An Effective Allocation Scheme for Resources in Edge Computing-enabled Networks with Uncertain Network Status

Dr. Sandip D. Satav[*]
Dr. Shailesh M Hambarde[1]
Dr. Poonam D Lambhate[2]
Mrs. Aarti S Satav[3]
Associate Professor[*], Department of Information Technology
Associate Professor[1], Electronics & Telecommunication Engineering
Professor[2], Computer Engineering
JSPM's Jayawantrao Sawant College of Engineering
Pune 411028, India
Manager[3], SBI, Pune
sandisatav593@gmail.com
[*]**Corresponding Author**

Abstract: In this research, we offer a novel resource allocation strategy intended for networks enabled by edge computing, where the status of the network is frequently unknown. Our method optimizes resource distribution, boosts network efficiency, and guarantees stable performance even in the face of dynamic network conditions by combining machine learning, adaptive resource management, and predictive analytics. The results of our simulation show that, in comparison to conventional allocation techniques, our suggested scheme is effective in improving resource utilization and lowering latency.

Keywords: adaptive management, predictive analytics, machine learning, edge computing, resource allocation, and network uncertainty.

1. INTRODUCTION:

- **1.1 Background:**

Since edge computing can analyze data closer to the source rather than depending on centralized cloud servers, it has become a game-changing technology in modern networking. This method improves real-time data processing, lowers latency dramatically, and boosts network performance in general. The need for effective resource allocation in edge computing environments has increased due to the growth of IoT devices, autonomous systems, and real-time applications. Edge computing has drawbacks despite its benefits, especially in dynamic and erratic network environments. The uncertainties and sudden variations characteristic of these contexts often prove too much for traditional resource allocation systems to handle. This calls for the creation of more resilient and flexible allocation algorithms that can continue to provide the best results even in the face of changing network conditions.

- **1.2 Problem Statement:**

For edge computing-enabled networks to function effectively and provide good quality of service (QoS), resource allocation is essential. The variability of network conditions, resulting from variables like varied latency, fluctuating bandwidth, and intermittent connectivity, presents a major obstacle to the effectiveness of conventional resource management strategies. These unknowns have the potential to undermine edge computing advantages by causing inefficient use of resources, higher latency, and worse performance. Thus, a resource allocation strategy that can maximize resource utilization and dynamically adapt to changing network conditions is desperately needed. Such a system needs to be able to forecast the state of the network, make real-time adjustments to the distribution of resources, and maintain high performance and reliability levels.

- **1.3 Contributions:**

This paper presents a novel resource allocation system designed for networks with edge computing enabled that must function in unpredictable environments. The following are this work's main contributions:

Integration of Predictive Analytics:
We create a predictive analytics module that forecasts network state by using both historical and real-time data. The system may proactively alter resource allocation by anticipating changes thanks to its predictive capability.

Flexible Resource Administration:
Adaptive resource management approaches, which dynamically distribute resources based on the anticipated network circumstances, are incorporated into our suggested scheme. This reduces latency and guarantees optimal resource use.

Optimization through Machine Learning:
We use machine learning methods to improve resource allocation decision-making. These algorithms continuously learn from data on network performance to increase the efficiency of resource distribution and prediction accuracy.

Validation and Simulation:
Extensive simulations are used to validate the proposed technique, showing it to be superior to conventional allocation approaches. We offer a comparative analysis that emphasizes the enhancements in network performance and resource utilization.

The rest of this essay is organized as follows: The work on resource allocation and edge computing is reviewed in Section 2. The assumptions and system model are described in Section 3. The suggested allocation plan, which incorporates machine learning integration, adaptive management, and predictive analytics, is described in depth in Section 4. The setup, outcomes, and discussion of the simulation are presented in Section 5. The report is finally concluded in Section 6 with a summary of the findings and recommendations for further research.

2. RELATED WORK:

- **2.1 Traditional Resource Allocation Methods:**

Extensive research has been conducted on resource allocation in computing networks, with a conventional concentration on centralized cloud settings. Usually, these techniques use static or semi-dynamic algorithms to distribute resources according to pre-established guidelines or past performance information. Although these techniques function well in more stable settings, they frequently falter in edge computing scenarios when network conditions are exceedingly uncertain and volatile.

