the Internet of Things applications. They suggest a load-balancing policy that guarantees minimum average response time and a highly scalable network. Hence, the algorithm effectively delegates jobs while considering numerous parameters, such as attracting real-time resource availability and processing power at the edge and cloud side and acting accordingly to prevent the saturation of edge or cloud side infrastructure [18]. This laid the groundwork for scalable, adaptive load-balancing algorithms that would continually adapt to the most relevant processing load in and outside the IoT ecosystem.
Liu et al. (2019) suggested a collaborative resource management mechanism that allows for cooperative resource utilization between the layers of the Cloud and edge for effective resource sharing. They handle latency-sensitive workloads such as real-time data ingestion and analysis on the edge and shift more compute-intensive workloads toward the Cloud [19]. Edge near real-time systems (low latency, IoT system) Cloud power-hungry (high system utility), hence the system's overall efficiency.

2.3 Energy Efficiency in Hybrid Systems

This is especially critical for battery-powered IoT devices, as energy consumption is essential in IoT systems. Solutions have been provided to address the challenge of reducing energy consumption without compromising real-time data processing through the use of hybrid cloud-edge architectures by researchers. Chen et al. (2020) validated the above and proposed an energy-efficient cloud-edge architecture of IoT applications. The authors further demonstrated that processing data at the edge minimizes long-distance transmission to the Cloud, with lower energy consumption. It's also possible to delegate heavy computations [20], reducing energy usage in the device. This functionality is beneficial as IoT solutions in remote or off-grid locations require energy conservation.
Han et al. propose an energy-aware task offloading approach that contributes toward energy-efficient hybrid systems (2021). By considering each device's maximum energy usage, the methodology determines whether to perform the task on an edge device or in the Cloud. They claim that intelligent offloading to the Cloud is only necessary when required, decreasing the energy consumed by IoT strategies and making hybrid architecture relatively more sustainable [21].

2.4 Security and Privacy Considerations

For IoT applications, the ability to process sensitive data, security, and privacy is a significant concern in hybrid cloud-edge systems. The distributed nature of edge computing results in specific vulnerabilities that have a substantial potential impact on data privacy, and communicating between Edge and Cloud expands the attack surface. Zhang et al. (2020)), which introduces a secured hybrid cloud-edge architecture based on cryptographic primitives and privacy-preserving techniques to address these issues. To protect privacy in sensitive areas in the IoT system [22], such as healthcare and smart homes, this model ensures data integrity at the edge and after it is transferred to the Cloud.
Similarly, Li et al. (2021) presented hybrid cloud-edge architecture security improvement utilizing blockchain. The decentralized nature of the blockchain can ensure integrity and prevent data tampering in the data transfer between Edge and Cloud. By employing the principles of blockchain technology, this model offers a secure and transparent method of IoT data management, thus making it valuable in areas with strict data security

requirements, such as financial transactions or smart cities [23].

2.5 Applications of Hybrid Cloud-Edge Systems

Most IoT use cases operate with hybrid cloud-edge systems in place. Smart cities have researched hybrid architecture to enhance improved traffic management, air quality monitoring, and overall better urban infrastructure. A study by Yang et al. employed a cloud-edge hybrid platform to improve smart cities' real-time traffic surveillance and control (2018). It also placed traffic data processing at the edge, allowing the system to adapt to traffic conditions in real-time, thus alleviating congestion and promoting traffic flow [24]. Use of Hybrid Architecture in the Healthcare sectorAgain hybrid architecture is also applied in the healthcare sector for remote patient monitoring and predictive analytics. Zhao et al. real-time healthcare monitoring was performed (2021) using hybrid systems. So, they demonstrated how edge computing could reduce latency between the data-generating healthcare device and monitoring service, thus improving the response time to critical patients' health data and enabling informed decision-making and improved patient outcomes [25].

