Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
January-June 2023 4(1), 4497–4503

EDGE COMPUTING FOR REAL-TIME DATA PROCESSING

Nikhat Raza Khan [1], Gulfishan Firdose Ahmed [2], Raju Barskar [3], Ankit Kumar Singh [4]

[1] Professor, Computer Science & Engineering, IES College of Technology, Bhopal, (M.P) India
[2] Assistant Professor Computer Science, JNKVV, College of Agriculture Powarkheda, Narmadapuram (M.P) India
[3] Associate Professor Computer Science & Engineering UIT –RGPV Bhopal (M.P) India
[4] Computer Science & Engineering, IES College Of Technology, Bhopal, (M.P) India

Corresponding Author
Nikhat Raza Khan,
nikhat.raza@gmail.com

DOI
10.29121/shodhkosh.v4.i1.2023.5645

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

With the rise of the Internet of Things (IoT), autonomous systems, and intelligent environments, the necessecity for real-time data processing has increased significantly. Traditional cloud computing models often struggle to meet the strict demands of low latency, high bandwidth, and consistent reliability required by real-time applications such as self-driving vehicles, industrial control systems, remote medical services, and smart city infrastructure. Edge computing offers a powerful solution to these challenges by shifting computational works closer to the data source—positioned at the network's edge—thereby enhancing speed and performance. This study investigates the structure, essential components, and core technologies that form the foundation of edge computing, with particular emphasis on its role in real-time data processing. It highlights how edge computing facilitates faster response times, minimizes the burden of data transfer, and improves privacy and security by reducing the dependence on transmitting sensitive data to centralized cloud systems. Furthermore, We examine various real-world scenarios where edge computing is utilized for real-time decision-making, such as in intelligent transport systems, smart energy grids, and video analysis. Additionally, this paper explores how edge computing is integrated with emerging technologies like 5G, artificial intelligence (AI), and machine learning (ML), which significantly improve its capability to handle data processing efficiently and intelligently. Finally, we address current challenges in deploying edge-based systems, including resource constraints, interoperability, security vulnerabilities, and management complexity. The paper concludes with an outlook on future research directions and the role of edge computing in building scalable, resilient, and intelligent infrastructures for next generation real-time applications.

1. INTRODUCTION
1.1. RESEARCH ISSUE AND IMPORTANCE

Edge computing has become a essential solution for enabling real-time data processing, especially as conventional centralized computing models struggle to meet the performance, responsiveness, and scalability required by today's advanced applications. However, despite its benefits, the widespread adoption and deployment of edge computing still encounter several research challenges that need to be overcome to realize its full potential.

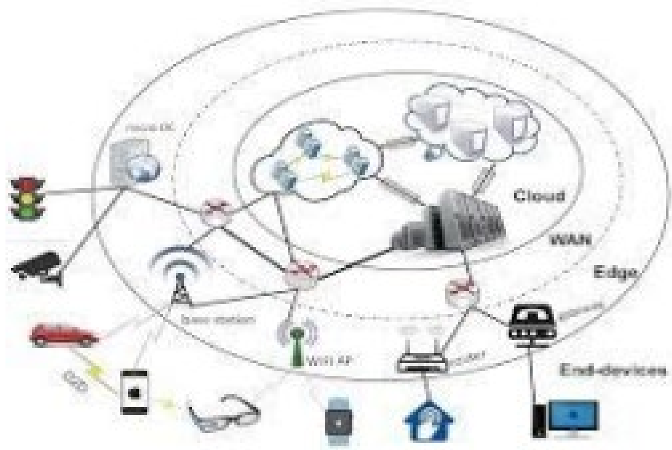
2. RESEARCH ISSUES

1) **Latency and Performance:** Ensuring fast and consistent responses with limited edge resources is a major challenge in real-time applications.

2) **Limited Resources:** Edge devices often have less computing power, storage, and energy, which makes managing and distributing tasks efficiently difficult.
3) **Security and Privacy:** Processing sensitive data locally increases the risk of security breaches. Protecting data on edge devices is an active research area.
4) **Data Compatibility:** Different devices and platforms generate data in various formats, making integration and communication complex.
5) **AI at the Edge:** Running machine learning models on edge devices requires optimization, as full-scale models are too large and slow for real-time use.
6) **Unstable Network Conditions:** Edge systems must handle poor or lost connectivity without affecting performance or losing data.

3. IMPORTANCE

Research in this area is important because edge computing supports many critical and time-sensitive applications. It enables faster responses in autonomous vehicles, real-time health monitoring, industrial automation, smart cities, and disaster response. Solving these issues will make future systems more responsive, secure, and creating a foundation for more intelligent and seamlessly connected systems and environments.