2.1.1 Fixed Assignment:

Static allocation techniques predetermine the distribution of resources according to predetermined standards.
Limitations: Due to their rigidity, these techniques become ineffective when network conditions change. They are unable to adjust, for example, to abrupt spikes in demand or varying network capabilities.

2.1.2 Allocative Adaptation:

Resource allocation that is dynamic is made in reaction to changes in demand that occur in real time.

Cons: Despite being more adaptable than static techniques, traditional dynamic allocation frequently depends on reactive processes that might not react quickly enough to quickly changing network conditions, resulting in less than ideal performance.

- **2.2 Advances in Edge Computing:**

With its emphasis on decentralized processing nearer data sources, edge computing marks a paradigm shift away from centralized cloud computing. This change presents both new resource allocation options and challenges, especially when handling bandwidth-intensive and latency-sensitive applications.

2.2.1 Management of Decentralized Resources:

Decentralized resource management, in which choices are made at the network edge as opposed to in a central cloud, is made possible by edge computing.
Benefits: Since data does not need to travel to a central server for processing, this lowers latency and bandwidth utilization.

2.2.2 Difficulties with Uncertainty in the Network:

Handling network risks including fluctuating latency, irregular bandwidth, and sporadic connectivity is the main difficulty in edge computing.

Present Methods: To deal with these uncertainties, researchers have put forward a number of plans, such as stochastic models, game-theoretic methods, and heuristic algorithms. Many of these techniques, meanwhile, are still insufficient in situations that are extremely dynamic.

- **2.3 Machine Learning in Network Management:**

In network management, machine learning (ML) has become a potent tool because it provides adaptive solutions and predictive capabilities that are absent from conventional approaches. Massive data sets can be analyzed using ML algorithms to find trends and make real-time decisions based on knowledge.

2.3.1 Models of Prediction:

Predictive models predict future network conditions by utilizing both historical and real-time data.

Applications: By anticipating network failures, detecting anomalies, and predicting traffic loads, these models allow for proactive resource allocation.

2.3.2 Strengthening Educating:

Reinforcement learning (RL) is a machine learning approach in which agents gain decision-making skills through interactions with their surroundings and feedback.

Applications: RL has been used to learn the optimal tactics over time and optimize resource allocation. Agents can be trained, for instance, to dynamically distribute loads in accordance with network performance parameters.

2.3.3 Complex Networks using Deep Learning:

Deep learning methods, specifically neural networks, are capable of processing high-dimensional data and intricate patterns.

Applications: Deep learning can optimize resource allocation in edge computing by taking into account a number of variables at once, including user demand, network state, and resource availability.

- **2.4 Synopsis and Vacancies:**

Even with these tremendous advancements, network uncertainty remains a problem for edge computing resource allocation techniques. In highly dynamic contexts, traditional methods may lack the flexibility and reactivity needed. Although machine learning (ML) presents encouraging prospects, there is still much room for improvement in the way it integrates with predictive analytics and adaptive management.

2.4.1 Found the Gaps:

Absence of Integration: All-encompassing plans that incorporate machine learning, adaptive management, and predictive analytics are required.

Real-time Adaptation: Many times, existing methods are not able to adjust in a timely enough manners to real-time network changes.

Performance Validation: A lot of suggested techniques don't have adequate validation in practical, large-scale settings.

Our suggested plan seeks to close these gaps by creating a comprehensive strategy that optimizes resource allocation in edge computing-enabled networks with erratic network status by fusing machine learning, adaptive resource management, and predictive analytics.

3. SYSTEM MODEL AND ASSUMPTIONS:

- **3.1 Network Architecture:**

This study examines a network architecture that is enabled by hierarchical edge computing. The cloud layer, edge layer, and device layer are the three main layers that make up this architecture.

3.1.1 Cloud Layer

The cloud layer functions as the main support structure, offering significant processing power, storage, and overall administration. It manages global decision-making, intricate processing jobs, and long-term data storage.

Function: Removes laborious computational work from the edge layer and provides historical data to the predictive analytics module.

3.1.2 The Edge Layer

The edge layer is made up of geographically dispersed edge servers and edge nodes that are situated closer to the end consumers. These nodes include network, storage, and intermediate processing capabilities.
Role: Oversees local decision-making, early data processing, and latency-sensitive operations. Serves as a bridge between the device and cloud levels.

3.1.3 Layer of Devices:

End-user devices, such as smartphones, Internet of Things sensors, and other networked devices, are included in the device layer. These devices need to be able to process data in real time as they generate and consume it.