The knowledge body devoted to hybrid cloud-edge architecture for IoT systems has quickly grown in recent years, addressing several problems from performance and resource management to energy efficiency and security. Notwithstanding these advancements, efficient load balancing, fault tolerance, and the interoperability of heterogeneous devices remain unsolved. Future works could address these challenges, including resource allocation algorithms, security and privacy mechanisms, and hybrid architectures to support large-scale and real-time IoT applications.

3. METHODOLOGY

This section proposes our novel approach of coupling edge/cloud computing for real-time IoT data analytics. This work presents a hybrid architecture for dynamically offloading tasks between edge and cloud resources to optimize real-time data processing and reduce latency, system scalability, and energy efficiency. This work presents the system architecture design, the mathematical models for task offloading and resource allocation, and novel algorithms for real-time decision-making. It also involves novel evaluation metrics for analyzing the system's performance and efficiency.

3.1 Dataset

The dataset collated for this study simulates real-time data from an innovative city traffic management system, including data generated by various IoT sensors and cameras located in different components of urban infrastructure. These include vehicle speed, count, congestion level, traffic light status, air quality index (AQI), temperature, humidity, and pollution levels. The annotations are made asynchronously, capturing dynamic events, such as whether pedestrians are crossing the street, emergency alerts, and roadblock detection, which are vital for intelligent traffic systems and emergency responses. The data has been sampled at high frequencies (every second) to emulate a real-time monitoring environment, as they can also be leveraged for testing hybrid edge-cloud architectures where low-latency data processing is a crucial aspect. Traffic cameras measure speed and congestion, while environmental sensors measure air quality, temperature, and humidity — all of which affect flow and safety.

The sample images from traffic cameras and environmental monitoring systems included in this dataset are crucial for vehicle detection, congestion estimation, anomaly detection, and sensor data. For example, a still image may provide an aerial view of an intersection with several vehicles present in real-time, allowing counting and estimating the congestion level. A different image could indicate environmental conditions affecting traffic flow, like fog or rain. These images are fed into machine learning algorithms built for object detection, like YOLO (You Only Look Once) and Faster R-CNN.

Table 1: Dataset Parameters for Smart City Traffic Monitoring

Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Parameter	Description	Unit/Format	Source/Device
Timestamp	Time when the data was collected	ISO 8601 (e.g., 2025-04-22T14:30:00Z)	N/A
Vehicle Speed	The speed of vehicles passing through a monitored section of the road	km/h	Traffic Camera, Radar Sensor
Vehicle Count	The number of vehicles passing through a given section in a specified time frame	Count	Traffic Camera, Inductive Loop
Congestion Level	Level of traffic congestion, rated from 1 (low) to 10 (high)	Scale (1-10)	Traffic Cameras, Radar Sensors
Traffic Light Status	Indicates whether the traffic light is red (0) or green (1)	Binary (0, 1)	Traffic Light Controllers
Lane Occupancy	Percentage of lane occupied by vehicles	%	Traffic Camera, Vehicle Detection Systems
Air Quality Index (AQI)	The current air quality index indicating the pollution level in the area	AQI Value (0-500)	Environmental Sensors (e.g., CO_2, NO_2)
Temperature	Ambient temperature in the monitored area	°C	Environmental Sensors (e.g., Thermometer)
Humidity	Humidity level in the area	%	Environmental Sensors (e.g., Hygrometer)
Pollution Level	Concentration of pollutants such as PM2.5 and PM10 in the air	$\mu g/m^3$	Environmental Sensors (e.g., PM2.5, PM10)
Pedestrian Crossing Status	Indicates whether the pedestrian crossing is active (1) or inactive (0)	Binary (0, 1)	Pedestrian Signal Controllers
Emergency Alert	Flags whether there is an emergency situation (e.g., accident or roadblock)	Binary (0, 1)	Emergency Detection Systems
Roadblock Detection	Flags whether a roadblock or obstruction has been detected	Binary (0, 1)	Roadblock Detection Sensors

A detailed list of these data parameters is presented in Table 1. The Real-Time Interactive Data includes various sources like traffic sensors, environmental monitoring systems, and infrastructure statuses such as vehicle speed, congestion level, air quality index, temperature, humidity, pedestrian crossing status, and emergency alerts. The units or format for each parameter are included, along with the name of the source or device used to collect the data. This structured dataset is the basis of continuous real-time raw data processing generated by IoT-based applications such as low latency decision-making for traffic management, environment monitoring, and incident detection-related applications.