1) Prior Investigations

In recent years, numerous researchers have explored the potential of edge computing to enhance real-time data processing. Their findings indicate that handling data near its point of origin significantly lowers latency and boosts overall system efficiency.For example, some research has focused on using edge computing in smart cities to manage traffic in real-time, while others have applied it in healthcare for faster diagnosis using wearable sensors. Several papers have explored combining edge computing with AI and machine learning to make smart decisions locally without sending all data to the cloud. Other studies have looked into optimizing task scheduling and resource management to make edge systems more efficient. Overall, past research highlights that edge computing can significantly enhance real-time applications, but also points out challenges like limited resources, security concerns, and the need for better frameworks to support large scale deployments.

2) Gaps in Existing Studies

While many studies highlight the benefits of edge computing, several important areas still need more research:

• **Limited Focus on Real-Time Performance:** Most research focuses on general edge computing benefits but lacks in-depth analysis of how to 1. guarantee real time performance under different conditions.

The main goal of this research is to explore how edge computing can improve real-time data processing by reducing latency, enhancing speed, and enabling faster decision-making.

Specific goals include:

1) To analyze the architecture and functions of edge computing systems.
2) To identify the advantages and limitations of using edge computing for real-time applications.

3) To study real-world use cases in areas like healthcare, transportation, and industry.
4) To explore how edge computing can work with AI, 5G, and IoT for better performance.
5) To highlight current challenges and suggest future research directions.

Edge computing significantly improves the speed and efficiency of real-time data processing compared to How does edge computing impact the performance, latency, and reliability of real-time data processing in time sensitive applicati

• **Resource Management Issues:** There is a gap in strategies for efficiently managing limited edge resources like CPU, memory, and power, especially in dynamic environments.

• **Security at the Edge:** Existing solutions do not fully address the unique security and privacy risks at the edge, such as physical attacks or unauthorized data access.

• **Lack of Standardization:** There is no common framework or standard for integrating edge devices and systems, which makes interoperability difficult.

• **Scalability and Deployment:** Few studies explore how edge systems can scale and perform effectively in large, real-world deployments.

• **AI and ML at the Edge:** More work is needed on lightweight AI models that can run efficiently on resource-constrained edge devices in real-time.

3) The Current Study's Contribution

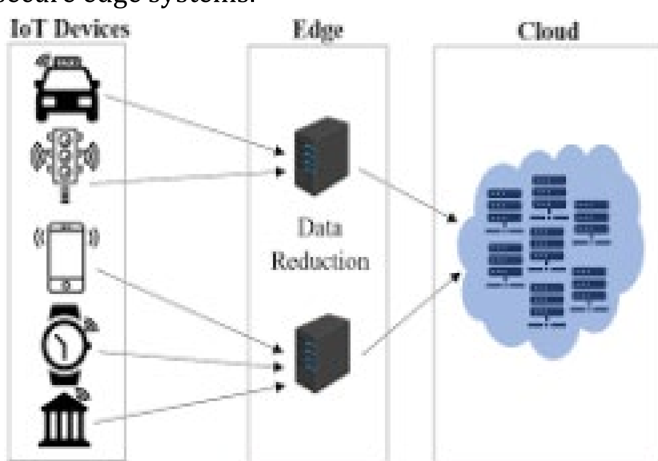
This research pupose to examine how edge computing enhances real-time data processing across different application domains. The key contributions of this study include:

• **Identifying Key Challenges:** This paper highlights the critical issues in edge computing, such as latency, resource management, security, and data integration, specifically for real-time processing.

• **Analyzing Real-World Applications:** The study examines several use cases where edge computing can significantly enhance performance, including

• **Proposing Solutions:** Based on current challenges, the research proposes practical approaches for optimizing edge resources, improving security, and ensuring efficient real-time data processing.

• **Future Directions:** The paper outlines potential future research areas, such as integrating AI with edge computing and developing more scalable and secure edge systems.



4. RESEARCH DESIGN &APPROACH

This research adopts a qualitative and experimental approach to explore the application of edge computing for real-time data processing. The study is divided into three main phases:

1) **Literature Review:** A comprehensive review of existing research on edge computing, real-time data processing, and related technologies (e.g., IoT, AI, and machine learning). This phase helps identify the current challenges, gaps, and opportunities in the field.

2) **System Design and Simulation:** Using simulation tools, we design and implement an edge computing architecture that processes real-time data. This includes setting up edge devices, local processing units, and cloud integration. The design will evaluate key factors such as latency, processing efficiency, and scalability.