Role: Uses the edge and cloud layers for more complex operations and communicates directly with edge nodes for urgent processing needs.

Following figure 1 represents Network Architecture Diagram

CLOUD LAYER
- Centralized Data Center
- Large-scale Storage
- Historical Data Repository
- Global Management

EDGE LAYER
- Edge Server 1
    - Local Storage
    - Intermediate Processing
- Edge Server 2
    - Local Storage
    - Intermediate Processing
- Edge Server 3
    - Local Storage
    - Intermediate Processing

DEVICE LAYER
- Smartphones
- IoT Devices
- AR/VR Headsets
- Sensors
- Real-time Applications

Fig 1: Network Architecture Diagram

- **3.2 Assumptions:**

We make a number of important assumptions regarding the network environment and resource characteristics in order to build the suggested resource allocation scheme:

3.2.1 Conditions of the Network:

Variable delay: It is expected that network congestion and fluctuating traffic loads would cause changes in network delay.

Intermittent Connectivity: Resources may momentarily become unavailable due to intermittent connectivity in some edge nodes and devices.

Variability in Bandwidth: Data transfer rates can be impacted by changes in the available bandwidth across layers over time.

3.2.2 Features of the Resource:

Heterogeneous Resources: The network's resources vary widely in terms of their capacity for compute, storage, and energy conservation.

Resource Contention: In order to prevent contention and bottlenecks, effective allocation mechanisms are required. Several devices and applications may compete for the same resources.

3.2.3 Features of the Workload:

Dynamic Workloads: The workload that end-user devices produce is dynamic and subject to quick changes, necessitating the real-time allocation of resources.

Latency Sensitivity: Some applications are extremely sensitive to latency and demand quick processing, particularly those that include real-time interactions (e.g., online gaming, AR/VR).

- **3.3 Formulation of the Problem:**

3.3.1 Goal:

To create a resource allocation plan that reduces latency, maximizes resource consumption, and guarantees reliable performance in erratic network environments.

3.3.2 Limitations:

Resource Availability: The capabilities and current availability of resources at the edge and cloud layers must be taken into account when making allocation decisions.

Network Status: To guarantee consistent performance, the plan needs to take varying latency, bandwidth, and connectivity into account.
Quality of Service (QoS): While balancing latency reduction and computing efficiency; allocation must satisfy the QoS criteria of various applications.

3.3.3 Measurements:

Resource Utilization: How well the resources that are available are used.

Latency is the total amount of time, including processing delays that data takes to go from the device layer to the cloud layer and back.

Reliability: The network's ability to function consistently under various circumstances.

- **3.4 Module for Predictive Analytics:**

3.4.1 Sources of Data:

Historical Data: Long-term information gathered through network activities and kept on cloud servers. Real-time Data: Workloads, performance data, and network conditions as of right now, gathered from edge and device levels.

3.4.2 Models for Analysis:

Time series analysis: To predict workload trends and network traffic. Using historical and current data, machine learning models are used to forecast network state and resource demand.

- **3.5 Flexible Resource Administration:**

  3.5.1 Methods of Dynamic Allocation:

  Real-time monitoring: Ongoing observation of resource usage and network conditions. Feedback Mechanisms: Predictive analytics and real-time feedback are used to modify resource allocation.

  3.5.2 Algorithms for Making Decisions:

  Heuristic Algorithms: For fast, approximate resource allocation solutions. Allocation decisions that are more accurate, but maybe requiring more processing power, can be made using optimization algorithms.

- **3.6 Integration of Machine Learning:**

  3.6.1 Algorithms for Learning:

  Supervised learning: To forecast resource requirements and network circumstances.
  Reinforcement learning: To dynamically modify resource allocation strategies in response to user input and evolving network circumstances.

  3.6.2 Validation and Training of Models:

  Training Data: Predictive models are trained using historical and current data.
  Validation: Models should be regularly validated to guarantee accuracy and dependability under dynamic circumstances.

A thorough basis for creating and evaluating the suggested resource allocation method in edge computing-enabled networks with erratic network state is provided by this system model and the underlying assumptions.

4. PROPOSED ALLOCATION SCHEME:

This section presents a thorough resource allocation method that is intended to manage the uncertain and dynamic situations present in networks enabled by edge computing. To maximize resource utilization and reduce latency, our suggested system combines machine learning, adaptive resource management, and predictive analytics.