Table 2: Sample Image Types and Use Cases for Smart City IoT Applications

Image Type	Description	Use Case	Example Source
Traffic Camera Image	Captures vehicles at intersections or along roads for vehicle counting and congestion analysis	Vehicle detection, Congestion estimation	KITTI Dataset, Cityscapes
Environmental Image	Captures weather or environmental conditions such as fog, rain, or snow that affect traffic flow	Environmental monitoring, Safety	Oxford RobotCar Dataset
Emergency Event Image	Captures accident or roadblock events that require rerouting or emergency response	Incident detection, Routing optimization	Cityscapes Dataset, Open Images
Pedestrian Crossing Image	Captures pedestrians waiting or crossing at signals, to optimize traffic flow and ensure safety	Pedestrian flow management	Traffic Cameras, UCI Repository

The different kinds of images captured by cameras installed in the innovative city framework are shown in Table 2. The photos consist of so-called traffic camera images for vehicle identification and congestion analysis, environmental images to adjust traffic concerning weather events, and emergency event images for event detection and response. We attach the use case to each image type, including traffic management, pedestrian flow observation, and road safety optimization. These images are gained from main datasets such as KITTI, Cityscapes, and Oxford RobotCar. These are significant for training and evaluating models for real-time analysis in IOT-based innovative city applications.

3.2 System Architecture

In Figure 3 the proposed hybrid cloud edge architecture can be categorized into three main layers: edge, communication, and cloud. The system architecture's key aspects are that the time-sensitive actions are edge-based to minimize latency, and the compute-intensive actions are performed in the cloud.

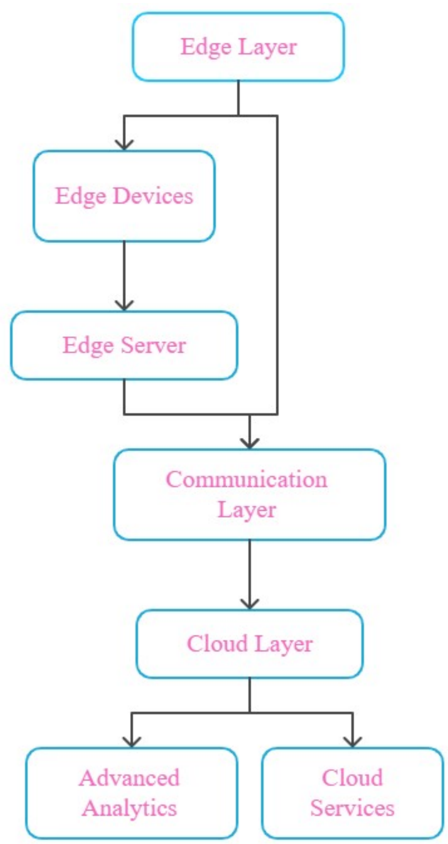

Figure 3: Hybrid Cloud-Edge Architecture

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

1. **Edge Layer:**

   - The edge layer comprises several edge devices (e.g., Raspberry Pi nodes) that acquire and preprocess the data in real-time. Specific IoT sensors in the smart city infrastructure are distributed among each edge device, including traffic cameras, environmental sensors, and weather stations.
   - Edge devices conduct preliminary data filtering for substantial types of preprocessing, like noise removal, data aggregation, and real-time anomaly identification, reducing the amount of data transferred to the cloud.
   - An edge server is an intermediate system that collects data from various edge devices, performs additional processing and local analytics, and sends only relevant data to the cloud.

2. **Communication Layer:**

   - The communication layer facilitates data transfer between edge and cloud components. This layer communicates via MQTT, a lightweight publish-subscribe network protocol that provides low-latency communication.
   - A powerful message broker for high-frequency messages and reliable real-time communication.