3) **Case Studies and Use Case Evaluation:** Real-world case studies will be examined to assess how this study explores the application of edge computing in industries such as healthcare, autonomous vehicles, and smart cities. These case studies serve to demonstrate both the practical advantages and the potential limitations of edge computing in real-time environments. By integrating theoretical insights with practical simulations and real-world examples, the research aims to offer a well-rounded understanding of edge computing's impact on real-time data processing.

4.1. DATA GATHERING TECHNIQUES

In edge computing for real-time data processing, effective data gathering is essential for ensuring accurate analysis and timely decision-making. Below are some common techniques used to gather data in such systems:

1) **Sensor Networks:** Edge devices, such as IoT sensors, collect real-time data from their environment. These sensors measure physical parameters like temperature, humidity, motion, or air quality, and transmit data to edge nodes for immediate processing.
2) **Edge Device Logs:** Devices at the edge often generate logs that contain valuable data about system performance, user interactions, or environmental conditions. These logs can be used to monitor real

time activities and detect anomalies.

3) Streaming Data: Real-time data streams from sources like cameras, microphones, or other video and audio sensors are processed locally on edge devices. This technique is commonly used in applications like surveillance and autonomous vehicles.
4) Crowdsourcing: In some cases, data is collected from multiple users or devices. Crowdsourced data can help gather diverse information for real-time decision making, such as traffic updates or weather monitoring.
5) Cloud-to-Edge Data Transfer: Some data gathering techniques involve transferring aggregated data from cloud systems to edge nodes for localized processing. This hybrid approach allows for efficient data handling in systems that need to process large volumes of information in real time.

4.2. ALGORITHMS AND MODELS

Real-time data processing in edge computing depends on a range of algorithms and models that effectively manage tasks, optimize resource usage, and maintain low-latency performance. These methods are specifically developed to address challenges such as constrained computational resources, rapidly changing conditions, and the demand for instant decision-making.

1) **Task Scheduling Algorithms**

Task scheduling algorithms play a vital role in deciding which tasks are to be executed locally on edge devices and which should be delegated to cloud servers. Some widely used approaches include:

• **Priority-based Scheduling:** Tasks with higher priority (e.g., urgent real-time data) are processed first.

• **Load Balancing:** Distributes tasks evenly across edge nodes to avoid overloading any single device and ensure optimal performance.

• **Deadline-aware Scheduling:** Tasks are scheduled based on their deadlines to guarantee real-time processing

2) **Resource Allocation Models**

These models are used to efficiently allocate computational resources (CPU, memory, storage, and energy) at the edge. Some common approaches include:

• **Dynamic Resource Allocation:** Resources are

allocated based on the current load and available resources at each edge node.

• **Energy-efficient Resource Allocation:** Focuses on minimizing energy consumption while meeting real time processing requirements, especially in battery powered edge devices.

3) Data Offloading Models

Edge computing requires determining which data should be handled locally and which should be forwarded to the cloud. The primary models used for this purpose include:

• **Edge-Cloud Hybrid Model:** A hybrid model that combines local and cloud-based processing, where time-critical data is handled at the edge, while non-urgent information is transmitted to the cloud for further analysis or storage.

• **Fog Computing Model:** An intermediate layer known as fog nodes is introduced between the edge and the cloud, assisting in processing data nearer to its origin and alleviating the burden on edge devices.

4) Machine Learning at the Edge

Machine learning models are used to process and analyze data on edge devices. Edge-specific machine learning approaches include:

• **TinyML:** Optimized machine learning models designed for edge devices with limited computational power.

• **Federated Learning:** A decentralized method in which edge devices work together to train a model without exchanging raw data, thereby preserving privacy while enhancing the model's performance

• Environment: Real-world or simulated environments like smart home, factory, or vehicle.

5) Fault Tolerance Models

Fault tolerance is crucial for real-time applications. These models ensure that edge devices can handle failures or network issues without affecting performance. Techniques include:

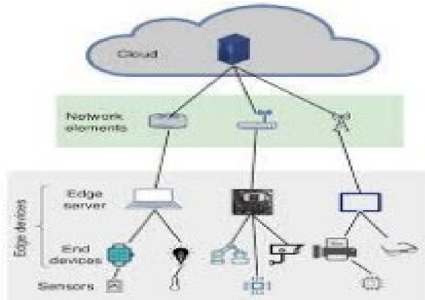
• **Redundancy:** Using backup systems or processes to ensure continuous operation.

• **Error Detection and Correction:** Identifying faults in real-time and implementing corrective measures quickly.

5. JUSTIFICATION OF METHODS

The methods chosen for this research focus on evaluating and optimizing edge computing systems for real-time data processing. To assess the effectiveness of edge computing, a combination of simulation models, experimental prototypes, and performance metrics will be used.