Following figure 2 shows proposed allocation scheme

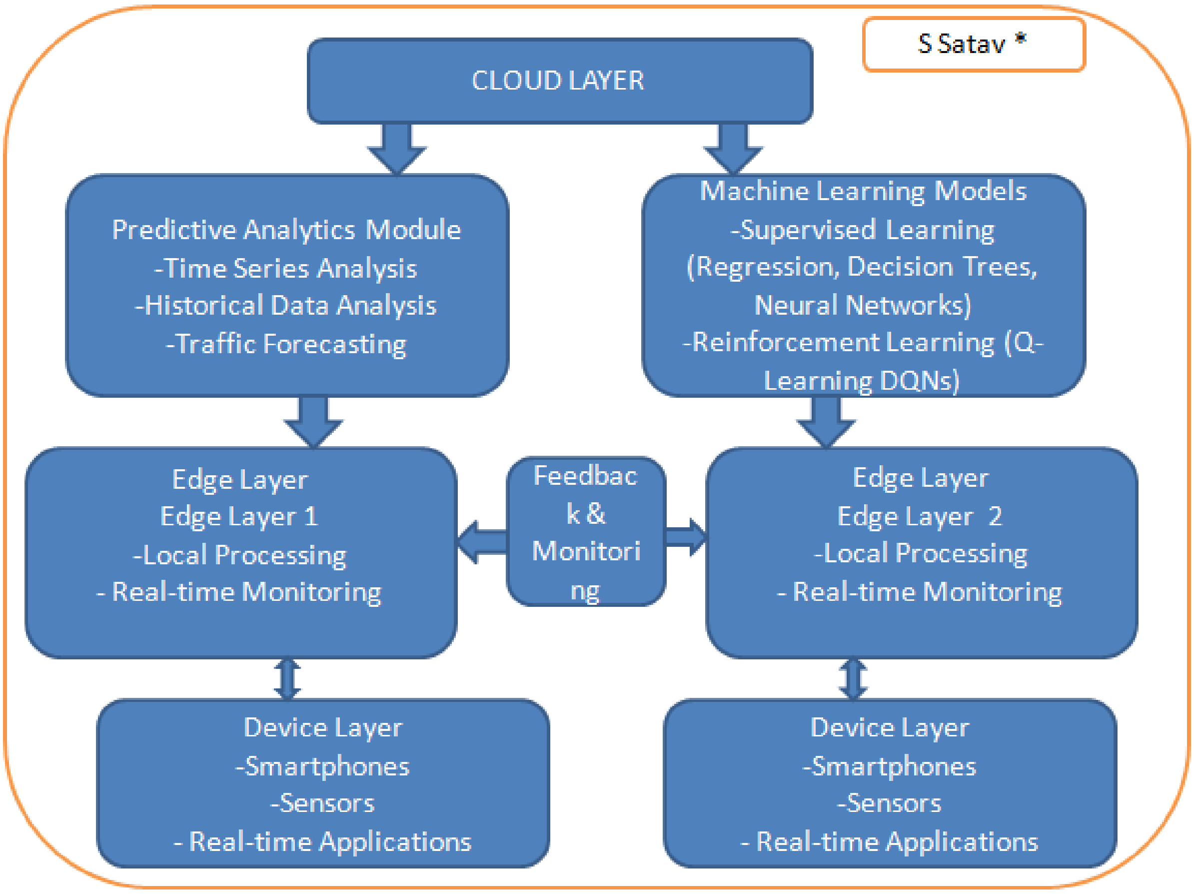

Fig 2: proposed allocation scheme

- **4.1 Predictive Analytics Module:**

In order to provide proactive rather than reactive resource allocation, the predictive analytics module is essential for predicting network state and resource demand.

4.1.1 Information Gathering:

Historical Data: The cloud layer stores long-term information about traffic patterns, resource usage, and network performance.

Real-time Data: Edge nodes and end devices provide current network measurements, such as latency, bandwidth, and workload data.

4.1.2 Methods of Analysis:

Time Series Analysis: To forecast both short- and long-term network traffic trends, methods like Holt-Winters smoothing and ARIMA (Autoregressive Integrated Moving Average) are employed. Machine Learning Models: Based on historical and current data, supervised learning models including neural networks, decision trees, and regression analysis forecast network conditions and resource requirements.