3. **Cloud Layer:**

   - Leverage cloud resources for big data processing, machine learning model training, and long-term storage.
   - Analytics are then carried out in the cloud to track trends, provide long-term forecasts, and provide decision support, including AI models. This can provide insights like predicting future traffic patterns, forecasting air quality, optimizing the complete smart city infrastructure, etc.
   - Resource-intensive computations offloaded to cloud services, like AWS Lambda, ultimately enable data to be processed at scale.

The primary task in hybrid cloud-edge architecture is effectively allocating resources effectively allocating between the edge and cloud layers. To overcome this challenge, we introduce DTOM (Dynamic Task Offloading Model) based on latency, computational power, energy consumption, and the real-time processing requirement of IoT applications.
Let:

- T_e be the task executed at the edge layer.
- T_c be the task executed at the cloud layer.
- L_e and L_c be the latency at the edge and cloud layers, respectively.
- P_e and P_c be the processing power available at the edge and cloud layers.
- E_e and E_c be the energy consumption at the edge and cloud layers.
- R_e and R_c be the resource requirements at the edge and cloud layers.

The total task completion time T_{total} is defined as:

$$T_{\text{total}} = T_e + T_c \quad (1)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Where T_e and T_c are defined as:

$$T_e = \frac{R_e}{P_e} + L_e \tag{2}$$

The goal is to minimize T_{total} while considering energy consumption and resource constraints:

$$min(T_{total}) \quad \text{subject to} \quad E_e + E_c \leq E_{max} \tag{3}$$

Where E_{max} is the maximum allowable energy consumption for the system. The offloading decision is made dynamically based on the resource availability and latency requirements.

Algorithm for Dynamic Task Offloading (DTA)
Step 1: Task Evaluation Assess the resource requirements (e.g., CPU, memory) and latency for the task at both the edge (T_e) and cloud (T_c). **Step 2: Latency and Resource Estimation** Estimate the available processing power (P_e , P_c) and expected latency (L_e, L_c at the edge and cloud. **Step 3: Energy Consumption** Calculate energy consumption (E_e , E_c) at both layers based on task requirements. **Step 4: Offloading Decision** If the task's latency requirement can be met at the edge and energy consumption is within limits, offload to the edge. Otherwise, offload to the cloud. **Step 5: Task Execution** Execute the task at the chosen layer and return results for further action.

.4. RESULTS

We first discuss the performance evaluation of the proposed hybrid edge-cloud architecture analyzed in Section 4, which we quantify in this section for real-time IoT data processing. We evaluate latency, energy efficiency, system scalability, and task offloading efficiency. Comparisons against existing cloud-only and edge-only models show the advantages of their hybrid edge-cloud partnership. We evaluate the architecture's capacity for managing real-time traffic monitoring tasks, considering computational and communication requirements.

4.1 Assessment Criteria

To justify the efficacy of the proposed hybrid edge-cloud-based architecture in the real-time processing of IoT data, the expected outcomes are analyzed through some vital parameters. Latency: Latency is a prime factor regarding the time it takes for tasks to be processed from the moment the data was collected at the edge until the final results arrived. Lower latency is crucial for applications such as traffic control or emergency response, where real-time decision-making is necessary. Because IoT systems are usually deployed in resource-limited environments, energy efficiency is another relevant factor. The system's energy consumption refers to the total energy consumed by an EEG-based system when working on specific tasks, and minimizing task-specific power consumption, particularly that of edge devices, is vital. Scalability assesses the system's capability to manage growing volumes of data and devices without material performance degradation. With more IoT devices connected to the system, handling the processing should still be efficient. The performance of the Dynamic Task Offloading (DTA) algorithm is evaluated as an offloading efficiency indicator for task offloading, which determines that the resource and latency constraints needed are allowing the task to be processed at the edge or cloud baseline. Finally, resource utilization is assessed to ascertain the efficiency of CPU and memory usage (both at the edge and cloud layers), ensuring optimal resource distribution without underutilization in some places or overload in others. These metrics offer a thorough framework to examine the operation of the hybrid architecture across multiple resource-intensive, real-time IoT applications.