1) **Simulation Models:** Using simulations allows the modeling of edge environments at scale without the need for extensive physical infrastructure. This method helps in understanding the behavior of edge systems under varying network conditions, load, and resource constraints.
2) **Experimental Prototypes: B**uilding small-scale edge computing prototypes enables real-world testing of the edge devices, such as edge servers or IoT devices. This hands-on approach will help to measure performance in real-time scenarios and capture practical issues related to latency, resource management, and security.
3) **Performance Metrics:** Key metrics such as response time, throughput, energy consumption, and resource utilization will be used to evaluate the efficiency of edge computing systems. These metrics are critical for assessing the system's ability to meet the real-time demands of different applications.
4) **Comparison with Cloud Models:** To confirm the benefits of edge computing, it will be compared with traditional cloud-based processing, focusing on metrics such as latency, bandwidth consumption, and processing speed. This comparison aims to emphasize the advantages of edge computing in situations where real-time data handling is essential.



Parameter Settings and Evaluation Metrics To assess the effectiveness of edge computing systems in handling real-time data processing, specific parameter configurations and evaluation metrics will be defined and analyzed, several key parameters and metrics are considered. These help in analyzing how well the system performs under different conditions.

5.1. PARAMETER SETTINGS

1) **Device Type:** Type of edge devices used (e.g., Raspberry Pi, Jetson Nano, industrial gateways).
2) **Network Type:** Wired or wireless connections such as Wi-Fi, 4G, or 5G.
3) **Data Size:** Volume of data being processed at the edge (e.g., in MB or GB).
4) **Processing Load:** Number and complexity of tasks handled by the edge device.
5) **Latency Threshold:** Maximum acceptable delay for processing and responding to data. Deployment

5.2. EVALUATION METRICS

1) **Latency:** Time taken from data generation to response. Lower latency is better for real-time processing.
2) **Throughput:** Amount of data processed in a given time. Higher throughput indicates better performance.
3) **Resource Usage:** CPU, memory, and energy consumed by edge devices. Lower usage means more efficiency.
4) **Accuracy:** For AI-based edge tasks, this measures how correct the outputs are.
5) **Reliability:** System stability over time and under various loads.
6) **Scalability:** Ability to handle more devices or data without performance drop.

Edge computing processes data close to where it is generated. Below are the typical steps involved in real time data processing at the edge:

- **Data Generation**

Sensors, IoT devices, or machines collect raw data (e.g., temperature, video, speed, etc.).

- **Data Transmission to Edge Device**

The collected data is sent to a nearby edge device, such as a gateway, edge server, or local processing unit.

- **Data Preprocessing**

The edge device filters, cleans, or compresses the data to remove noise and reduce its size.

- **Real-Time Analysis**

The edge device runs algorithms or lightweight AI models to analyze the data and make quick decisions (e.g., trigger alerts, control actions).

- **Action or Response**

Based on the analysis, the system performs an action

- **Optional Cloud Sync**

Significant data or outcomes can be transmitted to the cloud for long-term storage, advanced analysis, or generating reports.

6. FINDINGS AND CONCLUSION

This study illustrate that edge computing offers an effectual approach to managing real-time data processing demands. By bringing computation closer to where the data is generated, it reduces latency, enhances processing speed, and reduces the burden on centralized cloud infrastructure. Furthermore, it grant to improved data privacy and enables quicker decision-making, which is particularly valuable in time-sensitive fields such as healthcare, smart city systems, and industrial automation.

However, several challenges remain to be addressed. These include the limited computational capacity of edge devices, potential security vulnerabilities, the complexity of handling diverse data types, and the need to maintain reliable network connectivity. There is also a growing need to design AI models that service efficiently on edge devices.

In conclusion, edge computing is essential for the future of real-time systems. With continued research and technological improvements, it may become more secure, scalable, and effective, making it a key part of next

generation digital infrastructure.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Satyanarayanan, M. (2017). The emergence of edge computing. Computer, 50(1), 30–39. https://doi.org/10.1109/MC.2017.9

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. IEEE Communications Surveys & Tutorials, 19(3), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720

Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. Future Generation Computer Systems, 79, 849–861. https://doi.org/10.1016/j.future.2017.09.020

Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2018). Mobile edge computing: A survey. IEEE Internet of Things Journal, 5(1), 450–465. https://doi.org/10.1109/JIOT.2017.2750180

Chiang, M., & Zhang, T. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of Things Journal, 3(6), 854–864. https://doi.org/10.1109/JIOT.2016.2584538

Premsankar, G., Di Francesco, M., & Taleb, T. (2018). Edge computing for the Internet of Things: A case study. IEEE Internet of Things Journal, 5(2), 1275– 1284. https://doi.org/10.1109/JIOT.2018.280526