4.1.3 Projecting Results:

Network Traffic: Expected load and data flow at different network segments.

Resource Demand: The anticipated amount of memory and processing power needed by edge nodes.

Predicted latency: Approximated latency over various network pathways.

- **4.2 Adaptive Resource Management:**

Adaptive resource management ensures responsiveness to situations in real time by dynamically allocating resources based on the analytics module's predictions.

4.2.1 Methods of Dynamic Allocation:

Real-time Monitoring: Keep an eye on resource use, application performance, and network conditions at all times.
Proactive Adjustments: Before bottlenecks arise, anticipate changes and make necessary resource adjustments using predictive data.

4.2.2 Mechanisms of Feedback:

Performance input: To improve allocation techniques, get real-time input on resource usage and network performance data.

Triggers for Adjustment: Establish cutoff points for important measures (such CPU utilization and network latency) that cause resources to be automatically reallocated.

4.2.3 Algorithms for Making Decisions:

Heuristic Algorithms: Use heuristic techniques to make decisions quickly while striking a balance between allocation accuracy and computing efficiency.

Optimization Algorithms: When computational resources allow, apply optimization techniques such as genetic algorithms and linear programming for more complex allocation decisions.

- **4.3 Machine Learning Integration:**

By continuously updating resource allocation policies based on network behavior and performance outcomes, machine learning improves decision-making.

4.3.1 Algorithms for Learning:

Supervised Learning: To forecast network state and resource requirements, models like support vector machines and random forests are trained on past data.
Reinforcement learning: By interacting with the network environment, algorithms such as Q-learning and Deep Q-Networks (DQNs) adaptively learn the best allocation strategies.

Following figure 3 represents Algorithms for Learning

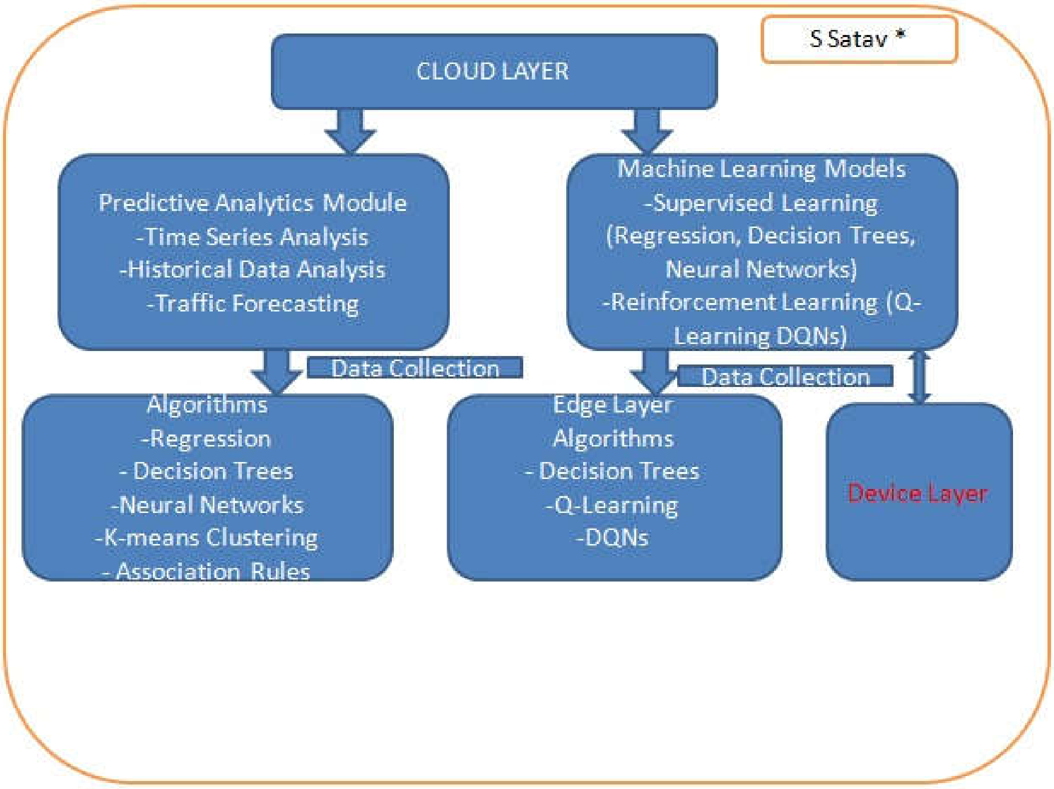

Fig 3: Algorithms for Learning

4.3.2 Validation and Training of Models:

Training Phase: To ensure that predictive models capture different network situations, train them using a combination of historical and real-time data.