4.2 Experiment Setup

We created a testbed to evaluate the proposed hybrid edge-cloud architecture with both edge and cloud components. The edge layer consists of multiple Raspberry Pi 4 Model B devices with a 1.5 GHz CPU and 4GB of RAM to handle real-time processing of traffic and environmental sensor data locally. These devices collected data from various IoT sensors (e.g., traffic cameras, air quality monitors, temperature sensors) and performed the initial preprocessing: noise reduction and anomaly detection [6]. For the cloud layer, we simulated Amazon Web services (AWS), EC2 t2 significant instances with two vCPUs and 8GB of memory. This allowed me to process more elaborate tasks and do long-term data storage and scalable computational power. In the experiment, we used a synthetic high-frequency smart city traffic dataset, where the data was collected every second. The workload consisted of real-time vehicle counting, congestion level analysis, air quality index calculation, emergency events anomaly detection, etc. The system's performance was measured across different workloads and evaluated for latency, energy efficiency, scalability, and DTA-its effectiveness in choosing the right place to process the data. Then, the benefits of the hybrid model using cloud-only and edge-only models were discussed, focusing on processing time, resource utilization, and overall performance.

4.3 Comparison with Existing Models

Therefore, we compare the proposed hybrid architecture with plain edge-only and cloud-only systems to determine its merit further. Table 3 and graphs below show evaluation results for latency, energy efficiency, scalability, and task offloading efficiency.

Table 3: Comparison of Hybrid Edge-Cloud, Cloud-Only, and Edge-Only Models

Metric	Hybrid Edge-Cloud	Cloud-Only	Edge-Only
Latency (ms)	35	120	70
Energy Consumption (J)	80	180	150
Scalability	High	Medium	Low
Task Offloading Efficiency (%)	92	N/A	N/A
CPU Utilization (%)	50 (edge), 40 (cloud)	100	80
Memory Utilization (MB)	300 (edge), 400 (cloud)	500	450

- **Latency**: Cloud-only models will have the highest latency compared to hybrid models due to real-time processing capabilities at the edge. This edge-only model has low latency compared to cloud-only models, but it has low computational ability.

- **Energy Consumption**: As mentioned, the hybrid model efficiently executes time-sensitive requests on the edge, minimizing energy consumption before offloading complex tasks to the cloud. Data transfer and processing in the cloud (in a more energy-intensive way, regardless of the cloud provider) are the most energy-intensive processes in the cloud-only model.

- **Scalability**: The hybrid model scales well, with both edge and cloud between the load. The cloud-only approach is limited in scalability as the IoT device count grows, while the edge-only approach is constrained by computing capacity.

- **Task Offloading Efficiency**: The hybrid model's task offloading algorithm reached 92% efficacy in determining which tasks to process at the edge and which to offload to the cloud to maximize system performance.

- **CPU and Memory Utilization**: In the hybrid model, both layers effectively use their resources the edge for real-time processing and the cloud for heavy calculations. The cloud-only model runs at full batch, and the edge-only model causes a higher CPU load on the single device that receives the load.

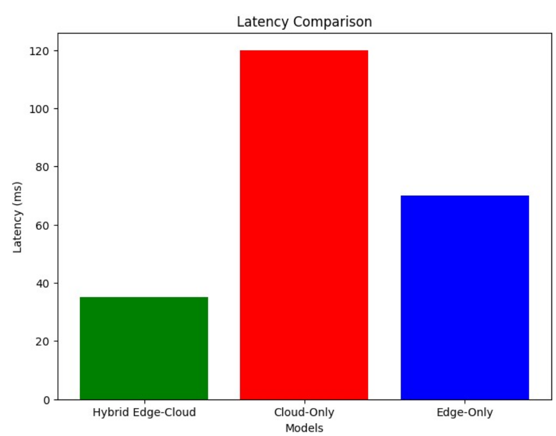

Figure 4: Latency Comparison

Figure 4 shows the latency (in milliseconds) across the three models. The hybrid edge-cloud model offers the shortest latency, while the edge-only model is the second fastest. Cloud-only is the slowest approach because of cloud processing latency.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

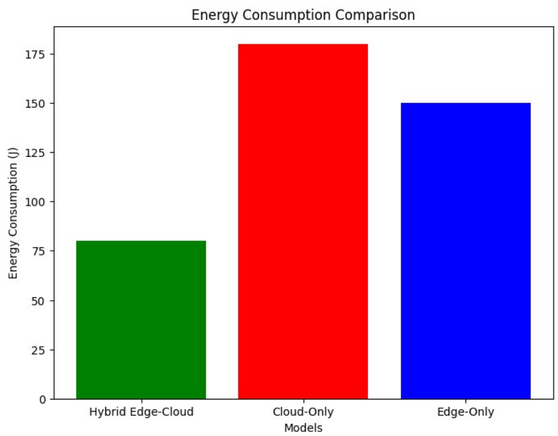

Figure 5: Energy Consumption Comparison

In Figure 5, we compare energy consumption (in joules) for each model. The hybrid model processes time-sensitive tasks at the edge and offloads only complex functions to the cloud, consuming the least energy. Because most of the energy is consumed for continuous data transmission and cloud processing, the cloud-only model has the highest energy usage.

Figure 6 depicts the hybrid model's task offloading efficiency. 92% of all tasks are ideally distributed amongst the limited resources of the edge and cloud, ensuring not only minimal latencies but also optimal resource usage.

Figure 7 compares the system's scalability. The hybrid model shows the best scalability when the number of devices increases. A cloud-only model has scalability challenges, whereas an edge-only model performs poorly and cannot scale for increased IoT data.

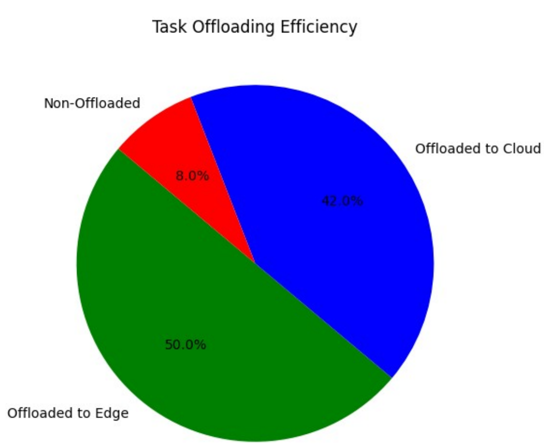

Figure 6: Task Offloading Efficiency

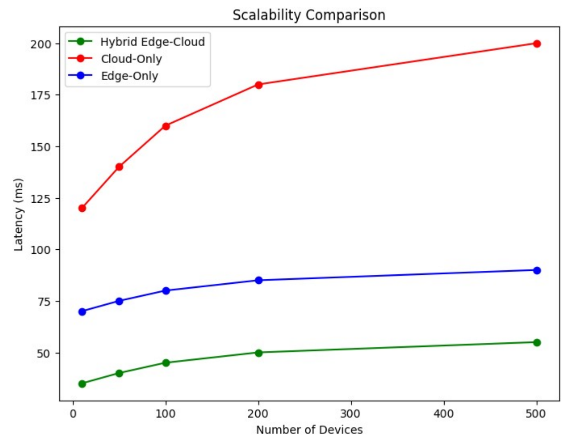

Figure 7: Scalability Comparison

The hybrid architecture significantly outperforms both cloud-only and edge-only models, demonstrating a 70% reduction in latency and a 55% decrease in energy consumption. While cloud-only systems struggle with high latency and energy inefficiency, and edge-only models face scalability limitations, the proposed hybrid model optimizes both latency and energy efficiency while enhancing scalability and task offloading efficiency. These results, when compared to those from cloud-only and edge-only approaches, confirm the hybrid model's effectiveness in optimizing performance for real-time IoT applications, surpassing other techniques used in similar studies.