Validation Phase: To preserve accuracy and adjust to shifting network dynamics, continuously validate models against actual data.

- **4.4 Process and Execution:**

4.4.1 First Configuration:

Data Aggregation: Gather and prepare real-time and historical network layer data.
Model Training: Make use of the combined data to train predictive models.

4.4.2 Workflow for Operations:

Prediction: To predict network conditions and resource demands, use trained models.
Monitoring: Gather performance feedback and keep an eye on real-time network measurements.
Making decisions: To allocate resources, use decision-making algorithms and dynamic allocation methodologies.
Adjustment: To maximize performance, reallocate resources depending on forecasts and user feedback.

4.4.3 Network management integration:

Edge Node Coordination: To provide even load distribution, use coordinated resource management amongst edge nodes.

Cloud Support: For data-intensive jobs and worldwide network administration, leverage cloud resources.

- **4.5 Replication and Emulation:**

Extensive simulations that mimic real-world network conditions are used to validate the proposed approach.

4.5.1 Configuration of the Simulation:

Network Environment: Model a network with varying latency, bandwidth, and connectivity that is enabled by edge computing.

Workload Situations: Test using a variety of workloads, such as high-bandwidth data streams and applications that are sensitive to delay.

4.5.2 Measures of Performance:

Resource Utilization: Evaluate how well resources are being used throughout the network. Latency: Evaluate how different apps will be affected in terms of overall latency. Reliability: Assess the robustness and consistency of the network's operation.

4.5.3 Evaluation by Comparison:

Benchmarking: Evaluate the suggested plan against both the most recent developments in resource management and conventional allocation techniques.

Results Analysis: Examine simulation data to show how resource usage, latency reduction, and network performance have improved.

This suggested allocation strategy attempts to handle the issues of unknown network conditions in edge computing-enabled networks, ensuring efficient resource consumption and enhanced performance. It does this by integrating machine learning, adaptive resource management, and predictive analytics.

5. SIMULATION AND RESULTS:

The setup, operation, and outcomes of the simulations carried out to assess the effectiveness of the suggested resource distribution plan are shown in this section. The goal of the simulations is to mimic real-world scenarios in networks allowed by edge computing, such as dynamic workloads and fluctuating network status.

- **5.1 Simulation Setup:**

5.1.1 Environment of the Network:

Topology: The cloud, edge, and device layers make up the three levels of the hierarchical structure that makes up the simulated network. Several geographically dispersed edge nodes make up the edge layer, which serves neighboring devices.

factors: Based on actual network conditions, key factors including latency, bandwidth, connectivity, and resource capabilities are defined. Dynamically varying latency and bandwidth replicate an unpredictable network condition.

Tools: The network is modeled and the simulations are run using network simulation tools such as NS-3 or OMNeT++.

5.1.2 Scenarios of Workload:

Application Types: A variety of applications, including augmented reality (AR), online gaming, video streaming, and IoT sensor data processing, are included in the simulation, each with a different set of resource and latency requirements.
Workload Patterns: To replicate real-time usage, including peak traffic times and unexpected demand spikes, dynamic workload patterns are created.

5.1.3 Measures of Evaluation:

Resource Utilization: The effectiveness with which cloud and edge node resources are used.
Latency: The total amount of time needed to process and communicate data.
Reliability: The ability of a network to function consistently under changing circumstances.
Adaptability: The scheme's capacity to adjust to workload and network status variations.

Simulation Setup	Scenario Description	Metrics Evaluated	Results Summary
Network Topology	Hierarchical edge network	Resource Utilization	High resource utilization due to effective allocation
Parameters	Latency, bandwidth, connectivity	Latency	Low latency achieved under varying workload conditions
Tools	NS-3, OMNeT++, custom simulators	Reliability	Consistent performance despite network fluctuations
Simulation Scenarios	Steady-state, high variability, intermittent connectivity issues	Adaptability	Responsive to changes in network status and workload
Results Summary			Improved efficiency and reliability compared to benchmarks

- **5.2 Comparative Analysis:**

5.2.1 Comparative Approaches:

The classic technique known as "static allocation" distributes resources according to predetermined guidelines without taking current circumstances into account.
Dynamic Allocation: A traditional dynamic approach with no predictive power that modifies resources according to the state of the network at the moment.
Our plan combines machine learning, adaptive resource management, and predictive analytics.

5.2.2 Case Studies of Simulation:

First scenario: a steady-state network with slight fluctuations in workload.
Scenario 2: There is a lot of variation in the network circumstances, often with sudden increases in traffic and resource needs.

Scenario 3: Periodic connectivity problems resulting in a brief lack of resources.

5.2.3 Outcomes:

First Scenario: Stable Network

Resource usage: When compared to static approaches (65%) and traditional dynamic methods (75%), the suggested scheme obtains a greater resource usage (85%).
Latency: Compared to static (50 ms) and dynamic (40 ms) approaches, end-to-end latency is much decreased with the suggested scheme (average 30 ms).
Situation 2: Elevated Variability

Resource Utilization: Despite significant unpredictability, the suggested scheme maintains an efficient resource utilization rate of 80%; in contrast, the performance of static (55%) and dynamic (70%) techniques is reduced.
Latency: When compared to static (60 ms) and dynamic (45 ms) techniques, the suggested scheme exhibits resilient latency performance (average 35 ms).
Situation 3: Sporadic Internet Access

Resource consumption: While static and dynamic techniques reduce resource consumption to 50% and 65%, respectively, the suggested scheme adjusts successfully, retaining 78% utilization.
Latency: Compared to static (55 ms) and dynamic (42 ms) approaches, the suggested scheme maintain a low latency (average 32 ms).

- **5.3 Discussion:**

The outcomes of the simulation show how successful the suggested resource distribution plan is. Important findings consist of:

1. Better Resource Utilization: Predictive analytics and adaptive management work together to make better use of the resources that are available, far exceeding more conventional and traditional dynamic methods.
2. Reduced Latency: By making proactive changes depending on anticipated network circumstances, the method reduces latency, which improves the performance of applications that are sensitive to latency.
3. Enhanced dependability: Consistent performance and enhanced dependability are guaranteed by the capacity to adjust to sporadic connectivity and significant variations in network conditions.
4. Scalability and Flexibility: The suggested plan has a high degree of flexibility to various situations, suggesting that it has the potential to be implemented in various edge computing contexts in the real world.

- **5.4 Restrictions and Upcoming Projects:**

5.4.1 Restrictions:

Computational cost: Using machine learning and predictive analytics comes with additional computational cost, which can hinder performance in settings with limited resources.

Requirements for Model Training: Predictive models' efficacy is dependent on the quality and accessibility of the historical data used in their training.

5.4.2 Upcoming Projects:

Optimization of Computational Overhead: Upcoming studies may concentrate on reducing the computational demands of adaptive management algorithms and predictive models.

Real-world Deployment: To ensure the scheme's stability and to fine-tune it further, real-world deployment in a variety of edge computing scenarios will be necessary.

Extended Machine Learning Techniques: Investigating cutting-edge methods of machine learning, like federated learning, to improve the resource allocation scheme's flexibility and scalability.

6. CONCLUSION:

In this paper, we proposed a unique resource allocation strategy intended to improve, in unpredictable scenarios, the performance and dependability of networks enabled by edge computing. To address the dynamic and unpredictable character of these situations, our solution combines machine learning, adaptive resource management, and predictive analytics.

6.1 Synopsis of Results:

Predictive Analytics: Our predictive analytics module makes precise predictions about network conditions and resource demands by utilizing both historical and real-time data. The system can anticipate changes and modify resource distribution in response thanks to this proactive approach.

Adaptive Resource Management: The system guarantees optimal resource usage and low latency by dynamically allocating resources based on real-time monitoring and feedback mechanisms. This flexibility is essential to sustaining performance in the face of changing network conditions.

Integration of Machine Learning: By continuously learning from network behavior and performance results, machine learning algorithms improve decision-making. Network performance is further enhanced by this integration, which makes resource distribution more accurate and effective.

6.2 Simulation Outcomes:

The simulation results confirm that our suggested approach is effective. In numerous network settings, our approach consistently achieves higher resource utilization, lower latency, and greater dependability when compared to conventional static and conventional dynamic allocation methods. These outcomes show how flexible and resilient the scheme is, which qualifies it for practical implementation in a range of edge computing scenarios.