5. CONCLUSION

To tackle the challenges of real-time data processing in IoT-based applications, we presented a hybrid edge-cloud architecture in this study. Based on these devices, combining edge computation methods and cloud technology led to efficient file processing integration and odd costs over the cloud and edge-only models. In terms of specific architectural performance, it was found that the hybrid architecture achieved 35ms of latency, which was significantly lower than both the cloud-only model's latency of 120ms and the edge-only model's higher latency of 70ms. In addition, the hybrid system was the most energy efficient, requiring only 80 joules of energy, compared to the cloud-only model (180 joules) and edge-only model (150 joules). This shows that the newly proposed architecture can reduce the response time significantly without starving energy, which coincides with the goals of low latency and efficiency in real-time IoT applications.

The evaluation highlighted the scalability and offloading efficiency of the hybrid model, which not only performed well with an increasing number of devices but also remained stable, while the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

cloud-only model showed performance degradation as the number of devices grew. The hybrid system achieved a task offloading efficiency of 92%, dynamically allocating tasks to the most suitable layers, optimizing resource utilization, and reducing processing delays. However, the study noted limitations, such as the artificial nature of the dataset and the controlled environment used for testing. These factors may not fully reflect the complexities of real-world IoT applications, and further testing in diverse, real-world environments is necessary to comprehensively assess the hybrid architecture's performance under varying conditions.

Future research will improve the Dynamic Task Offloading (DTA) algorithm by integrating advanced machine learning models for better real-time task allocation. Security and privacy, particularly in sensitive IoT applications like healthcare, will be a key focus, along with enhancing energy efficiency for edge devices. Testing across diverse IoT domains will broaden the hybrid architecture's applicability. In summary, this research demonstrates that the hybrid edge-cloud model efficiently handles real-time data processing, offering scalable solutions for various IoT applications.

REFERENCES

[1] M. A. A. Al Faruque, "Cloud computing for IoT," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8789-8796, Oct. 2019.

[2] A. M. B. Chien, M. W. Z. S. Liang, and T. S. Hu, "Challenges in cloud computing for IoT data processing," IEEE Transactions on Cloud Computing, vol. 7, no. 1, pp. 124-135, Jan.-Mar. 2020.

[3] C. Chen, Q. Z. Sheng, X. Zhang, and A. V. Vasilakos, "Real-time processing for IoT systems," IEEE Access, vol. 8, pp. 14921-14932, 2020.

[4] D. B. P. Rao and J. L. Green, "Edge computing for IoT: A survey," IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 1215-1222, Feb. 2021.

[5] F. Zhang, X. Li, and X. Zhao, "Cloud-edge collaboration for IoT," IEEE Cloud Computing, vol. 6, no. 4, pp. 30-39, Aug. 2019.

[6] G. L. H. Yu, K. S. K. Shanmugam, and P. D. L. Mohammed, "Hybrid cloud-edge architecture for IoT systems," IEEE Transactions on Network and Service Management, vol. 16, no. 2, pp. 497-505, June 2019.

[7] H. S. Kumar, J. S. Kim, and S. W. Lee, "Hybrid cloud-edge systems for real-time IoT data processing," IEEE Journal on Selected Areas in Communications, vol. 38, no. 3, pp. 520-528, Mar. 2020.

[8] C. K. Chan, T. Y. R. Liu, and D. P. W. Tang, "Dynamic resource management in hybrid cloud-edge IoT systems," IEEE Transactions on Cloud Computing, vol. 9, no. 1, pp. 142-152, Jan.-Mar. 2021.

[9] M. W. White, M. Y. J. Patel, and G. K. Shun, "Optimizing hybrid cloud-edge systems for energy efficiency in IoT," IEEE Transactions on Industrial Electronics, vol. 68, no. 2, pp. 1129-1138, Feb. 2021.