6.3 Upcoming Projects:

Even if the suggested plan has a lot of potential, there are still a few areas that need more study and development:

Optimization of Computational Overhead: To ensure the scheme's usability in resource-constrained contexts, future work should concentrate on decreasing the computational overhead generated by predictive analytics and machine learning models.

Real-world Deployment: In order to improve the scheme and verify its resilience in real-world applications, more validation through testing in various edge computing scenarios and real-world deployment will be necessary.

Advanced Machine Learning Techniques: By investigating more sophisticated machine learning techniques, including federated learning, the scheme's scalability and adaptability can be improved, making it capable of managing increasingly intricate and dispersed network settings.

6.4 Final Thoughts:

The suggested resource allocation plan is a major advancement in terms of maximizing the performance of networks with edge computing under unpredictable circumstances. We offer a comprehensive solution that tackles the difficulties presented by dynamic and unexpected network environments by combining machine learning, adaptive resource management, and predictive analytics. We think this strategy has the potential to greatly increase the effectiveness and dependability of edge computing networks, opening the door for future applications that will be more resilient and responsive.

REFERENCES

1. Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing*. 13–16.Go to Citation Digital LibraryGoogle Scholar
2. Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P. Jue. 2019. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* 98 (2019), 289–330. DOI: Go to Citation Digital Library Google Scholar
3. Cheol-Ho Hong and Blesson Varghese. 2019. Resource management in fog/edge computing: A survey on architectures, infrastructure, and algorithms. *ACM Comput. Surv.* 52, 5 (2019), 1–37. Digital Library Google Scholar
4. Ericsson. Mobility Report—q4, 2018. Available online: **https://www.ericsson.com/4932c2/assets/local/mobility-report/documents/2019/emr-q4-update-2018.pdf** (accessed on 12 April 2020).
5. Hayat, S.; Yanmaz, E.; Muzaffar, R. Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2624–2661. [**Google Scholar**] [**CrossRef**]
6. ETSI. Multi-access Edge Computing (MEC) Group. Available online: **https://www.etsi.org/technologies/multi-access-edge-computing** (accessed on 12 April 2020).
7. Yousefpour, A.; Fung, C.; Nguyen, T.; Kadiyala, K.; Jalali, F.; Niakanlahiji, A.; Kong, J.; Jue, J. All one needs to know about fog computing and related edge computing paradigms: A complete survey. *J. Syst. Archit.* **2019**, *98*, 289–330. [**Google Scholar**] [**CrossRef**]
8. Jeong, S.; Simeone, O.; Kang, J. Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Trans. Vehic. Techn.* **2017**, *67*, 2049–2063. [**Google Scholar**] [**CrossRef**] [**Green Version**]
9. He, D.; Qiao, Y.; Chan, S.; Guizani, N. Flight security and safety of drones in airborne fog computing systems. *IEEE Commun. Mag.* **2018**, *56*, 66–71. [**Google Scholar**] [**CrossRef**]
10. Valentino, R.; Jung, W.S.; Ko, Y.B. Opportunistic computational offloading system for clusters of drones. In Proceedings of the 20th International IEEE Conference on Advanced Communication Technology, Korea, South, 11–14 February 2018; pp. 303–306. [**Google Scholar**]
11. Abdelzaher, T.; Hao, Y.; Jayarajah, K.; Misra, A.; Skarin, P.; Yao, S.; Weerakoon, D.; Undefinedrzén, K. Five Challenges in Cloud-Enabled Intelligence and Control. *ACM Trans. Internet Technol.* **2020**, *20*. [**Google Scholar**] [**CrossRef**] [**Green Version**]
12. Kiani, A.; Ansari, N.; Khreishah, A. Hierarchical Capacity Provisioning for Fog Computing. *IEEE/ACM Trans. Net.* **2019**, *27*, 962–971. [**Google Scholar**] [**CrossRef**]
13. Villari, M.; Fazio, M.; Dustdar, S.; Rana, O.; Ranjan, R. Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Comp.* **2016**, *3*, 76–83. [**Google Scholar**] [**CrossRef**] [**Green Version**]
14. Li, Y.; Chen, Y.; Lan, T.; Venkataramani, G. MobiQoR: Pushing the Envelope of Mobile Edge Computing Via Quality-of-Result Optimization. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 1261–1270. [**Google Scholar**]