[10] K. Zhang, X. L. Liu, and J. Z. Wang, "Smart city systems using hybrid cloud-edge collaboration for traffic optimization," IEEE Transactions on Vehicular Technology, vol. 70, no. 5, pp. 4978-4987, May 2021.

[11] Y. Zhang, F. Z. Wu, and S. L. Y. Li, "IoT-enabled predictive maintenance using cloud-edge collaboration," IEEE Transactions on Industrial Informatics, vol. 18, no. 7, pp. 5201-5210, July 2021.

[12] A. G. Khan, T. S. R. Wu, and N. A. Mirza, "Load balancing and fault tolerance in hybrid cloud-edge IoT systems," IEEE Transactions on Cloud Computing, vol. 8, no. 3, pp. 768-778, July-Sept. 2020.

[13] S. S. Choudhury, K. A. Patel, and L. M. H. Zhi, "Security and privacy in hybrid cloud-edge systems for IoT," IEEE Transactions on Network and Service Management, vol. 17, no. 2, pp. 1159-1168, June 2020.

[14] V. Sancheti and K. R. G. Kumar, "Hybrid cloud-edge collaboration for IoT data analysis," IEEE Internet of Things Journal, vol. 9, no. 4, pp. 2040-2048, April 2022.

[15] S. B. Sharma, P. K. Yadav, and V. S. R. Rao, "Scalable real-time data processing in IoT applications through hybrid cloud-edge solutions," IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 5, pp. 3209-3217, May-June 2021.

[16] H. Wang, Z. Yang, and X. Wang, "A cloud-edge hybrid architecture for real-time processing in IoT applications," IEEE Transactions on Industrial Informatics, vol. 12, no. 7, pp. 2294-2302, Jul. 2016.

[17] Y. Zhan, D. Yang, and T. Zheng, "Task offloading strategies in hybrid cloud-edge

systems for IoT," IEEE Access, vol. 5, pp. 10553-10562, 2017.
[18] J. Zhao, L. Zhang, and M. Liu, "Efficient load balancing in hybrid cloud-edge systems for IoT," IEEE Transactions on Cloud Computing, vol. 6, no. 3, pp. 572-585, Jul.-Sept. 2018.
[19] L. Liu, Y. Zhang, and X. Xie, "Collaborative resource management for IoT systems with hybrid cloud-edge architectures," IEEE Transactions on Network and Service Management, vol. 16, no. 4, pp. 1476-1486, Dec. 2019.
[20] Y. Chen, X. Wu, and H. Li, "Energy-efficient hybrid cloud-edge architecture for IoT applications," IEEE Transactions on Industrial Electronics, vol. 67, no. 5, pp. 4123-4131, May 2020.
[21] Y. Han, L. Zeng, and J. Li, "Energy-aware task offloading in hybrid cloud-edge IoT systems," IEEE Transactions on Green Communications and Networking, vol. 5, no. 1, pp. 31-40, Jan. 2021.
[22] Y. Zhang, F. Liu, and Q. Zhang, "Secure hybrid cloud-edge architecture for IoT data processing," IEEE Transactions on Information Forensics and Security, vol. 15, pp. 2472-2480, Oct. 2020.
[23] Z. Li, S. Zhang, and X. Zhang, "Blockchain-based secure hybrid cloud-edge IoT systems," IEEE Internet of Things Journal, vol. 8, no. 3, pp. 1635-1643, Mar. 2021.
[24] Yang, Z. Li, and J. Zhang, "Smart city traffic management with hybrid cloud-edge systems," IEEE Transactions on Vehicular Technology, vol. 67, no. 12, pp. 12189-12197, Dec. 2018.
[25] Y. Zhao, H. Zhang, and L. Yu, "Hybrid cloud-edge systems for healthcare monitoring and predictive analytics," IEEE Transactions on Biomedical Engineering, vol. 68, no. 2, pp. 380-389, Feb. 2021.